

Micro Fiche Scan

Name of device(s) tested:

UDA

Test description:

PDP-11 UDA DRIVE FMTR

MAINDEC Number or Package Identifier (after SEP 1977):

CZUDED0

Fiche Document Part Number:

AH-S837D-MC

Fiche preparation date unknown, using copyright year:

1984

Image resolution:

8-bit gray levels, max. quality for archiving

COPYRIGHT (C) 1984 by d|i|g|i|t|a|l

1

.REM

.TITLE ZUDED0 PDP-11 UDA DRV FMTR

IDENTIFICATION  
-----

PRODUCT CODE: AC-S836D-MC  
PRODUCT NAME: CZUDED0 PDP-11 UDA DRV FMTR  
PRODUCT DATE: 24-MAY-83  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: DALE KECK

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

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

COPYRIGHT (C) 1981, 1983 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

	Page
1.0 GENERAL INFORMATION	3
1.1 PROGRAM ABSTRACT	3
1.2 SYSTEM REQUIREMENTS	4
2.0 OPERATING INSTRUCTIONS	4
2.1 COMMANDS	4
2.2 SWITCHES	5
2.3 FLAGS	6
2.4 HARDWARE QUESTIONS	7
2.5 SOFTWARE QUESTIONS	8
2.6 MANUAL INTERVENTION QUESTIONS	9
2.7 EXTENDED P-TABLE DIALOGUE	10
2.8 QUICK STARTUP PROCEDURE	12
3.0 ERROR INFORMATION	15
3.1 TYPES OF ERROR MESSAGES	15
3.2 SPECIFIC ERROR MESSAGES	16
3.2.1 HOST PROGRAM ERROR MESSAGES	16
3.2.2 DUP PROGRAM ERROR MESSAGES	24
4.0 PERFORMANCE AND PROGRESS REPORTS	28
5.0 TEST SUMMARIES	29

## 1.0 GENERAL INFORMATION

-----

### 1.1 PROGRAM ABSTRACT

-----

This program will format any disk drive connected to a UDA50 disk controller. At the time of this writing, there are three such drives in existence -- the RA60, RA80 and RA81. No changes to this program will be needed to format new disk drives as they become available.

There are three ways to format a disk with this program:

1. Reformat - Format the disk with the bad sector information that was written onto the disk at the factory. This is the normal way to format a disk.
2. Reconstruct - Format the disk without using any bad sector information. This should be used only when the bad sector information has been destroyed or for some reason can no longer be read from the disk. This method may also be specified in the disk drive's maintenance manual for special cases (eg. changing an RM/RA80 spare HDA from RM80 format to RA80 format).
3. Restore - Format the disk using bad sector information obtained from a disk file on the XXDP+ system load device. This method is provided for use by manufacturing. No files are provided, nor any method of obtaining the files, at this time.

The format operation is performed by a Diagnostic Utilities and Protocol (DUP) program loaded into the UDA50 disk controller. The host program simply downline loads the DUP program into the UDA50 and monitors its execution. The DUP program obtains parameters from the host program (eg. drive number and format mode) and requests the host program to print error and summary messages. The DUP program is also commonly called a "diagnostic machine" (DM) program.

This program can only format in one mode at a time. In RESTORE mode, only one disk may be selected in the hardware questions or an error message will result and the program will stop.

In REFORMAT and RECONSTRUCT modes, any number of disk drives may be selected. A UDA50 can only format one disk at a time, so each disk on a UDA50 will be selected sequentially. If the disk drives to be formatted are connected to different UDA50s, all UDA50s will be run simultaneously. For example, lets assume three units are selected for formatting in the hardware questions, units 1 and 2 are connected to one UDA50 and unit 3 is connected to a different UDA50 (Unibus addresses are different). This program will automatically start simultaneous format operations on units 1 and 3. When unit 1 finishes (or errors), unit 2 will be started. After units 2 and 3 are finished, the program stops.

This program will stop after each pass (all units formatted once). There is no need to specify a PASS switch on the command line to the Diagnostic Runtime Services (eg. START/PASS:1).

Special provisions have been made to allow this program to run under an APT system in manufacturing. This system does not allow questions to be asked of an operator. Such a condition also exists under XXDP, when the UAM flag is set. In this condition, only reformat mode can be selected. Selecting RECONSTRUCT or RESTORE will result in an error. Also, a date of 1-JAN-70 will be written on the disk.

## 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. The initial version will require the following:

- PDP-11 Unibus processor
- 28K words of memory (minimum)
- Console terminal
- XXDP, load media containing this program
- One or more UDA50 subsystems. The subsystem controllers type UDA50-A with microcode level 3 or greater.

A system clock - either type L or P - will be used to time the DUP program and report runtime, if available. If no system clock is available, this program cannot detect a hung DUP program.

## 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, 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)
IDU	Inhibit program dropping of units
LOT	Loop on test

\*Error messages are described in section 3.1

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

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

## 2.4 HARDWARE QUESTIONS

-----

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

### UNIBUS ADDRESS OF UDA (0) 172150 ?

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

### VECTOR (0) 154 ?

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

### BR LEVEL (0) 5 ?

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

### UNIBUS BURST RATE (0) 63 ?

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

### DRIVE NUMBER (0) 0 ?

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



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. You may change the default values with the SETUP utility.

REFORMAT USING EXISTING BAD SECTOR INFORMATION (L) Y ?

If this question is answered "YES", then the user wants the REFORMAT mode format operation. REFORMAT mode will use the bad sector information that is already on the disk. Any other mode will destroy this information. If this question is answered "NO", the following will be asked to be sure the user knows what he is doing.

NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR INFORMATION ON THE DISK.  
AGAIN - REFORMAT USING EXISTING BAD SECTOR INFORMATION (L) Y ?

This is asked to verify that the user does want to destroy the bad sector information on the disk and run another format mode. If this is answered "YES", then the user wants the REFORMAT mode format operation and use the existing bad block information. If again answered "NO", the following question will be asked.

RECONSTRUCT BAD SECTOR INFORMATION (L) Y ?

A "YES" answer will cause a reconstruct mode format operation. If answered "NO", the following will be asked to verify the user really wants the restore mode format.

DO YOU HAVE A FILE ON THE SYSTEM LOAD DEVICE  
CONTAINING BAD SECTOR INFORMATION (L) N ?

Note that such a file will not be provided with the diagnostic and this mode is not recommended. The format will begin only on a "YES" answer. Otherwise the following message will be printed and the program will abort.

YOU CANNOT PROCEED WITHOUT SUCH A FILE.  
RESTART PROGRAM AND SELECT TO REFORMAT OR RECONSTRUCT DISK.

## 2.6 MANUAL INTERVENTION QUESTIONS

-----

When the program starts a warning message is printed to warn of improper use of this formatter.

### WARNING:

THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC TOOL. RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK DRIVE'S SERVICE MANUAL.

ARE YOU SURE YOU WANT TO RUN THIS FORMATTER (L) N ?

You must answer "YES" or the program will abort immediately. This family of disk drives uses a powerful bad block revectoring mechanism to replace blocks that fall on defective areas of the disk media. As a disk is used and defective blocks are detected, DEC operating systems replace the blocks with other blocks on the disk (reserved for this purpose and otherwise inaccessible) so that the disk constantly appears to have its full storage capacity of error free disk blocks. Formatting a disk of this type destroys this history information and is absolutely not recommended except in the cases specifically described in the disk drive's service manual. These disks are fully formatted when shipped from the factory, therefore there is no reason to run this formatter program at installation.

Upon answering "YES" to the above question, the date will be asked for in the format used by the XXDP+ system.

ENTER DATE AS DD-MMM-YY (A) 1-JAN-70 ?

The default is provided so the user need not supply the date. The date question will normally only be asked one time. If an improper answer is typed, "INPUT ERROR" is printed and the question is asked again. A two or four digit year may be typed. A four digit year must be 1900 or greater (eg. 14-APR-1982). If only two digits are typed, the year is determined as follows:

1. If the number typed is 70 or greater, a 19 is prefixed.  
Eg., 1-JAN-70 translates to year 1970 and 25-DEC-99 translates to year 1999.
2. If the number typed is less than 70, a 20 is prefixed. Eg., 1-APR-21 is translated to year 2021.

IF RECONSTRUCT mode is selected, the following question will be asked for each disk before the format operation begins.

SERIAL NUMBER FOR UNIT xx UDA AT xxxxxx DRIVE xxx  
(A) ?

A decimal number in the range of 0 to 18446744073709551615 must be entered (no default).

If RESTORE mode is selected, the following question will be asked.

NAME OF FILE CONTAINING BAD SECTOR INFORMATION FOR  
DISK TO BE FORMATTED (A) ?

If the file named does not exist on the system load device,  
the program will abort back to the XXDP prompt after printing  
an error message.

## 2.7 EXTENDED P-TABLE DIALOGUE

-----

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

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

# UNITS (D) ? 8<CR>

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

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

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

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

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

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

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

```
UNIT 8  
CSR ADDRESS (0) 160000<CR>  
SUB-DEVICE # (0) ? 7<CR>  
Q-FACTOR (0) 1 ? <CR>
```

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

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

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

```
# UNITS (0) ? 8<CR>
```

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

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

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

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

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

```
# UNITS (D) ? 8<CR>
UNIT 1
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 0-7<CR>
Q-FACTOR (O) 0 ? 0.1,0,...,1.1<CR>
```

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

## 2.8 QUICK START-UP PROCEDURE

-----

To start-up this program:

1. Boot XXDP.
2. Give the date and answer the LSI and 50HZ (if there is a clock) questions
3. Type "R ZUDED0"
4. Type "START"
5. Answer the "CHANGE HW" question with "Y"
6. Answer all the hardware questions
7. Answer the "CHANGE SW" question with "N"
8. Answer "Y" to the "ARE YOU SURE ..." question following the warning. Please read the disk drive's service manual before answering this question.
9. Type today's date.

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

Sample of terminal dialogue to test two disks on one UDA50:

DR>STA

CHANGE HW (L) ? Y

# UNITS (D) ? 2

UNIT 0

UNIBUS ADDRESS OF UDA (0) 172150 ?

VECTOR (0) 154 ?

BR LEVEL (D) 5 ?

UNIBUS BURST RATE (D) 63 ?

DRIVE NUMBER (D) 0 ? 0,1

CHANGE SW (L) ? N

WARNING:

THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC  
TOOL. RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK  
DRIVE'S SERVICE MANUAL.

ARE YOU SURE YOU WANT TO RUN THIS FORMATTER (L) N ? Y

ENTER DATE AS DD-MMM-YY (A) 1-JAN-70 ? 14-APR-82

UNIT 0 UDA AT 172150 DRIVE 0 RUNTIME 0:00:20

Format begun Version 8

STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK  
UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN  
BROUGHT ONLINE.

UNIT 1 UDA AT 172150 DRIVE 1 RUNTIME 0:00:23

Format begun Version 8

STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK  
UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN  
BROUGHT ONLINE.

UNIT 0 UDA AT 172150 DRIVE 0 RUNTIME 0:42:20

Format completed

2 Revectorized LBNS

2 Primary revectorized LBNS

0 Secondary/tertiary revectorized LBNS

0 Bad RBNS

0 Bad blocks in the RCT area due to data errors

0 Bad blocks in the DBN area due to data errors

0 Bad blocks in the XBN area due to data errors

2 Blocks retried on the check pass

FCT used successfully

```

UNIT 1 UDA AT 172150 DRIVE 1  RUNTIME 1:25:18
Format completed
 131 Revectored LBNS
 131 Primary revectored LBNS
   0 Secondary/tertiary revectored LBNS
   0 Bad RBNS
   1 Bad blocks in the RCT area due to data errors
   0 Bad blocks in the DBN area due to data errors
   0 Bad blocks in the XBN area due to data errors
 249 Blocks retried on the check pass
FCT used successfully

```

```

CZUDE EOP 1
 0 CUMULATIVE ERRORS
DR>

```

Sample of terminal dialogue going through software questions.  
Only one disk is being tested.

```
DR>STA
```

```
CHANGE MW (L) ? N
```

```
CHANGE SW (L) ? Y
```

```
REFORMAT USING EXISTING BAD SECTOR INFORMATION (L) Y ? Y
```

```

WARNING:
  THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC
  TOOL.  RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK
  DRIVE'S SERVICE MANUAL.

```

```
ARE YOU SURE YOU WANT TO RUN THIS FORMATTER (L) N ? Y
```

```
ENTER DATA AS DD-MMM-YY (A) 1-JAN-70 ? 14-APR-82
```

```

RUNTIME 0:00:20
Format begun Version 8
STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK
UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN
BROUGHT ONLINE.

```

```

RUNTIME 1:33:45
Format completed
 2 Revectored LBNS
 2 Primary revectored LBNS
 0 Secondary/tertiary revectored LBNS
 0 Bad RBNS
 0 Bad blocks in the RCT area due to data errors
 0 Bad blocks in the DBN area due to data errors
 0 Bad blocks in the XBN area due to data errors
 2 Blocks retried on the check pass
FCT used successfully

```

```

CZUDE EOP 1
 0 CUMULATIVE ERRORS
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:

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

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

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

Device fatal errors (DVC FTL ERR) are used to report errors that are fatal to the device (may be either a UDA50 or disk drive). Testing stops on that device for the remainder of the current test.

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 UDA50 being used and the time of the error:

```
HOST PROGRAM UDA AT XXXXXX RUNTIME hhh:mm:ss
```

The host program (PDP-11) detected the error. UDA AT XXXXX identifies the address of the UDA50 being tested. It may be omitted if the error is not specific to one UDA50.



Sample error message:

```
CZUDE DVC FTL ERR 00021 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx      . general message
HOST PROGRAM UDA AT 172150 RUNTIME 0:00:12                       . basic message
UDA RESIDENT DIAGNOSTICS DETECTED FAILURE                        \
  UDASA CONTAINS 104041                                         \- extended message
REPLACE UDA MODULE M7485
```

The DUP program may also print error messages. They are printed exactly as presented by the DUP program and cannot be suppressed by any flags.

### 3.2 SPECIFIC ERROR MESSAGES

-----

#### 3.2.1 HOST PROGRAM ERROR MESSAGES

-----

Following is a list of the error messages that may be printed by the diagnostic program. In the list, some of the numbers that may vary with execution or program version are shown as "xxx". These include program counters and runtime. Other numbers, such as unit number, drive number, UDA50 address and data in registers are filled with sample numbers. Additional information about the error may follow the error message.

```
00001 CZUDE SYS FTL ERR 00001 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS
UDA HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE
```

When the hardware questions were answered, two units were selected with the same UNIBUS address but with a different vector, BR level or burst rate. A single UDA50 can have only one vector, BR level or burst rate. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

```
00002 CZUDE SYS FTL ERR 00002 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS
TWO UNITS SELECT THE SAME DRIVE
```

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

00003 CZUDE SYS FTL ERR 00003 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS  
 MORE THAN EIGHT DRIVES SELECTED ON THIS UDA

Up to four physical disk drives can be attached to a UDA50 at one time. A physical disk drive may be from one to four logical disk drives. Each logical disk drive is considered one unit to the diagnostic program. Even though more than eight logical disk drives can be attached to one UDA50, the UDA50 only supports eight. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00004 CZUDE SYS FTL ERR 00004 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM RUNTIME x:xx:xx  
 NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED  
 PLEASE START PROGRAM OVER AND TEST FEWER UNITS AT A TIME

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

00008 CZUDE SYS FTL ERR 00008 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS  
 TWO UDA'S USE THE SAME VECTOR

The hardware questions for two units specified different UDA50 Unibus addresses but identical vector addresses. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00009 CZUDE DVC FTL ERR 00009 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM RUNTIME x:xx:xx  
 ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE.  
 PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK.

If the operator chooses to run the formatter in RESTORE mode, then only one disk can be selected in the hardware questions. RESTORE mode is run in this way because a file containing the bad block information is used and that information matches only one drive.

00010 CZUDE DVC FTL ERR 00010 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM RUNTIME x:xx:xx  
 THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE

This program needs to ask questions of the operator. It refuses to run in RECONSTRUCT and RESTORE modes because the questions obtain data that is absolutely necessary. REFORMAT mode is allowed to run because only a date is needed. The default date of 1-JAN-70 is used.

00014 CZUDE DVC FTL ERR 00014 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
UDA50 CONTROLLER IS AT A REVISION LEVEL NO LONGER SUPPORTED  
BY THIS FORMATTER PROGRAM. THIS PROGRAM REQUIRES A UDA50-A  
CONTROLLER (MODEL 6) WITH MICROCODE VERSION AT 3 OR GREATER.

CONTROLLER REPORTED MODEL CODE xx AND MICROCODE VERSION xx

All UDA50-0's (modules M7161-2) are not supported by this diagnostic. The module set M7485-6 is the only one that can be tested by this diagnostic. If the controller is a UDA50-0 (M7161-2) it will not be tested. If the controller is a UDA50-A (M7485-6) and it has old microcode (the microcode version is less than 3) this message will be printed but testing will go on. If the controller consists of the M7161-2 modules, install one with M7485-6 modules. Do not intermix the two, it will not work!

00020 CZUDE DVC FTL ERR 00020 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
MEMORY ERROR TRYING TO READ UDA REGISTERS  
CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7485  
OR UNIBUS  
OR REPLACE UDA MODULE M7485

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

00021 CZUDE DVC FTL ERR 00021 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
UDA RESIDENT DIAGNOSTICS DETECTED FAILURE  
UDASA CONTAINS 105154  
REPLACE UGA MODULE M7486

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

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

Replace the board specified. M7485 is the Unibus interface board. M7486 is the SDI interface board.

00022 CZUDE DVC FTL ERR 00022 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION  
 STEP BIT EXPECTED 004000  
 UDASA CONTAINS 000000  
 REPLACE UDA MODULE M7485

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

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

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

00023 CZUDE DVC FTL ERR 00023 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION  
 6 WORDS WERE TO BE CLEARED STARTING AT ADDRESS 040644  
 FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):

ADDRESS	CONTENTS
040644	000010
040650	000010
040652	000010

REPLACE UDA MODULE M7485

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

00024 CZUDE DVC FTL ERR 00024 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 UDASA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION  
 PURGE/POLE DIAGNOSTICS WERE REQUESTED  
 UDASA CONTENTS 004400

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

00025 CZUDE DVC FTL ERR 00025 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION  
 UDASA EXPECTED 004400  
 UDASA CONTAINS 004000  
 REPLACE UDA MODULE M7485

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

00030 CZUDE DVC FTL ERR 00030 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE RUNNING DM PROGRAM  
 UDASA CONTAINS 100004

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

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

00031 CZUDE DVC FTL ERR 00031 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
 NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES  
 ASSUME PROGRAM IS HUNG

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

00032 CZUDE DVC FTL ERR 00032 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER  
MESSAGE BUFFER CONTAINS:  
000001 000002 000003 000004 000005 000006 000007  
000008 000009 000010 000011 000012 000013 000014  
000015 000016 000017 000018 000019 000020 000021  
000022 000023 000024 000025 000026 000027 000028  
000029 000030 000031 000032 000033 000034 000035

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

00033 CZUDE DVC FTL ERR 00033 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
00034 HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
RESPONSE PACKET FROM UDA DOES NOT CONTAIN EXPECTED DATA  
EITHER UDA RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED CORRECTLY  
COMMAND PACKET SENT                      RESPONSE PACKET RECEIVED  
000000 000020                              000000 000020  
000000 000000                              000000 000000  
000000 000002                              000000 000202  
000000 014336                              000000 014336  
000000 034674                              000000 034674  
000000 000000                              000000 000000  
000000 000000                              000000 000000  
000000 051232                              000000 051232  
000000 000000                              000000 000000  
000000 000000                              000000 000000  
000000 000000                              000000 000000  
000000 000000                              000000 000000

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

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

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

00036 CZUDE DVC FTL ERR 00036 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
NO INTERRUPT RECEIVED FROM UDA FOR 30 SECONDS  
WHILE LOADING DM PROGRAM

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

00037 CZUDE DVC FTL ERR 00037 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM  
UDASA CONTAINS 100004  
REPLACE UDA MODULE M7485

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

00100 CZUDE DVC FTL ERR 00100 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
DUP PROGRAM ASKED UNEXPECTED QUESTION (25)

The DUP program sends a value that corresponds to a specific question or message. If this value does not fit into the range of questions, then this error appears.

00101 CZUDE DVC FTL ERR 00101 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx  
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx  
DUP PROGRAM REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION

After the operator inputs the date/serial number, the DUP program will ask the host program for them. If for some reason the date/serial number was unacceptable to the DUP program, this error message will appear. Retry the program and if this error appears again, get out of the diagnostic runtime services and back to the XXDP\* prompt and reload the program.



### 3.2.2 DUP PROGRAM ERROR MESSAGES

Error messages returned by the UDA formatter are as follows:

#### GET STATUS failure

This could be caused by a number of reasons. Examples: the RUN/STOP switch is out, the WRITE PROTECT switch is in, or the DIAGNOSTIC REQUEST bit is set by the drive.

#### SDI send error

An attempt to send an SDI command failed. The signal RECEIVER READY was not asserted.

#### Unsuccessful SDI command

The response from an SDI command was unsuccessful and all commands should be successful for the formatter to work. There may be a cable problem, drive receiver problem or UDA transmitter problem.

#### SDI receive error

This message is presented for several reasons. The drive timed out, the first word from the drive was not a start frame, there was a framing error on the SDI level 0 read (cable/receiver/transmitter problem), checksum error, or the buffer size given by the formatter wasn't large enough for the UDA. Again, there may be a cable/receiver/transmitter problem.

#### UNIBUS read error

This is caused by one of two problems. While trying to read an overlay into the UDA buffer memory, the formatter came across a nonexistent memory error. Or, there was a failure while downline loading the bad block information. There may be something wrong with the UNIBUS or the UDA module M7485.

#### Formatter initialization error

For this error to occur, the UDA must be processing the DM code improperly.

#### Non-existent unit number

The desired disk drive wasn't attached to the UDA.

DBN/XBN format error (drive FORMAT command failed)

All attempts and retries to format a track failed. There may have been a timeout of drive signals, the drive dropped the READ/WRITE READY signal during the format operation or the drive clock timed out (which indicates cable/transmitter/receiver failures).

FCT does not have enough good copies of each block

There must at least two good copies of every block in the FCT. For this error to occur, the media is badly corrupted or the read/write logic is failing.

SEEK error

After a seek command completed successfully, the READ/WRITE READY signal was never set or the ATTENTION signal was set.

RCT does not have enough good copies of each block

There must be at least two good copies of every block in the RCT. For this error to occur, the media is badly corrupted or the read/write logic is failing.

LBN format error (drive FORMAT command failed)

All attempts and retries to format a track failed. There may have been a timeout of drive signals, the drive dropped the READ/WRITE READY signal during the format operation or the drive clock timed out (which indicates cable/transmitter/receiver failures).

FCT write error

A particular block failed to be written into every copy of the FCT. There is either terribly bad media or a write logic failure.

RCT read error

The formatter could not read at least one good copy of a particular block in the RCT area.

RCT write error

A particular block failed to be written into every copy of the RCT. There is either terribly bad media or a write logic failure.

**RCT full**

There were so many bad blocks on the media that the RCT area was filled and could not hold any more. There could be read/write logic failure or bad cable connection.

**FCT read error**

The formatter could not read at least one good copy of a particular block in the FCT area.

**FCT downline-load error**

The formatter was led to believe that a bad block information file was larger than it really was. There may be a UNIBUS or M7485 problem.

**Drive init timeout**

After the drive was initied, the RECEIVER READY signal never asserted.

**Illegal response to start-up question**

An overflow occurred when the serial number went over 64 bits.

**FCT corrupted - Format Invalid**

A problem was detected while using the data in the FCT. Either the data was not written properly or it has been corrupted since the last format. The format on the disk is no good and the disk will not be usable by any DEC operating system. Running the formatter again may have a slight chance of succeeding. Otherwise, replace the disk or HDA. If you do not have a spare disk or HDA you may try to format the disk in RECONSTRUCT mode. If the disk is not an R80, order a replacement disk or HDA immediately.

DRIVE ERROR ENCOUNTERED - STATUS RESPONSE:  
STATUS (R TO L): 1AF1 0304 E100 8B00 0080 0013 1000  
LAST BLOCK ACCESSED (16-BIT OCTAL): 000000 000000

The disk drive reported an error. You may see the drive's fault light come on. The formatter will attempt to clear the error in the drive and continue. This error does not mean that anything is necessarily wrong unless this error is printed many times. If you see many of these errors, you may wish to stop the format and run diagnostics on the disk drive. But remember, if you stop the formatter the disk will not be usable and the diagnostics will report the format is bad. The drive's status is presented in hexadecimal in the same format as the diagnostic programs. The last block accessed is a representation of the last block header written onto the disk.

#### MORE THAN 12.5% OF TRACK IS BAD

The formatter found more than one eighth of the blocks on a single track bad. This error does not mean that anything is necessarily wrong unless this error is printed many times. If you see many of these errors, you may wish to stop the format and run diagnostics on the disk drive. But remember, if you stop the formatter the disk will not be usable and the diagnostics will report the format is bad.

An example of how the errors are presented is below:

RUNTIME 0:00:18  
Non-existent unit number

#### 4.0 PERFORMANCE AND PROGRESS REPORTS

-----

There is no statistical report that can be printed using the Diagnostic Runtime Services PRINT command.

The DUP program issues the following messages upon normal completion:

Format completed

n Revectored LBNS

Where n is the number of LBNS revectored in the user data area.

n Primary revectored LBNS

Where n is the number of LBNS which were primary revector.

n Secondary/teritary revectored LBNS

Where n is the number of the LBNS which were secondary or tertiary revector.

n Bad RBNS

Where n is the number of RBNS which were bad.

n Bad blocks in the RCT area due to data errors

Where n is the number of blocks in the total RCT area which were bad.

n Bad blocks in the DBN area due to data errors

Where n is the number of blocks in the total DBN area which were bad.

n Bad blocks in the XBN area due to data errors

Where n is the number of blocks in the total XBN area which were bad.

n Blocks retried on the check pass

Where n is the number of blocks which had an error on the first read attempt after formatting.

FCT used successfully or  
FCT was not used

Depending on the answers to the software questions and the availability of the bad sector information (FCT), one of these messages will be printed.

An example of how the messages are presented is below.

```
RUNTIME 1:24:57
Format completed
  5 Revectored LBNS
  5 Primary revectored LBNS
  0 Secondary/tertiary revectored LBNS
  0 Bad RBNS
  0 Bad blocks in the RCT area due to data errors
  0 Bad blocks in the DBN area due to data errors
  0 Bad blocks in the XBN area due to data errors
  5 Blocks retried on the check pass
FCT was not used
```

## 5.0 TEST SUMMARIES

-----

There is only one test in this program - Test #1. Its only purpose is to load and run the format program in a UDA50.

```

1
25
26 002000
27
28
29
30
31
32 002000
33
34 002000
002000
002000 103
002001 132
002002 125
002003 104
002004 105
002005 000
002006 000
002007 000
002010
002010 104
002011
002011 060
002012 000001
002014 016040
002016 022620
002020 023006
002022 002130
002024 002144
002026 000124
002030 000000
002032 000000
002034 000001
002036 000000
002040 002124
002042 000340
002044 000000
002046 000000
002050 003
002051 003

```

```

.SBTTL PROGRAM HEADER
      BGNMOD
:
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:
      POINTER BGNSW, BGNSET, BGNSETUP
      HEADER CZUDE,D,0,7200.,1,PRI07

```

```

L$NAME::
      .ASCII /C/
      .ASCII /Z/
      .ASCII /U/
      .ASCII /D/
      .ASCII /E/
      .BYTE 0
      .BYTE 0
      .BYTE 0
L$REV::
      .ASCII /D/
L$DEPG::
      .ASCII /0/
L$UNIT::
      .WORD T$PTHV
L$TIML::
      .WORD 7200.
L$HPCP::
      .WORD L$HARD
L$SPCP::
      .WORD L$SOFT
L$HPTP::
      .WORD L$HW
L$SPTP::
      .WORD L$SW
L$LADP::
      .WORD L$LAST
L$STA::
      .WORD 0
L$CO::
      .WORD 0
L$DTYP::
      .WORD 1
L$APT::
      .WORD 0
L$DTP::
      .WORD L$DISPATCH
L$PRIO::
      .WORD PRI07
L$ENVI::
      .WORD 0
L$EXP1::
      .WORD 0
L$MREV::
      .BYTE C$REVISION
      .BYTE C$EDIT

```

```

002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 003456
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 000000
002072
002072 000000
002074
002074 000000
002076
002076 003500
002100
002100 104035
002102
002102 000000
002104
002104 021100
002106
002106 022116
002110
002110 022114
002112
002112 021072
002114
002114 000000
002116
002116 000000
002120
002120 000000

```

```

L$EF:: .WORD 0
          .WORD 0
L$SPC:: .WORD 0
L$DEVP:: .WORD L$DVTYP
L$REPP:: .WORD 0
L$EXP4:: .WORD 0
L$EXPS:: .WORD 0
L$AUT:: .WORD 0
L$DUT:: .WORD 0
L$LUN:: .WORD 0
L$DESP:: .WORD L$DESC
L$LOAD:: EMT E$LOAD
L$ETP:: .WORD 0
L$ICP:: .WORD L$INIT
L$CCP:: .WORD L$CLEAN
L$ACP:: .WORD L$AUTO
L$PRT:: .WORD L$PROT
L$TEST:: .WORD 0
L$DLY:: .WORD 0
L$HIME:: .WORD 0

```



1  
2  
3  
4  
5  
6  
7  
8  
9

.SBTTL DISPATCH TABLE

:::  
: THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.  
: IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.  
:--

DISPATCH 1

002122  
002122 000001  
002124  
002124 022130

.WORD 1  
L\$DISPATCH::  
.WORD T1

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17

.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.  
:--

002126		BGNHW	DFPTBL				
002126	000005					.WORD	L10000-L\$HW/2
002130						L\$HW::	
002130						DFPTBL::	
002130	172150	.WORD	172150				
002132	000154	.WORD	154				: UNIBUS ADDRESS
002134	000005	.WORD	5.				: VECTOR ADDRESS
002136	000077	.WORD	63.				: BR LEVEL
002140	000000	.WORD	0.				: UNIBUS BURST RATE
002142		ENDHW					: LOGICAL DRIVE NUMBER
002142							L10000:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16

.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.  
:--

002142  
002142 000001  
002144  
002144

BGNSW SFPTBL

.WORD L10001-L\$SW/2  
L\$SW::  
SFPTBL::

002144 000007  
002146  
002146

.WORD 7  
ENDSW

;OFFSET USE  
: 0. YES/NO ANSWERS

L10001:

002146

ENDMOD

1  
2  
3 002146  
4  
5  
6  
7  
8  
9  
10 002146

.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

EQUALS

; BIT DIFINITIONS

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

; BIT9== BIT09  
; BIT8== BIT08  
; BIT7== BIT07  
; BIT6== BIT06  
; BIT5== BIT05  
; BIT4== BIT04  
; BIT3== BIT03  
; BIT2== BIT02  
; BIT1== BIT01  
; BIT0== BIT00

; EVENT FLAG DEFINITIONS  
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START==	32.	; START COMMAND WAS ISSUED
000037	EF.RESTART==	31.	; RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE==	30.	; CONTINUE COMMAND WAS ISSUED
000035	EF.NEW==	29.	; A NEW PASS HAS BEEN STARTED
000034	EF.PWR==	28.	; A POWER-FAIL/POWER-UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

000340	PRI07==	340
000300	PRI06==	300
000240	PRI05==	240
000200	PRI04==	200

```

000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
;
; OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      JAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

```

```

11
12      000015

```

CR= 15

;VALUE TO PASS TO PRINT MACRO TO END LINE

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33

```

;MACRO DEFINITIONS FOR GLOBAL EQUATES
;THESE MACROS ARE USED TO DEFINE INDEXES INTO A TABLE
;CALLING SEQUENCE MUST BE
:
:   TABLE
:   ITEM   NAME   BYTES
:   ITEM   NAME   BYTES
:   ITEM   NAME   BYTES
:   END     SIZE
:
;TABLE DEFINES THAT A TABLE IS ABOUT TO BE DEFINED AND END TERMINATES THE DEFINITION.
;ANY NUMBER OF ITEM LINES CAN APPEAR. NAME IS THE NAME OF THE SYMBOL BEING EQUATED TO
;THE INDEX. THE INDEX ALWAYS STARTS AT ZERO. BYTES SPECIFIES THE SIZE OF THE VALUE TO BE
;STORED AT THAT INDEX IN BYTES. THE SIZE ARGUMENT TO THE END STATEMENT IS OPTIONAL, IT
;BE EQUATED TO THE SIZE OF THE TABLE IN BYTES. THE SYMBOL TINDEX IS USED TO KEEP TRACK
;OF THE INDEX VALUE AND WILL BE EQUAL TO THE SIZE OF THE TABLE AFTER THE END STATEMENT.

.MACRO TABLE
      TINDEX=0
.ENDM

.MACRO ITEM NAME BYTES
      NAME=TINDEX
      TINDEX=TINDEX*BYTES
.ENDM

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

```

```

1      ;UDA BIT DEFINITIONS
2
3      ;UDASA REGISTER UNIVERSAL READ BITS
4
5      004000      SA.S1= 004000      ;STEP 1 STATUS BIT
6      010000      SA.S2= 010000      ;STEP 2 STATUS BIT
7      020000      SA.S3= 020000      ;STEP 3 STATUS BIT
8      040000      SA.S4= 040000      ;STEP 4 STATUS BIT
9      100000      SA.ERR= 100000     ;ERROR INDICATOR
10
11     ;UDASA REGISTER ERROR STATUS BITS
12
13     003777      SA.ERC= 003777     ;ERROR CODE
14
15     ;UDASA REGISTER STEP ONE READ BITS
16
17     002000      SA.NV= 002000     ;NON SETTABLE INTERRUPT VECTOR
18     001000      SA.A2= 001000     ;22 BIT ADDRESS BUS
19     000400      SA.DI= 000400     ;ENHANCED DIAGNOSTICS
20     ;           ;           000377 ;ALL BITS RESERVED
21
22     ;UDASA REGISTER STEP ONE WRITE BITS
23
24     000177      SA.VEC= 000177     ;INTERRUPT VECTOR (DIVIDED BY 4)
25     000200      SA.INT= 000200     ;INTERRUPT ENABLE DURING INITIALIZATION
26     003400      SA.MSG= 003400     ;MESSAGE RING LENGTH
27     034000      SA.CMD= 034000     ;COMMAND RING LENGTH
28     040000      SA.WRP= 040000     ;WRAP BIT
29     100000      SA.STP= 100000     ;STEP - MUST ALWAYS BE WRITTEN A ONE
30
31     000400      SA.MS1= 000400     ;LSB OF MESSAGE RING LENGTH
32     004000      SA.CM1= 004000     ;LSB OF COMMAND RING LENGTH
33
34     ;UDASA REGISTER STEP TWO READ BITS
35
36     000007      SA.MSE= 000007     ;MESSAGE RING LENGTH ECHO
37     000070      SA.CME= 000070     ;COMMAND RING LENGTH ECHO
38     ;           ;           000100 ;RESERVED
39     000200      SA.STE= 000200     ;STEP ECHO
40     003400      SA.CTP= 003400     ;CONTROLLER TYPE
41
42     ;UDASA REGISTER STEP TWO WRITE BITS
43
44     000001      SA.PRG= 000001     ;ENABLE VAX UNIBUS ADAPTER PURGE INTERRUPT
45     ;           ;           177776 ;LOW ORDER MESSAGE RING BYTE ADDRESS

```

```

1      ;UDASA REGISTER STEP THREE READ BITS
2
3      000177      SA.VCE= 000177      ;INTERRUPT VECTOR ECHO
4      000200      SA.INE= 000200      ;INTERRUPT ENABLE ECHO
5      000400      SA.NVE= 000400      ;VECTOR NOT PROGRAMMABLE
6      ;          003000      ;RESERVED
7
8      ;UDASA REGISTER STEP THREE WRITE BITS
9
10     ;          077777      ;HIGH ORDER MESSAGE RING BYTE ADDRESS
11     100000      SA.TST= 100000      ;PURGE POLE TEST ENABLE
12
13     ;UDASA REGISTER STEP FOUR READ BITS
14
15     000017      SA.MCV= 000017      ;UDA MICROCODE VERSION
16     000360      SA.CNT= 000360      ;CONTROLLER MODEL
17     ;          003400      ;RESERVED
18
19     ;UDASA REGISTER STEP FOUR WRITE BITS
20
21     000001      SA.GO= 000001      ;GO BIT TO START UDA FIRMWARE
22     000002      SA.LFC= 000002      ;LAST FAILURE CODE REQUEST
23     000374      SA.BST= 000374      ;BURST LEVEL

```



```

1          ;COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS
2
3          100000      RG.OWN= 100000          ;SET WHEN UDA OWNS RING
4          040000      RG.FLG= 040000          ;FLAG BIT
5
6          ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
7          ;AND TWO PACKET AND BUFFER AREAS.
8
9          000004      HC.ISZ= 4.              ;SIZE OF INTERRUPT INDICATOR WORDS
10         000004      HC.RSZ= 4.              ;SIZE OF RING IN BYTES
11         000004      HC.ESZ= 4.              ;SIZE OF ENVELOPE WORDS BEFORE PACKET
12         000060      HC.PSZ= 48.             ;SIZE OF COMMAND AND MESSAGE PACKETS
13         000244      HC.BSZ= 164.            ;SIZE OF BUFFER
14
15         000000      HC.INT= 0.              ;INTERRUPT INDICATOR WORDS START
16         000004      HC.MSG= HC.INT+HC.ISZ   ;MESSAGE RING START
17         000006      HC.MCT= HC.MSG+2.       ;MESSAGE RING CONTROL WORD
18         000010      HC.CMD= HC.MSG+HC.RSZ   ;COMMAND RING START
19         000012      HC.CCT= HC.CMD+2.       ;COMMAND RING CONTROL WORDS
20         000014      HC.MEV= HC.CMD+HC.RSZ   ;MESSAGE ENVELOPE START
21         000020      HC.MPK= HC.MEV+HC.ESZ   ;MESSAGE PACKET START
22         000100      HC.CEV= HC.MPK+HC.PSZ   ;COMMAND ENVELOPE START
23         000104      HC.CPK= HC.CEV+HC.ESZ   ;COMMAND PACKET START
24         000164      HC.BF1= HC.CPK+HC.PSZ   ;FIRST BUFFER
25         000430      HC.BF2= HC.BF1+HC.BSZ   ;SECOND BUFFER
26
27         000674      HC.SIZ= HC.BF2+HC.BSZ   ;TOTAL SIZE OF HOST COMM AREA
28
29         ;VIRTUAL CIRCUIT IDENTIFIERS
30
31         000000      MSCP= 0                  ;MSCP CIRCUIT
32         000001      LOG= 1                  ;LOG CIRCUIT
33         177777      DIAG= -1                ;DIAGNOSTIC CIRCUIT
34         001000      DUP= 1000               ;DIAGNOSTIC AND UTILITIES PROTOCOL

```

1	:	HC.INT	:	INTERRUPT INDICATORS	:	4 BYTES
2	:		:		:	
3	:		:		:	
4	:		:		:	
5	:	HC.MSG	:	MESSAGE RING	:	4 BYTES
6	:	HC.MCT	:		:	
7	:		:		:	
8	:	HC.CMD	:	COMMAND RING	:	4 BYTES
9	:	HC.CCT	:		:	
10	:		:		:	
11	:	HC.MEV	:	MESSAGE ENVELOPE	:	52 BYTES
12	:	HC.MPK	:		:	
13	:		:		:	
14	:		:		:	
15	:		:		:	
16	:		:		:	
17	:		:		:	
18	:	HC.CEV	:	COMMAND ENVELOPE	:	52 BYTES
19	:	HC.CPK	:		:	
20	:		:		:	
21	:		:		:	
22	:		:		:	
23	:		:		:	
24	:		:		:	
25	:	HC.BF1	:	BUFFER # 1 (RESPONSE TO DM PROGRAM)	:	82 BYTES
26	:		:		:	
27	:		:		:	
28	:		:		:	
29	:		:		:	
30	:	HC.BF2	:	BUFFER # 2 (REQUEST FROM DM PROGRAM)	:	82 BYTES
31	:		:		:	
32	:		:		:	
33	:		:		:	
34	:		:		:	

```

1          ;COMMAND PACKET OPCODES
2
3          000001      OP.ABO= 1          ;ABORT COMMAND
4          000020      OP.ACC= 20         ;ACCESS COMMAND
5          000010      OP.AVL= 10         ;AVAILABLE COMMAND
6          000021      OP.CCD= 21         ;COMPARE CONTROLLER DATA COMMAND
7          000040      OP.CMP= 40         ;COMPARE HOST DATA COMMAND
8          000022      OP.ERS= 22         ;ERASE COMMAND
9          000023      OP.FLU= 23         ;FLUSH COMMAND
10         000002      OP.GCS= 2          ;GET COMMAND STATUS COMMAND
11         000003      OP.GUS= 3          ;GET UNIT STATUS COMMAND
12         000011      OP.ONL= 11         ;ONLINE COMMAND
13         000041      OP.RD= 41          ;READ COMMAND
14         000024      OP.RPL= 24         ;REPLACE COMMAND
15         000004      OP.SCC= 4          ;SET CONTROLLER CHARACTERISTICS COMMAND
16         000012      OP.SUC= 12         ;SET UNIT CHARACTERISTICS COMMAND
17         000042      OP.WR= 42          ;WRITE COMMAND
18         000030      OP.MRD= 30         ;MAINTENANCE READ COMMAND
19         000031      OP.MWR= 31         ;MAINTENANCE WRITE COMMAND
20         000200      OP.END= 200        ;END PACKET FLAG
21         000007      OP.SEX= 7          ;SERIOUS EXCEPTION END PACKET
22         000100      OP.AVA= 100        ;AVAILABLE ATTENTION MESSAGE
23         000101      OP.DUP= 101        ;DUPLICATE UNIT NUMBER ATTENTION MESSAGE
24         000102      OP.SMC= 102        ;SHADOW COPY COMPLETE ATTENTION MESSAGE
25         000103      OP.RLC= 103        ;RESET COMMAND LIMIT ATTENTION MESSAGE
26
27         000001      OP.GDS= 1          ;DUP GET DUST STATUS
28         000001      OP.GSS= 1          ;DUP GET DUST STATUS
29         000002      OP.ESP= 2          ;DUP EXECUTE SUPPLIED PROGRAM
30         000003      OP.ELP= 3          ;DUP EXECUTE LOCAL PROGRAM
31         000004      OP.SSD= 4          ;DUP SEND STUD DATA
32         000005      OP.RSD= 5          ;DUP RECEIVE STUD DATA
33
34         ;NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
35         ;PACKET FLAG TO THE COMMAND OPCODE. FOR EXAMPLE, A READ COMMAND'S END PACKET
36         ;CONTAINS THE VALUE OP.RD*OP.END IN ITS OPCODE FIELD. THE INVALID COMMAND END
37         ;PACKET CONTAINS JUST THE END PACKET FLAG (I.E., OP.END) IN ITS OPCODE FIELD.
38         ;THE SERIOUS EXCEPTION END PACKET CONTAINS THE SUM OF THE END PACKET FLAG
39         ;PLUS THE SERIOUS EXCEPTION OPCODE SHOWN ABOVE (I.E., OP.SEX*OP.END) IN ITS
40         ;OPCODE FIELD.
41         ;
42         ;COMMAND OPCODE BITS 3 THROUGH 5 INDICATE THE COMMAND CLASS, WHICH IS ENCODED
43         ;AS FOLLOWS:
44         ; 000 IMMEDIATE COMMANDS
45         ; 001 SEQUENTIAL COMMANDS
46         ; 010 NON-SEQUENTIAL COMMANDS THAT DO NOT INCLUDE A BUFFER DESCRIPTOR
47         ; 100 NON-SEQUENTIAL COMMANDS THAT DO INCLUDE A BUFFER DESCRIPTOR
    
```

```

1      ;COMMAND MODIFIERS
2
3      ;          = 020000
4      MD.CMP= 040000      ;CLEAR SERIOUS EXCEPTION
5      MD.EXP= 100000      ;COMPARE
6      MD.ERR= 010000      ;EXPRESS REQUEST
7      MD.SCH= 004000      ;FORCE ERROR
8      MD.SCL= 002000      ;SUPPRESS CACHING (HIGH SPEED)
9      MD.SEC= 000100      ;SUPPRESS CACHING (LOW SPEED)
10     MD.SER= 000400      ;SUPPRESS ERROR CORRECTION
11     MD.SSH= 000200      ;SUPPRESS ERROR RECOVERY
12     MD.WBN= 000100      ;SUPPRESS SHADOWING
13     MD.WBV= 000400      ;WRITE-BACK (NON-VOLATILE)
14     MD.SEQ= 000020      ;WRITE BACK (VOLATILE)
15     MD.SPD= 000001      ;WRITE SHADOW SET ONE UNIT AT A TIME
16     MD.FEU= 000001      ;SPIN-DOWN
17     MD.VOL= 000002      ;FLUSH ENTIRE UNIT
18     MD.NXU= 000001      ;VOLATILE ONLY
19     MD.RIP= 000001      ;NEXT UNIT
20     MD.IMF= 000002      ;ALLOW SELF DESTRUCTION
21     MD.SWP= 000004      ;IGNORE MEDIA FORMAT ERROR
22     MD.CWB= 000010      ;SET WRITE PROTECT
23     MD.PRI= 000001      ;CLEAR WRITE-BACK DATA LOST
24                                     ;PRIMARY REPLACEMENT BLOCK
25     ;END PACKET FLAGS
26
27     EF.BBR= 000200      ;BAD BLOCK REPORTED
28     EF.BBU= 000100      ;BAD BLOCK UNREPORTED
29     EF.LOG= 000040      ;ERROR LOG GENERATED
30     EF.SEX= 000020      ;SERIOUS EXCEPTION
31
32     ;CONTROLLER FLAGS
33
34     CF.ATN= 000200      ;ENABLE ATTENTION MESSAGES
35     CF.MSC= 000100      ;ENABLE MISCELLANEOUS ERROR LOG MESSAGES
36     CF.OTH= 000040      ;ENABLE OTHER HOST'S ERROR LOG MESSAGES
37     CF.THS= 000020      ;ENABLE THIS HOST'S ERROR LOG MESSAGES
38     CF.SHD= 000002      ;SHADOWING
39     CF.576= 000001      ;576 BYTE SECTORS

```



```

1          ;END PACKET OFFSETS
2
3          ;
4          000000      P.CRF = 0.          ;COMMAND REFERENCE NUMBER
5          000004      P.UNIT = 4.         ;UNIT NUMBER
6          000010      P.OPCD = 8.         ;OPCODE (ALSO CALLED ENDCODE)
7          000011      P.FLGS = 9.         ;END PACKET FLAGS
8          000012      P.STS = 10.        ;STATUS
9          000014      P.BCNT = 12.        ;BYTE COUNT
10         000034      P.FBBK = 28.        ;FIRST BAD BLOCK
11
12         ;
13         000014      P.OTRF = 12.        ;OUTSTANDING REFERENCE NUMBER
14         000020      P.CMST = 16.        ;COMMAND STATUS
15
16         ;
17         000014      P.MLUN = 12.        ;MULTI-UNIT CODE
18         000016      P.UNFL = 14.        ;UNIT FLAGS
19         000020      P.HSTI = 16.        ;HOST IDENTIFIER
20         000024      P.UNTI = 20.        ;UNIT IDENTIFIER
21         000034      P.MEDI = 28.        ;MEDIA TYPE IDENTIFIER
22         000040      P.SHUN = 32.        ;SHADOW UNIT
23         000042      P.SHST = 34.        ;SHADOW STATUS
24         000044      P.TRCK = 36.        ;TRACK SIZE
25         000046      P.GRP = 38.        ;GROUP SIZE
26         000050      P.CYL = 40.        ;CYLINDER SIZE
27         000054      P.RCTS = 44.        ;RCT TABLE SIZE
28         000056      P.RBNS = 46.        ;RBNS / TRACK
29         000057      P.RCTC = 47.        ;RCT COPIES
30
31         ;
32         ;
33         000014      P.MLUN = 12.        ;MULTI-UNIT CODE
34         000016      P.UNFL = 14.        ;UNIT FLAGS
35         000020      P.HSTI = 16.        ;HOST IDENTIFIER
36         000024      P.UNTI = 20.        ;UNIT IDENTIFIER
37         000034      P.MEDI = 28.        ;MEDIA TYPE IDENTIFIER
38         000040      P.SHUN = 32.        ;SHADOW UNIT
39         000042      P.SHST = 34.        ;SHADOW STATUS
40         000044      P.UNCL = 36.        ;UNIT COMMAND LIMIT
41         000050      P.UNSZ = 40.        ;UNIT SIZE
42         000054      P.VSER = 44.        ;VOLUME SERIAL NUMBER
43
44         ;
45         000014      P.VRSN = 12.        ;MSCP VERSION
46         000016      P.CNTF = 14.        ;CONTROLLER FLAGS
47         000020      P.CTMO = 16.        ;CONTROLLER TIMEOUT
48         000022      P.CNCL = 18.        ;CONTROLLER COMMAND LIMIT
49         000024      P.CNTI = 20.        ;CONTROLLER ID
50
51         ;
52         000014      P.DEXT = 12.        ;DUST PROGRAM EXTENSION
53         000017      P.DFLG = 15.        ;STATUS FLAGS
54         000020      P.DPI = 16.        ;PROGRESS INDICATOR
55         000024      P.DTO = 20.        ;TIMEOUT VALUE

```

```

1          ;STATUS AND EVENT CODE DEFINITIONS
2
3          000037      ST.MSK= 37          ;STATUS / EVENT CODE MASK
4          000040      ST.SUB= 40          ;SUB-CODE MULTIPLIER
5          000000      ST.SUC= 0           ;SUCCESS
6          000001      ST.CMD= 1           ;INVALID COMMAND
7          000002      ST.ABO= 2           ;COMMAND ABORTED
8          000003      ST.OFL= 3           ;UNIT-OFFLINE
9          000004      ST.AVL= 4           ;UNIT-AVAILABLE
10         000005      ST.MFE= 5           ;MEDIA FORMAT ERROR
11         000006      ST.WPR= 6           ;WRITE PROTECTED
12         000007      ST.CMP= 7           ;COMPARE ERROR
13         000010      ST.DAT= 10          ;DATA ERROR
14         000011      ST.HST= 11          ;HOST BUFFER ACCESS ERROR
15         000012      ST.CNT= 12          ;CONTROLLER ERROR
16         000013      ST.DRV= 13          ;DRIVE ERROR
17         000037      ST.DIA= 37          ;MESSAGE FROM AN INTERNAL DIAGNOSTIC
18
19         ;GET DUST STATUS FLAGS
20
21         000010      DF.ACT= 010         ;SET IF THIS DUST CURRENTLY ACTIVE
22         000004      DF.NES= 004         ;SET IF THIS DUST WILL NOT ACCEPT THE EXECUTE
23                                     ;SUPPLIED PROGRAM COMMAND
24         000002      DF.LCL= 002         ;SET IF THIS DUST HAS A LOCAL LOAD MEDIA FOR LOADING
25                                     ;DIAGNOSTICS AND OTHER UTILITIES
26         000001      DF.SA= 001         ;SET IF ANY PROGRAM EXECUTION UNDER THIS DUST
27                                     ;DISABLES THE OPERATION OF ALL OTHER SERVERS IN THE
28                                     ;SAME SYSTEM AS THE DUST
29
30         ;DUP MESSAGE TYPES
31
32         010000      DU.QUE = 10000      ;QUESTION
33         020000      DU.DFL = 20000      ;DEFAULT QUESTION
34         030000      DU.INF = 30000      ;INFORMATION
35         040000      DU.TER = 40000      ;TERMINATOR
36         050000      DU.FTL = 50000      ;FATAL ERROR
37         060000      DU.SPC = 60000      ;SPECIAL
38
39         170000      DU.TYP= 170000      ;MESSAGE TYPE FIELD
40
41         ;DM PROGRAM HEADER DEFINITIONS
42
43         000000      DMTRLN= 0           ;OFFSET TO SIZE OF PROGRAM NEEDING DOWNLINE LOAD
44         000004      DMOVRL= 4           ;OFFSET TO SIZE OF OVERLAY
45         000021      DMTMO= 21           ;TIMEOUT VALUE IN SECONDS (ONE BYTE)
46         000040      DMMAIN= 40          ;OFFSET TO FIRST WORD OF MAIN PROGRAM
47         001000      DMFRST= 1000        ;ADDRESS IN DM FILE CONTAINING FIRST BYTE OF HEADER

```

```

1      ;CONTROLLER TABLE DEFINITIONS
2      ;
3      ;ONE TABLE WILL BE SET UP BY INITIALIZE SECTION FOR EACH UDA SELECTED
4      ;FOR TESTING. TABLES ARE CONTIGUOUS. THE END OF THE TABLES IS
5      ;MARKED BY A WORD OF ZEROS.
6      ;
7      ;THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF CTABS.
8      ;THE NUMBER OF TABLES IS CONTAINED IN CTRLRS.
9
10     002146      TABLE      ;START A TABLE DEFINITION
11
12     002146      ITEM C.UADR    2      ;UNIBUS ADDRESS OF UDAIP REGISTER
13     002146      ITEM C.UNIT    2
14             000077          CT.UNT= 000077      ; LOGICAL UNIT NUMBER (FIRST)
15             100000          CT.AVL= BIT15        ; SET WHEN NOT AVAILABLE FOR TESTING
16     002146      ITEM C.VEC     2
17             000777          CT.VEC= 000777      ; VECTOR ADDRESS
18             007000          CT.BRL= 007000      ; BR LEVEL
19     002146      ITEM C.BST     2      ; BURST LEVEL
20     002146      ITEM C.JSR     2      ; INTERRUPT SERVICE ROUTINE FOR CONTROLLER
21     002146      ITEM C.JAD     2      ; THESE TWO WORDS LOADED WITH [JSR RO,UDASRV]
22     002146      ITEM C.FLG     2      ; FLAGS
23             000002          CT.RN= BIT1         ; DM PROGRAM RUNNING
24             000004          CT.CMD= BIT2        ; COMMAND ISSUED, WAITING FOR RESPONSE
25             000010          CT.MSG= BIT3        ; MESSAGE RESPONSE RECEIVED
26
27             000020          CT.REQ= BIT4        ; WHENEVER THIS BIT IS SET, CT.CMD IS CLEARED
28
29
30             000040          CT.STA= BIT5        ; BUFFER HAS BEEN GIVEN TO UDA FOR REQUEST
31             000100          CT.TM1= BIT6        ; SET WHENEVER READ STUD DATA COMMAND
32
33             000200          CT.TM2= BIT7        ; GIVEN TO UDA
34     002146      ITEM C.RING    2      ; GET DUST STATUS COMMAND HAS BEEN SENT
35     002146      ITEM C.DR0     2      ; ONE TIMEOUT PERIOD HAS EXPIRED BETWEEN SEND OR
36     002146      ITEM C.DR1     2      ; RECEIVE DATA RESPONSE
37     002146      ITEM C.DR2     2      ; SECOND TIMEOUT HAS EXPIRED
38     002146      ITEM C.DR3     2      ; RING BUFFER ADDRESS
39     002146      ITEM C.DR4     2      ; POINTER TO DRIVE TABLES
40     002146      ITEM C.DR5     2      ; IF ZERO, NO DRIVE TABLE EXISTS
41     002146      ITEM C.DR6     2
42     002146      ITEM C.DR7     2
43     002146      ITEM C.TO      2      ; TIMEOUT COUNTER
44     002146      ITEM C.TOM     2      ; (TWO WORDS)
45     002146      ITEM C.TOT     2      ; DUP PROGRAM TIMEOUT VALUE IN SECONDS
46     002146      ITEM C.PRI     4      ; DUP PROGRAM PROGRESS INDICATOR
47     002146      ITEM C.REF     2      ; COMMAND REFERENCE NUMBER
48
49     002146      END C.SIZE      ;SIZE OF CONTROLLER TABLE IN BYTES
    
```



```

1      ;DRIVE TABLE DEFINITIONS
2      ;
3      ;ONE DRIVE TABLE WILL BE SET UP BY THE INITIALIZE SECTION FOR EACH
4      ;DRIVE SELECTED FOR TESTING. EACH TABLE IS POINTED TO BY A
5      ;WORD IN THE CONTROLLER TABLE ON WHICH THE DRIVE EXISTS.
6
7 002146      TABLE      ;START A TABLE DEFINITION
8
9 002146      ITEM D.DRV    2      ;DRIVE NUMBER
10 002146     ITEM D.UNIT  2
11           DT.UNT= 000077      ; LOGICAL UNIT NUMBER OF DRIVE
12           DT.AVL= BIT15      ; SET WHEN NOT AVAILABLE FOR TESTING
13 002146     ITEM D.SERN  22.   ;DISK SERIAL NUMBER
14
15 002146     END D.SIZE      ;SIZE OF DRIVE TABLE IN BYTES

```

```

1      ;USEFUL INSTRUCTION DEFINITIONS
2
3      .MACRO AND ARG,ADR          ;LOGICAL AND INSTRUCTION
4      .LIST
5
6      .NLIST                      BIC #C<ARG>,ADR
7      .ENDM
8
9      .MACRO OR ARG,ADR          ;LOGICAL OR INSTRUCTION
10     .LIST
11
12     .NLIST                      BIS #ARG,ADR
13     .ENDM
14
15     .MACRO PUSH ARG            ;PUSH INSTRUCTION
16     .IRP X,<ARG>
17     .LIST
18
19     .NLIST                      MOV X,-(SP)
20     .ENDM
21     .ENDM
22
23     .MACRO POP ARG             ;POP INSTRUCTION
24     .IRP X,<ARG>
25     .LIST
26
27     .NLIST                      MOV (SP),X
28     .ENDM
29     .ENDM
30
31     .MACRO .BR ADR             ;A BRANCH TO THE NEXT LOCATION
32     .IF P2
33         .IF NE .-ADR
34             .ERROR ;ILLEGAL .BR TO ADR
35         .ENDC
36     .ENDC
37     .ENDM
38
39     .MACRO ASSUME FIRST CONDITION SECOND
40         .IF CONDITION <FIRST>-<SECOND>
41         .IFF
42             .ERROR ;BAD ASSUME OF <FIRST> CONDITION <SECOND>
43         .ENDC
44     .ENDM

```



```
1      ;PRINT FORMATTED MESSAGE MACROS
2      ; USE THESE MACROS TO PRINT A FORMATTED MESSAGE
3      ; FIRST ARGUMENT MUST BE ADDRESS OF FIRST CHARACTER OF MESSAGE STRING
4      ; TO BE PUT INTO WORD (.WORD ARG)
5      ; UP TO 8 SOURCE STATEMENTS MAY FOLLOW TO SPECIFY PARAMETERS TO BE
6      ; USED BY THE FORMAT
7
8      .MACRO PNTF ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
9          PNT... LPNTF ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
10     .ENDM
11     .MACRO PNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
12         PNT... LPNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
13     .ENDM
14     .MACRO PNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
15         PNT... LPNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
16     .ENDM
17     .MACRO PNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
18         PNT... LPNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
19     .ENDM
20     .MACRO PNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
21         PNT... LPNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
22     .ENDM
```

```

1      .SBTTL GLOBAL DATA SECTION
2
3      ;**
4      ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
5      ; IN MORE THAN ONE TEST.
6      ;--
7
8 002146 FFREE:: .BLKW 1 ;FIRST FREE WORD IN MEMORY
9 002150 FSIZE:: .BLKW 1 ;SIZE OF FREE MEMORY IN WORDS
10 002152 FMEM: .BLKW 1 ;COPY OF FFREE AT END OF INIT SECTION
11 002154 FMEMS: .BLKW 1 ;COPY OF FSIZE AT END OF INIT SECTION
12 002156 CTABS:: .BLKW 1 ;START OF CONTROLLER TABLE STORAGE
13 002160 CTRLRS: .BLKW 1 ;COUNT OF UDA CONTROLLERS IN PTABLES
14 002162 TSTTAB: .BLKW 1 ;POINTER TO FIRST CONTROLLER TABLE UNDER TEST
15      .GLOBL UDAFM
16 002164 000000G DMPROG: .WORD UDAFM ;START ADDRESS OF DM PROGRAM
17 002166 URUN: .BLKW 1 ;NUMBER OF UNITS TO RUN AT ONE TIME
18 002170 URNING: .BLKW 1 ;NUMBER OF UNITS STILL RUNNING
19 002172 UCNT: .BLKW 1 ;COUNTER OF UNITS UNDER TEST
20 002174 000000 FILOPN: .WORD 0 ;FILE OPEN
21 002176 UFREEZ: .BLKW 1 ;FREEZE ON UNIT WHEN NOT ZERO
22 002200 NXMAD: .BLKW 1 ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
23 002202 000000 FDATA: .WORD 0
24 002204 FCTBUF: .BLKB 512. ;STORAGE FOR FCT BLOCK
25 003204 FCTNUM: .BLKW 1 ;FCT BLOCK NUMBER
26 003206 MODE: .BLKW 1 ;MODE WORD, SAME BIT DEFS AS SO.BIT
27
28      ;CLOCK CONTROL
29
30 003210 000000 KW.CSR: .WORD 0 ;CSR OF CLOCK
31 003212 KW.BRL: .BLKW 1 ;BR LEVEL
32 003214 KW.VEC: .BLKW 1 ;VECTOR
33 003216 KW.HZ: .BLKW 1 ;HERTZ (50. OR 60.)
34 003220 KW.EL: .BLKW 2 ;ELAPSED TIME
35
36 003224 016104 PTYPE: .WORD PF ;PRINT TYPE
37 003226 000 000 ERRCHR: .BYTE 0,0 ;FIRST BYTE LOADED WITH OUTPUT CHARACTER
38 003230 000000 NULL: .WORD 0 ;USED TO PRINT A NULL CHARACTER
39 003232 FNAME: .BLKB 10.

```

```

1 003244          TEMP: .BLKB 22.          ;USED TO GET ANSWER FROM GMANID CALL
2 003272          061    055    112 DATEI: .ASCIZ\1-JAN-70\      ;DEFAULT DATE
3 003303          .BLKB 3
4 003306          000000 DATED: .WORD 0 ;DATE STRING IN FORMATTER FORMAT
5 003310          .BLKB 10.          ;(FIRST WORD ZERO SAYS NO DATE HERE YET)
6 003322          061    070    064 HIGHEST: .ASCIZ\18446744073709551615\ ;HIGHEST DISK SERIAL NUMBER
7 003347          104    105    103 MONTHS: .ASCII\DEC\          ;NAME OF MONTHS
8 003352          116    117    126 .ASCII\NOV\
9 003355          117    103    124 .ASCII\OCT\
10 003360         123    105    120 .ASCII\SEP\
11 003363         101    125    107 .ASCII\AUG\
12 003366         112    125    114 .ASCII\JUL\
13 003371         112    125    116 .ASCII\JUN\
14 003374         115    101    131 .ASCII\MAY\
15 003377         101    120    122 .ASCII\APR\
16 003402         115    101    122 .ASCII\MAR\
17 003405         106    105    102 .ASCII\FEB\
18 003410         112    101    116 .ASCII\JAN\
19 003413         037          DAYS: .BYTE 31.          ;NUMBER OF DAYS IN EACH MONTH
20 003414         035          .BYTE 29.
21 003415         037          .BYTE 31.
22 003416         036          .BYTE 30.
23 003417         037          .BYTE 31.
24 003420         036          .BYTE 30.
25 003421         037          .BYTE 31.
26 003422         037          .BYTE 31.
27 003423         036          .BYTE 30.
28 003424         037          .BYTE 31.
29 003425         036          .BYTE 30.
30 003426         037          .BYTE 31.
31 003427         061    071    000 YEAR19: .ASCIZ\19\
32 003432         062    060    000 YEAR20: .ASCIZ\20\
33          .EVEN
34 003436         000000 IPADRS: .WORD 0
35 003440         000000 .WORD 0
36 003442         000000 .WORD 0
37 003444         000000 .WORD 0
38 003446         000000 .WORD 0
39 003450         000000 .WORD 0
40 003452         000000 .WORD 0
41 003454         000000 .WORD 0

```

```

1      .SBTTL GLOBAL TEXT SECTION
2
3
4      ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5      ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
6      ; MORE THAN ONE TEST.
7
8
9
10     ; NAMES OF DEVICES SUPPORTED BY PROGRAM
11
12     DEVTYP <UDA-50 CONTROLLER>
13
14     ; TEST DESCRIPTION
15
16     DESCRIPT <CZUDED0 PDP-11 UDA DRV FMTR>
17
18     003456      125      104      101
19     003456
20     003456
21
22     003500      103      132      125
23     003500
24     003500
25     DRV FMTR/
26
27     L$DVTYP::
28     .ASCIZ /UDA-50 CONTROLLER/
29     .EVEN
30
31     L$DESC::
32     .ASCIZ /CZUDED0 PDP-11 UDA
33     .EVEN

```

```
1  
2  
3 003536      105      116      124 DATEQ: .ASCIZ\ENTER DATE AS DD-MMM-YY\  
4 003566      040      106      117 FILNAQ: .ASCIZ\ FOR DISK TO BE FORMATTED\  
5 003620      040      000      SERNG: .ASCIZ\ \  
6 003622      101      122      105 WNQUES: .ASCIZ\ARE YOU SURE YOU WANT TO RUN THIS FORMATTER\  

```



```

1          ; FORMAT STATEMENTS USED IN PRINT CALLS
2
3 003676   045   124   000  ERRONE: .ASCIZ\BT\
4 003701   045   116   000  ERRNL: .ASCIZ\BN\
5 003704   042   040   040  RNTIM: .ASCIZ\"  RUNTIME "D16";"\
6 003727   104   071   042  RNTIM1: .ASCIZ\D9";"\
7 003735   104   071   000  RNTIM2: .ASCIZ\D9\
8 003740   042   040   040  ERRME1: .ASCIZ\" * * * ERROR PROCESSING MESSAGE STRING * * *"\
9 004027   116   042   125  MESSG: .ASCIZ\N"UNIT "D6" UDA AT "D16" DRIVE "D9S\
10 004073  042   116   117  NOCLOCK: .ASCIZ\N"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\
11 004150  042   110   117  BASNO: .ASCIZ\N"HOST PROGRAM"\
12 004167  042   040   040  BASL2: .ASCIZ\"  UDA AT "D16\
13 004206  042   040   040  BASL3: .ASCIZ\"  DRIVE "D9\
14 004223  000                   BAS: .BYTE 0          ;NULL TO PRINT NOTHING
15
16 004224  122   066   122  BASLN: .ASCIZ\R6R6R6R6\          ;USED TO PRINT BASIC LINE OF ERROR MESSAGE
17 004235  116   042   123  SERNUM: .ASCIZ\N"SERIAL NUMBER FOR UNIT "D6" UDA AT "D16" DRIVE "D9\
18 004322  042   123   124  WNSTOP: .ASCIZ\N"STOPPING THIS FORMATTER AFTER THIS POINT WILL MAKE THE DISK"N\
19 004415  042   125   116  .ASCIZ\N"UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN"N\
20 004506  042   102   122  .ASCIZ\N"BROUGHT ONLINE."MN\
21 004532  116   042   127  WNSTRT: .ASCIZ\N"WARNING:"N\
22 004546  042   040   040  .ASCIZ\"  THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC"N\
23 004650  042   040   040  .ASCIZ\"  TOOL.  RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK"N\
24 004747  042   040   040  .ASCIZ\"  DRIVE'S SERVICE MANUAL."N\

```

1	005010				X1A:	
2	005010				X2A:	
3	005010				X3A:	
4	005010	042	111	040	X8A:	.ASCIZ\ "I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS" N\
5	005107	122	065	122	X1:	.ASCIZ\R5R6 "UDA HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE" N\
6	005203	122	065	122	X2:	.ASCIZ\R5R6 "TWO UNITS SELECT THE SAME DRIVE" N\
7	005252	122	065	122	X3:	.ASCIZ\R5R6 "MORE THAN EIGHT DRIVES SELECTED ON THIS UDA" N\
8	005335	122	064	042	X4:	.ASCII\R4 "NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED" N\
9	005426	042	120	114		.ASCIZ\ "PLEASE START PROGRAM OVER AND TEST FEWER UNITS AT A TIME" N\
10	005522	122	065	122	X8:	.ASCIZ\R5R6 "TWO UDA'S USE THE SAME VECTOR" N\
11	005567	122	064	042	X9:	.ASCII\R4 "ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE." N\
12	005672	042	120	114		.ASCIZ\ "PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK." N\
13	005761	122	064	042	X10:	.ASCIZ\R4 "THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE." N\
14	006060	122	065	042	X14:	.ASCII\R5 "UDASO CONTROLLER IS AT A REVISION LEVEL NO LONGER SUPPORTED BY" N\
15	006163	042	124	110		.ASCII\ "THIS FORMATTER PROGRAM. THIS PROGRAM REQUIRES A UDASO-A" N\
16	006256	042	103	117		.ASCII\ "CONTROLLER (MODEL 6) WITH MICROCODE VERSION AT 3 OR GREATER." N\
17	006356	042	103	117		.ASCIZ\ "CONTROLLER REPORTED MODEL CODE "D4" AND MICROCODE VERSION "D4"." N\
18	006461	122	065	042	X20:	.ASCII\R5 "MEMORY ERROR TRYING TO READ UDA REGISTERS" N\
19	006537	042	103	110		.ASCII\ "CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7485" N\
20	006625	042	117	122		.ASCII\ "OR UNIBUS" N\
21	006641	042	117	122		.ASCIZ\ "OR "R7\
22	006651	122	065	042	X21:	.ASCII\R5 "UDA RESIDENT DIAGNOSTICS DETECTED FAILURE" NR8\
23	006731	042	122	105		.ASCIZ\ "REPLACE UDA MODULE M748" O3N\
24	006766	122	065	042	X22:	.ASCII\R5 "STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION" N\
25	007067	042	123	124		.ASCIZ\ "STEP BIT EXPECTED "016NR8R7\
26	007124	122	065	042	X23A:	.ASCII\R5 "UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION" N\
27	007236	104	071	042		.ASCII\D9 " WORDS WERE TO BE CLEARED STARTING AT ADDRESS "016N\
28	007324	042	106	111		.ASCII\ "FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):" N\
29	007401	123	066	042		.ASCIZ\S6 "ADDRESS" S4 "CONTENTS" N\
30	007432	123	067	117	X23B:	.ASCIZ\S7016S5016N\
31	007446	122	065	042	X24:	.ASCII\R5 "UDASA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION" N\
32	007561	042	120	125		.ASCIZ\ "PURGE/POLE DIAGNOSTICS WERE REQUESTED" NR8R7\
33	007636	122	065	042	X25:	.ASCII\R5 "UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION" N\
34	007752	042	040	040		.ASCIZ\ " UDASA EXPECTED "016NR8R7\
35	010007	122	065	042	X30:	.ASCIZ\R5 "UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE RUNNING DM PROGRAM" NR8\
36	010122	122	065	042	X31:	.ASCIZ\R5 "DUP PROGRAM IS HUNG" N\
37	010153	122	065	042	X32:	.ASCIZ\R5 "MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER" N\
38	010264	122	065	042	X36:	.ASCII\R5 "NO INTERRUPT RECEIVED FROM UDA FOR 30 SECONDS" N\
39	010346	042	127	110		.ASCIZ\ "WHILE LOADING DM PROGRAM" N\
40	010402	122	065	042	X37:	.ASCIZ\R5 "UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM" NR8R7\
41	010517	122	065	042	X100:	.ASCIZ\R5 "DUP PROGRAM ASKED UNEXPECTED QUESTION ("D12")" N\
42	010602	122	065	042	X101:	.ASCIZ\R5 "DUP PROGRAM REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION" N\

1	010705	042	115	105	XMSG1:	.ASCIZ\ "MESSAGE BUFFER CONTAINS;"N\
2	010741	123	063	117	XMSG2:	.ASCIZ\S3016S1016S1016S1016S1016S1016S1016N\
3	011006	122	065	042	XPKT1:	.ASCII\R5"RESPONSE PACKET FROM UDA DOES NOT CONTAIN EXPECTED DATA"N\
4	011102	042	105	111		.ASCII\ "EITHER UDA RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED CORRECTLY"N\
5	011212	123	063	042		.ASCIZ\S3"COMMAND PACKET SENT"56"RESPONSE PACKET RECEIVED"N\
6	011277	123	066	117	XPKT2:	.ASCIZ\S6016S1016S14016S1016N\
7	011326	042	040	040	XSA:	.ASCIZ\ " UDASA CONTAINS "016N\
8	011357	042	122	105	XFRU:	.ASCIZ\ "REPLACE UDA MODULE M7485"N\
9						
10						
11	011413	045	101	111	SERNX:	.ASCIZ\MAINPUT ERROR. ANSWER WITH DECIMAL NUMBER LO= 0 HI= #T\
12	011503	042	111	116	DATEX:	.ASCIZ\ "INPUT ERROR."N\
13	011522	042	116	101	FILNAM:	.ASCIZ\ "NAME OF FILE CONTAINING BAD SECTOR INFORMATION"N\
14						.EVEN

```

1      .SBTTL GLOBAL ERROR REPORT SECTION
2
3      ;**
4      ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
5      ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PRINTB
6      ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
7      ;--
8      177777      SVCINS= -1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
9      177777      SVCTST= -1     ; LIST TEST TAGS, SHIFTED RIGHT
10     177777      SVCSUB= -1     ; LIST SUBTEST TAGS, SHIFTED RIGHT
11     177777      SVCGBL= -1    ; LIST GLOBAL TAGS, SHIFTED RIGHT
12     177777      SVCTAG= -1    ; LIST OTHER TAGS, SHIFTED RIGHT
13
14     011604      BGNMSG ERRO01
15     011604      PNTB X1,@X1A
16     011604      012746      005010      MOV @X1A,-(SP)
17     011610      004137      016236      JSR R1,LPNTB
18     011614      005107
19     011616      000002      .WORD X1
20     011620      ENDMSG      .WORD PNT.CT
21
22     011622      BGNMSG ERRO02
23     011622      PNTB X2,@X2A
24     011622      012746      005010      MOV @X2A,-(SP)
25     011626      004137      016236      JSR R1,LPNTB
26     011632      005203
27     011634      000002      .WORD X2
28     011636      ENDMSG      .WORD PNT.CT
29
30     011640      BGNMSG ERRO03
31     011640      PNTB X3,@X3A
32     011640      012746      005010      MOV @X3A,-(SP)
33     011644      004137      016236      JSR R1,LPNTB
34     011650      005252
35     011652      000002      .WORD X3
36     011654      ENDMSG      .WORD PNT.CT
37
38     011656      BGNMSG ERRO04
39     011656      PNTB X4
40     011656      004137      016236      JSR R1,LPNTB
41     011662      005335
42     011664      000000      .WORD X4
43     011666      ENDMSG      .WORD PNT.CT
44
45     011670      BGNMSG ERRO08
46     011670      PNTB X8,@X8A
47     011670      012746      005010      MOV @X8A,-(SP)
48     011674      004137      016236      JSR R1,LPNTB
49     011700      005522
50     011702      000002      .WORD X8
51     011704      ENDMSG      .WORD PNT.CT
52
53     011706      BGNMSG ERRO09
54     011706      PNTB X9
55     011706      004137      016236      JSR R1,LPNTB
56     011712      005567
57     011714      000000      .WORD X9
58     .WORD PNT.CT

```

36	011716			ENDMSG	
37					
38	011720			BGNMSG ERR010	
39	011720			PNTB X10	
	011720	004137	016236		JSR R1,LPNTB
	011724	005761			.WORD X10
	011726	000000			.WORD PNT.CT
40	011730			ENDMSG	
41					
42	011732			BGNMSG ERR014	
43	011732			PNTB X14,R3,R2	
	011732	010246			MOV R2,-(SP)
	011734	010346			MOV R3,-(SP)
	011736	004137	016236		JSR R1,LPNTB
	011742	006060			.WORD X14
	011744	000004			.WORD PNT.CT
44	011746			ENDMSG	
45					
46	011750			BGNMSG ERR020	
47	011750			PNTB X20	
	011750	004137	016236		JSR R1,LPNTB
	011754	006461			.WORD X20
	011756	000000			.WORD PNT.CT
48	011760			ENDMSG	
49					
50	011762			BGNMSG ERR021	
51	011762	010201		MOV R2,R1	
52	011764	000301		SWAB R1	
53	011766			AND 2,R1	
	011766	042701	177775		BIC #C<2>,R1
54	011772	006201		ASR R1	
55	011774	062701	000005	ADD #5,R1	
56	012000			PNTB X21,R2,R1	
	012000	010146			MOV R1,-(SP)
	012002	010246			MOV R2,-(SP)
	012004	004137	016236		JSR R1,LPNTB
	012010	006651			.WORD X21
	012012	000004			.WORD PNT.CT
57	012014			ENDMSG	
58					
59	012016			BGNMSG ERR022	
60	012016	042737	100000 020216	BIC #SA.ERR,UDARSD	
61	012024			PNTB X22,UDARSD,R2	
	012024	010246			MOV R2,-(SP)
	012026	013746	020216		MOV UDARSD,-(SP)
	012032	004137	016236		JSR R1,LPNTB
	012036	006766			.WORD X22
	012040	000004			.WORD PNT.CT
62	012042			ENDMSG	
63					
64	012044			BGNMSG ERR023	
65	012044			PNTB X23A,R1,FFREE	
	012044	013746	002146		MOV FFREE,-(SP)
	012050	010146			MOV R1,-(SP)
	012052	004137	016236		JSR R1,LPNTB
	012056	007124			.WORD X23A
	012060	000004			.WORD PNT.CT

66	012062	005742		TST -(R2)	
67	012064	005712		ERR23A: TST (R2)	
68	012066	001410		BEQ ERR23B	
69	012070			PNTB X23B,R2,(R2)	
	012070	011246			MOV (R2),-(SP)
	012072	010246			MOV R2,-(SP)
	012074	004137	016236		JSR R1,LPNTB
	012100	007432			.WORD X23B
	012102	000004			.WORD PNT,CT
70	012104	005304		DEC R4	
71	012106	001403		BEQ ERR23C	
72	012110	005722		ERR23B: TST (R2),	
73	012112	005303		DEC R3	
74	012114	001363		BNE ERR23A	
75	012116			ERR23C: PNTB XFRU	
	012116	004137	016236		JSR R1,LPNTB
	012122	011357			.WORD XFRU
	012124	000000			.WORD PNT,CT
76	012126			ENDMSG	
77					
78	012130			BGNMSG ERR024	
79	012130			PNTB X24,R2	
	012130	010246			MOV R2,-(SP)
	012132	004137	016236		JSR R1,LPNTB
	012136	007446			.WORD X24
	012140	000002			.WORD PNT,CT
80	012142			ENDMSG	
81					
82	012144			BGNMSG ERR025	
83	012144			PNTB X25,R1,R2	
	012144	010246			MOV R2,-(SP)
	012146	010146			MOV R1,-(SP)
	012150	004137	016236		JSR R1,LPNTB
	012154	007636			.WORD X25
	012156	000004			.WORD PNT,CT
84	012160			ENDMSG	
85					
86	012162			BGNMSG ERR030	
87	012162			PNTB X30,R1	
	012162	010146			MOV R1,-(SP)
	012164	004137	016236		JSR R1,LPNTB
	012170	010007			.WORD X30
	012172	000002			.WORD PNT,CT
88	012174			ENDMSG	
89					
90	012176			BGNMSG ERR031	
91	012176			PNTB X31	
	012176	004137	016236		JSR R1,LPNTB
	012202	010122			.WORD X31
	012204	000000			.WORD PNT,CT
92	012206			ENDMSG	
93					
94	012210			BGNMSG ERR032	
95	012210			PNTB X32	
	012210	004137	016236		JSR R1,LPNTB
	012214	010153			.WORD X32
	012216	000000			.WORD PNT,CT

96	012220	004737	012410		CALL MSGPKT	
97	012224			ENDMSG		
98						
99	012226			BGNMSG	ERR033	
100	012226	004737	012316		CALL PNTPKT	
101	012232			ENDMSG		
102						
103	012234			BGNMSG	ERR034	
104	012234	004737	012316		CALL PNTPKT	
105	012240			ENDMSG		
106						
107	012242			BGNMSG	ERR036	
108	012242				PNTB X36	
	012242	004137	016236			JSR R1,LPNTB
	012246	010264				.WORD X36
	012250	000000				.WORD PNT.CT
109	012252			ENDMSG		
110						
111	012254			BGNMSG	ERR037	
112	012254				PNTB X37,R1	
	012254	010146				MOV R1,-(SP)
	012256	004137	016236			JSR R1,LPNTB
	012262	010402				.WORD X37
	012264	000002				.WORD PNT.CT
113	012266			ENDMSG		
114						
115	012270			BGNMSG	ERR100	
116	012270				PNTB X100,(R4)	
	012270	011446				MOV (R4),-(SP)
	012272	004137	016236			JSR R1,LPNTB
	012276	010517				.WORD X100
	012300	000002				.WORD PNT.CT
117	012302			ENDMSG		
118						
119	012304			BGNMSG	ERR101	
120	012304				PNTB X101	
	012304	004137	016236			JSR R1,LPNTB
	012310	010602				.WORD X101
	012312	000000				.WORD PNT.CT
121	012314			ENDMSG		
122						
123	012316			PNTPKT:	PNTB XPKT1	
	012316	004137	016236			JSR R1,LPNTB
	012322	011006				.WORD XPKT1
	012324	000000				.WORD PNT.CT
124	012326	010401			MOV R4,R1	
125	012330	062701	000104		ADD #HC.CPK,R1	
126	012334	010402			MOV R4,R2	
127	012336	062702	000020		ADD #HC.MPK,R2	
128	012342	012703	000014		MOV #12.,R3	
129	012346			PNTPKL:	PNTB XPKT2,2(R1),(R1),2(R2),(R2)	
	012346	011246				MOV (R2),-(SP)
	012350	016246	000002			MOV 2(R2),-(SP)
	012354	011146				MOV (R1),-(SP)
	012356	016146	000002			MOV 2(R1),-(SP)
	012362	004137	016236			JSR R1,LPNTB
	012366	011277				.WORD XPKT2

	012370	000010				.WORD PNT.CT
130	012372	062701	000004	ADD #4,R1		
131	012376	062702	000004	ADD #4,R2		
132	012402	005303		DEC R3		
133	012404	001360		BNE PNTPKL		
134	012406	000207		RETURN		
135						
136	012410			MSGPKT: PNTB XMSG1		
	012410	004137	016236			JSR R1,LPNTB
	012414	010705				.WORD XMSG1
	012416	000000				.WORD PNT.CT
137	012420	016504	000016	MOV C.RING(R5),R4		
138	012424	062704	000430	ADD #HC.BF2,R4		
139	012430	012703	000005	MOV #5,R3		
140	012434			MSGPKL: PNTB XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)		
	012434	016446	000014			MOV 12.(R4),-(SP)
	012440	016446	000012			MOV 10.(R4),-(SP)
	012444	016446	000010			MOV 8.(R4),-(SP)
	012450	016446	000006			MOV 6(R4),-(SP)
	012454	016446	000004			MOV 4(R4),-(SP)
	012460	016446	000002			MOV 2(R4),-(SP)
	012464	011446				MOV (R4),-(SP)
	012466	004137	016236			JSR R1,LPNTB
	012472	010741				.WORD XMSG2
	012474	000016				.WORD PNT.CT
141	012476	062704	000016	ADD #14.,R4		
142	012502	005303		DEC R3		
143	012504	001353		BNE MSGPKL		
144	012506	000207		RETURN		



1	000001	SVCINS= 1	; LIST INSTRUCTIONS, SHIFTED RIGHT
2	000001	SVCTST= 1	; LIST TEST TAGS, SHIFTED RIGHT
3	000001	SVCSUB= 1	; LIST SUBTEST TAGS, SHIFTED RIGHT
4	000001	SVCGBL= 1	; LIST GLOBAL TAGS, SHIFTED RIGHT
5	000001	SVCTAG= 1	; LIST OTHER TAGS, SHIFTED RIGHT

```

1      .SBTTL GLOBAL SUBROUTINES SECTION
2
3      ;MEMORY ALLOCATION ERROR
4      ;
5      ;THIS ROUTINE PRINTS A SYSTEM FATAL ERROR AND EXITS THE TEST
6
7      FMERR:  ERRSF 4,,ERR004
           DOCLN                                ;ABORT
           TRAP  C$ERSF
           .WORD 4
           .WORD 0
           .WORD ERR004
8      012510 104454
           012512 000004
           012514 000000
           012516 011656
           012520 104444
           TRAP  C$DCLN

```

```

1      ;ALOCM
2      ;
3      ;ALLOCATE A BLOCK OF FREE MEMORY.  REPORT ERROR IF MEMORY EXHAUSTED.
4      ;
5      ;INPUTS:
6      ;      R1 - NUMBER OF WORDS TO ALLOCATE
7      ;      FFREE - FIRST FREE WORD IN MEMORY
8      ;      FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS
9      ;
10     ;OUTPUTS:
11     ;      R1 - ADDRESS OF FIRST WORD OF ALLOCATED MEMORY
12     ;      FFREE - NEW FIRST FREE WORD IN MEMORY
13     ;      FSIZE - SIZE OF FREE MEMORY LEFT AFTER ALLOCATION
14     ;SYSTEM FATAL ERROR WILL BE REPORTED IF NOT ENOUGH MEMORY AVAILABLE
15     ;AND ENTIRE PROGRAM WILL BE STOPPED.
16     ALOCM:  PUSH FFREE                ;SAVE FFREE AT ENTRY
17     ;                                MOV FFREE, -(SP)
18     ;                                ;REDUCE SIZE OF FREE MEMORY
19     ;                                ;REPORT ERROR IF NOT ENOUGH MEMORY
20     ;                                ;CHANGE WORDS TO BYTES
21     ;                                ;CALCULATE NEW START OF FREE MEMORY
22     ;                                ;GET START OF ALLOCATED MEMORY
23     ;                                MOV (SP)+,R1
24     ;
25     ;
26     ;
27     ;
28     ;
29     ;
30     ;
31     ;
32     ;
33     ;
34     ;
35     ;
36     ;
37     ;
38     ;
39     ;
40     ;
41     ;
42     ;
43     ;
44     ;
45     ;
46     ;
47     ;
48     ;
49     ;
50     ;
51     ;
52     ;
53     ;
54     ;
55     ;
56     ;
57     ;
58     ;
59     ;
60     ;
61     ;
62     ;
63     ;
64     ;
65     ;
66     ;
67     ;
68     ;
69     ;
70     ;
71     ;
72     ;
73     ;
74     ;
75     ;
76     ;
77     ;
78     ;
79     ;
80     ;
81     ;
82     ;
83     ;
84     ;
85     ;
86     ;
87     ;
88     ;
89     ;
90     ;
91     ;
92     ;
93     ;
94     ;
95     ;
96     ;
97     ;
98     ;
99     ;
100    ;
101    ;
102    ;
103    ;
104    ;
105    ;
106    ;
107    ;
108    ;
109    ;
110    ;
111    ;
112    ;
113    ;
114    ;
115    ;
116    ;
117    ;
118    ;
119    ;
120    ;
121    ;
122    ;
123    ;
124    ;
125    ;
126    ;
127    ;
128    ;
129    ;
130    ;
131    ;
132    ;
133    ;
134    ;
135    ;
136    ;
137    ;
138    ;
139    ;
140    ;
141    ;
142    ;
143    ;
144    ;
145    ;
146    ;
147    ;
148    ;
149    ;
150    ;
151    ;
152    ;
153    ;
154    ;
155    ;
156    ;
157    ;
158    ;
159    ;
160    ;
161    ;
162    ;
163    ;
164    ;
165    ;
166    ;
167    ;
168    ;
169    ;
170    ;
171    ;
172    ;
173    ;
174    ;
175    ;
176    ;
177    ;
178    ;
179    ;
180    ;
181    ;
182    ;
183    ;
184    ;
185    ;
186    ;
187    ;
188    ;
189    ;
190    ;
191    ;
192    ;
193    ;
194    ;
195    ;
196    ;
197    ;
198    ;
199    ;
200    ;
201    ;
202    ;
203    ;
204    ;
205    ;
206    ;
207    ;
208    ;
209    ;
210    ;
211    ;
212    ;
213    ;
214    ;
215    ;
216    ;
217    ;
218    ;
219    ;
220    ;
221    ;
222    ;
223    ;
224    ;
225    ;
226    ;
227    ;
228    ;
229    ;
230    ;
231    ;
232    ;
233    ;
234    ;
235    ;
236    ;
237    ;
238    ;
239    ;
240    ;
241    ;
242    ;
243    ;
244    ;
245    ;
246    ;
247    ;
248    ;
249    ;
250    ;
251    ;
252    ;
253    ;
254    ;
255    ;
256    ;
257    ;
258    ;
259    ;
260    ;
261    ;
262    ;
263    ;
264    ;
265    ;
266    ;
267    ;
268    ;
269    ;
270    ;
271    ;
272    ;
273    ;
274    ;
275    ;
276    ;
277    ;
278    ;
279    ;
280    ;
281    ;
282    ;
283    ;
284    ;
285    ;
286    ;
287    ;
288    ;
289    ;
290    ;
291    ;
292    ;
293    ;
294    ;
295    ;
296    ;
297    ;
298    ;
299    ;
300    ;
301    ;
302    ;
303    ;
304    ;
305    ;
306    ;
307    ;
308    ;
309    ;
310    ;
311    ;
312    ;
313    ;
314    ;
315    ;
316    ;
317    ;
318    ;
319    ;
320    ;
321    ;
322    ;
323    ;
324    ;
325    ;
326    ;
327    ;
328    ;
329    ;
330    ;
331    ;
332    ;
333    ;
334    ;
335    ;
336    ;
337    ;
338    ;
339    ;
340    ;
341    ;
342    ;
343    ;
344    ;
345    ;
346    ;
347    ;
348    ;
349    ;
350    ;
351    ;
352    ;
353    ;
354    ;
355    ;
356    ;
357    ;
358    ;
359    ;
360    ;
361    ;
362    ;
363    ;
364    ;
365    ;
366    ;
367    ;
368    ;
369    ;
370    ;
371    ;
372    ;
373    ;
374    ;
375    ;
376    ;
377    ;
378    ;
379    ;
380    ;
381    ;
382    ;
383    ;
384    ;
385    ;
386    ;
387    ;
388    ;
389    ;
390    ;
391    ;
392    ;
393    ;
394    ;
395    ;
396    ;
397    ;
398    ;
399    ;
400    ;
401    ;
402    ;
403    ;
404    ;
405    ;
406    ;
407    ;
408    ;
409    ;
410    ;
411    ;
412    ;
413    ;
414    ;
415    ;
416    ;
417    ;
418    ;
419    ;
420    ;
421    ;
422    ;
423    ;
424    ;
425    ;
426    ;
427    ;
428    ;
429    ;
430    ;
431    ;
432    ;
433    ;
434    ;
435    ;
436    ;
437    ;
438    ;
439    ;
440    ;
441    ;
442    ;
443    ;
444    ;
445    ;
446    ;
447    ;
448    ;
449    ;
450    ;
451    ;
452    ;
453    ;
454    ;
455    ;
456    ;
457    ;
458    ;
459    ;
460    ;
461    ;
462    ;
463    ;
464    ;
465    ;
466    ;
467    ;
468    ;
469    ;
470    ;
471    ;
472    ;
473    ;
474    ;
475    ;
476    ;
477    ;
478    ;
479    ;
480    ;
481    ;
482    ;
483    ;
484    ;
485    ;
486    ;
487    ;
488    ;
489    ;
490    ;
491    ;
492    ;
493    ;
494    ;
495    ;
496    ;
497    ;
498    ;
499    ;
500    ;
501    ;
502    ;
503    ;
504    ;
505    ;
506    ;
507    ;
508    ;
509    ;
510    ;
511    ;
512    ;
513    ;
514    ;
515    ;
516    ;
517    ;
518    ;
519    ;
520    ;
521    ;
522    ;
523    ;
524    ;
525    ;
526    ;
527    ;
528    ;
529    ;
530    ;
531    ;
532    ;
533    ;
534    ;
535    ;
536    ;
537    ;
538    ;
539    ;
540    ;
541    ;
542    ;
543    ;
544    ;
545    ;
546    ;
547    ;
548    ;
549    ;
550    ;
551    ;
552    ;
553    ;
554    ;
555    ;
556    ;
557    ;
558    ;
559    ;
560    ;
561    ;
562    ;
563    ;
564    ;
565    ;
566    ;
567    ;
568    ;
569    ;
570    ;
571    ;
572    ;
573    ;
574    ;
575    ;
576    ;
577    ;
578    ;
579    ;
580    ;
581    ;
582    ;
583    ;
584    ;
585    ;
586    ;
587    ;
588    ;
589    ;
590    ;
591    ;
592    ;
593    ;
594    ;
595    ;
596    ;
597    ;
598    ;
599    ;
600    ;
601    ;
602    ;
603    ;
604    ;
605    ;
606    ;
607    ;
608    ;
609    ;
610    ;
611    ;
612    ;
613    ;
614    ;
615    ;
616    ;
617    ;
618    ;
619    ;
620    ;
621    ;
622    ;
623    ;
624    ;
625    ;
626    ;
627    ;
628    ;
629    ;
630    ;
631    ;
632    ;
633    ;
634    ;
635    ;
636    ;
637    ;
638    ;
639    ;
640    ;
641    ;
642    ;
643    ;
644    ;
645    ;
646    ;
647    ;
648    ;
649    ;
650    ;
651    ;
652    ;
653    ;
654    ;
655    ;
656    ;
657    ;
658    ;
659    ;
660    ;
661    ;
662    ;
663    ;
664    ;
665    ;
666    ;
667    ;
668    ;
669    ;
670    ;
671    ;
672    ;
673    ;
674    ;
675    ;
676    ;
677    ;
678    ;
679    ;
680    ;
681    ;
682    ;
683    ;
684    ;
685    ;
686    ;
687    ;
688    ;
689    ;
690    ;
691    ;
692    ;
693    ;
694    ;
695    ;
696    ;
697    ;
698    ;
699    ;
700    ;
701    ;
702    ;
703    ;
704    ;
705    ;
706    ;
707    ;
708    ;
709    ;
710    ;
711    ;
712    ;
713    ;
714    ;
715    ;
716    ;
717    ;
718    ;
719    ;
720    ;
721    ;
722    ;
723    ;
724    ;
725    ;
726    ;
727    ;
728    ;
729    ;
730    ;
731    ;
732    ;
733    ;
734    ;
735    ;
736    ;
737    ;
738    ;
739    ;
740    ;
741    ;
742    ;
743    ;
744    ;
745    ;
746    ;
747    ;
748    ;
749    ;
750    ;
751    ;
752    ;
753    ;
754    ;
755    ;
756    ;
757    ;
758    ;
759    ;
760    ;
761    ;
762    ;
763    ;
764    ;
765    ;
766    ;
767    ;
768    ;
769    ;
770    ;
771    ;
772    ;
773    ;
774    ;
775    ;
776    ;
777    ;
778    ;
779    ;
780    ;
781    ;
782    ;
783    ;
784    ;
785    ;
786    ;
787    ;
788    ;
789    ;
790    ;
791    ;
792    ;
793    ;
794    ;
795    ;
796    ;
797    ;
798    ;
799    ;
800    ;
801    ;
802    ;
803    ;
804    ;
805    ;
806    ;
807    ;
808    ;
809    ;
810    ;
811    ;
812    ;
813    ;
814    ;
815    ;
816    ;
817    ;
818    ;
819    ;
820    ;
821    ;
822    ;
823    ;
824    ;
825    ;
826    ;
827    ;
828    ;
829    ;
830    ;
831    ;
832    ;
833    ;
834    ;
835    ;
836    ;
837    ;
838    ;
839    ;
840    ;
841    ;
842    ;
843    ;
844    ;
845    ;
846    ;
847    ;
848    ;
849    ;
850    ;
851    ;
852    ;
853    ;
854    ;
855    ;
856    ;
857    ;
858    ;
859    ;
860    ;
861    ;
862    ;
863    ;
864    ;
865    ;
866    ;
867    ;
868    ;
869    ;
870    ;
871    ;
872    ;
873    ;
874    ;
875    ;
876    ;
877    ;
878    ;
879    ;
880    ;
881    ;
882    ;
883    ;
884    ;
885    ;
886    ;
887    ;
888    ;
889    ;
890    ;
891    ;
892    ;
893    ;
894    ;
895    ;
896    ;
897    ;
898    ;
899    ;
900    ;
901    ;
902    ;
903    ;
904    ;
905    ;
906    ;
907    ;
908    ;
909    ;
910    ;
911    ;
912    ;
913    ;
914    ;
915    ;
916    ;
917    ;
918    ;
919    ;
920    ;
921    ;
922    ;
923    ;
924    ;
925    ;
926    ;
927    ;
928    ;
929    ;
930    ;
931    ;
932    ;
933    ;
934    ;
935    ;
936    ;
937    ;
938    ;
939    ;
940    ;
941    ;
942    ;
943    ;
944    ;
945    ;
946    ;
947    ;
948    ;
949    ;
950    ;
951    ;
952    ;
953    ;
954    ;
955    ;
956    ;
957    ;
958    ;
959    ;
960    ;
961    ;
962    ;
963    ;
964    ;
965    ;
966    ;
967    ;
968    ;
969    ;
970    ;
971    ;
972    ;
973    ;
974    ;
975    ;
976    ;
977    ;
978    ;
979    ;
980    ;
981    ;
982    ;
983    ;
984    ;
985    ;
986    ;
987    ;
988    ;
989    ;
990    ;
991    ;
992    ;
993    ;
994    ;
995    ;
996    ;
997    ;
998    ;
999    ;
1000 ;

```

```
1      ;HCOMM
2      ;
3      ; ALLOCATES MEMORY FOR HOST COMM AREA AND PACKET BUFFERS WITH ONE
4      ; DESCRIPTOR IN EACH RING, TO BE CALLED WHEN INITIALIZING
5      ; A CONTROLLER WITH SA.MSG=0 AND SA.CMD=0.
6      ;
7      ; INPUTS:
8      ; R5 - ADDRESS OF CONTROLLER TABLE
9      ;
10     ; OUTPUTS:
11     ; CONTROLLER TABLE POINTING TO HOST COMM AREA
12     ; R4 - ADDRESS OF HOST COMM AREA
13 012546 012701 000336 HCOMM: MOV #MC.SIZ/2,R1      ;GET SIZE OF AREA TO ALLOCATE
14 012552 004737 012522      CALL ALOCM          ;ALLOCATE THE MEMORY
15 012556 010165 000016      MOV R1,C.RING(R5)    ;GET ADDRESS OF HOST COMM AREA
16                                     ;PLACE IN CONTROLLER TABLE
17 012562 000207      RETURN
```

```

1      ;RESET
2      ; RESET ALL UDA-50S IN THE CONTROLLER TABLES
3
4      ; INPUTS:
5      ; IPADRS - CONTAINS ALL IP ADDRESSES
6      ; OUTPUTS:
7      ; NONE
8
9      RESET: PUSH <R3,R4>
10     MOV R3,-(SP)
11     MOV R4,-(SP)
12     CLR    NXMAD
13     SETVEC #4,#NXMI,#PRIO7
14
15     MOV    #PRIO7,-(SP)
16     MOV    #NXMI,-(SP)
17     MOV    #4,-(SP)
18     MOV    #3,-(SP)
19     TRAP   C$SVEC
20     ADD    #10,SP
21
22     BREAK
23
24     MOV    #8,R3      ; R3 = COUNTER OF ENTRIES
25     MOV    #IPADRS,R4 ; R4 -> IP ADDRESS
26     1$: TST    (R4)    ; IS THERE AN ENTRY?
27     BEQ    2$        ; IF NOT, DONE
28     CLR    @R4        ; INIT UDA
29     TST    NXMAD      ; WAS THERE AN ERROR?
30     BNE    3$        ; IF SO, EXIT
31     DEC    R3         ; MAKE SURE WE DO NOT EXTEND OVER AREA
32     BNE    1$        ; IF NOT DONE, BRANCH
33
34     2$: CLRVEC #4
35
36     MOV    #4,R0
37     TRAP   C$CVEC
38
39     POP    <R4,R3>
40
41     MOV (SP),R4
42     MOV (SP),R3
43
44     RETURN
45
46     3$: TST    -(R4)   ; R4 -> UDAIP THAT FAILED
47     MOV    R4,R5     ; SAVE IN R5 FOR REPORT
48     ERDF  20,ERR020
49
50     TRAP   C$ERDF
51     .WORD 20
52     .WORD 0
53     .WORD ERRO20
54
55     CLR    (R4)     ; DESTROY ENTRY SO NOT TO FALL INTO RESET ERROR LOOP
56
57     DOCLN
58
59     TRAP   C$DCLN
60
61     012564 010346
62     012564 010446
63     012570 005037 002200
64     012574 012746 000340
65     012600 012746 017146
66     012604 012746 000004
67     012610 012746 000003
68     012614 104437
69     012616 062706 000010
70     012622 104422
71     012624 012703 000010
72     012630 012704 003436
73     012634 005714
74     012636 001406
75     012640 005034
76     012642 005737 002200
77     012646 001010
78     012650 005303
79     012652 001370
80     012654 012700 000004
81     012660 104436
82     012662 012604
83     012664 012603
84     012666 000207
85     012670 005744
86     012672 010405
87     012674 104455
88     012676 000024
89     012700 000000
90     012702 011750
91     012704 005014
92     012706 104444

```

```

1      ;RUNDM
2
3      ;LOAD AND RUN A DM PROGRAM IN THE CONTROLLERS. RETURN WHEN ALL
4      ;DM PROGRAMS HAVE TERMINATED.
5
6      ;INPUTS:
7      ;   TSTTAB - POINTER TO FIRST CONTROLLER TABLE
8      ;   R1 - NUMBER OF CONTROLLERS TO TEST
9      ;IMPLICIT INPUTS:
10     ;   DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
11     ;OUTPUTS:
12     ;   Z SET IF NO CONTROLLERS SUCCESSFULLY STARTED
13     ;ALL REGISTERS ARE USED AND PREVIOUS CONTENTS DESTROYED.
14
15 012710 010137 002166      RUNDM:  MOV R1,URUN      ;SAVE NUMBER OF UNITS TO RUN
16 012714 005037 002170      CLR URNING      ;CLEAR NUMBER OF UNITS RUNNING
17
18     ;LOAD DM PROGRAM INTO EACH CONTROLLER
19
20 012720 013737 002166 002172      MOV URUN,UCNT      ;SET COUNTER OF UNITS
21 012726 013705 002162      MOV TSTTAB,R5      ;GET FIRST CONTROLLER TABLE
22 012732      LDDM:
23 012732 005065 000014      CLR C.FLG(R5)      ;CLEAR ALL FLAGS
24 012736 116537 000002 002074      MOVB C.UNIT(R5),L#LUN ;SEE IF UNIT TO BE TESTED
25 012744 005765 000002      TST C.UNIT(R5)
26 012750 100407      BMI LDNEXT      ;IF NOT, DON'T LOAD THIS UNIT
27 012752      ASSUME CT.AVL EQ BIT15
28 012752 004737 012546      CALL HCOMM      ;ALLOCATE SPACE FOR HOST COMM AREA
29 012756 004737 016370      CALL LOADDM      ;LOAD THE DM PROGRAM
30 012762 001402      BEQ LDNEXT      ;IF ERROR, GO TO NEXT CONTROLLER
31 012764 005237 002170      INC URNING      ;IF NO ERROR, COUNT UNIT RUNNING
32 012770 062705 000054      LDNEXT: ADD #C.SIZE,R5 ;MOVE TO NEXT CONTROLLER TABLE
33 012774 005337 002172      DEC UCNT      ;CHECK IF MORE CONTROLLERS
34 013000 001354      BNE LDDM      ;LOAD NEXT
35 013002 005037 002176      CLR UFREEZ      ;CLEAR UNIT FREEZE FLAG
36 013006 012737 177777 003204      MOV #-1,FCNUM ;INVALIDATE FCT BLOCK NUMBER (BLOCK IN MEMORY)
37
38     ;CHECK IF ANY CONTROLLERS LOADED
39
40 013014 005737 002170      TST URNING      ;ANY UNITS LOADED?
41
42     ;THE DM PROGRAMS ARE NOW IN CONTROL
43     ;RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
44
45 013020 000207      RETURN

```

```

1          ;CLOSEF
2          ;
3          ;CLOSE DATA FILE FOR DM PROGRAMS
4          ;
5          ;INPUTS:
6          ;      FILOPN - ZERO IF FILE NOT OPEN
7          ;OUTPUTS:
8          ;      NONE
9
10 013022 005737 002174  CLOSEF: TST FILOPN          ;SEE IF FILE CURRENTLY OPEN
11 013026 001403          BEQ 18          ; IF SO, CLOSE IT
12 013030          CLOSE          ;AND MARK AS SO          TRAP      C8CLOS
   01303C 104435
13 013032 005037 002174          CLR FILOPN
14 013036 000207          18:      RETURN

```

```

1      ;RESPDM
2
3      ;RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
4      ;HAVE TERMINATED.
5
6 013040 013705 002162      RESPDM: MOV TSTTAB,R5      ;GET CONTROLLER TABLE ADDRESS
7 013044 013737 002166 002172      MOV URUN,UCNT      ;SET COUNTER OF UNITS
8 013052      RESPCT: BREAK      ;ALLOW DRS TO SEE TERMINAL INPUT
9 013052 104422      ;
10 013054 016504 000016      MOV C.RING(R5),R4      ;GET MOST COMM AREA ADDRESS
11 013060 032765 000002 000014      BIT #CT.RN,C.FLG(R5)      ;CHECK IF PROGRAM RUNNING
12 013070 116537 000002 002074      BEQ RSPNXT      ;IF NOT, LOOK AT NEXT
13 013076 032765 000010 000014      MOVB C.UNIT(R5),L#LUN      ;STORE UNIT NUMBER UNDER TEST
14 013104 001150      BNE RSPIN      ;SEE IF INTERRUPT RECEIVED
15 013106 032765 000004 000014      BIT #CT.MSG,C.FLG(R5)      ;IF SO, LOOK AT PACKET
16 013114 001002      BNE 1#      ;SEE IF COMMAND HAS BEEN SENT
17 013116 000137 013664      JMP RSPOUT      ;IF NOT, SEND ONE
18
19      ;CHECK IF UDA STILL RUNNING
20
21 013122 011503      1#: MOV (R5),R3      ;GET ADDRESS OF UDAIP
22 013124 016301 000002      MOV 2(R3),R1      ;LOOK AT UDASA REGISTER
23 013130 001405      BEQ RSPTH      ;IF ZERO, UDA STILL RUNNING
24 013132      ERRDF 30,ERR030      ;REPORT UDA HAS FATAL ERROR
25 013132 104455      ;
26 013134 000036      ;
27 013136 000000      ;
28 013140 012162      ;
29 013142 000465      BR RSPDRP      ;DROP CONTROLLER FROM TESTING
30
31      ;CHECK FOR TIMEOUT OF RESPONSE
32
33 013144 005765 000044      RSPTH: TST C.TOT(R5)      ;SEE IF DUP PROGRAM TO BE TIMED
34 013150 001451      BEQ RSPNTO
35 013152 005737 003210      TST KW.CSR      ;SEE IF A CLOCK ON SYSTEM
36 013156 001446      BEQ RSPNTO      ;DON'T TIME IF NO CLOCK
37 013160 023765 003222 000042      CMP KW.EL*2,C.TOM(R5)      ;COMPARE TO TIMEOUT COUNTER
38 013166 101005      BHI RSPTHO
39 013170 001041      BNE RSPNTO
40 013172 023765 003220 000040      CMP KW.EL,C.TO(R5)
41 013200 103435      BLO RSPNTO
42 013202 032765 000040 000014      RSPTHO: BIT #CT.STA,C.FLG(R5)      ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
43 013210 001101      BNE RSPTOE      ;SEE IF A GET DUST STATUS COMMAND OUTSTANDING
44 013212 005764 000012      TST HC.CCT(R4)      ;REPORT ERROR IF SO
45 013216 100476      BMI RSPTOE      ;SEE IF UDA TOOK LAST COMMAND PACKET
46 013220 012700 000100      MOV #CT.TM1,R0      ;REPORT ERROR IF NOT
47 013224 032765 000100 000014      BIT #CT.TM1,C.FLG(R5)      ;SEE IF FIRST TIMEOUT ALREADY HAPPENED
48 013232 001401      BEQ 1#
49 013234 006300      ASL R0      ;IF SO,
50 013236 052700 000040      1#: BIS #CT.STA,R0      ;SET SECOND TIME OUT FLAG
51 013242 050065 000014      BIS R0,C.FLG(R5)      ;SET THE PROPER TIMEOUT BIT
52 013246 012700 000001      MOV #OP.GDS,R0      ; AND STATUS REQUESTED BIT
53 013252 004737 016570      CALL BLDCMD      ;BUILD GET DUST STATUS COMMAND
54 013256 012764 100000 000012      MOV #RG.OWN,HC.CCT(R4)      ;MARK COMMAND TO UDA
55 013264 005775 000000      TST #R5      ;TELL UDA COMMAND IS THERE
56 013270 000137 013744      JMP RSPOU4

```

C#BRK

C#ERDF  
30  
0  
ERR030



H6

53 013274

RSPNT0:

```

1          ;SWITCH TO NEXT CONTROLLER
2
3 013274 005737 002176  RSPNXT: TST UFREEZ      ;FROZEN TO ONE UNIT?
4 013300 001264          BNE RESPCT      ;STAY THERE IF SO
5 013302 062705 000054  ADD #C.SIZE,R5      ;MOVE TO NEXT TABLE
6 013306 005337 002172  DEC UCNT           ;CHECK IF MORE CONTROLLERS
7 013312 001257          BNE RESPCT      ;LOOK AT NEXT CONTROLLER
8 013314 000651          BR RESPDM       ;LOOK AT FIRST CONTROLLER AGAIN
9
10         ;REMOVE A CONTROLLER FROM TESTING
11
12 013316 005065 000014  RSPDRP: CLR C.FLG(R5)  ;CLEAR PROGRAM RUNNING
13 013322 005037 002176  CLR UFREEZ
14 013326 010504          MOV R5,R4
15 013330 062704 000020  ADD #C.DRO,R4
16 013334 012702 000010  MOV #8.,R2
17 013340 012403 1$:    MOV (R4),R3
18 013342 001420          BEQ 3$
19 013344 005763 000002  TST D.UNIT(R3)
20 013350          ASSUME DT.AVL EQ BIT15
21 013350 100003          BPL 2$
22 013352 005302          DEC R2
23 013354 001371          BNE 1$
24 013356 000412          BR 3$
25 013360 052763 100000 000002 2$:  BIS #DT.AVL,D.UNIT(R3)
26 013366 005302          DEC R2
27 013370 001405          BEQ 3$
28 013372 005714          TST (R4)
29 013374 001403          BEQ 3$
30 013376 004737 016370  CALL LOADDM      ;START DM PROGRAM AGAIN
31 013402 001223          BNE RESPCT
32 013404 005337 002170 3$:  DEC URNING      ;REDUCE RUNNING CONTROLLERS COUNT
33 013410 001331          BNE RSPNXT      ;IF ANY STILL RUNNING, LOOK AT THEM
34 013412 000207          RETURN      ;ELSE RETURN TO TEST SECTION
35
36 013414          RSPTOE: ERRDF 31.,ERR031 ;REPORT TIMEOUT ERROR
37 013414 104455          TRAP          C:ERRDF
38 013416 000037          .WORD          31
39 013420 000000          .WORD          0
40 013422 012176          .WORD          ERR031
41 013424 000734          BR RSPDRP      ;DROP CONTROLLER FROM TESTING

```

```

1          ;CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
2
3          ;CHECK FOR PROPER OPCODE IN END PACKET
4
5 013426 012700 000204          RSPIN: MOV #OP.END+OP.SSD,R0          ;GET SEND DATA END PACKET OPCODE
6 013432 032765 000020 000014  BIT #CT.REQ,C.FLG(R5)          ;LOOK IF SEND DATA OR RECEIVE DATA
7 013440 001402          BEQ RSPMWR
8 013442 012700 000205          MOV #OP.END+OP.RSD,R0          ;CHANGE TO RECEIVE DATA END PACKET OPCODE
9 013446 120064 000030          RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)      ;COMPARE TO OPCODE IN END PACKET
10 013452 001145          BNE RSPERR
11
12          ;LOOK AT STATUS CODE
13
14 013454 032764 000037 000032  BIT #ST.MSK,HC.MPK+P.STS(R4)      ;CHECK FOR STATUS CODE ST.SUC (ZERO)
15 013462 001004          BNE RSPERW
16
17          ;CHECK FOR EXPECTED REFERENCE NUMBER
18
19 013464 026564 000052 000020  CMP C.REF(R5),HC.MPK+P.CRF(R4)    ;CHECK IF CORRECT REF NUMBER
20 013472 001405          BEQ RSPPTW
21 013474          RSPERW: ERRDF 33.,ERR033
22 013474 104455          TRAP CERRDF
23 013476 000041          .WORD 33
24 013500 000000          .WORD 0
25 013502 012226          .WORD ERR033
26 013504 000704          BR RSPDRP          ;DROP UNIT FROM TESTING
27
28          ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
29
30 013506 032765 000020 000014  RSPPTW: BIT #CT.REQ,C.FLG(R5)    ;CHECK IF RESPONSE FROM DM PROGRAM
31 013514 001463          RSPOU: BEQ RSPOUT          ;LOOK AT REQUEST NUMBER IF 50

```

```

1      ;MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
2
3 013516 016401 000430 RSPPT2: MOV HC.BF2(R4),R1      ;GET REQUEST NUMBER
4 013522 042701 007777      BIC #C<DU.TYP>,R1      ;CHECK TYPE
5 013526 001403      BEQ 1$      ;IF ZERO, ERROR
6 013530 020127 060000      CMP R1,#DU.SPC      ;CHECK IF IN EXPECTED RANGE
7 013534 101405      BLOS RSPPT3
8 013536      1$: ERRDF 32,,ERR032      ;BAD REQUEST NUMBER
      013536 104455      TRAP C#ERDF
      013540 000040      .WORD 32
      013542 000000      .WORD 0
      013544 012210      .WORD ERR032
9 013546 000663      BR RSPDRP      ;DROP UNIT FROM TESTING
10
11 013550 016403 000034 RSPPT3: MOV HC.MPK.P.BCNT(R4),R3      ;GET BYTE COUNT OF CHARACTERS RECEIVED IN R3
12 013554 162703 000002      SUB #2,R3      ;(FIRST TWO CHARACTERS ARE TYPE WORD)
13 013560 012700 000004      MOV #OP.SSD,R0      ;BUILD A SEND DATA COMMAND PACKET
14 013564 004737 016570      CALL BLDCMD      ; FOR ANSWER TO DM PROGRAM
15 013570 012700 000164      MOV #HC.BF1,R0      ;POINT TO BUFFER IN PACKET
16 013574 004737 016732      CALL CLRBUF      ; AND CLEAR BUFFER
17 013600 010402      MOV R4,R2      ;R2 POINTS TO SEND BUFFER
18 013602 062704 000244      ADD #HC.BSZ,R4      ;R4 POINTS TO CHARACTERS IN RECEIVE BUFFER
19 013606 042724 170000      BIC #DU.TYP,(R4)      ;CLEAR TYPE FIELD IN BUFFER
20 013612 000301      SWAB R1      ;GET TYPE RIGHT JUSTIFIED
21 013614 006201      ASR R1      ;TIMES TWO
22 013616 006201      ASR R1
23 013620 006201      ASR R1
24 013622 010100      MOV R1,R0      ;COPY MESSAGE TYPE TO R0
25 013624 005001      CLR R1      ;R1 CONTAINS ZERO SEND BYTE COUNT
26 013626 004770 014112      CALL @RSPDSP-2(R0)      ;CALL REQUESTED ROUTINE
27 013632 001231      BNE RSPDRP      ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
28      ; Z SET IF UNIT TO CONTINUE RUNNING
29 013634 016504 000016      MOV C.RING(R5),R4      ;GET RING ADDRESS
30 013640 032701 000001      BIT #1,R1      ;LOOK AT CHARACTER COUNT TO SEND TO DUP PROGRAM
31 013644 001401      BEQ 1$      ;IF AN ODD COUNT
32 013647 005201      INC R1      ; INCREASE BY ONE
33 013650 010164 000120      1$: MOV R1,HC.CPK.P.BCNT(R4)      ;PUT CHARACTER COUNT IN COMMAND PACKET
34 013654 100003      BPL RSPOUT      ;IF NEGATIVE BYTE COUNT RETURNED
35 013656 042765 000020 000014      BIC #CT.REQ,C.FLG(R5)      ; DON'T SEND ANY DATA TO UDA
36
37      ;SEND COMMAND BACK TO UDA
38
39 013664 042765 000350 000014 RSPOUT: BIC #CT.MSG,CT.STA,CT.TM1,CT.TM2,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
40 013672 032765 000020 000014      BIT #CT.REQ,C.FLG(R5)      ;CHECK WHICH COMMAND TO SEND
41 013700 001014      BNE RSPDU2      ;BRANCH IF RESPONSE TO REQUEST
42
43 013702 012700 000005      MOV #OP.RSD,R0      ;BUILD RECEIVE DATA COMMAND
44 013706 004737 016570      CALL BLDCMD
45 013712 012700 000430      MOV #HC.BF2,R0      ;POINT TO MESSAGE BUFFER
46 013716 004737 016732      CALL CLRBUF      ; AND CLEAR IT
47 013722 052765 000020 000014      BIS #CT.REQ,C.FLG(R5)      ;SET REQUEST BIT
48 013730 000403      BR RSPDU3
49
50 013732 042765 000020 000014 RSPDU2: BIC #CT.REQ,C.FLG(R5)      ;CLEAR REQUEST BIT
51 013740      RSPDU3:
52 013740 004737 016654      CALL SNDCMD      ;SEND COMMAND TO UDA
53 013744 016500 000044      RSPDU4: MOV C.TOT(R5),R0      ;SET TIMEOUT
    
```

```

54 013750 010501          MOV R5,R1
55 013752 062701 000040    ADD #C.TO,R1          ;PUT TIME IN CONTROLLER TABLE
56 013756 004737 017166    CALL SETTO
57 013762 000137 013274    JMP RSPNXT          ;NOW WAIT FOR END PACKET
58 013766 122764 000201 000030 RSPERR: CMPB #OP.END+OP.GDS,HC.MPK+P.OPCD(R4) ;SEE IF GET DUST STATUS OPCODE
59 013774 001237          BNE RSPERW
60 013776 132764 000010 000037 BITB #DF.ACT,HC.MPK+P.DFLG(R4) ;IF DUST NO LONGER RUNNING
61 014004 001603          BEQ RSPTOE          ; REPORT ERROR
62 014006 042765 000050 000014 BIC #CT.STA+CT.MSG,C.FLG(R5) ;CLEAR CONTROL BITS
63 014014 032765 000200 000014 BIT #CT.TM2,C.FLG(R5) ;IF AT SECOND TIMEOUT
64 014022 001413          BEQ 1$
65 014024 026465 000040 000046 CMP HC.MPK+P.DPI(R4),C.PRI(R5) ;COMPARE PROGRESS INDICATOR
66 014032 001004          BNE 2$
67 014034 026465 000042 000050 CMP HC.MPK+P.DPI+2(R4),C.PRI+2(R5) ;COMPARE PROGRESS INDICATOR
68 014042 001422          BEQ 4$          ;REPORT ERROR IF NOT CHANGED
69 014044 042765 000200 000014 2$: BIC #CT.TM2,C.FLG(R5) ;CLEAR TIMEOUT 2 FLAG
70 014052 032765 000100 000014 1$: BIT #CT.TM1,C.FLG(R5) ;IF AT FIRST TIMEOUT
71 014060 001406          BEQ 3$
72 014062 016465 000040 000046 MOV HC.MPK+P.DPI(R4),C.PRI(R5) ;GET COPY OF PROGRESS INDICATOR
73 014070 016465 000042 000050 MOV HC.MPK+P.DPI+2(R4),C.PRI+2(R5) ;GET COPY OF PROGRESS INDICATOR
74 014076 012764 140000 000006 3$: MOV #RG.OWN+RG.FLG,HC.MCT(R4) ;GIVE MESSAGE BUFFER BACK TO UDA
75 014104 000137 013274          JMP RSPNXT
76 014110 000137 013414          4$: JMP RSPTOE

```

```
1  
2  
3 014114 014130  
4 014116 014202  
5 014120 014354  
6 014122 014502  
7 014124 014512  
8 014126 014522  
9 000006
```

;  
RESPONSE REQUEST DISPATCH TABLE

RSPDSP: .WORD QUEST  
.WORD DQUEST  
.WORD INFO  
.WORD TERM  
.WORD ERRTRM  
.WORD SPECL  
DSPSIZ=<.-RSPDSP>/2

;  
QUESTION  
;QUESTION WITH DEFAULT ANSWER  
;INFORMATION MESSAGE FOR OPERATOR  
;NORMAL TERMINATION  
;FATAL ERROR TERMINATION  
;SPECIAL  
;LEGAL NUMBERS ARE LOWER THAN THIS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38

;NORMAL DUP RECEIVE DATA BUFFER DESCRIPTION

;BYTE OFFSET FROM  
;START OF BUFFER

;	0	!	TYPE !	MESSAGE NUMBER	!
;	2	!		DATA BYTES	!
;	4	!		DATA BYTES	!
;	6	!		DATA BYTES	!
;	8	!		DATA BYTES	!
;	10	!		DATA BYTES	!
;	12	!		DATA BYTES	!
;	14	!		DATA BYTES	!
;	16	!		DATA BYTES	!
;	18	!		DATA BYTES	!
;	20	!		DATA BYTES	!
;	22	!		DATA BYTES	!
;	.	!		.	!
;	.	!		.	!
;	.	!		.	!
;	80	!		DATA BYTES	!

USED TO SELECT ROUTINE  
R4 CONTAINS THIS ADDRESS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38

;NORMAL DUP SEND DATA BUFFER DESCRIPTION GIVEN IN RESPONSE TO ABOVE PACKET

;BYTE OFFSET FROM  
;START OF BUFFER

:	0	:	DATA BYTES	:
:	2	:	DATA BYTES	:
:	4	:	DATA BYTES	:
:	6	:	DATA BYTES	:
:	8	:	DATA BYTES	:
:	10	:	DATA BYTES	:
:	12	:	DATA BYTES	:
:	14	:	DATA BYTES	:
:	16	:	DATA BYTES	:
:	18	:	DATA BYTES	:
:	20	:	DATA BYTES	:
:	22	:	DATA BYTES	:
:	.	:	.	:
:	.	:	.	:
:	.	:	.	:
:	80	:	DATA BYTES	:

R2 CONTAINS THIS ADDRESS



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 014130 004737 014654
17 014134 062700 000004
18 014140 014403
19 014142 001411
20 014144 020327 000007
21 014150 001410
22 014152
    014152 104455
    014154 000144
    014156 000000
    014160 012270
23 014162 000244
24 014164 000207
25
26 014166 012700 003306
27 014172
28 014172 005201
29 014174 112022
30 014176 001375
31 014200 000207

```

```

;MESSAGE TYPE 1
;
;ANSWER QUESTION FOR DUP PROGRAM
;
;INPUT:
;   R5 - ADDRESS OF CONTROLLER TABLE
;   R4 - POINTER TO DATA IN RECEIVE BUFFER
;   R3 - CHARACTER COUNT IN RECEIVE BUFFER
;   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
;   R1 - ZERO
;
;OUTPUT:
;   R1 - COUNT OF CHARACTERS IN SEND BUFFER
;   Z SET TO CONTINUE RUNNING DUP PROGRAM
;   Z CLEAR TO STOP THE DUP PROGRAM
;
QUEST: CALL GDRV   ;GET POINTER TO DRIVE TABLE
      ADD @D.SERN,R0 ;BUMP POINTER TO SERIAL NUMBER
      MOV -(R4),R3   ;GET QUESTION NUMBER
      BEQ QUE0       ;BRANCH IF QUESTION NUMBER 0
      CMP R3,#7      ;IF NOT, SEE IF QUESTION NUMBER 7
      BEQ QUE7
      ERDF 100.,ERR100 ;ANY OTHER NUMBER IS AN ERROR

      CLZ ;CLEAR Z TO STOP DUP PROGRAM
      RETURN

QUE0: MOV @DATE0,R0 ;POINT TO DATE STRING
QUE7:
QUEL: INC R1 ;COUNT THE CHARACTERS
      MOVB (R0)+,(R2)+ ;AND PUT THEM IN OUTPUT BUFFER
      BNE QUEL ;UNTIL A NUL CHARACTER FOUND
      RETURN ;RETURN WITH Z SET

```

```

TRAP C:ERDF
.WORD 100
.WORD 0
.WORD ERR100

```

```

1      ;MESSAGE TYPE 2
2
3      ;ANSWER QUESTION FOR DUP PROGRAM WITH DEFAULT ANSWER
4
5      ;INPUT:
6      ;      R5 - ADDRESS OF CONTROLLER TABLE
7      ;      R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;      R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;      R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;      R1 - ZERO
11     ;OUTPUT:
12     ;      R1 - COUNT OF CHARACTERS IN SEND BUFFER
13     ;      Z SET TO CONTINUE RUNNING DUP PROGRAM
14     ;      Z CLEAR TO STOP THE DUP PROGRAM
15
16 014202 004737 014654 DQUEST: CALL GTDRV      ;GET DRIVE TABLE ADDRESS INTO R0
17 014206 014403      MOV -(R4),R3      ;GET QUESTION NUMBER
18 014210 020327 000006      CMP R3,#DQUESZ
19 014214 101035      BHI DQUEX
20 014216 006303      ASL R3
21 014220 000173 014224      JMP @DQUEJP(R3)
22 014224 014310 DQUEJP: .WORD DQUEX      ; 0 (NOT USED)
23 014226 014242      .WORD DQUNIT      ; 1 ENTER UNIT NUMBER TO FORMAT
24 014230 014310      .WORD DQUEX      ; 2 (NOT USED)
25 014232 014310      .WORD DQUEX      ; 3 (NOT USED)
26 014234 014314      .WORD DQRFMT      ; 4 USE EXISTING BAD SECTOR INFORMATION
27 014236 014334      .WORD DQRSTR      ; 5 DOWN-LINE LOAD BAD SECTOR BLOCK INFORMATION
28 014240 014344      .WORD DQCONT      ; 6 CONTINUE IF BAD BLOCK INFO INACCESSIBLE
29      000006      DQUESZ=<<.-DQUEJP>/2>-1
30
31     ;ENTER UNIT NUMBER TO FORMAT
32
33 014242      DQUNIT: PUSH R5
34 014242 010546      CLR R4
35 014244 005004      MOV (R0),R3      ;GET DRIVE NUMBER
36 014246 011003      ASSUME D.DRV EQ 0
37 014250 012700 000012      MOV #10.,R0      ;RADIX 10.
38 014254 004737 016332 DQUNL1: CALL DIVIDE
39 014260      PUSH R5
40 014262 010546      MOV R5,-(SP)
41 014264 005201      INC R1
42 014266 005703      TST R3
43 014270 010100      BNE DQUNL1
44 014272      MOV R1,R0
45 014272 012605      DQUNL2: POP R5
46 014274 062705 000060      MOV (SP)+,R5
47 014300 110522      ADD #0,R5
48 014302 005300      MOVB R5,(R2)+
49 014304 001372      DEC R0
50 014306 012605      BNE DQUNL2
51 014310 000264      POP R5
52 014312 000207      MOV (SP)+,R5
53 014314 032737 000003 003206 DQRFMT: BIT #50.FMT,MODE

```

54	014322	001410					BEQ DQNO
55	014324	112712	000131			DQYES:	MOVB @'Y,(R2)
56	014330	005201					INC R1
57	014332	000766					BR DQUEX
58							
59	014334	032737	000010	003206	DQRSTR:		BIT #50,STR,MODE
60	014342	001370					BNE DQYES
61	014344				DQCONT:		
62	014344	112712	000116		DQNO:		MOVB @'N,(R2)
63	014350	005201					INC R1
64	014352	000756					BR DQUEX

```

1      ;MESSAGE TYPE 3
2      ;
3      ;PRINT INFORMATION FROM DUP PROGRAM
4      ;
5      ;INPUT:
6      ;
7      ;   R5 - POINTER TO CONTROLLER TABLE
8      ;   R4 - POINTER TO DATA IN RECEIVE BUFFER
9      ;   R3 - CHARACTER COUNT IN RECEIVE BUFFER
10     ;   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
11     ;   R1 - ZERO
12     ;OUTPUT:
13     ;   R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM
14     ;   Z SET TO CONTINUE RUNNING DUP PROGRAM
15     014354 016400 177776      INFO:  MOV -2(R4),R0      ;GET MESSAGE NUMBER
16     014360 001434              BEQ INFOB              ;IF ZERO, PRINT BEGUN MESSAGE
17     014362 020027 000100      CMP RO,#100           ;IF OCTAL 100
18     014366 001423              BEQ INFOE              ; PRINT ERROR MESSAGE
19     014370 020027 000200      CMP RO,#200           ;SEE IF 200 OR GREATER
20     014374 002005              BGE INFOH              ; IF SO, PRINT WITHOUT FREEZING
21     014376 005737 002176      TST UFREEZ
22     014402 001007              BNE INFOP
23     014404 005237 002176      INC UFREEZ
24     014410 004737 014654      INFOH: CALL GTDRV      ;
25     014414 010002              MOV RO,R2
26     014416 004737 014700      CALL HEADER
27     014422 004737 014620      INFOP: CALL MMSG      ;PRINT THE MESSAGE
28     014426 012701 100000      INFOX: MOV #BIT15,R1 ;RETURN A NEGATIVE BYTE COUNT
29     014432 000264              SEZ
30     014434 000207              RETURN                ;RETURN WITH Z SET
31
32     014436              INFOE: ERRDF 101,,ERR101 ;ANSWER WAS REJECTED BY DUP PROGRAM
33     014436 104455              TRAP C#ERDF
34     014440 000145              .WORD 101
35     014442 000000              .WORD 0
36     014444 012304              .WORD ERR101
37
38     014446 000244              CLZ ;RETURN WITH Z CLEAR TO STOP DUP PROGRAM
39     014450 000207              RETURN
40
41     014452 004737 014654      INFOB: CALL GTDRV      ;PRINT FORMAT BEGUN MESSAGE
42     014456 010002              MOV RO,R2
43     014460 004737 014700      CALL HEADER
44     014464 004737 014620      CALL MMSG
45     014470              PNT WNSTOP ;PRINT WARNING NOT TO STOP NOW
46     014470 004137 016264      JSR R1,LPNT
47     014474 004322              .WORD WNSTOP
48     014476 000000              .WORD PNT.CT
49
50     014500 000752              BR INFOX

```

```

1      ;MESSAGE TYPE 4
2      ;
3      ;TERMINATION MESSAGE
4      ;
5      ;INPUT:
6      ;   R5 - POINTER TO CONTROLLER TABLE
7      ;   R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;   R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;   R1 - ZERO
11     ;OUTPUT:
12     ;   Z CLEAR TO TERMINATE DUP PROGRAM
13
14 014502 004737 014354  TERM:  CALL INFO      ;PRINT THE MESSAGE
15 014506 000244          CLZ
16 014510 000207          RETURN                ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM

```

```

1      ;MESSAGE TYPE 5
2      ;
3      ;ERROR TERMINATION MESSAGE
4      ;
5      ;INPUT:
6      ;      R5 - POINTER TO CONTROLLER TABLE
7      ;      R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;      R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;      R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;      R1 - ZERO
11     ;OUTPUT:
12     ;      Z CLEAR TO TERMINATE DUP PROGRAM
13
14 014512 004737 014354  ERRTRM: CALL INFO
15 014516 000244          CLZ
16 014520 000207          RETURN                                ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM

```

```

1      ;MESSAGE TYPE 6
2      ;
3      ;SPECIAL TYPE - READ FCT BLOCK FROM FILE
4      ;
5      ;INPUT:
6      ;   R5 - POINTER TO CONTROLLER TABLE
7      ;   R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;   R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;   R1 - ZERO
11     ;OUTPUT:
12     ;   Z SET TO SEND DATA TO PROGRAM
13
14 014522 023714 003204   SPECL:  CMP FCTNUM,(R4) ;SEE IF DESIRED BLOCK IS IN MEMORY
15 014526 001425         BEQ SPECLX   ; IF SO, SEND TO DUP PROGRAM
16 014530 002407         BLT SPECLR   ; IF LOWER NUMBERED BLOCK IN MEMORY,
17                                     ; GO READ NEXT BLOCK
18 014532         SPECLC:
19 014532         CLOSE   ;OTHERWISE, START READING FROM BEGINNING AGAIN
20 014534 104435         OPEN  #FNAME
21 014534 012700 003232         MOV  #-1,FCTNUM
22 014540 104434         SPECLR:  MOV  #512.,R3   ;GET BYTE COUNT IN A BLOCK
23 014542 012737 177777 003204   MOV  #FCTBUF,R1 ;POINT TO STORAGE AREA
24 014550 012703 001000         SPECLL: GETBYTE (R1). ;READ THE FILE
25 014554 012701 002204         BNCOMPLETE SPECLE ;PRINT ERROR IF NO MORE BYTES IN FILE
26 014560 104426         DEC  R3   ;COUNT THE BYTES
27 014562 110021         BNE  SPECLL
28 014564 103005         INC  FCTNUM ;KEEP COUNT OF BLOCK IN MEMORY
29 014566 005303         BR   SPECL
30 014570 001373         BR   SPECL
31 014572 005237 003204         SPECL:  INC  (R2)   ;TELL DUP PROGRAM DATA NOT AVAILABLE
32 014576 000751         SPECLX:  MOV  #FCTBUF,2(R2) ;PUT ADDRESS OF DATA IN OUTPUT BUFFER
33 014600 005212         MOV  #6,R1   ;SEND 3 WORDS TO DUP PROGRAM
34 014602 012762 002204 000002   SEZ
35 014610 012701 000006         RETURN
36 014614 000264
37 014616 000207

```

```

TRAP  C#CLOS
MOV   #FNAME,R0
TRAP  C#OPEN
TRAP  C#GETB
MOV   R0,(R1)
BCC   SPECLE

```

```

;RETURN WITH Z SET TO SEND DATA TO DUP PROGRAM

```

```

1      ;PRINT A MESSAGE IN THE RECEIVE BUFFER FROM THE DUP PROGRAM
2      ;
3      ;INPUT:
4      ;      R4 - POINTER TO DATA IN RECEIVE BUFFER
5      ;      R3 - CHARACTER COUNT IN RECEIVE BUFFER
6      ;OUTPUT:
7      ;      R4 - POINTER TO CHARACTER AFTER MESSAGE IN RECEIVE BUFFER
8      ;      R3 - ZERO
9      ;      R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM
10     ;      R0 - CONTENTS DESTROYED
11     ;      Z SET TO CONTINUE RUNNING DUP PROGRAM
12
13 014620      MSG:
14 014620      112400      1$:      MOVB (R4),R0      ;PRINT CHARACTERS FROM DUP PROGRAM
15 014622      001405      BEQ 2$      ; DISCARDING LF AND NULL CHARACTERS
16 014624      020027      000012      CMP R0,#12
17 014630      001402      BEQ 2$
18 014632      014632      004737      016054      PRINT R0      CALL CPNT
19 014636      005303      2$:      DEC R3      ;COUNT THE CHARACTERS
20 014640      003367      BGT 1$
21 014642      014642      112700      000015      PRINT @CR      MOVB @CR,R0
22 014646      004737      016054      CALL CPNT
23 014652      000207      RETURN

```



```

1      ;GDRV
2      ;
3      ;GET DRIVE TABLE ADDRESS FROM CONTROLLER TABLE
4      ;
5      ;INPUTS:
6      ;      R5 - CONTROLLER TABLE ADDRESS
7      ;OUTPUTS:
8      ;      R0 - ADDRESS OF FIRST DRIVE TABLE AVAILABLE FOR TESTING
9      ;              (WITH DT.AVL BIT CLEAR)
10     ;
11 014654  GTDRV: PUSH R5
12 014654 010546                                MOV R5,-(SP)
13 014656 062705 000020                        ADD @C.DRO,R5
14 014662 012500                                GTDRV: MOV (R5)+,R0
15 014664 016037 000002 002074                MOV D.UNIT(R0),L&LUN
16 014672 100773                                ASSUME DT.AVL EQ BIT15
17 014674                                BMI GTDRV
18 014674 012605                                POP R5
18 014676 000207                                MOV (SP)+,R5
                                RETURN

```

```

1      ;HEADER
2      ;
3      ;PRINT A HEADER IN FRONT OF EACH MESSAGE FROM DUP PROGRAM.
4      ;A UDA ADDRESS IS PRINTED IF MORE THAN ONE UDA IS IN HARDWARE P-TABLE.
5      ;A RUNTIME IS PRINTED IF A CLOCK IS BEING USED TO TIME PROGRAM EXECUTION.
6      ;
7      ;INPUT:
8      ;      R5 - POINTER TO CONTROLLER TABLE
9      ;OUTPUT:
10     ;      R0 - POINTER TO DRIVE TABLE
11     ;      PRINTED MESSAGE
12     ;
13     014700 022737 000001 002012 HEADER: CMP #1,L#UNIT          ;IF MORE THAN ONE UNIT BEING TESTED
14     014706 001411                      BEQ 1$
15     014710 011246 000002 016226 PNTF MESSG,D.UNIT(R2),(R5),(R2) ;PRINT UDA ADDRESS
16     014710 011246                      MOV (R2),-(SP)
17     014712 011546                      MOV (R5),-(SP)
18     014714 016246 000002 016226 MOV D,UNIT(R2),-(SP)
19     014720 004137 016226 JSR R1,LPNTF
20     014724 004027                      .WORD MESSG
21     014726 000006                      .WORD PNT,CT
22     014730                      ASSUME C,UADR EQ 0
23     014730                      ASSUME D,DRV EQ 0
24     014730 000407 BR 2$
25     014732 005737 003210 1$: TST KW.CSR          ;IF NO CLOCK BEING USED
26     014736 001406                      BEQ 3$          ;BYPASS RUNTIME MESSAGE
27     014740 112700 000015 PRINT #CR
28     014744 004737 016054 MOV B #CR,R0
29     014750 004737 020242 2$: CALL RNTIME      ;PRINT RUNTIME IF A CLOCK IN USE
30     014754 112700 000015 3$: PRINT #CR
31     014754 112700 000015 MOV B #CR,R0
32     014760 004737 016054 CALL CPNT
33     014764 000207 RETURN

```

```

1      ;OSTRNG
2      ;
3      ;FORMAT OF THE ASCIZ STRING IS AS FOLLOWS:
4      ;
5      ;CHARACTERS ENCLOSED IN QUOTES ARE TO BE PRINTED AS THEY ARE.
6      ;
7      ;OTHERWISE CODE IS A SINGLE LETTER FOLLOWED BY AN OPTIONAL DECIMAL
8      ;NUMBER:
9      ; ON - PRINT OCTAL NUMBER. N REPRESENTS SIZE OF BINARY NUMBER PASSED
10     ;      IN PARAMETER IN BITS. MAY BE IN RANGE 1 TO 32. IF N>16, TWO PARAMETER
11     ;      WORDS ARE USED, OTHERWISE ONLY ONE WORD. LEADING ZEROS ARE PRINTED.
12     ;      N IS ALWAYS SPECIFIED.
13     ; DN - PRINT UNSIGNED DECIMAL NUMBER FROM N BIT PARAMETER. LEADING ZEROS
14     ;      ARE NOT PRINTED. A 16 BIT NUMBER EQUAL TO ZERO WILL PRINT "0".
15     ; HN - PRINT HEX NUMBER FROM PARAMETER OF N BITS. IF N>16 TWO PARAMETERS
16     ;      ARE USED, OTHERWISE ONLY ONE PARAMETER. LEADING ZEROS ARE PRINTED.
17     ; SN - PRINT N SPACES. N ASSUMED TO BE 1.
18     ; NN - START NEW LINE (CR-LF SEQUENCE). N ASSUMED TO BE 1.
19     ; AN - PRINT N ASCII CHARACTERS FROM PARAMETERS, N ASSUMED TO BE 1.
20     ;      N/2 PARAMETER WORDS USED.
21     ; RN - EXECUTE ROUTINE #N. N MUST BE GIVEN AND DEFINED IN HOST PROGRAM.
22     ;
23     ;A NULL CHARACTER MEANS END OF MESSAGE. A NULL AS FIRST CHARACTER IN STRING
24     ;MUST BE IGNORED.
25     ;
26     ;OUTPUT A MESSAGE ACCORDING TO A FORMAT STRING
27     ;
28     ;INPUTS:
29     ;      R2 - ADDRESS OF START OF FORMAT STRING
30     ;      R4 - ADDRESS OF PARAMETERS
31     ;OUTPUTS:
32     ;      R2 AND R4 UPDATED TO END OF STRING AND PARAMETERS
33
34 014766 112201      OSTRNG:  MOVB (R2),R1      ;GET CONTROL CHARACTER
35 014770 001421      BEQ OSTRE      ;EXIT IF NULL CHARACTER
36 014772 012700 015266  MOV #ERRC,R0      ;GET POINTER TO CHARACTER TABLE
37 014776 120110      NCONS:  CMPB R1,(R0)      ;COMPARE CHARACTER WITH TABLE ENTRY
38 015000 001407      BEQ NCONF      ;BRANCH IF MATCH FOUND
39 015002 105720      TSTB (R0),      ;INCREMENT POINTER
40 015004 001374      BNE NCONS      ;CONTINUE SEARCH IF NOT END OF TABLE
41 015006      PNTF ERRME1      ;REPORT BAD CONTROL CHARACTER
42      015006 004137 016226      JSR R1,LPNTF
43      015012 003740      .WORD ERRME1
44      015014 000000      .WORD PNT.CT
45 015016 000406      NCONF:  BR OSTRE      ;GET INCREMENT INTO TABLE
46 015020 162700 015266  SUB #ERRC,R0      ;DOUBLE TO WORD COUNT
47 015024 006300      ASL R0
48 015026 004770 015300  CALL @ERRD(R0)    ;DISPATCH TO PRINT ROUTINE
49 015032 000755      BR OSTRNG      ;GET NEXT
50 015034 000207      OSTRE:  RETURN
    
```

```

1          ;CONTROL CHARACTER WAS A QUOTE. PRINT ALL CHARACTERS TO THE NEXT QUOTE.
2
3 015036 112200          CON.QU: MOVB (R2),R0          ;GET CHARACTER
4 015040 120027 000042  CMPB R0,'"'          ;CHECK IF ENDING QUOTE
5 015044 001403          BEQ CON.QX          ;IF SO, GO GET NEXT CONTROL CHARACTER
6 015046          PRINT R0          ;PRINT THE CHARACTER
7 015052 004737 016054  BR CON.QU          ;CONTINUE PRINTING          CALL CPNT
8 015054 000207          CON.QX: RETURN
9
10         ;CONTROL CHARACTER WAS AN A. PRINT ASCII CHARACTERS FROM PARAMETERS.
11
12 015056 004737 015534  CON.A: CALL GETCNT          ;GET COUNT OF CHARACTERS
13 015062          CON.A1: PRINT (R4).          ;PRINT THE CHARACTER
14 015064 112400          MOVB (R4),R0          ;COUNT THE CHARACTERS          CALL CPNT
15 015070 005301          DEC R1          ;PRINT UNTIL COUNT REACHES ZERO
16 015072 001373          BNE CON.A1          ;CHECK IF R4 NOW ODD
17 015074 032704 000001  BIT #1,R4
18 015100 001401          BEQ CON.A2          ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
19 015102 005204          INC R4          ;NOW GET NEXT CONTROL CHARACTER
20 015104 000207          CON.A2: RETURN
21
22         ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.
23
24 015106 012701 000012  CON.D: MOV #10.,R1          ;LOAD RADIX
25 015112 004737 015612  CALL PNTNUM          ;PRINT NUMBER
26 015116 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
27
28         ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.
29
30 015120 012701 000020  CON.H: MOV #16.,R1          ;LOAD RADIX
31 015124 004737 015612  CALL PNTNUM          ;PRINT NUMBER
32 015130 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER

```

```

1
2
3 015132 012701 000010
4 015136 004737 015612
5 015142 000207
6
7
8
9 015144 004737 015534
10 015150
    015150 112700 000015
    015154 004737 016054
11 015160 005301
12 015162 001372
13 015164 000207
14
15
16
17 015166 004737 015534
18 015172 020127 000010
19 015176 101004
20 015200 060101
21 015202 004771 015244
22 015206 000207
23 015210
    015210 004137 016226
    015214 003740
    015216 000000
24 015220
    015220 012601
25 015222 000207
26
27
28
29 015224 004737 015534
30 015230
    015230 112700 000040
    015234 004737 016054
31 015240 005301
32 015242 001372
33 015244 000207

;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.
CON.O:  MOV @8.,R1          ;LOAD RADIX
        CALL PNTNUM        ;PRINT NUMBER
        RETURN            ;NOW GET NEXT CONTROL CHARACTER

;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.
CON.N:  CALL GETCNT        ;GET COUNT
CON.N1: PRINT @CR          ;PRINT NEW LINE SEQUENCE
                                MOV @CR,R0
                                CALL CPNT

        DEC R1             ;COUNT THE SEQUENCES
        BNE CON.N1
        RETURN            ;NOW GET NEXT CONTROL CHARACTER

;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.
CON.R:  CALL GETCNT        ;GET ROUTINE NUMBER
        CMP R1,@ERRRSZ    ;CHECK IF DEFINED ROUTINE NUMBER
        BHI CON.R1
        ADD R1,R1         ;DOUBLE COUNT TO GET WORD INDEX
        CALL @ERRRTB-2(R1);CALL ROUTINE
        RETURN            ;NOW GET NEXT CONTROL CHARACTER
CON.R1: PNTF ERRME1       ;REPORT BAD MESSAGE STRING
                                JSR R1,L.PNTF
                                .WORD ERRME1
                                .WORD PNT.CT

        POP R1            ;FIX THE STACK
        RETURN            MOV (SP),R1

;CONTROL CHARACTER WAS AN S. PRINT SPACES.
CON.S:  CALL GETCNT        ;GET COUNT
CON.S1: PRINT '<@>'       ;PRINT A SPACE
                                MOV @',R0
                                CALL CPNT

        DEC R1             ;COUNT THE SPACES
        BNE CON.S1
        RETURN            ;NOW GET NEXT CONTROL CHARACTER

```

```

1      ;ERROR ROUTINE DISPATCH TABLE
2
3 015246 015320  ERRRTB: .WORD CALRE          ;NOT USED
4 015250 015320          .WORD CALRE          ;NOT USED
5 015252 015320          .WORD CALRE          ;NOT USED
6 015254 015332          .WORD CALR4         ;PRINT BASIC LINE WITHOUT UDA ADDRESS
7 015256 015406          .WORD CALR5         ;PRINT BASIC LINE WITH UDA ADDRESS
8 015260 015464          .WORD CALR6         ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
9 015262 015500          .WORD CALR7         ;PRINT "REPLACE UDA MODULE M7485"
10 015264 015516         .WORD CALR8         ;PRINT " UDASA CONTAINS xxxxxx"
11          000010         ERRRSZ=<.-ERRRTB>/2
12
13      ;BUILD TWO TABLES
14      : FIRST CONTAINING CONTROL CHARACTERS
15      : SECOND CONTAINING ROUTINE ADDRESSES
16
17      .MACRO BUILD
18          ENTRY ".CON.QU
19          ENTRY A.CON.A
20          ENTRY D.CON.D
21          ENTRY H.CON.H
22          ENTRY O.CON.O
23          ENTRY N.CON.N
24          ENTRY R.CON.R
25          ENTRY S.CON.S
26      .ENDM

```

```

1
2
3
4
5
6
7
8
9 015266
   015266      042
   015267      101
   015270      104
   015271      110
   015272      117
   015273      116
   015274      122
   015275      123
10 015276      000
11
12
13
14
15
16
17
18
19
20
21 015300
   015300      015036
   015302      015056
   015304      015106
   015306      015120
   015310      015132
   015312      015144
   015314      015166
   015316      015224

```

```

;HERE IS FIRST TABLE
.MACRO ENTRY ARG1,ARG2
  .LIST
  .BYTE ' ARG1
  .NLIST
.ENDM
ERRC:  BUILD
       .BYTE ' "
       .BYTE ' A
       .BYTE ' D
       .BYTE ' H
       .BYTE ' O
       .BYTE ' N
       .BYTE ' R
       .BYTE ' S
       .BYTE 0
       .EVEN

```

;FOLLOW WITH A NULL BYTE

```

;HERE IS SECOND TABLE
.MACRO ENTRY ARG1,ARG2
  .LIST
  .WORD ARG2
  .NLIST
.ENDM
ERRD:  BUILD
       .WORD CON.QU
       .WORD CON.A
       .WORD CON.D
       .WORD CON.H
       .WORD CON.O
       .WORD CON.N
       .WORD CON.R
       .WORD CON.S

```

```

1
2
3
4 015320      ;PRE-PROGRAMMED ROUTINES 1, 2 AND 3
   015320      ;NOT USED - PRINTS ERROR MESSAGE
   015324      CALRE:  PNTF ERRME1
   015326      ;PRINT ERROR MESSAGE
   015326      JSR R1,LPNTF
5 015330      .WORD ERME1
   000000      .WORD PNT,CT
   000207      RETURN

```



```

1      ;PRE-PROGRAMMED ROUTINE 4
2      ;PRINT BASIC LINE FOR HOST PROGRAM ERROR WITHOUT UDA ADDRESS
3      ;THEN SWITCH TO EXTENDED FORMAT.
4
5      CALR4:  PNTB BASLN,@BASNO,@BAS,@BAS,@BAS
              MOV @BAS,-(SP)
              MOV @BAS,-(SP)
              MOV @BAS,-(SP)
              MOV @BASNO,-(SP)
              JSR R1,LPNTB
              .WORD BASLN
              .WORD PNT,CT
6      015332 012746 004223
7      015336 012746 004223
8      015342 012746 004223
9      015346 012746 004150
015352 004137 016236
015356 004224
015360 000010
6 015362 004737 020242      CALL RNTIME
7 015366                PRINT @CR
015366 112700 000015      MOV @CR,R0
015372 004737 016054      CALL CPNT
8 015376 012737 016154 003224      MOV @PX,PType
9 015404 000207                RETURN

```

```

1      ;PRE-PROGRAMMED ROUTINE 5
2      ;PRINT BASIC LINE FOR MOST PROGRAM ERROR WITH UDA ADDRESS
3      ;THEN SWITCH TO EXTENDED FORMAT
4
5      CALR5:  PNTB BASLN,@BASNO,@BASL2,(R5),@BAS,@BAS
6
7      MOV @BAS,-(SP)
      MOV @BAS,-(SP)
      MOV (R5),-(SP)
      MOV @BASL2,-(SP)
      MOV @BASNO,-(SP)
      JSR R1,LPNTB
      .WORD BASLN
      .WORD PNT.CT
8
9      CALL RNTIME
      PRINT @CR
      MOV @CR,R0
      CALL CPNT
10
11     MOV @PX,PTYPE
12     RETURN
13
14     015406 012746 004223
15     015412 012746 004223
16     015416 011546
17     015420 012746 004167
18     015424 012746 004150
19     015430 004137 016236
20     015434 004224
21     015436 000012
22     6 015440 004737 020242
23     7 015444
24     015444 112700 000015
25     015450 004737 016054
26     8 015454 012737 016154 003224
27     9 015462 000207

```

```
1 ;PRE-PROGRAMMED ROUTINE 6
2 ;CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
3
4 015464 CALR6: PUSH R2 ;SAVE CURRENT STRING POINTER
5 015464 010246 ;MOV R2, -(SP)
6 015466 012402 ;GET NEW STRING POINTER
7 015470 004737 014766 ;CALL OSTRING ;OUTPUT USING THIS STRING
8 015474 012602 ;POP R2 ;GET OLD POINTER BACK
9 015476 000207 ;MOV (SP), R2 ;NOW CONTINUE THE OLD STRING
10 RETURN
```

```
1  
2  
3  
4 015500 ;PRE-PROGRAMMED ROUTINE 7  
015500 010246 ;PRINT "REPLACE UDA MODULE M7485"  
5 015502 012702 011357  
6 015506 004737 014766 CALR7: PUSH R2  
7 015512 012602 MOV #XFRU,R2  
015514 000207 CALL OSTRNG  
8 015514 000207 POP R2  
MOV R2,-(SP)  
MOV (SP),R2  
RETURN
```

```
1 ;PRE-PROGRAMMED ROUTINE 8
2 ;PRINT " UDASA CONTAINS XXXXXX"
3
4 015516 CALR8: PUSH R2
5 015516 010246 MOV R2,-(SP)
6 015520 012702 011326 MOV #XSA,R2
7 015524 004737 014766 CALL OSTRNG
8 015530 POP R2
9 015532 012602 MOV (SP),R2
10 015532 000207 RETURN
```

```

1      ;GETCNT
2      ;
3      ;GET COUNT IN NEXT CHARACTERS OF STRING POINTED TO BY R2.
4      ;NUMBER WILL BE IN DECIMAL. IF NO NUMBER, RETURN A
5      ;DEFAULT OF 1.
6      ;
7      ;INPUTS:
8      ;   R2 - POINTER TO ASCII STRING
9      ;
10     ;OUTPUTS:
11     ;   R1 - NUMBER READ OR A ONE
12     ;   R2 - POINTING TO CHARACTER AFTER NUMBER
13     GETCNT: PUSH R0
14     015534      010046      MOV R0,-(SP)
15     015536      005001
16     015540      121227      000060      CLR R1
17     015544      103415      GETCNX: CMPB (R2),@'0
18     015546      121227      000071      BLO GETCDN
19     015552      101012      CMPB (R2),@'9
20     015556      010100      BHI GETCDN
21     015560      006301      ASL R1
22     015562      006301      MOV R1,R0
23     015564      060001      ASL R1
24     015566      112200      ADD R0,R1
25     015570      162700      000060      MOVB (R2),R0
26     015574      060001      SUB @'0,R0
27     015576      000760      ADD R0,R1
28     015600      005701      BR GETCNX
29     015602      001001      GETCDN: TST R1
30     015604      005201      BNE GETCXX
31     015606      012600      INC R1
32     015610      000207      GETCXX: POP R0
                                     RETURN
                                     MOV (SP),R0

```

```

1      ;PNTNUM
2      ;
3      ;PRINT A NUMBER
4      ;
5      ;INPUTS:
6      ;   R1 - RADIX OF NUMBER
7      ;   R2 - ASCII STRING TO COUNT OF BITS IN NUMBER
8      ;   R4 - POINTER TO NUMBER (LOW WORD)
9      ;OUTPUTS:
10     ;   NUMBER IS PRINTED. LEADING ZEROS ARE PRINTED EXCEPT FOR
11     ;   DECIMAL NUMBERS.
12     ;   R0 - CONTENTS DESTROYED
13
14 015612 010100      PNTNUM: MOV R1,R0          ;SAVE RADIX
15 015614 004737 015534      CALL GETCNT        ;GET COUNT OF BITS
16 015620      PNTNUS: PUSH <R2,R3,R5>
17     015620 010246      MOV R2,-(SP)
18     015622 010346      MOV R3,-(SP)
19     015624 010546      MOV R5,-(SP)
20 015626 012403      MOV (R4),R3          ;GET ONE PARAMETER WORD
21 015630 005005      CLR R5            ;CLEAR STORAGE FOR OTHER
22 015632 020127 000020      CMP R1,#16        ;MORE THAN 16 BITS IN NUMBER?
23 015636 003401      BLE 1$
24 015640 012405      MOV (R4),R5          ;YES, GET SECOND PARAMETER WORD
25 015642      1$: PUSH R4
26     015642 010446      MOV R4,-(SP)
27 015644 010504      MOV R5,R4          ;PUT HIGH WORD IN R4
28 015646 012702 000020      MOV #16,R2        ;COMPUTE BITS NOT WANTED
29 015652 160102      SUB R1,R2          ;BY SUBTRACTING BITS TO USE
30 015654 002002      BGE 2$            ;FROM 16.
31 015656 062702 000020      ADD #16,R2        ;IF NEGATIVE, ADD 16 FOR FIRST WORD
32 015662 001414      BEQ 6$            ;IF ZERO, NO BITS NEED BE CLEARED
33 015664 012705 100000      MOV #BIT15,R5    ;START MASK WITH SIGN BIT SET
34 015670 005302      3$: DEC R2        ;COUNT BITS IN MASK
35 015672 001402      BEQ 4$
36 015674 006205      ASR R5            ;SHIFT MORE BITS TO RIGHT
37 015676 000774      BR 3$
38 015700 020127 000020      4$: CMP R1,#16.     ;MORE THAN 16 BITS IN NUMBER?
39 015704 003402      BLE 5$
40 015706 040504      BIC R5,R4        ;YES, CLEAR IN HIGH WORD
41 015710 000401      BR 6$
42 015712 040503      5$: BIC R5,R3        ;NO, CLEAR IN LOW WORD
43 015714 004737 016332      6$: CALL DIVIDE    ;DIVIDE BY RADIX IN R0
44 015720      PUSH R5          ;PUSH REMAINDER ON STACK
45 015722 010546      MOV R5,-(SP)
46 015724 005202      INC R2            ;COUNT DIGITS ON STACK
47 015726 005703      TST R3           ;CHECK IF QUOTIENT IS ZERO
48 015728 001372      BNE 6$
49 015730 005704      TST R4
50 015732 001370      BNE 6$
    
```

1	015734	020027	000012		CMP R0,#10.		; IF RADIX IS DECIMAL
2	015740	001423			BEQ 10\$		; JUST GO PRINT DIGITS ON STACK
3	015742	010103			MOV R1,R3		; OTHERWISE COMPUTE NUMBER OF LEADING ZEROS
4	015744	162700	000014		SUB #12.,R0		; DIVIDEND IS BITS IN NUMBER
5	015750	003002			BGT 7\$		; DIVISOR IS BITS PER DIGIT PRINTED
6	015752	012700	000003		MOV #3,R0		; (3 OR 4)
7	015756	004737	016332	7\$:	CALL DIVIDE		
8	015762	005705			TST R5		; IF REMAINDER NOT ZERO
9	015764	001401			BEQ 8\$		; INCREMENT QUOTIENT
10	015766	005203			INC R3		
11	015770	160203		8\$:	SUB R2,R3		; SUBTRACT DIGITS ON STACK
12	015772	001406			BEQ 10\$		; NO LEADING ZEROS IF ZERO
13	015774			9\$:	PRINT #'0		; PRINT A ZERO
	015774	112700	000060				MOV #'0,R0
	016000	004737	016054				CALL CPNT
14	016004	005303			DEC R3		
15	016006	001372			BNE 9\$		; REPEAT UNTIL COUNT REACHES ZERO
16							
17	016010			10\$:	POP R5		; GET CHARACTER FROM STACK
	016010	012605					MOV (SP),R5
18	016012	062705	000060		ADD #'0,R5		; CONVERT TO ASCII DIGIT
19	016016	020527	000071		CMP R5,#'9		; IF GREATER THAN A 9
20	016022	003402			BLE 11\$		; CONVERT TO A OR HIGHER
21	016024	062705	000007		ADD #'A-'9-1,R5		; FOR HEX DIGIT
22	016030			11\$:	PRINT R5		; PRINT THE CHARACTER
	016030	110500					MOV R5,R0
	016032	004737	016054				CALL CPNT
23	016036	005302			DEC R2		; REPEAT FOR ALL DIGITS
24	016040	001363			BNE 10\$		; ON STACK
25	016042				POP <R4,R5,R3,R2>		
	016042	012604					MOV (SP),R4
	016044	012605					MOV (SP),R5
	016046	012603					MOV (SP),R3
	016050	012602					MOV (SP),R2
26	016052	000207			RETURN		



```

1      ;PRINT ONE CHARACTER
2
3      ;CALL WITH MACRO PRINT
4
5 016054 110037 003226      CPNT:  MOV B R0,ERRCHR
6 016060      010146      PUSH R1
7 016062 012701 003676      MOV @ERRONE,R1
8 016066 120027 000015      CMP B R0,@CR
9 016072 001002      BNE 1$
10 016074 012701 003701      MOV @ERRNL,R1
11 016100 000177 165120      1$:  JMP @PTYPE
12 016104      012746 003226      PF:  PRINTF R1,@ERRCHR
13 016126 000435      BR CPNTX
14 016130      012746 003226      PB:  PRINTB R1,@ERRCHR
15 016152 000423      BR CPNTX
16 016154      012746 003226      PX:  PRINTX R1,@ERRCHR
17 016176 000411      BR CPNTX
18 016200      012746 003226      PS:  PRINTS R1,@ERRCHR
19 016222      012601      CPNTX: POP R1
20 016224 000207      RETURN

```

```

MOV R1,-(SP)

MOV @ERRCHR,-(SP)
MOV R1,-(SP)
MOV @2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD @6,SP

MOV @ERRCHR,-(SP)
MOV R1,-(SP)
MOV @2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD @6,SP

MOV @ERRCHR,-(SP)
MOV R1,-(SP)
MOV @2,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD @6,SP

MOV @ERRCHR,-(SP)
MOV R1,-(SP)
MOV @2,-(SP)
MOV SP,R0
TRAP C$PNTS
ADD @6,SP

MOV (SP),R1

```

```

1          ;PRINT FORMATTED MESSAGE
2
3          ;CALL WITH MACRO PNT, PNTF, PNTB, PNTX, OR PNTS
4
5 016226 012737 016104 003224 LPNTF:  MOV @PF,PType
6 016234 000413                BR LPNT
7 016236 012737 016130 003224 LPNTB:  MOV @PB,PType
8 016244 000407                BR LPNT
9 016246 012737 016154 003224 LPNTX:  MOV @PX,PType
10 016254 000403                BR LPNT
11 016256 012737 016200 003224 LPNTS:  MOV @PS,PType
12 016264                LPNT:  PUSH <R2,R3,R4,R5>
    016264 010246                MOV R2,-(SP)
    016266 010346                MOV R3,-(SP)
    016270 010446                MOV R4,-(SP)
    016272 010546                MOV R5,-(SP)
13 016274 012102                MOV (R1),R2
14 016276 010604                MOV SP,R4
15 016300 062704 000012        ADD #10.,R4
16 016304                PUSH R1
    016304 010146                MOV R1,-(SP)
17 016306 004737 014766        CALL OSTRNG
18 016312                POP <R0,R5,R4,R3,R2,R1>
    016312 012600                MOV (SP),R0
    016314 012605                MOV (SP),R5
    016316 012604                MOV (SP),R4
    016320 012603                MOV (SP),R3
    016322 012602                MOV (SP),R2
    016324 012601                MOV (SP),R1
19 016326 062006                ADD (R0),SP
20 016330 000110                JMP @R0
    ;ADJUST STACK POINTER OVER ARGUMENTS
    ;RETURN

```

```

1      ;DIVIDE
2      ;
3      ;DIVIDE A 32 BIT UNSIGNED NUMBER BY A 16 BIT UNSIGNED NUMBER.
4      ;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
5      ;WILL NOT CHECK FOR DIVIDE BY ZERO.
6      ;
7      ;INPUTS:
8      ;   R3 - LOW 16 BITS OF DIVIDEND
9      ;   R4 - HIGH 16 BITS OF DIVIDEND
10     ;   R0 - DIVISOR
11     ;OUTPUTS:
12     ;   R3 - LOW 16 BITS OF QUOTIENT
13     ;   R4 - HIGH 16 BITS OF QUOTIENT
14     ;   R5 - REMAINDER
15
16 016332 DIVIDE: PUSH R2
17 016332 010246          MOV R2,-(SP)
18 016334 012702 000040  ;SET UP SHIFT COUNT
19 016340 005005          ;START WITH ZERO REMAINDER
20 016342 006303          ;SHIFT LEFT INTO R5
21 016344 006104
22 016346 006105
23 016350 020005          ;WILL DIVISOR GO INTO REMAINDER
24 016352 101002          ;ONLY SUBTRACT IF IT WILL
25 016354 160005          ;SUBTRACT DIVISOR
26 016356 005203          ;PUT A ONE INTO QUOTIENT
27 016360 005302          ;COUNT THE SHIFTS
28 016362 001367
29 016364 012602
30 016366 000207          RETURN

```

```

1      ;LOADDM
2
3      ;LOAD AND START A DM PROGRAM INTO A CONTROLLER
4
5      ;INPUTS:
6      ;       R5 - CONTROLLER TABLE ADDRESS
7      ;       DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
8
9      ;OUTPUTS:
10     ;       IF LOAD SUCCEEDS - Z CLEAR
11     ;                   CONTROLLER TABLE MARKED LOADED
12     ;       IF ERROR - Z SET
13 016370 013701 002164      LOADDM: MOV DMPROG,R1      ;GET STORAGE ADDRESS OF DM PROGRAM
14 016374 116165 000021 000044  MOVB DMTMO(R1),C.TOT(R5)  ;GET TIMEOUT VALUE
15 016402 105065 000045      CLR B C.TOT+1(R5)
16 016406 016504 000004      MOV C.VEC(R5),R4      ;GET VECTOR OF UDA
17 016412      AND CT.VEC,R4
18 016412 042704 177000      MOV R5,R1      BIC #C<CT.VEC>,R4
19 016416 010501      ;GET INTERRUPT SERVICE LINK
20 016420 062701 000010      ADD #C.JSR,R1
21 016424      SETVEC R4,R1,#PRI07      ;SET UP INTERRUPT VECTOR
22 016424 012746 000340      MOV #PRI07,-(SP)
23 016430 010146      MOV R1,-(SP)
24 016432 010446      MOV R4,-(SP)
25 016434 012746 000003      MOV #3,-(SP)
26 016440 104437      TRAP C$VEC
27 016442 062706 000010      ADD #10,SP
28
29      ;INITIALIZE UDA WITH SMALLEST
30      ; PING BUFFER AND INTERRUPTS ENABLED
31      ;BRANCH IF AN ERROR
32
33      CALL UDAINIT
34      BEQ LOADER

```

1	016454	012700	000002		MOV @OP.ESP,R0	;BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET
2	016460	004737	016570		CALL BLUCMD	
3	016464	013764	002164	000124	MOV DMPROG,HC.CPK.P.UADR(R4)	;LOAD MAIN PROGRAM ADDRESS
4	016472	017764	163466	000120	MOV @DMPROG,HC.CPK.P.BCNT(R4)	; AND SIZE
5	016500	013764	002164	000140	MOV DMPROG,HC.CPK.P.OVRL(R4)	;LOAD OVERLAY ADDRESS
6	016506	067764	163452	000140	ADD @DMPROG,HC.CPK.P.OVRL(R4)	
7	016514	004737	016654		CALL SDCMD	;SEND COMMAND TO UDA
8	016520	004737	016774		CALL WAITMS	;WAIT FOR MESSAGE RESPONSE
9	016524	001417			BEG LOADER	;ABORT IF NO RESPONSE
10	016526	032764	000037	000032	BIT @ST.MSK,HC.MPK.P.STS(R4)	;CHECK FOR ERRORS
11	016534	001007			BNE LOADE1	
12	016536	042765	000024	000014	BIC @CT.CMD.CT.REQ,C.FLG(R5)	;CLEAR COMMAND OUTSTANDING FLAG
13	016544	052765	000002	000014	BIS @CT.RN,C.FLG(R5)	;SET DM PROGRAM RUNNING FLAG
14	016552	000207			RETURN	

1  
2  
3 016554  
016554 104455  
016556 000042  
016560 000000  
016562 012234  
4 016564 000264  
5 016566 000207

;UDA FAILED TO DOWNLINE LOAD DM PROGRAM

LOADE1: ERRDF 34,,ERR034

LOADER: SEZ  
RETURN

TRAP	C\$ERDF
.WORD	34
.WORD	0
.WORD	ERR034

;SET Z TO INDICATE ERROR OCCURRED

```

1      ;BLDCMD
2      ;
3      ;BUILD A COMMAND IN COMMAND PACKET
4      ;
5      ;INPUTS:
6      ;      R5 - CONTROLLER TABLE ADDRESS
7      ;      R0 - COMMAND CODE
8      ;
9      ;OUTPUTS:
10     ;      R4 - ADDRESS OF HOST COMM AREA
11     ;      COMMAND PACKET CONTAINING REF NUMBER AND OPCODE, ALL OTHER FIELDS CLEARED.
12     ;      CMD REFERENCE NUMBER IN CONTROLLER TABLE INCREMENTED AND RESULT
13     ;      IN COMMAND PACKET.
14     ;      R0 - CONTENTS DESTROYED
15     BLDCMD: PUSH <R1,R0>
16     016570      016570      010146      MOV R1,-(SP)
17     016572      010046      MOV R0,-(SP)
18     016574      016504      000016      MOV C.RING(R5),R4 ;GET ADDRESS OF HOST COMM AREA
19     016600      010400      MOV R4,R0 ;COPY TO R0
20     016602      062700      000100      ADD #HC.CEV,R0 ;COMPUTE ADDRESS OF COMMAND ENVELOPE
21     016606      012720      000060      MOV #HC.PSZ,(R0) ;LOAD PACKET LENGTH
22     016612      012701      001000      MOV #DUP,R1 ;LOAD DIAG CIRCUIT IDENTIFIER
23     016616      022716      000031      CMP #OP.MWR,(SP) ;IF CODE IS MAINTENANCE WRITE
24     016622      001002      BNE BLDC0 ; GET OTHER CIRCUIT IDENTIFIER
25     016624      012701      177777      MOV #DIAG,R1
26     016630      010120      BLDC0: MOV R1,(R0) ;PUT IDENTIFIER INTO PACKET
27     016632      012701      000030      MOV #<HC.PSZ>/2,R1 ;GET WORDS TO CLEAR
28     016636      005020      BLDC1: CLR (R0) ;CLEAR PACKET
29     016640      005301      DEC R1
30     016642      001375      BNE BLDC1
31     016644      012664      000114      POP HC.CPK.P.OPCD(R4) ;PUT OPCODE IN PACKET
32     016650      012601      POP R1 ;RESTORE R1
33     016652      000207      RETURN ;MOV (SP)+,R1

```

```

1      ;SNDCMD
2      ;
3      ; SEND A COMMAND TO THE UDA.
4      ; MARK BOTH PACKETS AVAILABLE TO THE
5      ; UDA. SET COMMAND ISSUED BIT IN CONTROLLER TABLE AND INITIALIZE
6      ; TIMEOUT COUNTER.
7      ;
8      ; INPUTS:
9      ; R5 - CONTROLLER TABLE ADDRESS
10     ; OUTPUTS:
11     ; R4 - ADDRESS OF HOST COMM AREA
12
13
14 016654 SNDCMD: PUSH <R0,R1>
15 016654 010046 MOV R0,-(SP)
16 016656 010146 MOV R1,-(SP)
17 016660 016504 000016 MOV C.RING(R5),R4 ;LOAD R4 WITH HOST COMM AREA ADDRESS
18 016664 005265 000052 INC C.REF(R5) ;INCREMENT CMD REFERENCE NUMBER
19 016670 016564 000052 000104 MOV C.REF(R5),HC.CPK.P.CRF(R4) ;PUT IN PACKET
20 016676 012764 140000 000006 MOV @RG.OWN.RG.FLG,HC.MCT(R4) ;MARK MESSAGE PACKET AVAILABLE
21 016704 012764 100000 000012 MOV @RG.OWN,HC.CCT(R4) ;MARK COMMAND TO UDA
22 016712 005775 000000 TST @R5 ;TELL UDA COMMAND IS THERE
23 016716 052765 000004 000014 BIS @CT.CMD,C.FLG(R5) ;MARK COMMAND ISSUED
24 016724 POP <R1,R0>
25 016724 012601 MOV (SP)+,R1
26 016726 012600 MOV (SP)+,R0
27 016730 000207 RETURN

```



```

1      :CLRBUF
2      :
3      :CLEAR THE SPECIFIED DATA BUFFER IN THE HOST COMM AREA
4      :AND LOAD BUFFER DESCRIPTOR IN COMMAND PACKET TO THE BUFFER
5      :
6      :INPUTS:
7      :   R5 - CONTROLLER TABLE ADDRESS
8      :   R4 - ADDRESS OF HOST COMM AREA
9      :   R0 - OFFSET INTO HOST COMM AREA TO DATA BUFFER
10     :OUTPUTS:
11     :   DATA BUFFER CLEARED
12     :   COMMAND PACKET POINTING TO BUFFER
13     :   BYTE COUNT SET TO SIZE OF BUFFER
14     :   R4 - ADDRESS OF DATA BUFFER
15
16 016732 CLRBUF: PUSH <R0,R1>
17 016732 010046
18 016734 010146
19 016736 060400
20 016740 010064 000124
21 016744 012764 000244 000120
22 016752 010004
23 016754 012701 000122
24 016760 005020
25 016762 005301
26 016764 001375
27 016766 012601
28 016770 012600
29 016772 000207

      ADD R4,R0
      MOV RO,HC.CPK+P.UADR(R4)
      MOV #HC.BSZ,HC.CPK+P.BCNT(R4)
      MOV RO,R4
      MOV #HC.BSZ/2,R1
      CLRBFL: CLR (R0)
      DEC R1
      BNE CLRBFL
      POP <R1,R0>

      MOV RO,-(SP)
      MOV R1,-(SP)
      ;ADD START OF HOST COMM AREA TO OFFSET
      ;PUT BUFFER ADDRESS IN COMMAND PACKET
      ;PUT SIZE OF BUFFER IN COMMAND PACKET
      ;PUT BUFFER ADDRESS IN R4
      ;GET SIZE OF BUFFER IN WORDS
      ;CLEAR ALL THE WORDS

      MOV (SP),R1
      MOV (SP),R0

      RETURN
    
```

```

1      ;WAITMS
2      ;
3      ;WAIT FOR UDA TO RESPOND WITH A MESSAGE PACKET
4      ;
5      ;INPUTS:
6      ;      R5 - ADDRESS OF CONTROLLER TABLE
7      ;OUTPUTS:
8      ;      Z CLEAR IF NO ERROR
9      ;      Z SET IF ERROR, MESSAGE PRINTED
10     ;
11     016774      WAITMS: PUSH <R0,R1>
12     016774      010046
13     016776      010146
14     017000      012700      000036
15     017004      010501
16     017006      062701      000040
17     017012      004737      017166
18     017016      011500
19     017020      032765      000010      000014
20     017026      001030
21     017030      016001      000002
22     017034      001034
23     017036      017036
24     017040      104422
25     017044      005737      003210
26     017046      001764
27     017054      023765      003222      000042
28     017056      101005
29     017060      001357
30     017066      023765      003220      000040
31     017070      103753
32     017070      104455
33     017072      000044
34     017074      000000
35     017076      012242
36     017100      POP <R1,R0>
37     017102      012601
38     017104      000264
39     017106      000207
40     SEZ
41     RETURN

;SET TIME OUT VALUE OF 30 SECONDS
;POINT TO TIME OUT COUNTER

;GET ADDRESS OF UDAIP REGISTER
;LOOK IF INTERRUPT OCCURRED
;BRANCH IF SO
;LOOK AT UDASA REGISTER
;BRANCH IF ERROR CODE PRESENT

TRAP C$BRK
;SEE IF A CLOCK ON SYSTEM

;CHECK IF TIMEOUT HAS HAPPENED

TRAP C$ERDF
.WORD 36
.WORD 0
.WORD ERRO36
MOV RO,-(SP)
MOV R1,-(SP)
MOV (R5),R0
MOV #30.,R0
MOV R5,R1
ADD #C.TO,R1
CALL SETTO
BIT #CT.MSG,C.FLG(R5)
BNE 3$
MOV 2(R0),R1
BNE 4$
BREAK
TST KW.CSR
BEQ 1$
CMP KW.EL+2,C.TOH(R5)
BHI 2$
BNE 1$
CMP KW.EL.C.TO(R5)
BLO 1$
ERRDF 36,.,ERRO36

```

```

1 017110 042765 000010 000014 3$: BIC @CT.MSG.C.FLG(R5) ;CLEAR MESSAGE RECEIVED FLAG
2 017116 012601 POP <R1,R0> MOV (SP),R1
017120 012600 MOV (SP),R0
3 017122 000244 CLZ ;GIVE NO ERROR RETURN
4 017124 000207 RETURN
5 017126 104455 4$: ERRDF 37,,ERR037 TRAP C$ERDF
017130 000045 .WORD 37
017132 000000 .WORD 0
017134 012254 .WORD ERR037
6 017136 POP <R1,R0> MOV (SP),R1
017136 012601 MOV (SP),R0
017140 012600
7 017142 000264 SEZ
8 017144 000207 RETURN

```

```

1      ;NXMI
2      ;
3      ;NON-EXISTANT MEMORY SERVICE ROUTINE
4      ;
5      ;INPUTS:
6      ;      NXMAD SET TO ZERO
7      ;OUTPUTS:
8      ;      NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
9
10     017146      BGNSRV NXMI
11     017146
12     017146      012737 177777 002200      MOV e-1,NXMAD
13
14     017154      ENDSRV
15     017154
16     017154      000002

```

NXMI::

L10031: RTI

```

1      ;UDASRV
2      ;
3      ;UDA INTERRUPT SERVICE ROUTINE. MARKS UDA CONTROLLER TABLE THAT AN
4      ;INTERRUPT HAS BEEN RECEIVED.
5      ;
6      ;THIS ROUTINE IS CALLED BY A [JSR RO,UDASRV] INSTRUCTION FROM WITHIN
7      ;THE CONTROLLER TABLE. THE PC STORED IN RO IS THE ADDRESS OF THE C.FLG
8      ;WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
9      ;OF RO FOLLOWED BY THE INTERRUPTED PC AND PS.
10     ;
11     ;INPUTS:
12     ;   RO - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
13     ;   STACK - SAVED CONTENTS OF RO
14     ;OUTPUTS:
15     ;   CT.CMD CLEARED AND CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
16     ;   RO - RESTORED FROM STACK
17
18     BGNSRV UDASRV
19     017156      052710  000010      BIS #CT.MSG,(RO)      ;SET CT.MSG
20     017162      012600                      POP RO              ;RESTORE RO
21     017164
22     017164
23     017164      000002
24
25     ENDSRV
26
27     UDASRV::
28
29     L10032:
30     RTI
31
32     MOV (SP)+,RO

```

```

1      ;SETTO
2
3      ;SET TIMEOUT COUNTER TO SOME NUMBER OF SECONDS FROM CURRENT TIME.
4
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     ;COMPUTE CLOCK TICKS TIL TIMEOUT
14     SETTO:  PUSH <R2,R3>
15             MOV R2,-(SP)
16             MOV R3,-(SP)
17
18             CLR R2
19             MOV KW,HZ,R3
20             ;CLEAR PRODUCT
21             ;GET MULTIPLICAND
22             ;SHIFT MULTIPLIER TO RIGHT
23             ;IF A ONE BIT SHIFTED OUT
24             ; ADD MULTIPLICAND TO PRODUCT
25             ;DOUBLE THE MULTIPLICAND
26
27             SET00:  ASR R0
28                     BCC SET01
29                     ADD R3,R2
30                     ;CONTINUE UNTIL MULTIPLIER IS ZERO
31
32             SET01:  ASL R3
33                     TST R0
34                     BNE SET00
35
36             ;GET CURRENT TIME
37
38             SET02:  MOV KW,EL,R0
39                     MOV KW,EL+2,R3
40                     CMP R0,KW,EL
41                     BNE SET02
42
43             ;ADD TIME TIL TIMEOUT
44
45             ADD R2,R0
46             ;ADD
47             ADC R3
48
49             ;PUT RESULT IN STORAGE
50
51             MOV R0,(R1)+
52             MOV R3,(R1)
53
54             POP <R3,R2>
55
56             MOV (SP)+,R3
57             MOV (SP)+,R2
58
59             RETURN
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
    
```

```

14 017166
   017166 010246
   017170 010346
15 017172 005002
16 017174 013703 003216
17 017200 006200
18 017202 103001
19 017204 060302
20 017206 006303
21 017210 005700
22 017212 001372
23
24
25
26 017214 013700 003220
27 017220 013703 003222
28 017224 020037 003220
29 017230 001371
30
31
32
33 017232 060200
34 017234 005503
35
36
37
38 017236 010021
39 017240 010311
40
41 017242
   017242 012603
   017244 012602
42 017246 000207
    
```

```

1      ;UDAINI
2      ;
3      ;FUNCTIONAL DESCRIPTION:
4      ;   SUBROUTINE TO INITIALIZE A UDA AND BRING IT ON-LINE.
5      ;   ALL STEPS ARE CHECKED. AN ERROR MESSAGE IS REPORTED IF ANY ERROR
6      ;   DETECTED.
7      ;
8      ;INPUTS:
9      ;   R5 - ADDRESS OF CONTROLLER TABLE.
10     ;IMPLICIT INPUTS:
11     ;   C.RING(R5) - ADDRESS GIVEN TO UDA AS START OF RING BUFFER.
12     ;   LENGTH OF RING STRUCTURE IS ONE ENTRY EACH.
13     ;OUTPUTS:
14     ;   CONDITION Z - SET IF ANY ERROR REPORTED. CLEAR IF NO ERROR.
15     ;   R4 - ADDRESS OF UDAIP REGISTER IN UDA
16     ;   R5 - UNCHANGED.
17
18     ;FILL MOST COMMUNICATION AREA WITH ALL ONES
19
20 017250 016502 000016      UDAINI: MOV C.RING(R5),R2                ;GET FIRST ADDRESS OF RING BUFFER
21 017254 012703 000006      MOV #<HC.RSZ*2+HC.ISZ>/2,R3          ;GET SIZE OF RING BUFFER
22 017260 012722 177777      UDAI1L: MOV #-1,(R2)                ;WRITE ONES TO BUFFER
23 017264 005303              DEC R3                                    ;COUNT THE WORDS IN BUFFER
24 017266 003374              BGT UDAI1L                               ;LOOP UNTIL ENTIRE BUFFER WRITTEN
25
26     ;DO THE INITIALIZATION
27
28 017270 004737 017534      CALL UDAIST                            ;DO FIRST THREE STEPS
29 017274 103515              BCS UDAIEX                               ;GET OUT IF UDA MICROCODE REPORTED FAILURE
30 017276 012364 000002      MOV (R3),2(R4)                          ;WRITE NEXT WORD TO UDASA REGISTER
31 017302 012703 000310      MOV #200,,R3                             ;GET TRY COUNTER
32 017306 016402 000002      UDAI1A: MOV 2(R4),R2                       ;LOOK AT UDASA
33 017312 001410              BEQ UDAI1C
34 017314 100005              BPL UDAI1B
35 017316              ERRDF 24,,ERR024
36 017316 104455              TRAP C$ERDF
37 017320 000030              .WORD 24
38 017322 000000              .WORD 0
39 017324 012130              .WORD ERR024
40
41 017326 000500              BR UDAIEX
42 017330 005303      UDAI1B: DEC R3
43 017332 001365      BNE UDAI1A
44 017334 010264 000002      UDAI1C: MOV R2,2(R4)                ;WRITE 0 TO UDASA (PURGE)
45 017340 011402      MOV (R4),R2                ;READ FROM UDAIP (POLL)
46 017342 004737 020060      CALL UDARSP                            ;WAIT FOR STEP OR ERROR BIT
47 017346 103470      BCS UDAIEX                               ;GET OUT IF UDA MICROCODE REPORTED FAILURE
48 017350 010203      MOV R2,R3                                    ;GET COPY OF MODEL CODE
49 017352 042703 177417      BIC #<C<SA.CNT>,R3                       ;CLEAR OTHER BITS
50 017356 006203      ASR R3                                    ;MOVE TO RIGHT OF REGISTER
51 017360 006203      ASR R3
52 017362 006203      ASR R3
53 017364 006203      ASR R3
54 017366 042702 177760      BIC #<C<SA.MCV>,R2                       ;MICROCODE VERSION IN R2
55 017372 020327 000006      CMP R3,#6                               ;CONTROLLER MODEL MUST BE 6
56 017376 001003      BNE UDAI1D
57 017400 020227 000003      CMP R2,#3
58 017404 002005      BGE UDAI2                               ;MICROCODE VERSION MUST BE
; 3 OR GREATER

```

54 017406  
017406 104455  
017410 000016  
017412 000000  
017414 011732  
55 017416 000444

UDAI1D: ERRDF 14,,ERR014

;REPORT CONTROLLER NEEDS NEW REVISION  
TRAP C#ERRDF  
.WORD 14  
.WORD 0  
.WORD ERR014

BR UDAIEX



```

1          ;CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
2
3 017420 016502 000016      UDAI2:  MOV C.RING(R5),R2          ;GET FIRST ADDRESS OF RING BUFFER
4 017424 012703 000006      MOV @<HC.RSZ*2+HC.ISZ>/2,R3      ;GET SIZE OF RING BUFFER
5 017430 005722              UDAI2L: TST (R2)          ;CHECK WORD IN BUFFER
6 017432 001003              BNE UDAI2E          ;GO TO ERROR REPORTER IF NOT ZERO
7 017434 005303              DEC R3          ;COUNT THE WORDS IN BUFFER
8 017436 003374              BGT UDAI2L          ;LOOP UNTIL ALL WORDS CHECKED
9 017440 000405              BR UDAI3
10
11 017442              UDAI2E: ERRDF 23,,ERR023      ;REPORT BUFFER NOT CLEARED
    017442 104455              TRAP C$ERDF
    017444 000027              .WORD 23
    017446 000000              .WORD 0
    017450 012044              .WORD ERR023
12 017452 000426              BR UDAIEX
13
14          ;SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION
15
16 017454              UDAI3:
17 017454 016500 000006      MOV C.BST(R5),R0          ;GET BURST VALUE
18 017460 006300              ASL R0          ;SHIFT TO POSITION
19 017462 006300              ASL R0
20 017464 052700 000001      BIS @SA.GO,R0          ;SET THE GO BIT
21 017470 010064 000002      MOV R0,2(R4)          ;SEND TO UDA
22 017474 016501 000016      MOV C.RING(R5),R1
23 017500 010161 000004      MOV R1,HC.MSG(R1)
24 017504 062761 000020 000004  ADD @HC.MPK,HC.MSG(R1)
25 017512 010161 000010      MOV R1,HC.CMD(R1)
26 017516 062761 000104 000010  ADD @HC.CPK,HC.CMD(R1)
27 017524 000244              CLZ          ;CLEAR Z AS NO ERROR INDICATION
28 017526 000207              RETURN
29
30          ;ERROR RETURN
31
32 017530 000264      UDAIEX: SEZ          ;SET Z TO INDICATE ERROR OCCURRED
33 017532 000207              RETURN

```

```

1      ;UDAIST
2      ;
3      ;START THE INITIALIZATION PROCESS ON THE SELECTED UDA.
4      ;STOP BEFORE WRITING THE THIRD WORD SO UDA DOES NOT
5      ;ATTEMPT ANY UNIBUS TRANSFERS.
6      ;
7      ;INPUTS:
8      ;   R5 - ADDRESS OF CONTROLLER TABLE
9      ;
10     ;LOAD TABLE OF DATA TO SEND TO UDASA REGISTER
11
12     017534      UDAIST: BREAK
13     017534      104422      TRAP      C$BRK
14     017536      010146      PUSH R1
15     017540      016504      000004      MOV C.VEC(R5),R4
16     017544      042704      177000      AND CT.VEC,R4
17     017550      006204      ASR R4
18     017552      006204      ASR R4
19     017554      052704      100000      BIS #SA.STP,R4
20     017560      010437      017752      MOV R4,UDAID1
21     017564      016537      000016      017756      MOV C.RING(R5),UDAID2
22     017572      062737      000004      017756      ADD #MC.MSG,UDAID2
23     ;SET STEP BIT IN DATA WORD
24     ;LOAD INTERRUPT VECTOR
25     ;LOAD MEMORY ADDRESS
26     ; OF FIRST RESPONSE RING
27     ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER
28     MOV C.UADR(R5),R4      ;GET ADDRESS OF UDAIP REGISTER
29     CLR NXMAD      ;CLEAR MEMORY ERROR FLAG
30     SETVEC #4,#NXMI,#PRI07      ;SET UP VECTOR 4
31     MOV #PRI07,-(SP)
32     MOV #NXMI,-(SP)
33     MOV #4,-(SP)
34     MOV #3,-(SP)
35     TRAP C$SVEC
36     ADD #10,SP
37     TST 2(R4)      ;ACCESS UDASA REGISTER
38     CLR (R4)      ;WRITE TO UDAIP
39     CLRVEC #4      ;GIVE UP THE VECTOR
40     MOV #4,R0
41     TRAP C$CVEC
42     TST NXMAD      ;SEE IF A MEMORY ERROR OCCURRED
43     BEQ UDAISG
44     ERDF 20,,ERR020
45     TRAP C$ERDF
46     .WORD 20
47     .WORD 0
48     .WORD ERR020
49
50     SEC
51     BR UDAISE

```

```

1          ;SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
2
3 017674 012737 004000 020216 UDAISG: MOV @SA.S1,UDARSD          ;STORE RESPONSE MASK
4 017702 012703 017750          MOV @UDAIDT,R3          ;AND INDEX TO TABLE
5
6          ;WAIT FOR AND CHECK RESPONSE DATA
7
8 017706 04737 020060          UDAISL: CALL UDARSP          ;WAIT FOR STEP OR ERROR BITS
9 017712 103414          BCS UDAISE          ;EXIT IF ERROR
10 017714 004733          CALL @R3          ;CALL RESPONSE CHECKER FOR STEP
11 017716 103412          BCS UDAISE          ;GET OUT IF ERROR
12 017720 006337 020216          ASL UDARSD          ;SHIFT TO NEXT STEP BIT
13 017724 032737 040000 020216          BIT @SA.S4,UDARSD          ;CHECK IF NOW AT STEP 4
14 017732 001003          BNE UDAISX          ;GET OUT IF SO
15 017734 012364 000002          MOV (R3),2(R4)          ;WRITE DATA TO UDASA REGISTER
16 017740 000762          BR UDAISL          ;STAY IN LOOP
17
18 017742 000241          UDAISX: CLC          ;CLEAR CARRY FOR NO ERROR INDICATION
19 017744          UDAISE: POP R1
          017744 012601          MOV (SP),R1
20 017746 000207          RETURN

```

```

1          ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION
2
3 017750 017766      UDAID1: .WORD UDAIR1          ;FIRST WORD RESPONSE CHECK ROUTINE
4 017752 000000      UDAID1: .WORD 0              ;FIRST WORD TO SEND TO UDASA
5 017754 017774      UDAID2: .WORD UDAIR2          ;SECOND WORD RESPONSE CHECK ROUTINE
6 017756 000000      UDAID2: .WORD 0              ;SECOND WORD TO SEND TO UDASA
7 017760 020014      UDAID3: .WORD UDAIR3          ;THIRD WORD RESPONSE CHECK ROUTINE
8 017762 100000      UDAID3: .WORD SA.TST         ;THIRD WORD TO SEND TO UDASA
9 017764 020032      UDAID3: .WORD UDAIR4          ;FOURTH WORD RESPONSE CHECK ROUTINE
10
11          ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
12          ;CHECK FOR PROPER CONTROLLER TYPE
13
14 017766 012701 004400  UDAIR1: MOV #SA.S1+SA.DI,R1      ;SET STEP ONE BIT
15 017772 000422          BR UDAIRC                      ;NOW COMPARE
16
17          ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
18          ;CHECK FOR ECHO OF INTI AND VECTOR
19
20 017774 013701 017752  UDAIR2: MOV UDAID1,R1          ;GET WORD SENT TO UDASA
21 020000 000301          SWAB R1                        ;GET HIGH 8 BITS
22 020002 042701 177400  BIC #177400,R1
23 020006 052701 010000  BIS #SA.S2,R1                ;SET STEP 2 BIT
24 020012 000412          BR UDAIRC                      ;NOW COMPARE
25
26          ;RESPONSE CHECK FOR THIRD WORD FROM UDASA
27          ;CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
28
29 020014 013701 017752  UDAIR3: MOV UDAID1,R1          ;GET WORD SENT TO UDASA
30 020020 042701 177400  BIC #177400,R1                ;JUST LOW 8 BITS
31 020024 052701 020000  BIS #SA.S3,R1                ;SET STEP 3 BIT
32 020030 000403          BR UDAIRC                      ;NOW COMPARE
33
34          ;RESPONSE CHECK FOR FOURTH WORD FROM UDASA
35          ;CHECK FOR ECHO OF PURGE AND LFAIL BITS
36
37 020032 010201          UDAIR4: MOV R2,R1              ;GET RESPONSE FROM UDA
38 020034 042701 137760  BIC #C<SA.S4+SA.MCV>,R1        ;KEEP MICROCODE VERSION AND STEP 4
39
40          ;COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
41
42 020040 020102      UDAIRC: CMP R1,R2                ;COMPARE THE DATA
43 020042 001405      BEQ UDAIRX                       ;EXIT IF COMPARED CORRECTLY
44 020044          ERRDF 25,,ERR025                    ;REPORT ERROR
45 020044 104455      TRAP C:ERRDF
46 020046 000031      .WORD 25
47 020050 000000      .WORD 0
48 020052 012144      .WORD ERR025
49 020054 000261
50 020056 000207      SEC
51          UDAIRX: RETURN

```



```

1
2
3 020170 016402 000002      ;CHECK IF ERROR BIT SET
4 020174 100006      UDARS2: MOV 2(R4),R2      ;GET REGISTER CONTENTS
5 020176      BPL UDARSX      ;EXIT IF ERROR NOT SET
6 020176 104455      ERRDF 21,,ERR021      ;REPORT ERROR INFO
7 020176 000025      TRAP      C$ERDF
8 020200 000000      .WORD 21
9 020202 011762      .WORD 0
10 020204 000261      .WORD ERR021
11 020206 000207      UDARSE: SEC
12 020210 000207      RETURN
13
14      ;NORMAL EXIT
15
16 020212 000241      UDARSX: CLC      ;CLEAR CARRY AS NO ERROR INDICATION
17 020214 000207      RETURN
18
19      ;LOCATION FOR STEP BIT MASK
20
21 020216 000000      UDARSD: .WORD 0      ;LOAD BY CALLING ROUTINE

```

```

1      ;KW11I
2      ;
3      ;CLOCK INTERRUPT SERVICE ROUTINE
4
5      BGNSRV KW11I
6      020220      062737      000001      003220      ADD #1,KW.EL      ;COUNT THE INTERRUPT      KW11I::
7      020226      005537      003222      ADC KW.EL+2
8      020232      012777      000105      162750      MOV #KWOUT.,@KW.CSR      ;RESTART THE CLOCK
9      020240      ENDSRV
      020240      000002      L10033:
      RTI

```

```

1      ;RNTIME
2
3      ;PRINT RUNTIME
4
5      ;INPUTS:
6      ;       KW.EL - CONTAINS ELAPSED TIME
7      ;       KW.HZ - HERTZ OF CLOCK
8
9      ;OUTPUTS:
10     ;       IF CLOCK ON SYSTEM:
11     ;           " RUNTIME HH:MM:SS " PRINTED
12     ;       IF NO CLOCK: ONE SPACE IS PRINTED
13 020242 005737 003210  RNTIME: TST KW.CSR           ;CHECK IF A CLOCK PRESENT
14 020246 001465                BEQ RNTIMX           ;BRANCH IF NOT
15 020250                PUSH <R0,R3,R4,R5>
16 020250 010046                MOV R0,-(SP)
17 020252 010346                MOV R3,-(SP)
18 020254 010446                MOV R4,-(SP)
19 020256 010546                MOV R5,-(SP)
20 020260 013703 003220        MOV KW.EL,R3           ;GET ELAPSED TIME
21 020264 013704 003222        MOV KW.EL+2,R4
22 020270 013700 003216        MOV KW.HZ,R0           ;GET SPEED OF CLOCK
23 020274 004737 016332        CALL DIVIDE           ;COMPUTE SECONDS OF ELAPSED TIME
24 020300 012700 000074        MOV #60,R0           ;NOW DIVIDE BY 60
25 020304 004737 016332        CALL DIVIDE           ; TO COMPUTE MINUTES
26 020310                PUSH R5           ;SAVE REMAINDER AS SECONDS
27 020310 010546                MOV R5,-(SP)
28 020312 004737 016332        CALL DIVIDE           ;DIVIDE BY 60 AGAIN
29 020316                PNT RNTIM,R3       ;PRINT HOURS
30 020316 010346                MOV R3,-(SP)
31 020320 004137 016264        JSR R1,LPNT
32 020324 003704                .WORD RNTIM
33 020326 000002                .WORD PNT.CT
34 020330 020527 000011        CMP R5,#9           ;IF MINUTES 9 OR LESS
35 020334 003004                BGT 1$
36 020336                PRINT #'0       ;PRINT A LEADING ZERO
37 020336 112700 000060        MOVB #'0,R0
38 020342 004737 016054        CALL CPNT
39 020346                PNT RNTIM1,R5       ;NOW PRINT MINUTES
40 020346 010546                MOV R5,-(SP)
41 020350 004137 016264        JSR R1,LPNT
42 020354 003727                .WORD RNTIM1
43 020356 000002                .WORD PNT.CT
44 020360                POP R5           ;GET SECONDS
45 020360 012605                MOV (SP),R5
46 020362 020527 000011        CMP R5,#9           ;IF 9 OR LESS
47 020366 003004                BGT 2$
48 020370                PRINT #'0       ;PRINT A LEADING ZERO
49 020370 112700 000060        MOVB #'0,R0
50 020374 004737 016054        CALL CPNT
51 020400                PNT RNTIM2,R5       ;NOW PRINT SECONDS
52 020400 010546                MOV R5,-(SP)
53 020402 004137 016264        JSR R1,LPNT
54 020406 003735                .WORD RNTIM2
55 020410 000002                .WORD PNT.CT
56 020412                POP <R5,R4,R3,R0> ;HOURS IN R3
57 020412 012605                MOV (SP),R5
    
```



020414 012604  
020416 012603  
020420 012600  
35 020422 112700 000040  
020426 004737 016054  
36 020432 000207

RNTIMX: PRINT '<0' >

;PRINT A SPACE

RETURN

MOV (SP),R4  
MOV (SP),R3  
MOV (SP),R0

MOVB 0,R0  
CALL CPNT

```

1 020434      DATE:  GMANID DATEQ,DATEI,A,-1,1,11.,YES      ;GET DATE
  020434      104443
  020436      000406
  020440      003272
  020442      000152
  020444      003536
  020446      177777
  020450      000001
  020452      000013
  020454
2 020454      012705      003272      MOV @DATEI,R5      ;GET POINTER TO ANSWER
3 020460      121527      000060      CMPB (R5),@'0
4 020464      103443      BLO DERR
5 020466      122527      000071      DAY:  CMPB (R5),@'9
6 020472      101040      BHI DERR
7 020474      121527      000055      CMPB (R5),@'-
8 020500      001406      BEQ DAS1
9 020502      121527      000060      CMPB (R5),@'0
10 020506     103432      BLO DERR
11 020510     122527      000071      CMPB (R5),@'9
12 020514     101027      BHI DERR
13 020516     122527      000055      DAS1:  CMPB (R5),@'-
14 020522     001024      BNE DERR
15 020524     012704      000014      MOV @12.,R4      ;GET NUMBER OF MONTH
16 020530     012703      003347      MON1:  MOV @MONTHS,R3   ;GET POINTER TO MONTH NAMES
17 020534     005000      CLR R0
18 020536     121523      CMPB (R5),(R3).
19 020540     001401      BEQ MON2
20 020542     005200      INC R0
21 020544     126523      000001      MON2:  CMPB 1(R5),(R3).
22 020550     001401      BEQ MON3
23 020552     005200      INC R0
24 020554     126523      000002      MON3:  CMPB 2(R5),(R3).
25 020560     001401      BEQ MON4
26 020562     005200      INC R0
27 020564     005700      MON4:  TST R0
28 020566     001407      BEQ MON5
29 020570     005304      DEC R4
30 020572     001360      BNE MON1
31 020574     004137      016226      DERR:  PNTF DATEX
  020574     011503
  020600     000000
  020602     000713
32 020604     012701      003306      MON5:  BR DATE
33 020606     010403      MOV @DATEO,R1   ;GET POINTER TO DATE FOR FORMATTER
34 020612     020327      000012      MON6:  MOV R4,R3      ;GET COPY OF MONTH NUMBER
35 020614     103404      CMP R3,@10.     ; IF 10 OR GREATER
36 020620     112721      000061      BLO MON6
37 020622     162703      000012      MOVB #'1,(R1). ;PUT A "1" IN OUTPUT
38 020626     062703      000060      SUB @10.,R3
39 020632     110321      000055      MON6:  ADD #'0,R3     ;CONVERT MONTH NUMBER TO ASCII
40 020636     062704      003412      MOVB R3,(R1).  ;PUT A NUMBER IN OUTPUT
41 020640     112721      000055      MOVB #'-(R1). ;PUT A "-" IN OUTPUT
42 020644     012703      003272      ADD @DAYS-1,R4 ;GET POINTER TO DAYS IN MONTH
43                                     ;INDEXED BY NUMBER OF MONTH
44 020650     005000      MOV @DATEI,R3   ;GET POINTER TO DATE INPUT
45 020654

```

10000\$:

```

TRAP
BR
.WORD
.WORD
.WORD
.WORD
.WORD
.WORD

```

```

JSR R1,LPNTF
.WORD DATEX
.WORD PNT.CT

```

```

46 020656 121327 000055      DAY1:  CMPB (R3),@' -
47 020662 001413              BEQ DAY2
48 020664 111321              MOVB (R3),(R1). ;PUT DAY CHARACTER IN OUTPUT
49 020666 006300              ASL R0
50 020670 010002              MOV R0,R2
51 020672 006300              ASL R0
52 020674 006300              ASL R0
53 020676 060200              ADD R2,R0
54 020700 112302              MOVB (R3),R2
55 020702 162702 000060      SUB @'0,R2
56 020706 060200              ADD R2,R0
57 020710 000762              BR DAY1
58 020712 120014      DAY2:  CMPB R0,(R4)
59 020714 101327              BHI DERR
60 020716 005700              TST R0 ;SEE IF DATE IS ZERO
61 020720 001725              BEQ DERR ;ERROR IF SO
62 020722 062705 000003      ADD @3,R5
63 020726 121527 000055      CMPB (R5),@' - ;CHECK FOR "-" BETWEEN DAY
64 020732 001320              BNE DERR ; AND YEAR IN OUTPUT
65 020734 112521              MOVB (R5),R1. ;PUT "-" IN OUTPUT
66 020736 010504              MOV R5,R4 ;GET COPY OF INPUT STRING POINTER
67 020740 005000              CLR R0
68 020742 005002              CLR R2
69 020744 121427 000060      YER1:  CMPB (R4),@'0
70 020750 103416              BLO YER2
71 020752 121427 000071      CMPB (R4),@'9
72 020756 101013              BHI YER2
73 020760 006300              ASL R0
74 020762 010003              MOV R0,R3
75 020764 006300              ASL R0
76 020766 006300              ASL R0
77 020770 060300              ADD R3,R0
78 020772 112403              MOVB (R4),R3
79 020774 162703 000060      SUB @'0,R3
80 021000 060300              ADD R3,R0
81 021002 005202              INC R2
82 021004 000757              BR YER1
83 021006 105714      YER2:  TSTB (R4)
84 021010 001271              BNE DERR
85 021012 020227 000002      CMP R2,@2
86 021016 001407              BEQ YER3
87 021020 020227 000004      CMP R2,@4
88 021024 001263              BNE DERR
89 021026 020027 003554      CMP R0,@1900.
90 021032 103660              BLO DERR
91 021034 000413              BR YER5
92 021036 012702 003427      YER3:  MOV @YEAR19,R2
93 021042 020027 000106      CMP R0,@70.
94 021046 103002              BHIS YER4
95 021050 012702 003432      MOV @YEAR20,R2
96 021054 105712      YER4:  TSTB (R2)
97 021056 001402              BEQ YER5
98 021060 112221              MOVB (R2),R1.
99 021062 000774              BR YER4
100 021064 112521      YER5:  MOVB (R5),R1.
101 021066 001376              BNE YER5
102 021070 000207              RETURN

```

103  
104 021072

ENDMOD

```

1          .SBTTL  PROTECTION TABLE
2
3 021072          BGNMOD
4
5          : **
6          : THIS TABLE IS USED BY THE RUNTIME SERVICES
7          : TO PROTECT THE LOAD MEDIA.
8          : **
9
10 021072         BGNPROT
10 021072
11
12 021072 177777          -1          ;OFFSET INTO P-TABLE FOR CSR ADDRESS
13 021074 177777          -1          ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
14 021076 177777          -1          ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
15
16 021100         ENDPROT
17

```

L\$PROT::

```

1      .SBTTL INITIALIZE SECTION
2
3
4      ;**
5      ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
6      ; AT THE BEGINNING OF EACH PASS.
7      ;**
8      021100      BGNINIT
9
10     021100      READEF @EF.START      ;CHECK IF STARTED BY OPERATOR
11     021100      012700 000040      MOV @EF.START,RO
12     021104      104447      TRAP C$REFG
13     021106      BCOMPLETE INIT1      ; IF NOT,
14     021106      103415      BCS INIT1
15     021110      READEF @EF.RESTART
16     021110      012700 000037      MOV @EF.RESTART,RO
17     021114      104447      TRAP C$REFG
18     021116      BCOMPLETE INIT1
19     021116      103411      BCS INIT1
20     021120      READEF @EF.CONTINUE
21     021120      012700 000036      MOV @EF.CONTINUE,RO
22     021124      104447      TRAP C$REFG
23     021126      BCOMPLETE INIT1
24     021126      103405      BCS INIT1
25     021130      READEF @EF.PWR
26     021130      012700 000034      MOV @EF.PWR,RO
27     021134      104447      TRAP C$REFG
28     021136      BCOMPLETE INIT1
29     021136      103401      BCS INIT1
30     021140      INITQT: DOCLN      ; ABORT PROGRAM
31     021140      104444      TRAP C$DCLN
32
33     021142      012700 000003      INIT1: MOV @SO.FMT,RO ;BUILD MODE WORD FROM SOFTWARE QUESTIONS
34     021146      030037 002144      BIT RO,SFPTBL ;SEE IF REFORMAT
35     021152      001011      BNE INIT2 ; BRANCH IF SO
36     021154      012700 000004      MOV @SO.CNS,RO ;SEE IF RECONSTRUCT
37     021160      030037 002144      BIT RO,SFPTBL
38     021164      001004      BNE INIT2 ; BRANCH IF SO
39     021166      006300      ASL RO ;SEE IF RESTORE
40     021170      ;ASSUME SO.STR EQ SO.CNS*2
41     021170      030037 002144      BIT RO,SFPTBL
42     021174      001761      BEQ INITQT ;IF NOT, ABORT PROGRAM
43     021176      010037 003206      INIT2: MOV RO,MODE ;SAVE MODE FLAGS
44     021202      004737 012564      INIT3: CALL RESET ;RESET ALL UNITS
45     021206      MEMORY FFREE ;RESET START OF FREE MEMORY
46     021206      104431      TRAP C$MEM
47     021210      010037 002146      MOV RO,FFREE
48     021214      017737 160726 002150      MOV @FFREE,FSIZE ;RESET SIZE OF FREE MEMORY
49
50     ;INITIALIZE CLOCK
51
52     000105      KWOUT.=105 ;DATA TO SEND TO KW11 TO START CLOCK
53     021222      005037 003210      CLR KW.CSR ;MARK CLOCK AS NOT ON SYSTEM
54     021226      005037 003220      CLR KW.EL ;CLEAR ELAPSED TIME
55     021232      005037 003222      CLR KW.EL*2
56     021236      CLOCK L,RO ;SEE IF AN L CLOCK PRESENT

```

```

021236 012700 000114          MOV    @'L,RO
021242 104462          TRAP   C1CLK
42 021244          BCOMPLETE KYES
021244 103413          ;SEE IF A P CLOCK PRESENT
43 021246          BCS    KYES
021246 012700 000120          MOV    @'P,RO
021252 104462          TRAP   C1CLK
44 021254          BCOMPLETE KYES
021254 103407          ;IF NEITHER, CLEAR CSR STORAGE WORD
45 021256 005037 003210          CLR   KW.CSR
46 021262          PNTF  NOCLOCK
021262 004137 016226          JSR   R1,LPNTF
021266 004073          .WORD NOCLOCK
021270 000000          .WORD PNT.CT
47 021272 000426          BR    KNO
48 021274 012037 003210          KYES:  MOV   (RO)+,KW.CSR
49 021300 012037 003212          MOV   (RO)+,KW.BRL
50 021304 012037 003214          MOV   (RO)+,KW.VEC
51 021310 012037 003216          MOV   (RO)+,KW.MZ
52 021314          SETVEC KW.VEC,@KW1'I,KW.BRL
021314 013746 003212          ;STORE DATA RETURNED
021320 012746 020220          ;SET THE VECTOR
021324 013746 003214          MOV   KW.BRL,-(SP)
021330 012746 000003          MOV   @KW1'I,-(SP)
021334 104437          MOV   KW.VEC,-(SP)
021336 062706 000010          MOV   @3,-(SP)
53 021342 012777 000105 161640          TRAP  C1SVEC
54 021350          ADD   @10,SP
          MOV @KWOUT.,@KW.CSR
          ;START THE CLOCK

```

```

1          ;INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
2
3 021350 013737 002146 002156      MOV FFREE,CTABS          ;STORE START OF CONTROLLER TABLES
4 021356 005077 160574              CLR @CTABS             ;ZEROS MARKS END CONTROLLER TABLES
5 021362 005037 002160              CLR CTRLRS            ;CLEAR CONTROLLER COUNT
6 021366 012701 003436              MOV @IPADRS,R1         ; R1 -> IP ADDRESS
7 021372 012702 000010              MOV @B.,R2            ; R2 IS A COUNTER
8 021376 005021 1$: CLR (R1).        ; CLEAR ENTRY
9 021400 005302                      DEC R2                 ; DONE?
10 021402 001375                      BNE 1$                 ; IF NOT, BRANCH
11
12          ;GET A P-TABLE FROM DRS
13
14 021404 005002                      CLR R2                 ;LOGICAL UNIT NUMBER IN R2
15 021406                      INIT4: GPHARD R2,R0      ;GET POINTER TO A P-TABLE
16 021406 010200                      MOV R2,R0              ;
17 021410 104442                      TRAP C:GPHARD          ;
18 021412                      BNCOMPLETE NXTTAB      ;IGNORE IF NO TABLE RETURNED
19 021412 103110                      BCC NXTTAB             ;
20
21          ;SEE IF A CONTROLLER TABLE ALREADY EXISTS FOR CONTROLLER IN P-TABLE
22
23 021414 013703 002156              MOV CTABS,R3           ;GET ADDRESS OF CONTROLLER TABLES
24 021420 005713                      INIT5: TST (R3)         ;CHECK IF ANY MORE TABLES
25 021422 001416                      BEQ NEWTAB             ;BUILD NEW TABLE IF FOUND ZERO WORD
26 021424 021013                      CMP (R0),(R3)          ;CHECK IF SAME UNIBUS ADDRESS
27 021426                      ASSUME C.UADR EQ 0
28 021426                      ASSUME MO.UBA EQ 0
29 021426 001463                      BEQ SAMTAB             ;CHECK TABLE IF ALREADY EXISTS
30 021430 016301 000004              MOV C.VEC(R3),R1       ;GET VECTOR FROM EXISTING CONTROLLER TABLE
31 021434 042701 177000              BIC @C<CT.VEC>,R1
32 021440 026001 000002              CMP MO.VEC(R0),R1
33 021444 001002                      BNE 1$                 ;SEE IF DIFFERENT VECTOR
34 021446 000137 022076              JMP SAMVEC             ;ERROR, CAN'T HAVE TWO UDA'S WITH SAME VECTOR
35 021452 062703 000054              1$: ADD @C.SIZE,R3
36 021456 000760                      BR INIT5

```



```

1
2
3 021460 012703 000010
4 021464 012704 003436
5 021470 005714
6 021472 001404
7 021474 005724
8 021476 005303
9 021500 001373
10 021502 000401
11 021504 011014
12 021506 012701 000026
13 021512 004737 012522
14 021516 011021
15 021520 010221
16 021522 016004 000004
17 021526 000304
18 021530 006104
19 021532 056004 000002
20 021536 010421
21 021540 016021 000006
22 021544 012721 004037
23 021550 012721 017156
24 021554 012703 000020
25
26 021560 005021
27 021562 005303
28 021564 001375
29 021566 005237 002160
30 021572 005011
31 021574 000417

;BUILD A CONTROLLER TABLE
NEWTAB: MOV #8.,R3
MOV #IPADRS,R4
1$: TST (R4)
BEQ 2$
TST (R4).
DEC R3
BNE 1$
BR 3$
2$: MOV (R0),(R4)
3$: MOV #C.SIZE/2,R1
CALL ALOCM
MOV (R0),(R1).
MOV R2,(R1).
MOV HO.BRL(R0),R4
SWAB R4
ROL R4
BIS HO.VEC(R0),R4
MOV R4,(R1).
MOV HO.BST(R0),(R1).
MOV #4037,(R1).
MOV #UDASRV,(R1).
MOV #16.,R3

INIT7: CLR (R1).
DEC R3
BNE INIT7
INC CTRLRS
CLR (R1)
BR NXTTAB

;R3 IS A COUNTER
;R4 -> IP ADDRESSES
; FOUND AN OPEN ENTRY?
; IF SO, GO FILL ENTRY
; NEXT ENTRY
; SEARCH THROUGH ENTIRE TABLE?
; IF NOT, BRANCH
; ELSE, TABLE FULL
; STORE ENTRY INTO TABLE
;GET WORDS IN CONTROLLER TABLE
;ALLOCATE SPACE FOR IT
;STORE UNIBUS ADDRESS
;UNIT NUMBER
;GET BR LEVEL
;SWAP TO HIGH BYTE
;SHIFT ONE MORE TO LEFT
;ADD VECTOR ADDRESS
; TO TABLE

;PUT [JSR R0,UDASRV]
; INTO TABLE
;CLEAR POINT'RS TO DRIVE TABLES.
; TIMEOUT COUNTER, FLAGS, REF. NUMBER

;LOOP TIL ALL CLEARED
;COUNT THE CONTROLLER
;CLEAR TABLE END MARKER
;NOW GO TO NEXT P-TABLE

```

```

1          ;SHOULD BE SAME CONTROLLER, CHECK THAT OTHER PARAMETERS MATCH
2
3 021576 016004 000004      SAMTAB: MOV MO.BRL(R0),R4      ;GET BR LEVEL FROM P-TABLE
4 021602 000304              SWAB R4                ;SWAP TO HIGH BYTE
5 021604 006104              ROL R4                  ;SHIFT ONE MORE TO LEFT
6 021606 056004 000002      BIS MO.VEC(R0),R4        ;ADD VECTOR ADDRESS
7 021612 020463 000004      CMP R4,C.VEC(R3)        ;COMPARE WITH CONTROLLER TABLE
8 021616 001004              BNE 1$
9 021620 026063 000006 000006  CMP MO.BST(R0),C.BST(R3)    ;COMPARE BURST RATES
10 021626 001402              BEQ NXTTAB
11 021630 000137 022026      1$: JMP CTABER          ;FATAL ERROR IF NOT SAME
12
13          ;GET NEXT P-TABLE
14
15 021634 005202              NXTTAB: INC R2                ;INCREMENT LOGICAL UNIT NUMBER
16 021636 023702 002012      CMP L$UNIT,R2          ;CHECK IF GOT ALL TABLES
17 021642 003261              BGT INIT4                ;IF NOT, GO BACK FOR NEXT
18
19 021644 012701 000001      MOV #1,R1              ;ALLOCATE SPACE FOR ZERO END WORD
20 021650 004737 012522      CALL ALOCM           ;AFTER CONTROLLER TABLES

```

111

```
1      ;NOW BUILD DRIVE TABLES
2
3 021654 005002
4 021656 010200      CLR R2
                    ;LOGICAL UNIT NUMBER IN R2
                    ;GET POINTER TO A P-TABLE
5 021660 104442      INIT8: GPHARD R2,R0
                    ;IF NOT AVAILABLE, GO GET NEXT
6 021662 103040      BNCOMPLETE INIT14
                    ;FIND CONTROLLER TABLE
7
8
9 021664 013703 002156      MOV CTABS,R3
                    ;GET ADDRESS OF CONTROLLER TABLES
10 021670 021013      INIT10: CMP (R0),(R3)
                    ;CHECK IF SAME UNIBUS ADDRESS
11 021672 001403      BEQ INIT11
                    ;BRANCH IF TABLE FOUND
12 021674 062703 000054      ADD #C.SIZE,R3
                    ;MOVE TO NEXT TABLE
13 021700 000773      BR INIT10
```

```

1          ;BUILD DRIVE TABLE
2
3 021702 012701 000015  INIT11: MOV #D.SIZE/2,R1      ;GET SIZE OF DRIVE TABLE
4 021706 004737 012522      CALL ALOCM          ;ALLOCATE SPACE FROM FREE MEMORY
5          :          RO POINTS TO P-TABLE
6          :          R1 POINTS TO DRIVE TABLE
7          :          R3 POINTS TO CONTROLLER TABLE
8          :          R2 IS UNIT NUMBER
9 021712 010337 003244      MOV R3,TEMP          ;SAVE CONTROLLER TABLE ADDRESS
10         ;IN CASE AN ERROR IS DETECTED
11 021716 062703 000020      ADD #C.DR0,R3       ;BUILD POINTER TO C.DR ENTRY IN CONTROLLER TABLE
12 021722 012704 000010      MOV #8.,R4         ;GET MAX COUNT OF DRIVES ON ONE CONTROLLER
13 021726 005713          INIT12: TST (R3)          ;CHECK IF ENTRY CONTAINS POINTER TO DRIVE TABLE
14 021730 001411          BEQ INIT13
15 021732 026033 000010      CMP #0.LDR(R0),@(R3). ;CHECK DRIVE NUMBER IN DRIVE TABLE
16 021736 001002          BNE 1$
17 021740 000137 022042      JMP MLDRER         ;IF SAME, TWO P-TABLES POINT TO SAME DRIVE
18 021744 005304          1$: DEC R4          ;COUNT DRIVES
19 021746 001367          BNE INIT12        ;IF EIGHT DRIVE TABLES EXIST,
20 021750 000137 022060      JMP TOOMER        ; THEN REPORT ERROR
21 021754 010113          INIT13: MOV R1,(R3)      ;LOAD DRIVE TABLE POINTER
22 021756 016021 000010      MOV #0.LDR(R0),(R1). ;LOAD DRIVE NUMBER
23 021762 010221          MOV R2,(R1).      ;LOAD UNIT NUMBER

```

```

1          ;GO TO NEXT DRIVE TABLE
2
3 021764 005202          INIT14: INC R2          ;INCREMENT LOGICAL UNIT NUMBER
4 021766 023702 002012      CMP L$UNIT,R2          ;CHECK IF GOT ALL TABLES
5 021772 003331          BGT INIT8          ;IF NOT, GET NEXT TABLE
6
7          ;SAVE CURRENT PARAMETERS TO FREE MEMORY
8
9 021774 013737 002146 002152 INIT15: MOV FFREE,FMEM      ;SAVE START ADDRESS
10 022002 013737 002150 002154      MOV FSIZE,FMEMS      ;SAVE SIZE
11
12 022010          INITXX: SETPRI @PRI00          ; SET RUNNING PRIORITY TO ZERO
    022010 012700 000000          MOV @PRI00,R0
    022014 104441          TRAP C$SPRI
13 022016 004737 013022          CALL CLOSEF          ;MAKE SURE DATA FILE IS CLOSED
14 022022          EXIT INIT
    022022 104432          TRAP C$EXIT
    022024 000066          .WORD L10035-

```

INITIALIZE SECTION

```

1
2 022026 010305 ;DIFFERENT VECTORS, BR LEVELS OR BURST RATES FOR ONE CONTROLLER
3 022030 104454 CTABER: MOV R3,R5 ;GET CONTROLLER ADDRESS
022030 000001 ERRSF 1.,ERR001
022032 000001 TRAP C$ERSF
022034 000000 .WORD 1
022036 011604 .WORD 0
4 022040 DOCLN .WORD ERR001
022040 104444 TRAP C$DCLN
5
6 ;TWO P-TABLES FOR SAME DRIVE
7 022042 013705 003244 MLDRE: MOV TEMP,R5 ;GET CONTROLLER ADDRESS
8 022046 104454 ERRSF 2.,ERR002
022046 000002 TRAP C$ERSF
022050 000002 .WORD 2
022052 000000 .WORD 0
022054 011622 .WORD ERR002
9 022056 DOCLN TRAP C$DCLN
022056 104444
10
11 ;MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER
12
13 022060 013705 003244 TOOMER: MOV TEMP,R5 ;GET CONTROLLER ADDRESS
14 022064 104454 ERRSF 3.,ERR003
022064 000003 TRAP C$ERSF
022066 000003 .WORD 3
022070 000000 .WORD 0
022072 011640 .WORD ERR003
15 022074 DOCLN TRAP C$DCLN
022074 104444
16
17 ;TWO UDA'S USE THE SAME VECTOR
18
19 022076 010305 SAMVEC: MOV R3,R5 ;GET CONTROLLER ADDRESS
20 022100 104454 ERRSF 8.,ERR008
022100 000010 TRAP C$ERSF
022102 000010 .WORD 8
022104 011670 .WORD ERR008
022106 000000 .WORD 0
21 022110 DOCLN TRAP C$DCLN
022110 104444
22
23 022112 ENDINIT
022112 L10035: TRAP C$INIT
022112 104411

```

```

1      .SBTTL  AUTODROP SECTION
2
3      : **
4      : THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
5      : THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
6      : SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
7      : DROPPED FROM TESTING.
8      : --
9
10     022114      BGNAUTO
11     022114
12     022114      ENDAUTO
13     022114      L$AUTO::
14     022114      L10036: TRAP  C$AUTO
15     022114 104461

```

```
1          .SBTTL  CLEANUP CODING SECTION
2
3          : **
4          : THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
5          : AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
6          : **
7
8 022116          BGNCLN
9 022116          L$CLEAN::
10 022116 004737 013022          CALL CLOSEF          ;CLOSE DATA FILE
11 022122 004737 012564          CALL   RESET          ;RESET ALL UDAS
12
13 022126          ENDCLN
14 022126          L10037:          TRAP          C$CLEAN
15 022130          ENDMOD
```



```

1          .SBTTL TEST 1: DUP PROGRAM DRIVER
2
3 022130          BGNMOD
4
5 022130          BGNST
6 022130          PNTX WNSTRT          ;PRINT WARNING MESSAGE          T1::
   022130 004137 016246          JSR R1,LPNTX
   022134 004532          .WORD WNSTRT
   022136 000000          .WORD PNT.CT
7 022140          MANUAL          ;SEE IF MANUAL INTERVENTION ALLOWED
   022140 104450          TRAP          C$MANI
8 022142          BNCOMPLETE TIMODE          ;IF NOT, JUST RUN THE PROGRAM
   022142 103020          BCC          TIMODE
9 022144 005037 003244          CLR TEMP          ;CLEAR WORD FOR ANSWER
10 022150          GMANIL WNQUES,TEMP,1,YES          ;ASK IF STILL WANT TO RUN
   022150 104443          TRAP          C$GMAN
   022152 000404          BR          10000$
   022154 003244          .WORD          TEMP
   022156 000130          .WORD          T$CODE
   022160 003622          .WORD          WNQUES
   022162 000001          .WORD          1
   022164          10000$:
11 022164 005737 003244          TST TEMP          ;LOOK AT ANSWER
12 022170 001417          BEQ T1QUIT          ;IF NO, QUIT NOW
13 022172 005737 003306          TST DATED          ;SEE IF ALREADY ASKED FOR DATE
14 022176 001002          BNE TIMODE
15 022200 004737 020434          CALL DATE          ;IF NOT, GET IT NOW
16
17 022204 032737 000003 003206 TIMODE: BIT #50,FMT,MODE
18 022212 001164          BNE T1FMT
19 022214          MANUAL
   022214 104450          TRAP          C$MANI
20 022216          BCOMPLETE T1GO
   022216 103406          BCS          T1GO
21 022220          ERRSF 10,,ERR010
   022220 104454          TRAP          C$ERSF
   022222 000012          .WORD          10
   022224 000000          .WORD          0
   022226 011720          .WORD          ERR010
22 022230          T1QUIT: EXIT TST
   022230 104432          TRAP          C$EXIT
   022232 000362          .WORD          L10040-.
23 022234 032737 000010 003206 T1GO: BIT #50,STR,MODE
24 022242 001435          BEQ T1CNS
25 022244 023727 002012 000001          CMP L$UNIT,#1
26 022252 001406          BEQ T1RST
27 022254          ERRSF 9,,ERR009
   022254 104454          TRAP          C$ERSF
   022256 000011          .WORD          9
   022260 000000          .WORD          0
   022262 011706          .WORD          ERR009
28 022264          EXIT TST
   022264 104432          TRAP          C$EXIT
   022266 000326          .WORD          L10040-.
29
30 022270          T1RST: PNTF FILNAM

```

```

022270 004137 016226
022274 011522
022276 000000
31 022300          GMANID FILNAQ,FNAME,A,-1,1,10.,NO ;GET FILE NAME
022300 104443
022302 000406
022304 003232
022306 000142
022310 003566
022312 177777
022314 000001
022316 000012
022320
32 022320          OPEN #FNAME
022320 012700 003232
022324 104434
33 022326 012737 177777 002174      MOV #-1,FILOPN ;MARK FLAG AS FILE OPEN
34 022334 000513
35 022336 013705 002156      T1CNS: MOV CTABS,R5
36 022342 010504
37 022344 062704 000020      T1SER1: MOV R5,R4
38 022350 012703 000010      ADD #C.DRO,R4
39 022354 011402
40 022356 001476
41 022360          T1SER2: MOV (R4),R2 ;GET DRIVE TABLE POINTER
022360 011246          BEQ T1SERN
022362 011546          PNTF SERNUM,D.UNIT(R2),(R5),(R2)
022364 016246 000002
022370 004137 016226
022374 004235
022376 000006
42 022400          ASSUME C.UADR EQ 0
43 022400          ASSUME D.DRV EQ 0
44 022400          T1SER3: GMANID SERNO,TEMP,A,-1,1,20.,NO ;GET SERIAL NUMBER
022400 104443
022402 000406
022404 003244
022406 000142
022410 003620
022412 177777
022414 000001
022416 000024
022420
45 022420 012701 003244
46 022424 005000
47 022426 105711
48 022430 001410
49 022432 005200
50 022434 121127 000060
51 022440 103420
52 022442 122127 000071
53 022446 101767
54 022450 000414
55 022452 020027 000024
56 022456 103424
57 022460 012701 003244
58 022464 012700 003322

```

```

JSR R1,LPNTF
.WORD FILNAM
.WORD PNT,CT

```

```

TRAP C$GMAN
BR 10001$
.WORD FNAME
.WORD T$CODE
.WORD FILNAQ
.WORD -1
.WORD T$LOLIM
.WORD T$HILIM

```

10001\$:

```

MOV #FNAME,R0
TRAP C$OPEN

```

```

MOV (R2),-(SP)
MOV (R5),-(SP)
MOV D.UNIT(R2),-(SP)
JSR R1,LPNTF
.WORD SERNUM
.WORD PNT,CT

```

```

TRAP C$GMAN
BR 10002$
.WORD TEMP
.WORD T$CODE
.WORD SERNO
.WORD -1
.WORD T$LOLIM
.WORD T$HILIM

```

10002\$:

```

MOV #TEMP,R1
CLR R0
T1SER4: TSTB (R1)
BEQ T1SER5
INC R0
CMPB (R1),#0
BLO T1SER7
CMPB (R1),#9
BLOS T1SER4
BR T1SER7
T1SER5: CMP R0,#20
BLO T1SER8
MOV #TEMP,R1
MOV #HIGHEST,R0

```

```

59 022470 105710
60 022472 001416
61 022474 122120
62 022476 001774
63 022500 103413
64 022502
   022502 012746 003322
   022506 012746 011413
   022512 012746 000002
   022516 010600
   022520 104417
   022522 062706 000006
65 022526 000724
66 022530 062702 000004
67 022534 012701 003244
68 022540 112122
69 022542 001376
70 022544 005303
71 022546 001402
72 022550 005724
73 022552 000700
74 022554 062705 000054
75 022560 005715
76 022562 001267
77 022564 013737 002156 002162
78 022572 013701 002160
79 022576 004737 012710
80 022602 001402
81 022604 004737 013040
82 022610
   022610 104432
   022612 000002
83 022614
   022614
   022614 104401
84 022616

```

```

T1SER6: TSTB (R0)
        BEQ T1SER8
        CMPB (R1), (R0)
        BEQ T1SER6
        BLO T1SER8
T1SER7: PRINTF @SERNX, @HIGHEST

        BR T1SER3
T1SER8: ADD @D.SERN, R2 ;PUT ANSWER INTO DRIVE TABLE
        MOV @TEMP, R1
T1SER9: MOVB (R1), (R2)
        BNE T1SER9
        DEC R3
        BEQ T1SERN
        TST (R4)
        BR T1SER2
T1SERN: ADD @C.SIZE, R5
        TST (R5)
        BNE T1SER1
T1FMT: MOV CTABS, TSTTAB
        MOV CTRLRS, R1
        CALL RUNDM
        BEQ 68
        CALL RESPDM
68:     EXIT TST

        ENDTST

        ENDMOD

```

```

MOV     @HIGHEST, -(SP)
MOV     @SERNX, -(SP)
MOV     @2, -(SP)
MOV     SP, R0
TRAP   C$PNTF
ADD     @6, SP

```

```

;GET FIRST TABLE ADDRESS
;RUN DM PROGRAM ON ALL CONTROLLERS
; RUN ALL CONTROLLERS OF ONE TYPE AT ONCE

```

```

TRAP   C$EXIT
        .WORD   L10040-
L10040: TRAP   C$ETST

```

1  
2  
3 022616  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14 022616  
022616 000027  
022620  
15  
16  
17  
18 022620  
19  
20 022620  
21 022620  
22 022620  
23 022620  
24 022620  
25 022620

.SBTTL HARDWARE PARAMETER CODING SECTION

BGNMOD

\*\*\*  
; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS  
; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE  
; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE  
; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE  
; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS  
; WITH THE OPERATOR.  
;--

BGNHRD

.WORD L10041-L\$HARD/2  
L\$HARD::

;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:

TABLE		;START A TEBLE DEFINITION
ITEM NO.UBA	2	; UNIBUS ADDRESS
ITEM NO.VEC	2	; UDA VECTOR
ITEM NO.BRL	2	; BR LEVEL
ITEM NO.BST	2	; BURST RATE
ITEM NO.LDR	2	; DRIVE NUMBER
END		

```

1 022620          GPRMA  H.UBA,HO.UBA,0,160000,177774,YES      ;BUS ADDRESS
  022620 000031          .WORD  T%CODE
  022622 022676          .WORD  H.UBA
  022624 160000          .WORD  T%LOLIM
  022626 177774          .WORD  T%HILIM
2 022630          GPRMA  H.VEC,HO.VEC,0,4,774,YES      ; VECTOR
  022630 001031          .WORD  T%CODE
  022632 022724          .WORD  H.VEC
  022634 000004          .WORD  T%LOLIM
  022636 000774          .WORD  T%HILIM
3 022640          GPRMD  H.BRL,HO.BRL,0,-1,4,,7,,YES      ; BR LEVEL
  022640 002052          .WORD  T%CODE
  022642 022733          .WORD  H.BRL
  022644 177777          .WORD  -1
  022646 000004          .WORD  T%LOLIM
  022650 000007          .WORD  T%HILIM
4 022652          GPRMD  H.BST,HO.BST,D,-1,0,,63,,YES      ; BURST RATE
  022652 003052          .WORD  T%CODE
  022654 022744          .WORD  H.BST
  022656 177777          .WORD  -1
  022660 000000          .WORD  T%LOLIM
  022662 000077          .WORD  T%HILIM
5 022664          GPRMD  H.LDR,HO.LDR,D,-1,0,,255,,YES      ; DRIVE SELECT NUMBER
  022664 004052          .WORD  T%CODE
  022666 022766          .WORD  H.LDR
  022670 177777          .WORD  -1
  022672 000000          .WORD  T%LOLIM
  022674 000377          .WORD  T%HILIM
6 022676          ENDMRD
                                .EVEN
                                L10041:

```

```

7
8 022676      125      116      111  H.UBA:  .ASCIZ  \UNIBUS ADDRESS OF UDA\
9 022724      126      105      103  H.VEC:  .ASCIZ  \VECTOR\
10 022733     102      122      040  H.BRL:  .ASCIZ  \BR LEVEL\
11 022744     125      116      111  H.BST:  .ASCIZ  \UNIBUS BURST RATE\
12 022766     104      122      111  H.LDR:  .ASCIZ  \DRIVE NUMBER\
13           .EVEN

```

```
1      .SBTTL  SOFTWARE PARAMETER CODING SECTION
2
3      ;**
4      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
5      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
6      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
9      ; WITH THE OPERATOR.
10     ;--
11
12     023004      BGNSFT
13     023004      000022
14     023006
15
16     ;FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:
17
18     TABLE
19     ITEM SO.BIT      2
20     SO.FM1 = BIT0
21     SO.FM2 = BIT1
22     SO.FMT = SO.FM1+SO.FM2
23     SO.CNS = BIT2
24     SO.STR = BIT3
25     END
26
27     ;START A TABLE DEFINITION
28
29     ;YES/NO ANSWERS
30     ; REFORMAT MODE
31     ; (AGAIN)
32
33     ; RECONSTRUCT MODE
34     ; RESTORE MODE
35
36     .WORD L10042-L$SOFT/2
37     L$SOFT;;
```

```

1 023006          GPRML S.FMT,SO.BIT,SO.FM1,YES ;REFORMAT?          .WORD T$CODE
  023006 000130          .WORD S.FMT
  023010 023223          .WORD SO.FM1
2 023014          XFERT SWEND          .WORD T$CODE
  023014 017024          GPRML S.NRF,SO.BIT,SO.FM2,YES ;AGAIN - REFORMAT?    .WORD T$CODE
3 023016          .WORD S.NRF
  023016 000130          .WORD SO.FM2
  023020 023052          XFERT SWEND          .WORD T$CODE
  023022 000002          GPRML S.CNS,SO.BIT,SO.CNS,YES ;RECONSTRUCT          .WORD T$CODE
4 023024          .WORD S.CNS
  023024 013024          .WORD SO.CNS
5 023026          XFERT SWEND          .WORD T$CODE
  023026 000130          GPRML S.RST,SO.BIT,SO.STR,YES ;RESTORE?          .WORD T$CODE
  023030 023302          .WORD S.RST
  023032 000004          .WORD SO.STR
6 023034          XFERT SWEND          .WORD T$CODE
  023034 007024          GPRML S.NOF ;WARNING          .WORD T$CODE
7 023036          DISPLAY S.NOF ;WARNING          .WORD S.NOF
  023036 000130          .WORD T$CODE
  023040 023345          .WORD S.RST
  023042 000010          .WORD SO.STR
8 023044          XFERT SWEND          .WORD T$CODE
  023044 003024          DISPLAY S.NOF ;WARNING          .WORD T$CODE
9 023046          .WORD S.NOF
  023046 000003          .WORD T$CODE
  023050 023466          .WORD S.NOF
10 023052          SWEND: ENDSFT          .EVEN
      023052          L10042:

```

```

11
12 023052          015 012 S.NRF: .BYTE 15,12
13 023054          116 117 124 .ASCII\NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR\
14 023156          015 012 .BYTE 15,12
15 023160          111 116 106 .ASCII\INFORMATION ON THE DISKS.\
16 023211          015 012 .BYTE 15,12
17 023213          101 107 101 .ASCII\AGAIN - \
18 023223          122 105 106 S.FMT: .ASCIZ\REFORMAT USING EXISTING BAD SECTOR INFORMATION\
19 023302          122 105 103 S.CNS: .ASCIZ\RECONSTRUCT BAD SECTOR INFORMATION\
20 023345          104 117 040 S.RST: .ASCII\DO YOU HAVE A FILE ON THE SYSTEM LOAD DEVICE\
21 023421          015 012 .BYTE 15,12
22 023423          040 103 117 .ASCIZ\ CONTAINING BAD SECTOR INFORMATION\
23 023466          131 117 125 S.NOF: .ASCIZ\YOU CANNOT PROCEED WITHOUT SUCH A FILE.\
24 023536          122 105 123 .ASCIZ\RESTART PROGRAM AND SELECT TO REFORMAT OR RECONSTRUCT DISK.\
25 023632          000 .BYTE 0
26          .EVEN
27
28          .DSABL AMA
29 000000          .PSECT END

```

1  
2  
3 000000  
4  
5  
6  
7  
8 000120  
  
000120 000142'  
000122 000007  
000124  
9  
10 000124

.SBTTL PATCH AREA  
  
\$PATCH::  
.REPT 40.  
.WORD 0  
.ENDR  
  
LASTAD  
  
L\$LAST::  
  
ENDMOD

.EVEN  
.WORD T\$FREE  
.WORD T\$SIZE



```

1 000124          BGNSETUP          1
2
3 000124          BGNPTAB
   000124 000000
   000126 000005
   000130
4
5 000130 172150   .WORD 172150      ; UNIBUS ADDRESS
6 000132 000154   .WORD 154          ; VECTOR ADDRESS
7 000134 000005   .WORD 5            ; BR LEVEL
8 000136 000077   .WORD 63           ; UNIBUS BURST RATE
9 000140 000000   .WORD 0            ; LOGICAL DRIVE NUMBER
10
11 000142          ENDPTAB
   000142
12
13 000142          ENDSETUP
14
15
16
17
18
19
20
21          000001          .END

```

L10043:            .WORD 0  
                  .L10045-./2-1  
  
L10045:

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 276  
Work file writes: 268  
Size of work file: 28936 Words ( 114 Pages)  
Size of core pool: 17152 Words ( 67 Pages)  
Operating system: RT-11 (Under RSTS/E)

Elapsed time: 00:02:44.47  
ZUDED0.OBJ,ZUDED0/C-[20,0]SVC34R.MLB/P:1,ZUDED0.DOC,ZUDED0

\$PATCH	135-30																			
ADR	34-100																			
ALOCM	60-160	61-14	122-13	123-20	125-4															
ASSEMB	30-8	30-8																		
BAS	54-140	87-5	87-5	87-5	88-5	88-5														
BASL2	54-120	88-5																		
BASL3	54-130																			
BASLN	54-160	87-5	88-5																	
BASNO	54-110	87-5	88-5																	
BIT0	34-100	133-19																		
BIT00	34-10	34-100																		
BIT01	34-10	34-100																		
BIT02	34-10	34-100																		
BIT03	34-10	34-100																		
BIT04	34-10	34-100																		
BIT05	34-10	34-100																		
BIT06	34-10	34-100																		
BIT07	34-10	34-100																		
BIT08	34-10	34-100																		
BIT09	34-10	34-100																		
BIT1	34-100	45-23	133-20																	
BIT10	34-100																			
BIT11	34-100																			
BIT12	34-100																			
BIT13	34-100																			
BIT14	34-100																			
BIT15	34-100	45-15	46-12	63-27	66-20	74-28	79-15	93-29												
BIT2	34-100	45-24	133-22																	
BIT3	34-100	45-25	133-23																	
BIT4	34-100	45-27																		
BIT5	34-100	45-30																		
BIT6	34-100	45-31																		
BIT7	34-100	45-33																		
BIT8	34-100																			
BIT9	34-100																			
BLDC0	101-22	101-240																		
BLDC1	101-260	101-28																		
BLDCMD	65-49	68-14	68-44	99-2	101-150															
BOE	34-100																			
C#AU	30-80																			
C#AUTO	30-80	128-12																		
C#BRK	30-80	62-12	65-8	104-21	111-12	114-27														
C#BSEG	30-80																			
C#BSUB	30-80																			
C#CEFG	30-80																			
C#CLCK	30-80	120-41	120-43																	
C#CLEA	30-80	129-13																		
C#CLOS	30-80	64-12	77-19																	
C#CLP1	30-80																			
C#CVEC	30-80	62-22	111-30																	
C#DCLN	30-80	59-8	62-30	120-18	127-4	127-9	127-15	127-21												
C#DODU	30-80																			
C#DRPT	30-80																			
C#DU	30-80																			
C#EDIT	30-80	30-34																		
C#ERDF	30-80	62-28	65-24	66-36	67-21	68-8	72-22	74-32	100-3	104-29	105-5	109-35	109-54	110-11						

















L\$HIME	30-340				
L\$HPCF	30-340				
L\$HPTP	30-340				
L\$HW	30-34	32-10	32-100		
L\$ICP	30-340				
L\$INIT	30-34	120-80			
L\$LADP	30-340				
L\$LAST	30-34	135-80	136-13		
L\$LOAD	30-340				
L\$LUN	30-340	63-24*	65-12*	79-14*	
L\$MREV	30-340				
L\$NAME	30-340				
L\$PRIO	30-340				
L\$PROT	30-34	119-100			
L\$PRT	30-340				
L\$REPP	30-340				
L\$REV	30-340				
L\$SOFT	30-34	133-12	133-120		
L\$SPC	30-340				
L\$SPCP	30-340				
L\$SPTP	30-340				
L\$STA	30-340				
L\$SW	30-34	33-10	33-100		
L\$TEST	30-340				
L\$TIML	30-340				
L\$UNIT	30-340	80-13	123-16	126-4	130-25
L10000	32-10	32-170			
L10001	33-10	33-140			
L10002	57-160				
L10003	57-200				
L10004	57-240				
L10005	57-280				
L10006	57-320				
L10007	57-360				
L10010	57-400				
L10011	57-440				
L10012	57-480				
L10013	57-570				
L10014	57-620				
L10015	57-760				
L10016	57-800				
L10017	57-840				
L10020	57-880				
L10021	57-920				
L10022	57-970				
L10023	57-1010				
L10024	57-1050				
L10025	57-1090				
L10026	57-1130				
L10027	57-1170				
L10030	57-1210				
L10031	106-140				
L10032	107-210				
L10033	116-90				
L10035	126-14	127-230			
L10036	128-120				



NCONS	81-37*	81-40				
NEWTAB	121-22	122-3*				
NOCLOC	54-10*	120-46				
NULL	50-38*					
NXMAD	50-22*	62-10*	62-18	106-12*	111-26*	111-31
NXMI	62-11	106-10*	111-27			
NXTTAB	121-16	122-31	123-10	123-15*		
O\$APTS	30-8*	30-34				
O\$AU	30-8*	30-34				
O\$BGNR	30-8*	30-34				
O\$BGHS	30-8*	30-32*	30-34			
O\$DU	30-8*	30-34				
O\$ERRT	30-8*	30-34				
O\$GNSW	30-8*	30-32*	30-34			
O\$POIN	30-8*	30-32	30-32*	30-32*	30-32*	30-34
O\$SETU	30-8*	30-32*	30-34	135-8		
OP.ABO	40-3*					
OP.ACC	40-4*					
OP.AVA	40-22*					
OP.AVL	40-5*					
OP.CCD	40-6*					
OP.CMP	40-7*					
OP.DUP	40-23*					
OP.ELP	40-30*					
OP.END	40-20*	67-5	67-8	68-58		
OP.ERS	40-8*					
OP.ESP	40-29*	99-1				
OP.FLU	40-9*					
OP.GCS	40-10*					
OP.GDS	40-27*	65-48	68-58			
OP.GSS	40-28*					
OP.GUS	40-11*					
OP.MRD	40-18*					
OP.MWR	40-19*	101-21				
OP.ONL	40-12*					
OP.RD	40-13*					
OP.RLC	40-25*					
OP.RPL	40-14*					
OP.RSD	40-32*	67-8	68-43			
OP.SCC	40-15*					
OP.SEX	40-21*					
OP.SHC	40-24*					
OP.SSD	40-31*	67-5	68-13			
OP.SJC	40-16*					
OP.WR	40-17*					
OSTRE	81-35	81-42	81-47*			
OSTRNG	81-34*	81-46	89-6	90-6	91-6	96-17
P.BCNT	42-21*	43-9*	68-11	68-33*	99-4*	103-19*
P.BUFF	42-22*					
P.CMST	43-14*					
P.CNCL	43-48*					
P.CNTF	42-40*	43-46*				
P.CNTI	43-49*					
P.CPSP	42-34*					
P.CRF	42-17*	43-4*	67-19	102-17*		
P.CTMO	43-47*					













SVCTST	30-80	30-110	57-90	58-20	130-5									
SWEND	134-2	134-4	134-6	134-8	134-100									
T\$AUT	128-100	128-12												
T\$CLE	129-80	129-13												
T\$DAT	136-3	136-30	136-11											
T\$HAR	131-14	131-140	132-6											
T\$HW	32-10	32-100	32-17											
T\$INI	120-80	126-14	127-23											
T\$MSG	57-140	57-16	57-180	57-20	57-220	57-24	57-260	57-28	57-300	57-32	57-340	57-36	57-380	57-40
	57-420	57-44	57-460	57-48	57-500	57-57	57-590	57-62	57-640	57-76	57-780	57-80	57-820	57-84
	57-860	57-88	57-900	57-92	57-940	57-97	57-990	57-101	57-1030	57-105	57-1070	57-109	57-1110	57-113
	57-1150	57-117	57-1190	57-121										
T\$PC	136-10	136-13												
T\$PRO	119-100													
T\$PTA	136-10	136-3	136-30											
T\$SOF	133-12	133-120	134-10											
T\$SRV	106-100	106-14	107-180	107-21	116-50	116-9								
T\$SW	33-10	33-100	33-14											
T\$TES	130-50	130-22	130-28	130-82	130-83									
T\$ARGC	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-340	30-340	30-340
	30-340	30-340	30-340	95-12	95-12	95-12	95-120	95-120	95-14	95-14	95-14	95-140	95-140	95-16
	95-16	95-16	95-160	95-160	95-18	95-18	95-18	95-180	95-180	130-64	130-64	130-64	130-640	130-640
T\$CODE	118-1	118-1	118-1	118-10	118-10	118-10	130-10	130-10	130-10	130-100	130-100	130-100	130-31	130-31
	130-31	130-310	130-310	130-310	130-44	130-44	130-44	130-440	130-440	130-440	132-1	132-1	132-1	132-10
	132-10	132-10	132-2	132-2	132-2	132-20	132-20	132-20	132-3	132-3	132-3	132-30	132-30	132-30
	132-4	132-4	132-4	132-40	132-40	132-40	132-5	132-5	132-5	132-50	132-50	132-50	134-1	134-1
	134-1	134-10	134-10	134-10	134-2	134-2	134-2	134-2	134-2	134-2	134-20	134-20	134-20	134-20
	134-3	134-3	134-3	134-30	134-30	134-30	134-4	134-4	134-4	134-4	134-4	134-4	134-40	134-40
	134-40	134-40	134-5	134-5	134-5	134-50	134-50	134-50	134-6	134-6	134-6	134-6	134-6	134-6
	134-60	134-60	134-60	134-60	134-7	134-7	134-7	134-70	134-70	134-70	134-8	134-8	134-8	134-8
	134-8	134-8	134-80	134-80	134-80	134-80	134-9	134-90						
T\$ERRN	30-80	59-7	59-70	62-28	62-280	65-24	65-240	66-36	66-360	67-21	67-210	68-8	68-80	72-22
	72-220	74-32	74-320	100-3	100-30	104-29	104-290	105-5	105-50	109-35	109-350	109-54	109-540	110-11
	110-110	111-33	111-330	113-44	113-440	114-36	114-360	115-5	115-50	127-3	127-30	127-8	127-80	127-14
	127-140	127-20	127-200	130-21	130-210	130-27	130-270							
T\$EXCP	118-1	118-10	130-31	130-310	130-44	130-440	132-1	132-10	132-2	132-20	132-3	132-30	132-4	132-40
	132-5	132-50												
T\$FLAG	126-14	126-14	126-140	126-140	130-22	130-22	130-220	130-220	130-28	130-28	130-280	130-280	130-82	130-82
	130-820	130-820												
T\$FREE	135-8	136-130												
T\$GMAN	30-80	118-1	118-10	118-10	130-31	130-310	130-310	130-44	130-440	130-440				
T\$HILI	118-1	118-10	130-31	130-310	130-44	130-440	132-1	132-10	132-2	132-20	132-3	132-30	132-4	132-40
	132-5	132-50												
T\$LAST	30-80	135-80	136-1											
T\$LOLI	118-1	118-10	130-31	130-310	130-44	130-440	132-1	132-10	132-2	132-20	132-3	132-30	132-4	132-40
	132-5	132-50												
T\$LSYM	30-8	30-80	32-17	33-14	57-16	57-20	57-24	57-28	57-32	57-36	57-40	57-44	57-48	57-57
	57-62	57-76	57-80	57-84	57-88	57-92	57-97	57-101	57-105	57-109	57-113	57-117	57-121	106-14
	107-21	116-9	127-23	128-12	129-13	130-83	132-6	134-10						
T\$LTNO	135-80													
T\$NEST	30-80	30-26	30-26	30-260	32-10	32-10	32-100	32-17	32-17	32-17	32-170	33-10	33-10	33-100
	33-14	33-14	33-14	33-140	33-16	33-16	33-160	33-16	33-160	34-3	34-30	57-14	57-14	57-140
	57-16	57-16	57-16	57-160	57-18	57-18	57-180	57-20	57-20	57-20	57-200	57-22	57-22	57-220
	57-24	57-24	57-24	57-240	57-26	57-26	57-260	57-28	57-28	57-28	57-280	57-30	57-30	57-300
	57-32	57-32	57-32	57-320	57-34	57-34	57-340	57-36	57-36	57-36	57-360	57-38	57-38	57-380
	57-40	57-40	57-40	57-400	57-42	57-42	57-420	57-44	57-44	57-44	57-440	57-46	57-46	57-460

	57-48	57-48	57-48	57-480	57-50	57-50	57-500	57-57	57-57	57-57	57-570	57-59	57-59	57-590
	57-62	57-62	57-62	57-620	57-64	57-64	57-640	57-76	57-76	57-76	57-760	57-78	57-78	57-780
	57-80	57-80	57-80	57-800	57-82	57-82	57-820	57-84	57-84	57-84	57-840	57-86	57-86	57-860
	57-88	57-88	57-88	57-880	57-90	57-90	57-900	57-92	57-92	57-92	57-920	57-94	57-94	57-940
	57-97	57-97	57-97	57-970	57-99	57-99	57-990	57-101	57-101	57-101	57-1010	57-103	57-103	57-1030
	57-105	57-105	57-105	57-1050	57-107	57-107	57-1070	57-109	57-109	57-109	57-1090	57-111	57-111	57-1110
	57-113	57-113	57-113	57-1130	57-115	57-115	57-1150	57-117	57-117	57-117	57-1170	57-119	57-119	57-1190
	57-121	57-121	57-121	57-1210	106-10	106-10	106-100	106-14	106-14	106-14	106-140	107-18	107-18	107-180
	107-21	107-21	107-21	107-210	116-5	116-5	116-50	116-9	116-9	116-9	116-90	118-104	118-104	118-104
	118-1040	119-3	119-3	119-30	119-10	119-10	119-100	119-16	119-16	119-16	119-160	120-8	120-8	120-80
	127-23	127-23	127-23	127-230	128-10	128-10	128-100	128-12	128-12	128-12	128-120	129-8	129-8	129-80
	129-13	129-13	129-13	129-130	129-15	129-15	129-15	129-150	130-3	130-3	130-30	130-5	130-5	130-50
	130-83	130-83	130-83	130-830	130-84	130-84	130-84	130-840	131-3	131-3	131-30	131-14	131-14	131-140
	132-6	132-6	132-6	132-60	133-12	133-12	133-120	134-2	134-2	134-6	134-8	134-10	134-10	134-10
	134-100	135-10	135-10	135-10	135-100									
T&NSO	30-260	33-16	34-30	118-104	119-30	129-15	130-30	130-84	131-30	135-10				
T&NSI	32-100	32-17	33-100	33-14	57-140	57-16	57-180	57-20	57-220	57-24	57-260	57-28	57-300	57-32
	57-340	57-36	57-380	57-40	57-420	57-44	57-460	57-48	57-500	57-57	57-590	57-62	57-640	57-76
	57-780	57-80	57-820	57-84	57-860	57-88	57-900	57-92	57-940	57-97	57-990	57-101	57-1030	57-105
	57-1070	57-109	57-1110	57-113	57-1150	57-117	57-1190	57-121	106-100	106-14	107-180	107-21	116-50	116-9
	119-100	119-16	120-80	127-23	128-100	128-12	129-80	129-13	130-50	130-83	131-140	132-6	133-120	134-2
	134-4	134-6	134-8	134-10										
T&PCNT	136-10	136-3	136-3	136-30										
T&PTAB	136-3	136-30												
T&PTHV	30-34	136-130												
T&PTNU	30-80	136-3	136-30	136-13	136-13									
T&SAVL	30-80													
T&SEGL	30-80													
T&SIZE	135-8	136-130												
T&SUBN	30-80	130-50												
T&TAGL	30-80													
T&TAGN	30-80	32-10	32-10	32-100	33-10	33-10	33-100	57-14	57-14	57-140	57-18	57-18	57-180	57-22
	57-22	57-220	57-26	57-26	57-260	57-30	57-30	57-300	57-34	57-34	57-340	57-38	57-38	57-380
	57-42	57-42	57-420	57-46	57-46	57-460	57-50	57-50	57-500	57-59	57-59	57-590	57-64	57-64
	57-640	57-78	57-78	57-780	57-82	57-82	57-820	57-86	57-86	57-860	57-90	57-90	57-900	57-94
	57-94	57-940	57-99	57-99	57-990	57-103	57-103	57-1030	57-107	57-107	57-1070	57-111	57-111	57-1110
	57-115	57-115	57-1150	57-119	57-119	57-1190	106-10	106-10	106-100	107-18	107-18	107-180	116-5	116-5
	116-50	119-10	119-10	119-100	120-8	120-8	120-80	128-10	128-10	128-100	129-8	129-8	129-80	130-5
	130-5	130-50	131-14	131-14	131-140	133-12	133-12	133-120	136-1	136-1	136-10	136-3	136-3	136-3
	136-3	136-30	136-30											
T&TEMP	31-9	31-9	31-90	31-90	32-17	32-170	33-14	33-140	33-16	33-160	57-16	57-160	57-20	57-200
	57-24	57-240	57-28	57-280	57-32	57-320	57-36	57-360	57-40	57-400	57-44	57-440	57-48	57-480
	57-57	57-570	57-62	57-620	57-76	57-760	57-80	57-800	57-84	57-840	57-88	57-880	57-92	57-920
	57-97	57-970	57-101	57-1010	57-105	57-1050	57-109	57-1090	57-113	57-1130	57-117	57-1170	57-121	57-1210
	106-14	106-140	107-21	107-210	116-9	116-90	118-1	118-1	118-1	118-10	118-10	118-10	118-104	118-1040
	119-16	119-160	126-14	126-140	127-23	127-230	128-12	128-120	129-13	129-130	129-15	129-150	130-10	130-10
	130-10	130-100	130-100	130-100	130-22	130-220	130-28	130-280	130-31	130-31	130-31	130-310	130-310	130-310
	130-44	130-44	130-44	130-440	130-440	130-440	130-82	130-820	130-83	130-830	130-84	130-840	132-1	132-1
	132-1	132-10	132-10	132-10	132-2	132-2	132-2	132-20	132-20	132-20	132-3	132-3	132-3	132-30
	132-30	132-30	132-4	132-4	132-4	132-40	132-40	132-40	132-5	132-5	132-5	132-50	132-50	132-50
	132-6	132-60	134-1	134-1	134-1	134-10	134-10	134-10	134-3	134-3	134-3	134-30	134-30	134-30
	134-5	134-5	134-5	134-50	134-50	134-50	134-7	134-7	134-7	134-70	134-70	134-70	134-10	134-100
	135-10	135-100												
T&TEST	30-80	130-5	130-5	130-50	135-8									
T&TSTM	30-80	57-16	57-20	57-24	57-28	57-32	57-36	57-40	57-44	57-48	57-57	57-62	57-76	57-80
	57-84	57-88	57-92	57-97	57-101	57-105	57-109	57-113	57-117	57-121	59-7	59-8	62-11	62-12



UDAIR4	113-9	113-370					
UDAIRC	113-15	113-24	113-32	113-420			
UDAIRX	113-43	113-460					
UDAISE	111-35	112-9	112-11	112-190			
UDAISG	111-32	112-30					
UDAISL	112-80	112-16					
UDAIST	109-28	111-120					
UDAISX	112-14	112-180					
UDARS1	114-250	114-29	114-32	114-34			
UDARS2	114-26	115-30					
UDARSD	57-60*	57-61	112-3*	112-12*	112-13	114-19*	114-25 115-160
UDARSE	114-37	115-60					
UDARSP	109-41	112-8	114-180				
UDARSX	115-4	115-110					
UDASRV	107-180	122-23					
UF.576	42-120						
UF.CMR	42-30						
UF.CMW	42-40						
UF.INA	42-60						
UF.RPL	42-50						
UF.SCH	42-70						
UF.SCL	42-80						
UF.WBN	42-90						
UF.WPH	42-100						
UF.WPS	42-110						
UFREEZ	50-210	63-35*	66-3	66-13*	74-21	74-23*	
URNING	50-180	63-16*	63-31*	63-40	66-32*		
URUN	50-170	63-15*	63-20	65-7			
WAITMS	99-8	104-110					
WINQUES	53-60	130-10					
WINSTOP	54-180	74-40					
WINSTR	54-210	130-6					
X\$ALWA	30-80						
X\$FALS	30-80						
X\$OFFS	30-80	134-2	134-4	134-6	134-8		
X\$TRUE	30-80	134-2	134-4	134-6	134-8		
X1	55-50	57-15					
X10	55-130	57-39					
X100	55-410	57-116					
X101	55-420	57-120					
X14	55-140	57-43					
X1A	55-10	57-15					
X2	55-60	57-19					
X20	55-180	57-47					
X21	55-220	57-56					
X22	55-240	57-61					
X23A	55-260	57-65					
X23B	55-300	57-69					
X24	55-310	57-79					
X25	55-330	57-83					
X2A	55-20	57-19					
X3	55-70	57-23					
X30	55-350	57-87					
X31	55-360	57-91					
X32	55-370	57-95					
X36	55-380	57-108					

X37	55-40#	57-112	
X3A	55-3#	57-23	
X4	55-8#	57-27	
X8	55-10#	57-31	
X8A	55-4#	57-31	
X9	55-11#	57-35	
XFRU	56-8#	57-75	90-5
XMSG1	56-1#	57-136	
XMSG2	56-2#	57-140	
XPKT1	56-3#	57-123	
XPKT2	56-6#	57-129	
XSA	56-7#	91-5	
YEAR19	51-31#	118-92	
YEAR20	51-32#	118-95	
YER1	118-69#	118-82	
YER2	118-70	118-72	118-83#
YER3	118-86	118-92#	
YER4	118-94	118-96#	118-99
YER5	118-91	118-97	118-100# 118-101



GPHARD	121-15	124-4												
GPRMA	132-1	132-2												
GPRMD	118-1	118-10	130-31	130-310	130-44	130-440	132-3	132-4	132-5					
GPRML	130-10	130-100	134-1	134-3	134-5	134-7								
HEADER	30-34													
ITEM	35-240	45-12	45-13	45-16	45-19	45-20	45-21	45-22	45-34	45-35	45-36	45-37	45-38	45-39
	45-40	45-41	45-42	45-43	45-44	45-45	45-46	45-47	46-9	46-10	46-13	131-20	131-21	131-22
	131-23	131-24	133-18											
LASTAD	135-8													
M#BYTE	30-34	30-34	30-34	30-340										
M#CMC	126-14	126-140	130-22	130-220	130-28	130-280	130-82	130-820						
M#CNT0	118-1	118-10	130-10	130-100	130-31	130-310	130-44	130-440	132-1	132-10	132-2	132-20	132-3	132-30
	132-4	132-40	132-5	132-50	134-1	134-10	134-3	134-30	134-5	134-50	134-7	134-70		
M#COUN	95-12	95-120	95-14	95-140	95-16	95-160	95-18	95-180	130-64	130-640				
M#DATA	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34
	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34
	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-340	30-340	52-12	52-120	52-16
M#DECR	52-160													
	32-17	32-170	33-14	33-140	33-16	33-160	57-16	57-160	57-20	57-200	57-24	57-240	57-28	57-280
	57-32	57-320	57-36	57-360	57-40	57-400	57-44	57-440	57-48	57-480	57-57	57-570	57-62	57-620
	57-76	57-760	57-80	57-800	57-84	57-840	57-88	57-880	57-92	57-920	57-97	57-970	57-101	57-1010
	57-105	57-1050	57-109	57-1090	57-113	57-1130	57-117	57-1170	57-121	57-1210	106-14	106-140	107-21	107-210
	116-9	116-90	118-104	118-1040	119-16	119-160	127-23	127-230	128-12	128-120	129-13	129-130	129-15	129-150
M#DEFA	130-83	130-830	130-84	130-840	132-6	132-60	134-10	134-100	135-10	135-100	136-3	136-30		
	118-1	118-10	130-10	130-100	130-31	130-310	130-44	130-440	132-1	132-10	132-2	132-20	132-3	132-30
	132-4	132-40	132-5	132-50	134-1	134-10	134-3	134-30	134-5	134-50	134-7	134-70		
M#ENDE	32-170	33-140	33-160	57-160	57-200	57-240	57-280	57-320	57-360	57-400	57-440	57-480	57-570	57-620
	57-760	57-800	57-840	57-880	57-920	57-970	57-1010	57-1050	57-1090	57-1130	57-1170	57-1210	106-140	107-210
	116-90	118-1040	127-230	128-120	129-130	129-150	130-830	130-840	132-60	134-100	135-100			
M#ERRI	59-7	59-70	62-28	62-280	65-24	65-240	66-36	66-360	67-21	67-210	68-8	68-80	72-22	72-220
	74-32	74-320	100-3	100-30	104-29	104-290	105-5	105-50	109-35	109-350	109-54	109-540	110-11	110-110
	111-33	111-330	113-44	113-440	114-36	114-360	115-5	115-50	127-3	127-30	127-8	127-80	127-14	127-140
	127-20	127-200	130-21	130-210	130-27	130-270								
M#EXCP	118-1	118-1	118-10	130-31	130-31	130-310	130-44	130-44	130-440	132-1	132-1	132-10	132-2	132-2
	132-20	132-3	132-3	132-30	132-4	132-4	132-40	132-5	132-5	132-50				
M#EXIT	126-14	126-140	130-22	130-220	130-28	130-280	130-82	130-820						
M#EXSE	126-140	130-220	130-280	130-820										
M#EXTJ	126-140	130-220	130-280	130-820										
M#GEN	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34
	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34
	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-34	30-340	30-340	30-340	30-340
	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340
	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340	30-340
	30-340	30-340	30-340	30-340	30-340	30-340	31-9	31-90	32-10	32-100	32-100	32-17	32-170	32-170
	33-10	33-10	33-100	33-100	33-14	33-140	52-12	52-120	52-16	52-160	57-14	57-140	57-16	57-160
	57-18	57-180	57-20	57-200	57-22	57-220	57-24	57-240	57-26	57-260	57-28	57-280	57-30	57-300
	57-32	57-320	57-34	57-340	57-36	57-360	57-38	57-380	57-40	57-400	57-42	57-420	57-44	57-440
	57-46	57-460	57-48	57-480	57-50	57-500	57-57	57-570	57-59	57-590	57-62	57-620	57-64	57-640
	57-76	57-760	57-78	57-780	57-80	57-800	57-82	57-820	57-84	57-840	57-86	57-860	57-88	57-880
	57-90	57-900	57-92	57-920	57-94	57-940	57-97	57-970	57-99	57-990	57-101	57-1010	57-103	57-1030
	57-105	57-1050	57-107	57-1070	57-109	57-1090	57-111	57-1110	57-113	57-1130	57-115	57-1150	57-117	57-1170
	57-119	57-1190	57-121	57-1210	106-10	106-100	106-14	106-140	107-18	107-180	107-21	107-210	116-5	116-50
	116-9	116-90	118-1	118-10	119-10	119-100	120-8	120-80	127-23	127-230	128-10	128-100	128-12	128-120
	129-8	129-80	129-13	129-130	130-5	130-50	130-10	130-100	130-31	130-310	130-44	130-440	130-83	130-830
	131-14	131-140	132-6	132-60	133-12	133-120	134-10	134-100	135-8	135-80	136-3	136-30	136-11	136-110
M#GENE	118-1	118-10	130-10	130-100	130-31	130-310	130-44	130-440						





	121-15	121-15	121-15	121-15	121-15	121-16	121-16	124-4	124-4	124-4	124-4	124-4	124-5	124-5
	126-12	126-12	126-12	126-12	126-14	126-14	126-14	127-3	127-3	127-3	127-3	127-3	127-3	127-3
	127-3	127-3	127-3	127-4	127-4	127-8	127-8	127-8	127-8	127-8	127-8	127-8	127-8	127-8
	127-9	127-9	127-14	127-14	127-14	127-14	127-14	127-14	127-14	127-14	127-14	127-15	127-15	127-20
	127-20	127-20	127-20	127-20	127-20	127-20	127-20	127-21	127-21	127-23	127-23	127-23	128-12	128-12
	129-13	129-13	130-7	130-7	130-8	130-8	130-10	130-10	130-10	130-10	130-10	130-10	130-10	130-10
	130-10	130-10	130-19	130-19	130-20	130-20	130-21	130-21	130-21	130-21	130-21	130-21	130-21	130-21
	130-21	130-22	130-22	130-22	130-22	130-27	130-27	130-27	130-27	130-27	130-27	130-27	130-27	130-27
	130-28	130-28	130-28	130-28	130-31	130-31	130-31	130-31	130-31	130-31	130-31	130-31	130-31	130-31
	130-31	130-31	130-32	130-32	130-32	130-32	130-44	130-44	130-44	130-44	130-44	130-44	130-44	130-44
	130-44	130-44	130-44	130-44	130-64	130-64	130-64	130-64	130-64	130-64	130-64	130-64	130-64	130-64
	130-64	130-82	130-82	130-82	130-82	130-83	130-83	131-14	131-14	132-1	132-1	132-1	132-1	132-1
	132-2	132-2	132-2	132-2	132-2	132-3	132-3	132-3	132-3	132-3	132-3	132-4	132-4	132-4
	132-4	132-4	132-4	132-5	132-5	132-5	132-5	132-5	132-5	132-6	132-6	133-12	133-12	134-1
	134-1	134-1	134-1	134-2	134-2	134-3	134-3	134-3	134-3	134-4	134-4	134-5	134-5	134-5
	134-5	134-6	134-6	134-7	134-7	134-7	134-7	134-8	134-8	134-9	134-9	134-9	134-9	134-10
	134-10	135-8	135-8	135-8	135-8	136-3	136-3	136-3	136-3					
M&GNLS	118-1	118-1	130-10	130-10	130-31	130-31	130-44	130-44						
M&GNTA	32-17	32-17	33-14	33-14	57-16	57-16	57-20	57-20	57-24	57-24	57-28	57-28	57-32	57-32
	57-36	57-36	57-40	57-40	57-44	57-44	57-48	57-48	57-57	57-57	57-62	57-62	57-76	57-76
	57-80	57-80	57-84	57-84	57-88	57-88	57-92	57-92	57-97	57-97	57-101	57-101	57-105	57-105
	57-109	57-109	57-113	57-113	57-117	57-117	57-121	57-121	106-14	106-14	107-21	107-21	116-9	116-9
	127-23	127-23	128-12	128-12	129-13	129-13	130-83	130-83	132-6	132-6	134-10	134-10	136-3	136-3
	136-11	136-11												
M&GNTE	130-5	130-5												
M&HAPT	30-34	30-34												
M&HNAP	30-34	30-34												
M&INCR	30-26	30-26	32-10	32-10	32-10	32-10	33-10	33-10	33-10	33-10	34-3	34-3	57-14	57-14
	57-14	57-14	57-16	57-18	57-18	57-18	57-18	57-20	57-22	57-22	57-22	57-22	57-24	57-26
	57-26	57-26	57-26	57-28	57-30	57-30	57-30	57-30	57-32	57-34	57-34	57-34	57-34	57-36
	57-38	57-38	57-38	57-38	57-40	57-42	57-42	57-42	57-42	57-44	57-46	57-46	57-46	57-46
	57-48	57-50	57-50	57-50	57-50	57-57	57-59	57-59	57-59	57-59	57-62	57-64	57-64	57-64
	57-64	57-76	57-78	57-78	57-78	57-78	57-80	57-82	57-82	57-82	57-82	57-84	57-86	57-86
	57-86	57-86	57-88	57-90	57-90	57-90	57-90	57-92	57-94	57-94	57-94	57-94	57-97	57-99
	57-99	57-99	57-99	57-101	57-103	57-103	57-103	57-103	57-105	57-107	57-107	57-107	57-107	57-109
	57-111	57-111	57-111	57-111	57-113	57-115	57-115	57-115	57-115	57-117	57-119	57-119	57-119	57-119
	57-121	59-7	59-8	62-11	62-12	62-22	62-28	62-30	64-12	65-8	65-24	66-36	67-21	68-8
	72-22	74-32	77-19	77-20	77-24	95-12	95-14	95-16	95-18	98-20	100-3	104-21	104-29	105-5
	106-10	106-10	106-10	106-10	107-18	107-18	107-18	107-18	109-35	109-54	110-11	111-12	111-27	111-30
	111-33	113-44	114-27	114-36	115-5	116-5	116-5	116-5	116-5	118-1	118-1	118-1	119-3	119-3
	119-10	119-10	119-10	119-10	120-8	120-8	120-8	120-8	120-10	120-12	120-14	120-16	120-18	120-32
	120-41	120-43	120-52	121-15	124-4	126-12	126-14	127-3	127-4	127-8	127-9	127-14	127-15	127-20
	127-21	127-23	128-10	128-10	128-10	128-10	128-12	129-8	129-8	129-8	129-8	129-13	130-3	130-3
	130-5	130-5	130-5	130-5	130-5	130-5	130-7	130-10	130-10	130-10	130-19	130-21	130-22	130-27
	130-28	130-31	130-31	130-31	130-32	130-44	130-44	130-44	130-64	130-82	130-83	131-3	131-3	131-14
	131-14	131-14	131-14	133-12	133-12	133-12	133-12	136-1	136-1	136-3	136-3	136-3	136-3	136-3
M&LDRO	62-22	62-22	77-20	77-20	111-30	111-30	120-10	120-10	120-12	120-12	120-14	120-14	120-16	120-16
	120-41	120-41	120-43	120-43	121-15	121-15	124-4	124-4	126-12	126-12	130-32	130-32		
M&MCHI	30-8	30-8												
M&MCLO	30-8	30-8												
M&POP	32-17	32-17	33-14	33-14	33-16	33-16	57-16	57-16	57-20	57-20	57-24	57-24	57-28	57-28
	57-32	57-32	57-36	57-36	57-40	57-40	57-44	57-44	57-48	57-48	57-57	57-57	57-62	57-62
	57-76	57-76	57-80	57-80	57-84	57-84	57-88	57-88	57-92	57-92	57-97	57-97	57-101	57-101
	57-105	57-105	57-109	57-109	57-113	57-113	57-117	57-117	57-121	57-121	106-14	106-14	107-21	107-21
	116-9	116-9	118-104	118-104	119-16	119-16	119-16	127-23	127-23	128-12	128-12	129-13	129-13	129-15
	130-83	130-83	130-84	130-84	132-6	132-6	134-10	134-10	135-10	135-10				

M\$PRIN	95-12	95-120	95-14	95-140	95-16	95-160	95-18	95-180	130-64	130-640				
M\$PUSH	30-26	30-260	32-10	32-100	33-10	33-100	34-3	34-30	57-14	57-140	57-18	57-180	57-22	57-220
	57-26	57-260	57-30	57-300	57-34	57-340	57-38	57-380	57-42	57-420	57-46	57-460	57-50	57-500
	57-59	57-590	57-64	57-640	57-78	57-780	57-82	57-820	57-86	57-860	57-90	57-900	57-94	57-940
	57-99	57-990	57-103	57-1030	57-107	57-1070	57-111	57-1110	57-115	57-1150	57-119	57-1190	106-10	106-100
	107-18	107-180	116-5	116-50	119-3	119-30	119-10	119-100	120-8	120-80	128-10	128-100	129-8	129-80
	130-3	130-30	130-5	130-50	131-3	131-30	131-14	131-140	133-12	133-120				
M\$PUT	62-11	62-11	62-11	62-11	62-110	95-12	95-12	95-12	95-120	95-14	95-14	95-14	95-140	95-16
	95-16	95-16	95-160	95-18	95-18	95-18	95-180	98-20	98-20	98-20	98-20	98-200	111-27	111-27
	111-27	111-27	111-270	120-52	120-52	120-52	120-52	120-520	130-64	130-64	130-64	130-640		
M\$PUT1	62-11	62-11	62-11	62-11	62-110	62-110	62-110	62-110	95-12	95-12	95-12	95-120	95-120	95-120
	95-14	95-14	95-14	95-140	95-140	95-140	95-16	95-16	95-16	95-160	95-160	95-160	95-18	95-18
	95-18	95-180	95-180	95-180	98-20	98-20	98-20	98-20	98-200	98-200	98-200	98-200	111-27	111-27
	111-27	111-27	111-270	111-270	111-270	111-270	120-52	120-52	120-52	120-52	120-520	120-520	120-520	120-520
	130-64	130-64	130-64	130-640	130-640	130-640								
M\$RADI	118-1	118-10	130-10	130-100	130-31	130-310	130-44	130-440	132-1	132-10	132-2	132-20	132-3	132-30
	132-4	132-40	132-5	132-50	134-1	134-10	134-3	134-30	134-5	134-50	134-7	134-70		
M\$RBRO	77-24	77-240												
M\$RNRO	120-32	120-320	120-41	120-410	120-43	120-430	121-15	121-150	124-4	124-40				
M\$SETS	30-26	30-260	32-10	32-100	33-10	33-100	34-3	34-30	57-14	57-140	57-18	57-180	57-22	57-220
	57-26	57-260	57-30	57-300	57-34	57-340	57-38	57-380	57-42	57-420	57-46	57-460	57-50	57-500
	57-59	57-590	57-64	57-640	57-78	57-780	57-82	57-820	57-86	57-860	57-90	57-900	57-94	57-940
	57-99	57-990	57-103	57-1030	57-107	57-1070	57-111	57-1110	57-115	57-1150	57-119	57-1190	106-10	106-100
	107-18	107-180	116-5	116-50	119-3	119-30	119-10	119-100	120-8	120-80	128-10	128-100	129-8	129-80
	130-3	130-30	130-5	130-50	131-3	131-30	131-14	131-140	133-12	133-120				
M\$SVC	57-16	57-160	57-20	57-200	57-24	57-240	57-28	57-280	57-32	57-320	57-36	57-360	57-40	57-400
	57-44	57-440	57-48	57-480	57-57	57-570	57-62	57-620	57-76	57-760	57-80	57-800	57-84	57-840
	57-88	57-880	57-92	57-920	57-97	57-970	57-101	57-1010	57-105	57-1050	57-109	57-1090	57-113	57-1130
	57-117	57-1170	57-121	57-1210	59-7	59-8	59-80	62-11	62-110	62-12	62-120	62-22	62-220	62-28
	62-30	62-300	64-12	64-120	65-8	65-80	65-24	66-36	67-21	68-8	72-22	74-32	77-19	77-190
	77-20	77-200	77-24	77-240	95-12	95-120	95-14	95-140	95-16	95-160	95-18	95-180	98-20	98-200
	100-3	104-21	104-210	104-29	105-5	109-35	109-54	110-11	111-12	111-120	111-27	111-270	111-30	111-300
	111-33	113-44	114-27	114-270	114-36	115-5	118-1	118-10	120-10	120-100	120-12	120-120	120-14	120-140
	120-16	120-160	120-18	120-180	120-32	120-320	120-41	120-410	120-43	120-430	120-52	120-520	121-15	121-150
	124-4	124-40	126-12	126-120	126-14	126-140	127-3	127-4	127-40	127-8	127-9	127-90	127-14	127-15
	127-150	127-20	127-21	127-210	127-23	127-230	128-12	128-120	129-13	129-130	130-7	130-70	130-10	130-100
	130-19	130-190	130-21	130-22	130-220	130-27	130-28	130-280	130-31	130-310	130-32	130-320	130-44	130-440
	130-64	130-640	130-82	130-820	130-83	130-830								
M\$TLAB	57-160	57-200	57-240	57-280	57-320	57-360	57-400	57-440	57-480	57-570	57-620	57-760	57-800	57-840
	57-880	57-920	57-970	57-1010	57-1050	57-1090	57-1130	57-1170	57-1210	59-70	59-80	62-110	62-120	62-220
	62-280	62-300	64-120	65-80	65-240	66-360	67-210	68-80	72-220	74-320	77-190	77-200	77-240	95-120
	95-140	95-160	95-180	98-200	100-30	104-210	104-290	105-50	109-350	109-540	110-110	111-120	111-270	111-300
	111-330	113-440	114-270	114-360	115-50	118-10	120-100	120-120	120-140	120-160	120-180	120-320	120-410	120-430
	120-520	121-150	124-40	126-120	126-140	127-30	127-40	127-80	127-90	127-140	127-150	127-200	127-210	127-230
	128-120	129-130	130-70	130-100	130-190	130-210	130-220	130-270	130-280	130-310	130-320	130-440	130-640	130-820
	130-830													
M\$TSTL	57-16	57-160	57-20	57-200	57-24	57-240	57-28	57-280	57-32	57-320	57-36	57-360	57-40	57-400
	57-44	57-440	57-48	57-480	57-57	57-570	57-62	57-620	57-76	57-760	57-80	57-800	57-84	57-840
	57-88	57-880	57-92	57-920	57-97	57-970	57-101	57-1010	57-105	57-1050	57-109	57-1090	57-113	57-1130
	57-117	57-1170	57-121	57-1210	59-7	59-70	59-70	59-8	59-80	62-11	62-110	62-12	62-120	62-22
	62-220	62-28	62-280	62-280	62-30	62-300	64-12	64-120	65-8	65-80	65-24	65-240	65-240	66-36
	66-360	66-360	67-21	67-210	67-210	68-8	68-80	68-80	72-22	72-220	72-220	74-32	74-320	74-320
	77-19	77-190	77-20	77-200	77-24	77-240	95-12	95-120	95-14	95-140	95-16	95-160	95-18	95-180
	98-20	98-200	100-3	100-30	100-30	104-21	104-210	104-29	104-290	104-290	105-5	105-50	105-50	109-35
	109-350	109-350	109-54	109-540	109-540	110-11	110-110	110-110	111-12	111-120	111-27	111-270	111-30	111-300
	111-33	111-330	111-330	113-44	113-440	113-440	114-27	114-270	114-36	114-360	114-360	115-5	115-50	115-50

