

RABO

PDP-11 UDA DRV FMTR
CZUDECO

AH-S837C-MC
FICHE 1 OF 1

JAN 1983
COPYRIGHT © 81-82
MADE IN USA



.REM ~

IDENTIFICATION

PRODUCT CODE: AC-S836C-MC
PRODUCT NAME: CZUDECO UDA DISK FORMATTER
PRODUCT DATE: 27-AUG-82
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: MATT TEDONE

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

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

COPYRIGHT (C) 1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSEUS

.REM ~

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	9
2.8 QUICK STARTUP PROCEDURE	12
3.0 ERROR INFORMATION	14
3.1 TYPES OF ERROR MESSAGES	14
3.2 SPECIFIC ERROR MESSAGES	15
3.2.1 HOST PROGRAM ERROR MESSAGES	15
3.2.2 DUP PROGRAM ERROR MESSAGES	23
4.0 PERFORMANCE AND PROGRESS REPORTS	26
5.0 TEST SUMMARIES	27

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

This program will format any disk drive connected to a UDA-50 disk controller. 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/RAB0 spare HDA from RM80 format to RAB0 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 UDA-50 disk controller. The host program simply downline loads the DUP program in the UDA-50 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 UDA-50 can only format one disk at a time, so each disk on a UDA-50 will be selected sequentially. If the disk drives to be formatted are connected to different UDA-50s, all UDA-50s will be run simultaneously. For example, let's assume three units are selected for formatting in the hardware questions, units 1 and 2 are connected to one UDA-50 and unit 3 is connected to a different UDA-50 (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 RESURE 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 UDA-50 subsystems
- XXDP+ load media containing this program and the ZUDGA0.PAK data file.

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.

The diagnostic program requires that the data file ZUDGA0.PAK be on the XXDP+ system device. This data file is ordered under the name CZUDEB0. The XXDP+ system device must remain on-line during the execution of the this diagnostic.

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

COMMAND	EFFECT
START	Start the diagnostic from an initial state
RESTART	Start the diagnostic without initializing
CONTINUE	Continue at test that was interrupted (after ^C)

PRGCEED	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

Once the program is started, 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 of be formatted before the format operation begins.

SERIAL NUMBER FOR UNIT xx LDA 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 (O) ? 160000<CR>
SUB-DEVICE # (O) ? 5<CR>
Q-FACTOR (O) 0 ? <CR>

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

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

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

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

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

UNITS (D) ? 8<CR>

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

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

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

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

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

UNITS (D) ? 8<CR>

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

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

2.8 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 ZUDECO'
4. Type 'START'
5. Answer the 'CHANGE HW' question with 'Y'
6. Answer all the hardware questions
7. Answer the 'CHANGE SW' question with 'N'

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

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

DR>STA

CHANGE HW (L) ? Y

UNITS (D) ? 2

UNIT 0

UNIBUS ADDRESS OF UDA (D) 172150 ?

VECTOR (D) 154 ?

BR LEVEL (D) 5 ?

UNIBUS BURST RATE (D) 63 ?

DRIVE NUMBER (D) 0 ? 0,1

CHANGE SW (L) ? N

ENTER DATE AS DD-MMM-YY (A) 1-JAN-70 ? 14-APR-82
UNIT 0 UDA AT 172150 DRIVE 0 RUNTIME 0:12:20
Format completed
2 Rectorored LBNS
2 Primary revectorored LBNS
0 Secondary/tertiary revectorored LBNS
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 0:25:18
Format completed
6 Rectorored LBNS
5 Primary revectorored LBNS
1 Secondary/tertiary revectorored LBNS
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
4 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 HW (L) ? N
CHANGE SW (L) ? Y
REFORMAT USING EXISTING BAD SECTOR INFORMATION (L) Y ? Y
ENTER DATA AS DD-MMM-YY (A) 1-JAN-70 ? 14-APR-82
RUNTIME 0:12:45
Format completed
2 Rectorored LBNS
2 Primary revectorored LBNS
0 Secondary/tertiary revectorored LBNS
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 UDA-50 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 UDA-50 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 xxxxxx identifies the address of the UDA-50 being tested. It may be omitted if the error is not specific to one UDA-50.

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 M7161 /
```

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, UDA-50 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 UDA-50 can have only one vector, BR level or burst rate. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

```
00002 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 UDA-50 at one time. A physical disk drive may be from one to four logical disk drives. Each logical disk drive is considered one unit to the diagnostic program. Even though more than eight logical disk drives can be attached to one UDA-50, the UDA-50 only supports eight. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00004 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 UDA-50 Unibus addresses but identical vector addresses. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00009 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.

00011 CZUDE DVC FTL ERR 00011 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
WRONG APT FORMATTER IS BEING USED WITH THIS CONTROLLER.
USE CIUDx.

There are two APT compatible formatters. CIUDK is used for UDA-50s with the M7161-2 modules. CIUDI is used for UDA-50s with the M7485-6 modules. If one formatter program is used with the wrong UDA-50 module set, this error will appear. Use the other APT formatter program to successfully format.

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 M7161 OR M7485
OR UNIBUS
OR REPLACE UDA MODULE M7161 OR 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 M7161 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
UDA RESIDENT DIAGNOSTICS DETECTED FAIL
UDASA CONTAINS 105154
REPLACE UDA MODULE M7162 OR M7486

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

- 104000 - Fatal 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. M7161 and M7485 are the Unibus interface boards. M7162 and M7486 are the SDI interface boards. Module M7161 only works with the module M7182. The same for modules M7485 and M7486. Do not intermix the two boards. It will not work!

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 M7161 OR M7485

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

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

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

00023 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 M7161 OR M7485

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

00024 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 M7161 module or 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 M7161 OR M7485

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

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 000020
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 M7161 OR M7485

While loading the DM program to the UDA, the UDASA became non-zero. When this occurs, it signifies that the UDA microcode has run across a fatal error. The displayed value is in octal. Check the error code with the list 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 M7161 or M7485.

Formatter initialization error

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

Nonexistent 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 M7161 or M7485 problem.

Drive init timeout

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

Illegal response to start-up question

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

An example of how the errors are presented is below:

RUNTIME 0:00:18
Nonexistant 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 Revectorized LBNS

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

n Primary revectorized LBNS

Where n is the number of LBNS in message #2 which were primary revectorized.

n Secondary/tertiary revectorized LBNS

Where n is the number of the LBNS in message #2 which were secondary or tertiary revectorized.

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 0:24:57
Format completed
  5 Revectoring LBNS
  5 Primary revectoring LBNS
  0 Secondary/tertiary revectoring LBNS
  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 UDA-50.

57
58
59

```

1          .SBTTL  PROGRAM HEADER
25
26 002000          BGNMOD
27          :++
28          : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
29          : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
30          :--
31
32 002000          POINTER BGNSW, BGNSFT, BGNSETUP
33
34 002000          HEADER CZUDE,C,0,0,1,PRI07      ;FIELD SERVICE
002000
002000          103
002001          132
002002          125
002003          104
002004          105
002005          000
002006          000
002007          000
002010
002010          103
002011
002011          060
002012
002012          000001
002014
002014          000000
002016
002016          022762
002020
002020          023150
002022
002022          002130
002024
002024          002144
002026
002026          000124'
002030
002030          000000
002032
002032          000000
002034
002034          000001
002036
002036          000000
002040
002040          002124
002042
002042          000340
002044
002044          000000
002046
002046          000000
002050
002050          003
002051          003

```

```

LSNAME::
          .ASCII /C/
          .ASCII /Z/
          .ASCII /U/
          .ASCII /D/
          .ASCII /E/
          .BYTE 0
          .BYTE 0
          .BYTE 0
LSREV::
          .ASCII /C/
LSDEPO::
          .ASCII /O/
LSUNIT::
          .WORD  T$PTHV
L$TIML::
          .WORD  0
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 003514
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 000000
002072
002072 000000
002074
002074 000000
002076
002076 003536
002100
002100 104035
002102
002102 000000
002104
002104 020376
002106
002106 022072
002110
002110 022070
002112
002112 020370
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\$EXP5::
 .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 022102

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

```
1      .SBTTL  DEFAULT HARDWARE P-TABLE
2
3      ;++
4      ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
5      ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
6      ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
7      ; AND IS USED AS A 'TEMPLATE' FOR BUILDING THE P-TABLES.
8      ;--
9
10     002126      BGNHW  DFPTBL
11     002126      000005
12     002130
13     002130      LSHW::      .WORD  L10000-LSHW/2
14     002130      DFPTBL::
15
16     002130      172150      .WORD  172150      ; UNIBUS ADDRESS
17     002132      000154      .WORD  154        ; VECTOR ADDRESS
18     002134      000005      .WORD  5          ; BR LEVEL
19     002136      000077      .WORD  63         ; UNIBUS BURST RATE
20     002140      000000      .WORD  0          ; LOGICAL DRIVE NUMBER
21     002142
22     002142      ENDHW
23
24     L10000:
```

```
1
2
3
4
5
6
7
8
9
10 002142      BGNSW  SFPTBL
    002142      000001
    002144
    002144
11
12
13 002144      000007
14 002146
    002146
15
16 002146

                                LSSW:: .WORD  L10001-LSSW/2
                                SFPTBL::

                                ;OFFSET  USE
                                ; 0.    YES/NO ANSWERS
                                L10001:

                                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 DEFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1
001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

; EVENT FLAG DEFINITIONS

EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

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

; PRIORITY LEVEL DEFINITIONS

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

000140	PRI03==	140
000100	PRI02==	100
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	LIAM==	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      ;MACRO DEFINITIONS FOR GLOBAL EQUATES
2
3      ;THESE MACROS ARE USED TO DEFINE INDEXES INTO A TABLE
4
5      ;CALLING SEQUENCE MUST BE
6
7      :
8      :      TABLE
9      :      ITEM      NAME      BYTES
10     :      ITEM      NAME      BYTES
11     :      ITEM      NAME      BYTES
12     :      END      SIZE
13     :
14     ;TABLE DEFINES THAT A TABLE IS ABOUT TO BE DEFINED AND END TERMINATES THE DEFINITION.
15     ;ANY NUMBER OF ITEM LINES CAN APPEAR. NAME IS THE NAME OF THE SYMBOL BEING EQUATED TO
16     ;THE INDEX. THE INDEX ALWAYS STARTS AT ZERO. BYTES SPECIFIES THE SIZE OF THE VALUE TO BE
17     ;STORED AT THAT INDEX IN BYTES. THE SIZE ARGUMENT TO THE END STATEMENT IS OPTIONAL, IT
18     ;BE EQUATED TO THE SIZE OF THE TABLE IN BYTES. THE SYMBOL TINDEX IS USED TO KEEP TRACK
19     ;OF THE INDEX VALUE AND WILL BE EQUAL TO THE SIZE OF THE TABLE AFTER THE END STATEMENT.
20
21     .MACRO TABLE
22         TINDEX=0
23     .ENDM
24
25     .MACRO ITEM NAME BYTES
26         NAME=TINDEX
27         TINDEX=TINDEX+BYTES
28     .ENDM
29
30     .MACRO END SIZE
31         .IF NB SIZE
32         SIZE=TINDEX
33         .ENDC
34     .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		;	;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		;	;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		;	;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     000122      HC.BSZ= 82.     ;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     000306      HC.BF2= HC.BF1+HC.BSZ ;SECOND BUFFER
26
27     000430      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				
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				

HC.INT	INTERRUPT INDICATORS	4 BYTES
HC.MSG HC.MCT	MESSAGE RING	4 BYTES
HC.CMD HC.CCT	COMMAND RING	4 BYTES
HC.MEV HC.MPK	MESSAGE ENVELOPE	52 BYTES
HC.CEV HC.CPK	COMMAND ENVELOPE	52 BYTES
HC.BF1	BUFFER # 1 (RESPONSE TO DM PROGRAM)	82 BYTES
HC.BF2	BUFFER # 2 (REQUEST FROM DM PROGRAM)	82 BYTES

```

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

1		;UNIT FLAGS	
2			
3	000001	UF.CMR= 000001	;COMPARE READS
4	000002	UF.CMW= 000002	;COMPARE WRITES
5	100000	UF.RPL= 100000	;HOST INITIATED BAD BLOCK REPLACEMENT
6	040000	UF.INA= 040000	;INACTIVE SHADOW SET UNIT
7	004000	UF.SCH= 004000	;SUPPRESS CACHING (HIGH SPEED)
8	002000	UF.SCL= 002000	;SUPPRESS CACHING (LOW SPEED)
9	000100	UF.WBN= 000100	;WRITE-BACK (NON-VOLATILE)
10	020000	UF.WPH= 020000	;WRITE PROTECT (HARDWARE)
11	001000	UF.WPS= 001000	;WRITE PROTECT (SOFTWARE OR VOLUME)
12	000004	UF.576= 000004	;576 BYTE SECTORS
13			
14		;COMMAND PACKET OFFSETS	
15			
16			GENERIC COMMAND PACKET OFFSETS:
17	000000	P.CRF= 0.	;COMMAND REFERENCE NUMBER
18	000004	P.UNIT= 4.	;UNIT NUMBER
19	000010	P.OPCD= 8.	;OPCODE
20	000012	P.MOD= 10.	;MODIFIERS
21	000014	P.BCNT= 12.	;BYTE COUNT
22	000020	P.BUFF= 16.	;BUFFER DESCRIPTOR
23	000020	P.UADR= 16.	;UNIBUS ADDRESS OF BUFFER DESCRIPTOR
24	000034	P.LBN= 28.	;LOGICAL BLOCK NUMBER
25			
26			ABORT AND GET COMMAND STATUS COMMAND PACKET OFFSETS:
27	000014	P.OTRF= 12.	;OUTSTANDING REFERENCE NUMBER
28			
29			ONLINE AND SET UNIT CHARACTERISTICS COMMAND PACKET OFFSETS:
30	000016	P.UNFL= 14.	;UNIT FLAGS
31	000020	P.HSTI= 16.	;HOST IDENTIFIER / RESERVED
32	000034	P.ELGF= 28.	;ERROR LOG FLAGS
33	000040	P.SHUN= 32.	;SHADOW UNIT
34	000042	P.CPSP= 34.	;COPY SPEED
35			REPLACE COMMAND PACKET OFFSETS:
36	000014	P.RBN= 12.	;REPLACEMENT BLOCK NUMBER
37			
38			SET CONTROLLER CHARACTERISTICS COMMAND PACKET OFFSETS:
39	000014	P.VRSN= 12.	;MSCP VERSION
40	000016	P.CNTF= 14.	;CONTROLLER FLAGS
41	000020	P.HTMO= 16.	;HOST TIMEOUT
42	000022	P.USEF= 18.	;USE FRACTION
43	000024	P.TIME= 20.	;QUAD-WORD TIME AND DATE
44			
45			MAINTENANCE READ AND MAINTENANCE WRITE COMMAND PACKET OFFSETS:
46	000034	P.RGID= 28.	;REGION ID
47	000040	P.RGOF= 32.	;REGION OFFSET
48			
49			EXECUTE SUPPLIED PROGRAM COMMAND PACKET OFFSETS:
50	000024	P.DMDT= 20.	;DMDT TERMINAL ADDRESS (MAINT WRITE ONLY)
51	000034	P.OVRL= 28.	;BUFFER DESCRIPTOR FOR OVERLAYS

1		:END PACKET OFFSETS	
2			
3		:	GENERIC END PACKET OFFSETS:
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			GET COMMAND STATUS END PACKET OFFSETS:
13	000014	P.OTRF= 12.	:OUTSTANDING REFERENCE NUMBER
14	000020	P.CMST= 16.	:COMMAND STATUS
15		:	
16			GET UNIT STATUS END PACKET OFFSETS:
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		:	ONLINE AND SET UNIT CHARACTERISTICS END PACKET AND AVAILABLE
32		:	ATTENTION MESSAGE OFFSETS:
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		:	SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS:
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		:	GET DUST STATUS END PACKET OFFSETS:
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	DMOVR= 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 R0,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                                     ;WHENEVER THIS BIT IS SET, CT.CMD IS CLEARED
27             000020              CT.REQ= BIT4         ;BUFFER HAS BEEN GIVEN TO UDA FOR REQUEST
28                                     ;SET WHENEVER READ STUD DATA COMMAND
29                                     ;GIVEN TO UDA
30             000040              CT.STA= BIT5         ;GET DUST STATUS COMMAND HAS BEEN SENT
31             000100              CT.TM1= BIT6         ;ONE TIMEOUT PERIOD HAS EXPIRED BETWEEN SEND OR
32                                     ;RECEIVE DATA RESPONSE
33             000200              CT.TM2= BIT7         ;SECOND TIMEOUT HAS EXPIRED
34             000400              CT.U52= BIT8         ; UDA52 BIT
35             001000              CT.U50= BIT9         ; UDA50 BIT
36     002146          ITEM C.RING                2         ;RING BUFFER ADDRESS
37     002146          ITEM C.DR0                 2         ;POINTER TO DRIVE TABLES
38     002146          ITEM C.DR1                 2         ; IF ZERO, NO DRIVE TABLE EXISTS
39     002146          ITEM C.DR2                 2
40     002146          ITEM C.DR3                 2
41     002146          ITEM C.DR4                 2
42     002146          ITEM C.DR5                 2
43     002146          ITEM C.DR6                 2
44     002146          ITEM C.DR7                 2
45     002146          ITEM C.TO                  2         ;TIMEOUT COUNTER
46     002146          ITEM C.TOH                 2         ; (TWO WORDS)
47     002146          ITEM C.TOT                 2         ;DUP PROGRAM TIMEOUT VALUE IN SECONDS
48     002146          ITEM C.PRI                 4         ;DUP PROGRAM PROGRESS INDICATOR
49     002146          ITEM C.REF                 2         ;COMMAND REFERENCE NUMBER
50
51     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
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
39
40
41
42
43
44

;USEFUL INSTRUCTION DEFINITIONS

```
.MACRO AND ARG,ADR                ;LOGICAL AND INSTRUCTION
.LIST                                BIC #^C<ARG>,ADR

.NLIST

.ENDM

.MACRO OR ARG,ADR                 ;LOGICAL OR INSTRUCTION
.LIST                                BIS #ARG,ADR

.NLIST

.ENDM

.MACRO PUSH ARG                   ;PUSH INSTRUCTION
.IRP X,<ARG>
.LIST                                MOV X,-(SP)

.NLIST
.ENDM

.MACRO POP ARG                    ;POP INSTRUCTION
.IRP X,<ARG>
.LIST                                MOV (SP)+,X

.NLIST
.ENDM

.MACRO .BR ADR                     ;A BRANCH TO THE NEXT LOCATION
.IF P2
.IF NE .-ADR
.ERROR ;ILLEGAL .BR TO ADR
.ENDC
.ENDC

.ENDM

.MACRO ASSUME FIRST CONDITION SECOND
.IF CONDITION <FIRST>-<SECOND>
.IFF
.ERROR ;BAD ASSUME OF <FIRST> CONDITION <SECOND>
.ENDC

.ENDM
```

```

1      ;PRINT CHARACTER
2      ; ARGUMENT MUST BE SOURCE STATEMENT TO MOVE CHARACTER TO PRINT (MOV ARG,R0)
3      ; EX: 'PRINT R1' WILL PRINT THE CHARACTER IN R1
4      ; SPECIAL CASE: 'PRINT #CR' WILL PRINT END OF LINE SEQUENCE
5      ; THE PRINTING IS DONE AT THE MODE OF THE LAST PRINT LINE CALL
6      ; IE., PNTX, PNTB, PNTX, PNTS
7
8      .MACRO PRINT ARG1
9          .IF DIF <ARG1>,R0
10             .LIST
11
12             .NLIST
13
14             .ENDC
15             .LIST
16
17             .NLIST
18
19             .ENDM
20
21      ;PROCESSING MACRO FOR NEXT SET OF FORMATTED MESSAGE MACROS
22      .MACRO PNT... RTN,ADR,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
23          PNT.CT=0
24          .IRP AA,<ARG8,ARG7,ARG6,ARG5,ARG4,ARG3,ARG2,ARG1>
25              .IF NB,<AA>
26                  .LIST
27
28                  .NLIST
29                  PNT.CT=PNT.CT+2
30
31              .ENDC
32
33              .ENDM
34              .LIST
35
36              .NLIST
37
38              .ENDM
39
40          .NLIST
41
42          .ENDM
43
44          .NLIST
45
46          .ENDM
47
48          .NLIST
49
50          .ENDM
51
52          .NLIST
53
54          .ENDM
55
56          .NLIST
57
58          .ENDM
59
60          .NLIST
61
62          .ENDM
63
64          .NLIST
65
66          .ENDM
67
68          .NLIST
69
70          .ENDM
71
72          .NLIST
73
74          .ENDM
75
76          .NLIST
77
78          .ENDM
79
80          .NLIST
81
82          .ENDM
83
84          .NLIST
85
86          .ENDM
87
88          .NLIST
89
90          .ENDM
91
92          .NLIST
93
94          .ENDM
95
96          .NLIST
97
98          .ENDM
99
100         .NLIST
101
102         .ENDM
103
104         .NLIST
105
106         .ENDM
107
108         .NLIST
109
110         .ENDM
111
112         .NLIST
113
114         .ENDM
115
116         .NLIST
117
118         .ENDM
119
120         .NLIST
121
122         .ENDM
123
124         .NLIST
125
126         .ENDM
127
128         .NLIST
129
130         .ENDM
131
132         .NLIST
133
134         .ENDM
135
136         .NLIST
137
138         .ENDM
139
140         .NLIST
141
142         .ENDM
143
144         .NLIST
145
146         .ENDM
147
148         .NLIST
149
150         .ENDM
151
152         .NLIST
153
154         .ENDM
155
156         .NLIST
157
158         .ENDM
159
160         .NLIST
161
162         .ENDM
163
164         .NLIST
165
166         .ENDM
167
168         .NLIST
169
170         .ENDM
171
172         .NLIST
173
174         .ENDM
175
176         .NLIST
177
178         .ENDM
179
180         .NLIST
181
182         .ENDM
183
184         .NLIST
185
186         .ENDM
187
188         .NLIST
189
190         .ENDM
191
192         .NLIST
193
194         .ENDM
195
196         .NLIST
197
198         .ENDM
199
200         .NLIST
201
202         .ENDM
203
204         .NLIST
205
206         .ENDM
207
208         .NLIST
209
210         .ENDM
211
212         .NLIST
213
214         .ENDM
215
216         .NLIST
217
218         .ENDM
219
220         .NLIST
221
222         .ENDM
223
224         .NLIST
225
226         .ENDM
227
228         .NLIST
229
230         .ENDM
231
232         .NLIST
233
234         .ENDM
235
236         .NLIST
237
238         .ENDM
239
240         .NLIST
241
242         .ENDM
243
244         .NLIST
245
246         .ENDM
247
248         .NLIST
249
250         .ENDM
251
252         .NLIST
253
254         .ENDM
255
256         .NLIST
257
258         .ENDM
259
260         .NLIST
261
262         .ENDM
263
264         .NLIST
265
266         .ENDM
267
268         .NLIST
269
270         .ENDM
271
272         .NLIST
273
274         .ENDM
275
276         .NLIST
277
278         .ENDM
279
280         .NLIST
281
282         .ENDM
283
284         .NLIST
285
286         .ENDM
287
288         .NLIST
289
290         .ENDM
291
292         .NLIST
293
294         .ENDM
295
296         .NLIST
297
298         .ENDM
299
300         .NLIST
301
302         .ENDM
303
304         .NLIST
305
306         .ENDM
307
308         .NLIST
309
310         .ENDM
311
312         .NLIST
313
314         .ENDM
315
316         .NLIST
317
318         .ENDM
319
320         .NLIST
321
322         .ENDM
323
324         .NLIST
325
326         .ENDM
327
328         .NLIST
329
330         .ENDM
331
332         .NLIST
333
334         .ENDM
335
336         .NLIST
337
338         .ENDM
339
340         .NLIST
341
342         .ENDM
343
344         .NLIST
345
346         .ENDM
347
348         .NLIST
349
350         .ENDM
351
352         .NLIST
353
354         .ENDM
355
356         .NLIST
357
358         .ENDM
359
360         .NLIST
361
362         .ENDM
363
364         .NLIST
365
366         .ENDM
367
368         .NLIST
369
370         .ENDM
371
372         .NLIST
373
374         .ENDM
375
376         .NLIST
377
378         .ENDM
379
380         .NLIST
381
382         .ENDM
383
384         .NLIST
385
386         .ENDM
387
388         .NLIST
389
390         .ENDM
391
392         .NLIST
393
394         .ENDM
395
396         .NLIST
397
398         .ENDM
399
400         .NLIST
401
402         .ENDM
403
404         .NLIST
405
406         .ENDM
407
408         .NLIST
409
410         .ENDM
411
412         .NLIST
413
414         .ENDM
415
416         .NLIST
417
418         .ENDM
419
420         .NLIST
421
422         .ENDM
423
424         .NLIST
425
426         .ENDM
427
428         .NLIST
429
430         .ENDM
431
432         .NLIST
433
434         .ENDM
435
436         .NLIST
437
438         .ENDM
439
440         .NLIST
441
442         .ENDM
443
444         .NLIST
445
446         .ENDM
447
448         .NLIST
449
450         .ENDM
451
452         .NLIST
453
454         .ENDM
455
456         .NLIST
457
458         .ENDM
459
460         .NLIST
461
462         .ENDM
463
464         .NLIST
465
466         .ENDM
467
468         .NLIST
469
470         .ENDM
471
472         .NLIST
473
474         .ENDM
475
476         .NLIST
477
478         .ENDM
479
480         .NLIST
481
482         .ENDM
483
484         .NLIST
485
486         .ENDM
487
488         .NLIST
489
490         .ENDM
491
492         .NLIST
493
494         .ENDM
495
496         .NLIST
497
498         .ENDM
499
500         .NLIST
501
502         .ENDM
503
504         .NLIST
505
506         .ENDM
507
508         .NLIST
509
510         .ENDM
511
512         .NLIST
513
514         .ENDM
515
516         .NLIST
517
518         .ENDM
519
520         .NLIST
521
522         .ENDM
523
524         .NLIST
525
526         .ENDM
527
528         .NLIST
529
530         .ENDM
531
532         .NLIST
533
534         .ENDM
535
536         .NLIST
537
538         .ENDM
539
540         .NLIST
541
542         .ENDM
543
544         .NLIST
545
546         .ENDM
547
548         .NLIST
549
550         .ENDM
551
552         .NLIST
553
554         .ENDM
555
556         .NLIST
557
558         .ENDM
559
560         .NLIST
561
562         .ENDM
563
564         .NLIST
565
566         .ENDM
567
568         .NLIST
569
570         .ENDM
571
572         .NLIST
573
574         .ENDM
575
576         .NLIST
577
578         .ENDM
579
580         .NLIST
581
582         .ENDM
583
584         .NLIST
585
586         .ENDM
587
588         .NLIST
589
590         .ENDM
591
592         .NLIST
593
594         .ENDM
595
596         .NLIST
597
598         .ENDM
599
600         .NLIST
601
602         .ENDM
603
604         .NLIST
605
606         .ENDM
607
608         .NLIST
609
610         .ENDM
611
612         .NLIST
613
614         .ENDM
615
616         .NLIST
617
618         .ENDM
619
620         .NLIST
621
622         .ENDM
623
624         .NLIST
625
626         .ENDM
627
628         .NLIST
629
630         .ENDM
631
632         .NLIST
633
634         .ENDM
635
636         .NLIST
637
638         .ENDM
639
640         .NLIST
641
642         .ENDM
643
644         .NLIST
645
646         .ENDM
647
648         .NLIST
649
650         .ENDM
651
652         .NLIST
653
654         .ENDM
655
656         .NLIST
657
658         .ENDM
659
660         .NLIST
661
662         .ENDM
663
664         .NLIST
665
666         .ENDM
667
668         .NLIST
669
670         .ENDM
671
672         .NLIST
673
674         .ENDM
675
676         .NLIST
677
678         .ENDM
679
680         .NLIST
681
682         .ENDM
683
684         .NLIST
685
686         .ENDM
687
688         .NLIST
689
690         .ENDM
691
692         .NLIST
693
694         .ENDM
695
696         .NLIST
697
698         .ENDM
699
700         .NLIST
701
702         .ENDM
703
704         .NLIST
705
706         .ENDM
707
708         .NLIST
709
710         .ENDM
711
712         .NLIST
713
714         .ENDM
715
716         .NLIST
717
718         .ENDM
719
720         .NLIST
721
722         .ENDM
723
724         .NLIST
725
726         .ENDM
727
728         .NLIST
729
730         .ENDM
731
732         .NLIST
733
734         .ENDM
735
736         .NLIST
737
738         .ENDM
739
740         .NLIST
741
742         .ENDM
743
744         .NLIST
745
746         .ENDM
747
748         .NLIST
749
750         .ENDM
751
752         .NLIST
753
754         .ENDM
755
756         .NLIST
757
758         .ENDM
759
760         .NLIST
761
762         .ENDM
763
764         .NLIST
765
766         .ENDM
767
768         .NLIST
769
770         .ENDM
771
772         .NLIST
773
774         .ENDM
775
776         .NLIST
777
778         .ENDM
779
780         .NLIST
781
782         .ENDM
783
784         .NLIST
785
786         .ENDM
787
788         .NLIST
789
790         .ENDM
791
792         .NLIST
793
794         .ENDM
795
796         .NLIST
797
798         .ENDM
799
800         .NLIST
801
802         .ENDM
803
804         .NLIST
805
806         .ENDM
807
808         .NLIST
809
810         .ENDM
811
812         .NLIST
813
814         .ENDM
815
816         .NLIST
817
818         .ENDM
819
820         .NLIST
821
822         .ENDM
823
824         .NLIST
825
826         .ENDM
827
828         .NLIST
829
830         .ENDM
831
832         .NLIST
833
834         .ENDM
835
836         .NLIST
837
838         .ENDM
839
840         .NLIST
841
842         .ENDM
843
844         .NLIST
845
846         .ENDM
847
848         .NLIST
849
850         .ENDM
851
852         .NLIST
853
854         .ENDM
855
856         .NLIST
857
858         .ENDM
859
860         .NLIST
861
862         .ENDM
863
864         .NLIST
865
866         .ENDM
867
868         .NLIST
869
870         .ENDM
871
872         .NLIST
873
874         .ENDM
875
876         .NLIST
877
878         .ENDM
879
880         .NLIST
881
882         .ENDM
883
884         .NLIST
885
886         .ENDM
887
888         .NLIST
889
890         .ENDM
891
892         .NLIST
893
894         .ENDM
895
896         .NLIST
897
898         .ENDM
899
900         .NLIST
901
902         .ENDM
903
904         .NLIST
905
906         .ENDM
907
908         .NLIST
909
910         .ENDM
911
912         .NLIST
913
914         .ENDM
915
916         .NLIST
917
918         .ENDM
919
920         .NLIST
921
922         .ENDM
923
924         .NLIST
925
926         .ENDM
927
928         .NLIST
929
930         .ENDM
931
932         .NLIST
933
934         .ENDM
935
936         .NLIST
937
938         .ENDM
939
940         .NLIST
941
942         .ENDM
943
944         .NLIST
945
946         .ENDM
947
948         .NLIST
949
950         .ENDM
951
952         .NLIST
953
954         .ENDM
955
956         .NLIST
957
958         .ENDM
959
960         .NLIST
961
962         .ENDM
963
964         .NLIST
965
966         .ENDM
967
968         .NLIST
969
970         .ENDM
971
972         .NLIST
973
974         .ENDM
975
976         .NLIST
977
978         .ENDM
979
980         .NLIST
981
982         .ENDM
983
984         .NLIST
985
986         .ENDM
987
988         .NLIST
989
990         .ENDM
991
992         .NLIST
993
994         .ENDM
995
996         .NLIST
997
998         .ENDM
999
1000        .NLIST
1001
1002        .ENDM
1003
1004        .NLIST
1005
1006        .ENDM
1007
1008        .NLIST
1009
1010        .ENDM
1011
1012        .NLIST
1013
1014        .ENDM
1015
1016        .NLIST
1017
1018        .ENDM
1019
1020        .NLIST
1021
1022        .ENDM
1023
1024        .NLIST
1025
1026        .ENDM
1027
1028        .NLIST
1029
1030        .ENDM
1031
1032        .NLIST
1033
1034        .ENDM
1035
1036        .NLIST
1037
1038        .ENDM
1039
1040        .NLIST
1041
1042        .ENDM
1043
1044        .NLIST
1045
1046        .ENDM
1047
1048        .NLIST
1049
1050        .ENDM
1051
1052        .NLIST
1053
1054        .ENDM
1055
1056        .NLIST
1057
1058        .ENDM
1059
1060        .NLIST
1061
1062        .ENDM
1063
1064        .NLIST
1065
1066        .ENDM
1067
1068        .NLIST
1069
1070        .ENDM
1071
1072        .NLIST
1073
1074        .ENDM
1075
1076        .NLIST
1077
1078        .ENDM
1079
1080        .NLIST
1081
1082        .ENDM
1083
1084        .NLIST
1085
1086        .ENDM
1087
1088        .NLIST
1089
1090        .ENDM
1091
1092        .NLIST
1093
1094        .ENDM
1095
1096        .NLIST
1097
1098        .ENDM
1099
1100        .NLIST
1101
1102        .ENDM
1103
1104        .NLIST
1105
1106        .ENDM
1107
1108        .NLIST
1109
1110        .ENDM
1111
1112        .NLIST
1113
1114        .ENDM
1115
1116        .NLIST
1117
1118        .ENDM
1119
1120        .NLIST
1121
1122        .ENDM
1123
1124        .NLIST
1125
1126        .ENDM
1127
1128        .NLIST
1129
1130        .ENDM
1131
1132        .NLIST
1133
1134        .ENDM
1135
1136        .NLIST
1137
1138        .ENDM
1139
1140        .NLIST
1141
1142        .ENDM
1143
1144        .NLIST
1145
1146        .ENDM
1147
1148        .NLIST
1149
1150        .ENDM
1151
1152        .NLIST
1153
1154        .ENDM
1155
1156        .NLIST
1157
1158        .ENDM
1159
1160        .NLIST
1161
1162        .ENDM
1163
1164        .NLIST
1165
1166        .ENDM
1167
1168        .NLIST
1169
1170        .ENDM
1171
1172        .NLIST
1173
1174        .ENDM
1175
1176        .NLIST
1177
1178        .ENDM
1179
1180        .NLIST
1181
1182        .ENDM
1183
1184        .NLIST
1185
1186        .ENDM
1187
1188        .NLIST
1189
1190        .ENDM
1191
1192        .NLIST
1193
1194        .ENDM
1195
1196        .NLIST
1197
1198        .ENDM
1199
1200        .NLIST
1201
1202        .ENDM
1203
1204        .NLIST
1205
1206        .ENDM
1207
1208        .NLIST
1209
1210        .ENDM
1211
1212        .NLIST
1213
1214        .ENDM
1215
1216        .NLIST
1217
1218        .ENDM
1219
1220        .NLIST
1221
1222        .ENDM
1223
1224        .NLIST
1225
1226        .ENDM
1227
1228        .NLIST
1229
1230        .ENDM
1231
1232        .NLIST
1233
1234        .ENDM
1235
1236        .NLIST
1237
1238        .ENDM
1239
1240        .NLIST
1241
1242        .ENDM
1243
1244        .NLIST
1245
1246        .ENDM
1247
1248        .NLIST
1249
1250        .ENDM
1251
1252        .NLIST
1253
1254        .ENDM
1255
1256        .NLIST
1257
1258        .ENDM
1259
1260        .NLIST
1261
1262        .ENDM
1263
1264        .NLIST
1265
1266        .ENDM
1267
1268        .NLIST
1269
1270        .ENDM
1271
1272        .NLIST
1273
1274        .ENDM
1275
1276        .NLIST
1277
1278        .ENDM
1279
1280        .NLIST
1281
1282        .ENDM
1283
1284        .NLIST
1285
1286        .ENDM
1287
1288        .NLIST
1289
1290        .ENDM
1291
1292        .NLIST
1293
1294        .ENDM
1295
1296        .NLIST
1297
1298        .ENDM
1299
1300        .NLIST
1301
1302        .ENDM
1303
1304        .NLIST
1305
1306        .ENDM
1307
1308        .NLIST
1309
1310        .ENDM
1311
1312        .NLIST
1313
1314        .ENDM
1315
1316        .NLIST
1317
1318        .ENDM
1319
1320        .NLIST
1321
1322        .ENDM
1323
1324        .NLIST
1325
1326        .ENDM
1327
1328        .NLIST
1329
1330        .ENDM
1331
1332        .NLIST
1333
1334        .ENDM
1335
1336        .NLIST
1337
1338        .ENDM
1339
1340        .NLIST
1341
1342        .ENDM
1343
1344        .NLIST
1345
1346        .ENDM
1347
1348        .NLIST
1349
1350        .ENDM
1351
1352        .NLIST
1353
1354        .ENDM
1355
1356        .NLIST
1357
1358        .ENDM
1359
1360        .NLIST
1361
1362        .ENDM
1363
1364        .NLIST
1365
1366        .ENDM
1367
1368        .NLIST
1369
1370        .ENDM
1371
1372        .NLIST
1373
1374        .ENDM
1375
1376        .NLIST
1377
1378        .ENDM
1379
1380        .NLIST
1381
1382        .ENDM
1383
1384        .NLIST
1385
1386        .ENDM
1387
1388        .NLIST
1389
1390        .ENDM
1391
1392        .NLIST
1393
1394        .ENDM
1395
1396        .NLIST
1397
1398        .ENDM
1399
1400        .NLIST
1401
1402        .ENDM
1403
1404        .NLIST
1405
1406        .ENDM
1407
1408        .NLIST
1409
1410        .ENDM
1411
1412        .NLIST
1413
1414        .ENDM
1415
1416        .NLIST
1417
1418        .ENDM
1419
1420        .NLIST
1421
1422        .ENDM
1423
1424        .NLIST
1425
1426        .ENDM
1427
1428        .NLIST
1429
1430        .ENDM
1431
1432        .NLIST
1433
1434        .ENDM
1435
1436        .NLIST
1437
1438        .ENDM
1439
1440        .NLIST
1441
1442        .ENDM
1443
1444        .NLIST
1445
1446        .ENDM
1447
1448        .NLIST
1449
1450        .ENDM
1451
1452        .NLIST
1453
1454        .ENDM
1455
1456        .NLIST
1457
1458        .ENDM
1459
1460        .NLIST
1461
1462        .ENDM
1463
1464        .NLIST
1465
1466        .ENDM
1467
1468        .NLIST
1469
1470        .ENDM
1471
1472        .NLIST
1473
1474        .ENDM
1475
1476        .NLIST
1477
1478        .ENDM
1479
1480        .NLIST
1481
1482        .ENDM
1483
1484        .NLIST
1485
1486        .ENDM
1487
1488        .NLIST
1489
1490        .ENDM
1491
1492        .NLIST
1493
1494        .ENDM
1495
1496        .NLIST
1497
1498        .ENDM
1499
1500        .NLIST
1501
1502        .ENDM
1503
1504        .NLIST
1505
1506        .ENDM
1507
1508        .NLIST
1509
1510        .ENDM
1511
1512        .NLIST
1513
1514        .ENDM
1515
1516        .NLIST
1517
1518        .ENDM
1519
1520        .NLIST
1521
1522        .ENDM
1523
1524        .NLIST
1525
1526        .ENDM
1527
1528        .NLIST
1529
1530        .ENDM
1531
1532        .NLIST
1533
1534        .ENDM
1535
1536        .NLIST
1537
1538        .ENDM
1539
1540        .NLIST
1541
1542        .ENDM
1543
1544        .NLIST
1545
1546        .ENDM
1547
1548        .NLIST
1549
1550        .ENDM
1551
1552        .NLIST
1553
1554        .ENDM
1555
1556        .NLIST
1557
1558        .ENDM
1559
1560        .NLIST
1561
1562        .ENDM
1563
1564        .NLIST
1565
1566        .ENDM
1567
1568        .NLIST
1569
1570        .ENDM
1571
1572        .NLIST
1573
1574        .ENDM
1575
1576        .NLIST
1577
1578        .ENDM
1579
1580        .NLIST
1581
1582        .ENDM
1583
1584        .NLIST
1585
1586        .ENDM
1587
1588        .NLIST
1589
1590        .ENDM
1591
1592        .NLIST
1593
1594        .ENDM
1595
1596        .NLIST
1597
1598        .ENDM
1599
1600        .NLIST
1601
1602        .ENDM
1603
1604        .NLIST
1605
1606        .ENDM
1607
1608        .NLIST
1609
1610        .ENDM
1611
1612        .NLIST
1613
1614        .ENDM
1615
1616        .NLIST
1617
1618        .ENDM
1619
1620        .NLIST
1621
1622        .ENDM
1623
1624        .NLIST
1625
1626        .ENDM
1627
1628        .NLIST
1629
1630        .ENDM
1631
1632        .NLIST
1633
1634        .ENDM
1635
1636        .NLIST
1637
1638        .ENDM
1639
1640        .NLIST
1641
1642        .ENDM
1643
1644        .NLIST
1645
1646        .ENDM
1647
1648        .NLIST
1649
1650        .ENDM
1651
1652        .NLIST
1653
1654        .ENDM
1655
1656        .NLIST
1657
1658        .ENDM
1659
1660        .NLIST
1661
1662        .ENDM
1663
1664        .NLIST
1665
1666        .ENDM
1667
1668        .NLIST
1669
1670        .ENDM
1671
1672        .NLIST
1673
1674        .ENDM
1675
1676        .NLIST
1677
1678        .ENDM
1679
1680        .NLIST
1681
1682        .ENDM
1683
1684        .NLIST
1685
1686        .ENDM
1687
1688        .NLIST
1689
1690        .ENDM
1691
1692        .NLIST
1693
1694        .ENDM
1695
1696        .NLIST
1697
1698        .ENDM
1699
1700        .NLIST
1701
1702        .ENDM
1703
1704        .NLIST
1705
1706        .ENDM
1707
1708        .NLIST
1709
1710        .ENDM
1711
1712        .NLIST
1713
1714        .ENDM
1715
1716        .NLIST
1717
1718        .ENDM
1719
1720        .NLIST
1721
1722        .ENDM
1723
1724        .NLIST
1725
1726        .ENDM
1727
1728        .NLIST
1729
1730        .ENDM
1731
1732        .NLIST
1733
1734        .ENDM
1735
1736        .NLIST
1737
1738        .ENDM
1739
1740        .NLIST
1741
1742        .ENDM
1743
1744        .NLIST
1745
1746        .ENDM
1747
1748        .NLIST
1749
1750        .ENDM
1751
1752        .NLIST
1753
1754        .ENDM
1755
1756        .NLIST
1757
1758        .ENDM
1759
1760        .NLIST
1761
1762        .ENDM
1763
1764        .NLIST
1765
1766        .ENDM
1767
1768        .NLIST
1769
1770        .ENDM
1771
1772        .NLIST
1773
1774        .ENDM
1775
1776        .NLIST
1777
1778        .ENDM
1779
1780        .NLIST
1781
1782        .ENDM
1783
1784        .NLIST
1785
1786        .ENDM
1787
1788        .NLIST
1789
1790        .ENDM
1791
1792        .NLIST
1793
1794        .ENDM
1795
1796        .NLIST
1797
1798        .ENDM
1799
1800        .NLIST
1801
1802        .ENDM
1803
1804        .NLIST
1805
1806        .ENDM
1807
1808        .NLIST
1809
1810        .ENDM
1811
1812        .NLIST
1813
1814        .ENDM
1815
1816        .NLIST
1817
1818        .ENDM
1819
1820        .NLIST
1821
1822        .ENDM
1823
1824        .NLIST
1825
1826        .ENDM
1827
1828        .NLIST
1829
1830        .ENDM
1831
1832        .NLIST
1833
1834        .ENDM
1835
1836        .NLIST
1837
1838        .ENDM
1839
1840        .NLIST
1841
1842        .ENDM
1843
1844        .NLIST
1845
1846        .ENDM
1847
1848        .NLIST
1849
1850        .ENDM
1851
1852        .NLIST
1853
1854        .ENDM
1855
1856        .NLIST
1857
1858        .ENDM
1859
1860        .NLIST
1861
1862        .ENDM
1863
1864        .NLIST
1865
1866        .ENDM
1867
1868        .NLIST
1869
1870        .ENDM
1871
1872        .NLIST
1873
1874        .ENDM
1875
1876        .NLIST
1877
187
```

```
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
10
11 002146  FFREE:: .BLKW 1      ;FIRST FREE WORD IN MEMORY
12 002150  FSIZE:: .BLKW 1      ;SIZE OF FREE MEMORY IN WORDS
13 002152  FMEM:   .BLKW 1      ;COPY OF FFREE AT END OF INIT SECTION
14 002154  FMEMS:  .BLKW 1      ;COPY OF FSIZE AT END OF INIT SECTION
15 002156  CTABS:: .BLKW 1      ;START OF CONTROLLER TABLE STORAGE
16 002160  CTRLRS: .BLKW 1      ;COUNT OF UDA CONTROLLERS IN PTABLES
17 002162  TSTTAB: .BLKW 1      ;POINTER TO FIRST CONTROLLER TABLE UNDER TEST
18      :
19 002164  000000  DMPROG: .WORD 0      ;START ADDRESS OF DM PROGRAM
20 002166  000000  DMSIZE: .WORD 0      ;SIZE OF DM PROGRAM
21 002170  DMEND:  .BLKW 1      ;END ADDRESS OF DM PROGRAM(FIRST FREE MEMORY ADR)
22 002172  DMENDS: .BLKW 1      ;FREE MEMORY SIZE FROM END OF DM PROGRAM
23 002174  URUN:   .BLKW 1      ;NUMBER OF UNITS TO RUN AT ONE TIME
24 002176  URNING: .BLKW 1      ;NUMBER OF UNITS STILL RUNNING
25 002200  UCNT:   .BLKW 1      ;COUNTER OF UNITS UNDER TEST
26 002202  000000  FILOPN: .WORD 0      ; FILE OPEN
27 002204  000000  U50UNI: .WORD 0      ; KEEPS TRACK OF ANY UDA 50 CONTROLLERS
28 002206  125      065      062  U52EXT: .ASCII\U52\
29      :
30 002212  000000  TSTTYP: .WORD 0      ; CONTROLLER TYPE
31 002214  000000  FNUM:   .WORD 0      ; FILE NUMBER
32 002216  000000  INMEM:  .WORD 0      ; WHICH DM PROG IS IN MEMORY
33 002220  UFREEZ: .BLKW 1      ;FREEZE ON UNIT WHEN NOT ZERO
34 002222  NXMAD:  .BLKW 1      ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
35 002224  000000  FDATA:  .WORD 0
36 002226  FCTBUF: .BLKB 512.      ;STORAGE FOR FCT BLOCK
37 003226  FCTNUM: .BLKW 1      ;FCT BLOCK NUMBER
38 003230  MODE:   .BLKW 1 ;MODE WORD, SAME BIT DEFS AS SO.BIT
39
40      ;CLOCK CONTROL
41
42 003232  000000  KW.CSR: .WORD 0      ;CSR OF CLOCK
43 003234  KW.BRL: .BLKW 1      ;BR LEVEL
44 003236  KW.VEC: .BLKW 1      ;VECTOR
45 003240  KW.HZ:  .BLKW 1      ;HERTZ (50. OR 60.)
46 003242  KW.EL:  .BLKW 2      ;ELAPSED TIME
47
48 003246  016106  PTYPE:  .WORD PF      ;PRINT TYPE
49 003250  000      000      ERRCHR: .BYTE 0,0      ;FIRST BYTE LOADED WITH OUTPUT CHARACTER
50 003252  000000  NULL:   .WORD 0      ;USED TO PRINT A NULL CHARACTER
51 003254  FNAME2: .BLKB 10.
52 003266  132      125      104  FNAME: .ASCII\ZUDGA0.PAK\
53      :

```

1	003302			TEMP:	.BLKB 22.	:USED TO GET ANSWER FROM GMANID CALL
2	003330	061	055	112	DATE1:	.ASCIZ\1-JAN-70\ ;DEFAULT DATE
3	003341				.BLKB 3	
4	003344	000000			DATEO:	.WORD 0 ;DATE STRING IN FORMATTER FORMAT
5	003346				.BLKB 10.	:(FIRST WORD ZERO SAYS NO DATE HERE YET)
6	003360	061	070	064	HIGHEST:	.ASCIZ\18446744073709551615\ ;HIGHEST DISK SERIAL NUMBER
7	003405	104	105	103	MONTHS:	.ASCII\DEC\ ;NAME OF MONTHS
8	003410	116	117	126		.ASCII\NOV\
9	003413	117	103	124		.ASCII\OCT\
10	003416	123	105	120		.ASCII\SEP\
11	003421	101	125	107		.ASCII\AUG\
12	003424	112	125	114		.ASCII\JUL\
13	003427	112	125	116		.ASCII\JUN\
14	003432	115	101	131		.ASCII\MAY\
15	003435	101	120	122		.ASCII\APR\
16	003440	115	101	122		.ASCII\MAR\
17	003443	106	105	102		.ASCII\FEB\
18	003446	112	101	116		.ASCII\JAN\
19	003451	037			DAYS:	.BYTE 31. ;NUMBER OF DAYS IN EACH MONTH
20	003452	035				.BYTE 29.
21	003453	037				.BYTE 31.
22	003454	036				.BYTE 30.
23	003455	037				.BYTE 31.
24	003456	036				.BYTE 30.
25	003457	037				.BYTE 31.
26	003460	037				.BYTE 31.
27	003461	036				.BYTE 30.
28	003462	037				.BYTE 31.
29	003463	036				.BYTE 30.
30	003464	037				.BYTE 31.
31	003465	061	071	000	YEAR19:	.ASCIZ\19\
32	003470	062	060	000	YEAR20:	.ASCIZ\20\
33					.EVEN	
34	003474	000000			IPADRS:	.WORD 0
35	003476	000000				.WORD 0
36	003500	000000				.WORD 0
37	003502	000000				.WORD 0
38	003504	000000				.WORD 0
39	003506	000000				.WORD 0
40	003510	000000				.WORD 0
41	003512	000000				.WORD 0

```
1      .SBTTL  GLOBAL TEXT SECTION
2
3
4      :++
5      : THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
6      : MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
7      : MORE THAN ONE TEST.
8      :--
9
10     :
11     : NAMES OF DEVICES SUPPORTED BY PROGRAM
12     :
13     003514      DEVTYP <UDA-50 CONTROLLER>
14     003514      125      104      101
15     003514
16
17     : TEST DESCRIPTION
18     :
19     003536      DESCRIPT <CZUDECO UDA-50 DISK DRV FORMATTER>
20     003536      103      132      125
21     003536
```

L\$DVTYP::
 .ASCIZ /UDA-50 CONTROLLER/
 .EVEN

L\$DESC::
 .ASCIZ /CZUDECO UDA-50 DISK
 .EVEN

1					
2					
3	003600	105	116	124	DATEQ: .ASCIZ\ENTER DATE AS DD-MMM-YY\
4	003630	040	106	117	FILNAQ: .ASCIZ\ FOR DISK TO BE FORMATTED\
5	003662	040	000		SERNQ: .ASCIZ\ \

```

1          ; FORMAT STATEMENTS USED IN PRINT CALLS
2
3 003664    045    124    000  ERRONE: .ASCIZ\%T\
4 003667    045    116    000  ERRNL: .ASCIZ\%N\
5 003672    042    040    040  RNTIM: .ASCIZ\''  RUNTIME 'D16'':\
6 003715    104    071    042  RNTIM1: .ASCIZ\D9'':\
7 003723    104    071    000  RNTIM2: .ASCIZ\D9\
8 003726    042    040    040  ERRME1: .ASCIZ\'' * * * ERROR PROCESSING MESSAGE STRING * * *\
9 004015    116    042    125  MESSG: .ASCIZ\N'UNIT 'D6'' UDA AT 'D16'' DRIVE 'D9S\
10          ;:NOCLOCK:.ASCIZ\NO LINE CLOCK AVAILABLE'\
11 004061    042    116    117  NOCLOCK: .ASCIZ\NO LINE CLOCK AVAILABLE FOR TIMING EVENTS'\
12 004136    042    110    117  BASN0: .ASCIZ\HOST PROGRAM'\
13 004155    042    040    040  BASL2: .ASCIZ\'' UDA AT 'D16\
14 004174    042    040    040  BASL3: .ASCIZ\'' DRIVE 'D9\
15 004211    000          BAS:      .BYTE 0                      ;NULL TO PRINT NOTHING
16
17 004212    122    066    122  BASLN: .ASCIZ\R6R6R6R6\
18 004223    116    042    123  SERNUM: .ASCIZ\N'SERIAL NUMBER FOR UNIT 'D6'' UDA AT 'D16'' DRIVE 'D9\

```

1	004310			X1A:	
2	004310			X2A:	
3	004310			X3A:	
4	004310	042	111	040	X8A: .ASCIZ\''I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS'\
5	004407	122	065	122	X1: .ASCIZ\R5R6'UDA HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE'\
6	004503	122	065	122	X2: .ASCIZ\R5R6'TWO UNITS SELECT THE SAME DRIVE'\
7	004552	122	065	122	X3: .ASCIZ\R5R6'MORE THAN EIGHT DRIVES SELECTED ON THIS UDA'\
8	004635	122	064	042	X4: .ASCII\R4'NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED'\
9	004726	042	120	114	.ASCIZ\''PLEASE START PROGRAM OVER AND TEST FEWER UNITS AT A TIME'\
10	005022	122	064	042	X5: .ASCIZ\R4'CHECKSUM ERROR IN DM PROGRAM FILE '\
11	005072	122	064	042	X7: .ASCIZ\R4'ERROR IN DM PROGRAM FILE. DM PROGRAM NOT FOUND'\
12	005156	122	065	122	X8: .ASCIZ\R5R6'TWO UDA'S USE THE SAME VECTOR'\
13	005223	122	064	042	X9: .ASCII\R4'ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE.''\
14	005326	042	120	114	.ASCIZ\''PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK.''\
15	005415	122	064	042	X10: .ASCIZ\R4'THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE.''\
16	005514	122	065	042	X11: .ASCII\R5'WRONG APT FORMATTER IS BEING USED WITH THIS CONTROLLER.''\
23	005610	122	065	042	X20: .ASCII\R5'MEMORY ERROR TRYING TO READ UDA REGISTERS'\
24	005666	042	103	110	.ASCII\''CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7161 OR M7485'\
25	005765	042	117	122	.ASCII\''OR UNIBUS'\
26	006001	042	117	122	.ASCIZ\''OR 'R7'\
27	006011	122	065	042	X21: .ASCII\R5'UDA RESIDENT DIAGNOSTICS DETECTED FAILURE'NR8\
28	006071	042	122	105	.ASCIZ\''REPLACE UDA MODULE M716'D2'' OR M748'D3N\
29	006142	122	065	042	X22: .ASCII\R5'STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION'\
30	006243	042	123	124	.ASCIZ\''STEP BIT EXPECTED 'D16NR8R7\
31	006300	122	065	042	X23A: .ASCII\R5'UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION'\
32	006412	104	071	042	.ASCII\D9'' WORDS WERE TO BE CLEARED STARTING AT ADDRESS 'D16N\
33	006500	042	106	111	.ASCII\''FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):'\
34	006555	123	066	042	.ASCIZ\S6'ADDRESS'S4'CONTENTS'\
35	006606	123	067	117	X23B: .ASCIZ\S7016S5016N\
36	006622	122	065	042	X24: .ASCII\R5'UDASA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION'\
37	006735	042	120	125	.ASCIZ\''PURGE/POLE DIAGNOSTICS WERE REQUESTED'NR8R7\
38	007012	122	065	042	X25: .ASCII\R5'UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION'\
39	007126	042	040	040	.ASCIZ\'' UDASA EXPECTED 'D16NR8R7\
40	007163	122	065	042	X30: .ASCIZ\R5'UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE RUNNING DM PROGRAM'NR8\
41	007276	122	065	042	X31: .ASCIZ\R5'DUP PROGRAM IS HUNG'\
42	007327	122	065	042	X32: .ASCIZ\R5'MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER'\
43	007440	122	065	042	X36: .ASCII\R5'NO INTERRUPT RECEIVED FROM UDA FOR 30 SECONDS'\
44	007522	042	127	110	.ASCIZ\''WHILE LOADING DM PROGRAM'\
45	007556	122	065	042	X37: .ASCIZ\R5'UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM'NR8R7\
46	007673	122	065	042	X100: .ASCIZ\R5'DUP PROGRAM ASKED UNEXPECTED QUESTION ('D12')'\
47	007756	122	065	042	X101: .ASCIZ\R5'DUP PROGRAM REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION'\

1	010061	042	115	105	XMSG1:	.ASCIZ\MESSAGE BUFFER CONTAINS:'N\
2	010115	123	063	117	XMSG2:	.ASCIZ\S3016S1016S1016S1016S1016S1016S1016N\
3	010162	122	065	042	XPKT1:	.ASCII\R5'RESPONSE PACKET FROM UDA DOES NOT CONTAIN EXPECTED DATA'N\
4	010256	042	105	111		.ASCII\EITHER UDA RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED CORRECTLY'N\
5	010366	123	063	042		.ASCIZ\S3'COMMAND PACKET SENT'S6'RESPONSE PACKET RECEIVED'N\
6	010453	123	066	117	XPKT2:	.ASCIZ\S6016S1016S14016S1016N\
7	010502	042	040	040	XSA:	.ASCIZ\ UDASA CONTAINS '016N\
8	010533	042	122	105	XFRU:	.ASCIZ\REPLACE UDA MODULE M7161 OR M7485'N\
9						
10						
11	010600	045	101	111	SERNX:	.ASCIZ\%INPUT ERROR. ANSWER WITH DECIMAL NUMBER LO= 0 HI= %T\
12	010670	042	111	116	DATEX:	.ASCIZ\INPUT ERROR.'\
13	010707	042	116	101	FILNAM:	.ASCIZ\NAME OF FILE CONTAINING BAD SECTOR INFORMATION'\
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     010770      BGNMSG ERR001
15     010770      PNTB X1,#X1A
16     010770      012746 004310      MOV #X1A,-(SP)
17     010774      004137 016240      JSR R1,LPNTB
18     011000      004407      .WORD X1
19     011002      000002      .WORD PNT.CT
20     011004      ENDMSG
21
22     011006      BGNMSG ERR002
23     011006      PNTB X2,#X2A
24     011006      012746 004310      MOV #X2A,-(SP)
25     011012      004137 016240      JSR R1,LPNTB
26     011016      004503      .WORD X2
27     011020      000002      .WORD PNT.CT
28     011022      ENDMSG
29
30     011024      BGNMSG ERR003
31     011024      PNTB X3,#X3A
32     011024      012746 004310      MOV #X3A,-(SP)
33     011030      004137 016240      JSR R1,LPNTB
34     011034      004552      .WORD X3
35     011036      000002      .WORD PNT.CT
36     011040      ENDMSG
37
38     011042      BGNMSG ERR004
39     011042      PNTB X4
40     011042      004137 016240      JSR R1,LPNTB
41     011046      004635      .WORD X4
42     011050      000000      .WORD PNT.CT
43     011052      ENDMSG
44
45     011054      BGNMSG ERR005
46     011054      PNTB X5
47     011054      004137 016240      JSR R1,LPNTB
48     011060      005022      .WORD X5
49     011062      000000      .WORD PNT.CT
50     011064      ENDMSG
51
52     011066      BGNMSG ERR007
53     011066      PNTB X7
54     011066      004137 016240      JSR R1,LPNTB
55     011072      005072      .WORD X7
56     011074      000000      .WORD PNT.CT
57     011076      ENDMSG

```

37						
38	011100			BGNMSG	ERR008	
39	011100				PNTB X8,#X8A	
	011100	012746	004310			MOV #X8A,-(SP)
	011104	004137	016240			JSR R1,LPNTB
	011110	005156				.WORD X8
	011112	000002				.WORD PNT.CT
40	011114			ENDMSG		
41						
42	011116			BGNMSG	ERR009	
43	011116				PNTB X9	
	011116	004137	016240			JSR R1,LPNTB
	011122	005223				.WORD X9
	011124	000000				.WORD PNT.CT
44	011126			ENDMSG		
45						
46	011130			BGNMSG	ERR010	
47	011130				PNTB X10	
	011130	004137	016240			JSR R1,LPNTB
	011134	005415				.WORD X10
	011136	000000				.WORD PNT.CT
48	011140			ENDMSG		
49						
50	011142			BGNMSG	ERR011	
51	011142				PNTB X11	
	011142	004137	016240			JSR R1,LPNTB
	011146	005514				.WORD X11
	011150	000000				.WORD PNT.CT
52	011152			ENDMSG		
53						
54	011154			BGNMSG	ERR020	
55	011154				PNTB X20	
	011154	004137	016240			JSR R1,LPNTB
	011160	005610				.WORD X20
	011162	000000				.WORD PNT.CT
56	011164			ENDMSG		
57						
58	011166			BGNMSG	ERR021	
59	011166	010201			MOV R2,R1	
60	011170	000301			SWAB R1	
61	011172				AND 2,R1	
	011172	042701	177775			BIC #^C<2>,R1
62	011176	006201			ASR R1	
63	011200	005201			INC R1	
64	011202				PNTB X21,R2,R1	
	011202	010146				MOV R1,-(SP)
	011204	010246				MOV R2,-(SP)
	011206	004137	016240			JSR R1,LPNTB
	011212	006011				.WORD X21
	011214	000004				.WORD PNT.CT
65	011216			ENDMSG		
66						
67	011220			BGNMSG	ERR022	
68	011220	042737	100000 020152		BIC #SA.ERR,UDARSD	
69	011226				PNTB X22,UDARSD,R2	
	011226	010246				MOV R2,-(SP)
	011230	013746	020152			MOV UDARSD,-(SP)

011234	004137	016240		JSR R1,LPNTB
011240	006142			.WORD X22
011242	000004			.WORD PNT.CT
70 011244			PRINTX #XFRU	
71 011264			ENDMSG	
72				
73 011266			BGNMSG ERR023	
74 011266			PNTB X23A,R1,FFREE	
011266	013746	002146		MOV FFREE,-(SP)
011272	010146			MOV R1,-(SP)
011274	004137	016240		JSR R1,LPNTB
011300	006300			.WORD X23A
011302	000004			.WORD PNT.CT
75 011304	005742		TST -(R2)	
76 011306	005712		ERR23A: TST (R2)	
77 011310	001410		BEQ ERR23B	
78 011312			PNTB X23B,R2,(R2)	
011312	011246			MOV (R2),-(SP)
011314	010246			MOV R2,-(SP)
011316	004137	016240		JSR R1,LPNTB
011322	006606			.WORD X23B
011324	000004			.WORD PNT.CT
79 011326	005304		DEC R4	
80 011330	001403		BEQ ERR23C	
81 011332	005722		ERR23B: TST (R2)+	
82 011334	005303		DEC R3	
83 011336	001363		BNE ERR23A	
84 011340			ERR23C: PNTB XFRU	
011340	004137	016240		JSR R1,LPNTB
011344	010533			.WORD XFRU
011346	000000			.WORD PNT.CT
85 011350			ENDMSG	
86				
87 011352			BGNMSG ERR024	
88 011352			PNTB X24,R2	
011352	010246			MOV R2,-(SP)
011354	004137	016240		JSR R1,LPNTB
011360	006622			.WORD X24
011362	000002			.WORD PNT.CT
89 011364			ENDMSG	
90				
91 011366			BGNMSG ERR025	
92 011366			PNTB X25,R1,R2	
011366	010246			MOV R2,-(SP)
011370	010146			MOV R1,-(SP)
011372	004137	016240		JSR R1,LPNTB
011376	007012			.WORD X25
011400	000004			.WORD PNT.CT
93 011402			ENDMSG	
94				
95 011404			BGNMSG ERR030	
96 011404			PNTB X30,R1	
011404	010146			MOV R1,-(SP)
011406	004137	016240		JSR R1,LPNTB
011412	007163			.WORD X30
011414	000002			.WORD PNT.CT
97 011416			ENDMSG	

98					
99	011420			BGNMSG ERR031	
100	011420			PNTB X31	
	011420	004137	016240		JSR R1,LPNTB
	011424	007276			.WORD X31
	011426	000000			.WORD PNT.CT
101	011430			ENDMSG	
102					
103	011432			BGNMSG ERR032	
104	011432			PNTB X32	
	011432	004137	016240		JSR R1,LPNTB
	011436	007327			.WORD X32
	011440	000000			.WORD PNT.CT
105	011442	004737	011632	CALL MSGPKT	
106	011446			ENDMSG	
107					
108	011450			BGNMSG ERR033	
109	011450	004737	011540	CALL PNTPKT	
110	011454			ENDMSG	
111					
112	011456			BGNMSG ERR034	
113	011456	004737	011540	CALL PNTPKT	
114	011462			ENDMSG	
115					
116	011464			BGNMSG ERR036	
117	011464			PNTB X36	
	011464	004137	016240		JSR R1,LPNTB
	011470	007440			.WORD X36
	011472	000000			.WORD PNT.CT
118	011474			ENDMSG	
119					
120	011476			BGNMSG ERR037	
121	011476			PNTB X37,R1	
	011476	010146			MOV R1,-(SP)
	011500	004137	016240		JSR R1,LPNTB
	011504	007556			.WORD X37
	011506	000002			.WORD PNT.CT
122	011510			ENDMSG	
123					
124	011512			BGNMSG ERR100	
125	011512			PNTB X100,(R4)	
	011512	011446			MOV (R4),-(SP)
	011514	004137	016240		JSR R1,LPNTB
	011520	007673			.WORD X100
	011522	000002			.WORD PNT.CT
126	011524			ENDMSG	
127					
128	011526			BGNMSG ERR101	
129	011526			PNTB X101	
	011526	004137	016240		JSR R1,LPNTB
	011532	007756			.WORD X101
	011534	000000			.WORD PNT.CT
130	011536			ENDMSG	
131					
132	011540			PNTPKT: PNTB XPKT1	
	011540	004137	016240		JSR R1,LPNTB
	011544	010162			.WORD XPKT1

	011546	000000			.WORD PNT.CT
133	011550	010401		MOV R4,R1	
134	011552	062701	000104	ADD #HC.CPK,R1	
135	011556	010402		MOV R4,R2	
136	011560	062702	000020	ADD #HC.MPK,R2	
137	011564	012703	000014	MOV #12.,R3	
138	011570			PNTPKL: PNTB XPKT2,2(R1),(R1),2(R2),(R2)	
	011570	011246			MOV (R2),-(SP)
	011572	016246	000002		MOV 2(R2),-(SP)
	011576	011146			MOV (R1),-(SP)
	011600	016146	000002		MOV 2(R1),-(SP)
	011604	004137	016240		JSR R1,LPNTB
	011610	010453			.WORD XPKT2
	011612	000010			.WORD PNT.CT
139	011614	062701	000004	ADD #4,R1	
140	011620	062702	000004	ADD #4,R2	
141	011624	005303		DEC R3	
142	011626	001360		BNE PNTPKL	
143	011630	000207		RETURN	
144					
145	011632			MSGPKT: PNTB XMSG1	
	011632	004137	016240		JSR R1,LPNTB
	011636	010061			.WORD XMSG1
	011640	000000			.WORD PNT.CT
146	011642	016504	000016	MOV C.RING(R5),R4	
147	011646	062704	000306	ADD #HC.BF2,R4	
148	011652	012703	000005	MOV #5,R3	
149	011656			MSGPKL: PNTB XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)	
	011656	016446	000014		MOV 12.(R4),-(SP)
	011662	016446	000012		MOV 10.(R4),-(SP)
	011666	016446	000010		MOV 8.(R4),-(SP)
	011672	016446	000006		MOV 6(R4),-(SP)
	011676	016446	000004		MOV 4(R4),-(SP)
	011702	016446	000002		MOV 2(R4),-(SP)
	011706	011446			MOV (R4),-(SP)
	011710	004137	016240		JSR R1,LPNTB
	011714	010115			.WORD XMSG2
	011716	000016			.WORD PNT.CT
150	011720	062704	000016	ADD #14.,R4	
151	011724	005303		DEC R3	
152	011726	001353		BNE MSGPKL	
153	011730	000207		RETURN	

1	000001	SVCINS= 1
2	000001	SVCTST= 1
3	000001	SVCSUB= 1
4	000001	SVCGBL= 1
5	000001	SVCTAG= 1

; LIST INSTRUCTIONS, SHIFTED RIGHT
; LIST TEST TAGS, SHIFTED RIGHT
; LIST SUBTEST TAGS, SHIFTED RIGHT
; LIST GLOBAL TAGS, SHIFTED RIGHT
; LIST OTHER TAGS, SHIFTED RIGHT

1
2
3
4
5
6
7
8

.SBTTL GLOBAL SUBROUTINES SECTION
;MEMORY ALLOCATION ERROR
;THIS ROUTINE PRINTS A SYSTEM FATAL ERROR AND EXITS THE TEST
FMERR: ERRSF 4,,ERR004

011732
011732 104454
011734 000004
011736 000000
011740 011042
011742
011742 104444

DOCLN ;ABORT

TRAP CSERSF
.WORD 4
.WORD 0
.WORD ERR004
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
17 ALOCM: PUSH FFREE                                ;SAVE FFREE AT ENTRY
18                                     MOV FFREE,-(SP)
19                                     SUB R1,FSIZE        ;REDUCE SIZE OF FREE MEMORY
20                                     BLT FMERR            ;REPORT ERROR IF NOT ENOUGH MEMORY
21                                     ADD R1,R1           ;CHANGE WORDS TO BYTES
22                                     ADD R1,FFREE        ;CALCULATE NEW START OF FREE MEMORY
23                                     POP R1              ;GET START OF ALLOCATED MEMORY
24                                     MOV (SP)+,R1
25
26                                     RETURN

```

```
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      ;OUTPUTS:
10     ;      CONTROLLER TABLE POINTING TO HOST COMM AREA
11     ;      R4 - ADDRESS OF HOST COMM AREA
12     ;
13 011770 012701 000214      HCOMM:  MOV #HC.SIZ/2,R1          ;GET SIZE OF AREA TO ALLOCATE
14 011774 004737 011744      CALL ALOCM          ;ALLOCATE THE MEMORY
15 012000 010165 000016      MOV R1,C.RING(R5)      ;GET ADDRESS OF HOST COMM AREA
16                                     ;PLACE IN CONTROLLER TABLE
17 012004 000207      RETURN
```

```

2      :TINIT
3      :
4      :INITIALIZE VARIABLES FOR TEST
5      :
6      :INPUTS:
7      :      R1 - TEST NUMBER
8      :
9      :OUTPUTS:
10     :      LBUFS - CLEARED (DELETES ERROR LOG)
11     :      FFREE - FROM FMEM
12     :      FSIZE - FROM FMEMS
13     :      TNUM - TEST NUMBER FROM R1
14     :      ALL REGISTERS CLOBERED
15 012006 004737 012064      TINIT: CALL RESET          ;RESET ALL DEVICES
16 012012 013737 002152 002146  MOV FMEM,FFREE        ;INIT FFREE
17 012020 013737 002154 002150  MOV FMEMS,FSIZE       ;INIT FSIZE
18 012026 020137 002214      CMP R1,FNUM             ;SEE IF SAME TEST RUNNING
19 012032 001007      BNE TINITR                     ;IF NOT, GO TO READ DM PROGRAM
20 012034 013737 002170 002146  MOV DMEND,FFREE       ;CHANGE FREE MEMORY TO LEAVE
21 012042 013737 002172 002150  MOV DMENDS,FSIZE      ; DM PROGRAM ALLOCATED
22 012050 000207      RETURN
23
24 012052 004737 012210      TINITR: CALL READDM        ;READ DM PROGRAM
25 012056 010137 002214      MOV R1,FNUM             ;STORE TEST NUMBER TO SHOW DM PROGRAM IN MEMORY
26 012062 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
11      CLR     NXMAD
12      SETVEC  #4,#NXM1,#PRI07
13
14      MOV     R3,-(SP)
15      MOV     R4,-(SP)
16
17      MOV     #PRI07,-(SP)
18      MOV     #NXM1,-(SP)
19      MOV     #4,-(SP)
20      MOV     #3,-(SP)
21      TRAP    C$SVEC
22      ADD     #10,SP
23
24      TRAP     C$BRK
25
26      1$:  MOV     #8,R3      ; R3 = COUNTER OF ENTRIES
27      MOV     #IPADRS,R4    ; R4 -> IP ADDRESS
28      TST     (R4)          ; IS THERE AN ENTRY?
29      BEQ     2$            ; IF NOT, DONE
30      CLR     @R4+          ; INIT UDA
31      TST     NXMAD         ; WAS THERE AN ERROR?
32      BNE     3$            ; IF SO, EXIT
33      DEC     R3            ; MAKE SURE WE DO NOT EXTEND OVER AREA
34      BNE     1$            ; IF NOT DONE, BRANCH
35
36      2$:  CLRVEC  #4
37
38      MOV     #4,R0
39      TRAP    C$CVEC
40
41      POP     <R4,R3>
42
43      MOV (SP)+,R4
44      MOV (SP)+,R3
45
46      RETURN
47
48      3$:  TST     -(R4)      ; R4 -> UDAIP THAT FAILED
49      MOV     R4,R5          ; SAVE IN R5 FOR REPORT
50      ERDF    20,,ERR020
51
52      TRAP    C$ERDF
53      .WORD   20
54      .WORD   0
55      .WORD   ERR020
56
57      CLR     (R4)          ; DESTROY ENTRY SO NOT TO FALL INTO RESET ERROR LOOP
58      DOCLN
59
60      TRAP    C$DCLN
61
62      012064 010346
63      012064 010446
64      012066 005037 002222
65      012070 005037 002222
66      012074 012746 000340
67      012100 012746 017152
68      012104 012746 000004
69      012110 012746 000003
70      012114 104437
71      012116 062706 000010
72      012122 104422
73      012124 012703 000010
74      012130 012704 003474
75      012134 005714
76      012136 001406
77      012140 005034
78      012142 005737 002222
79      012146 001010
80      012150 005303
81      012152 001370
82      012154 012700 000004
83      012160 104436
84      012162 012604
85      012164 012603
86      012166 000207
87      012170 005744
88      012172 010405
89      012174 104455
90      012176 000024
91      012200 000000
92      012202 011154
93      012204 005014
94      012206 104444

```

```

1      ;READDM
2
3      ;READ A DM PROGRAM INTO FREE MEMORY
4
5      ;INPUTS:
6      ;   R1 - TEST NUMBER
7
8      ;OUTPUTS:
9      ;   DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
10     ;   R1 - UNCHANGED
11     ;   CARRY CLEAR IF NO ERROR, CARRY SET IF PROGRAM NOT FOUND
12     ;ALL REGISTERS BUT R1 ARE USED AND PREVIOUS CONTENTS DESTROYED
13 012210 013737 002146 002164 READDM: MOV FFREE,DMPROG      ;GET STORAGE ADDRESS
14 012216 004737 012516      CALL RDREC
15 012222 103407      BCS README      ;CHECK IF ERROR
16 012224 013737 002146 002170      MOV FFREE,DMEND      ;SAVE END OF ADDRESS OF DM PROGRAM
17 012232 013737 002150 002172      MOV FSIZE,DMENDS      ; AND CURRENT SIZE OF FREE MEMORY
18 012240 000207      RETURN
19
20      README: ERRSF 7,,ERR007      ;REPORT DM PROGRAM NOT FOUND
21      DOCLN

```

012242	104454			TRAP	C\$ERSF
012244	000007			.WORD	7
012246	000000			.WORD	0
012250	011066			.WORD	ERR007
012252	104444			TRAP	C\$DCLN

```

1      ;GTDUST
2
3      ;GET DUST STATUS
4
5 012254 016504 000004      GTDUST: MOV C.VEC(R5),R4
6 012260      AND CT.VEC,R4
7 012264 042704 177000
8 012266 010501      MOV R5,R1
9 012272 062701 000010      ADD #C.JSR,R1
      SETVEC R4,R1,#PRI07
      012272 012746 000340
      012276 010146
      012300 010446
      012302 012746 000003
      012306 104437
      012310 062706 000010
10
11 012314 006204      ASR R4
12 012316 006204      ASR R4
13 012320 004737 011770      CALL HCOMM
14 012324 004737 017254      CALL UDANT
15 012330 001002      BNE 1$
16 012332 000137 016570      JMP LOADER
17 012336 012700 000001      1$: MOV #OP.GSS,R0
18 012342 004737 016574      CALL BLDCMD
19 012346 004737 016660      CALL SNDCMD
20 012352 004737 017000      CALL WAITMS
21 012356 000207      RETURN

```

;GET VECTOR OF UDA

BIC #^C<CT.VEC>,R4
;GET INTERRUPT SERVICE LINK

;SET UP INTERRUPT VECTOR

```

MOV #PRI07,-(SP)
MOV R1,-(SP)
MOV R4,-(SP)
MOV #3,-(SP)
TRAP C$SVEC
ADD #10,SP

```

;INITIALIZE UDA WITH SMALLEST
;POSITION VECTOR FOR UDA

;ALLOCATE SPACE FOR HOST COMM AREA
; RING BUFFER AND INTERRUPTS ENABLED

;BRANCH IF AN ERROR
; R0 HAS OPCODE

```

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 012360 010137 002174 RUNDM: MOV R1,URUN          ;SAVE NUMBER OF UNITS TO RUN
16 012364 005037 002176      CLR URNING          ;CLEAR NUMBER OF UNITS RUNNING
17
18     ;LOAD DM PROGRAM INTO EACH CONTROLLER
19
20 012370 013737 002174 002200      MOV URUN,UCNT          ;SET COUNTER OF UNITS
21 012376 013705 002162      MOV TSTTAB,R5          ;GET FIRST CONTROLLER TABLE
22 012402      LDDM:
23 012402 042765 176377 000014      BIC #*C<CT.U52+CT.U50>,C.FLG(R5) ;CLEAR ALL FLAGS
24 012410 116537 000002 002074      MOVB C.UNIT(R5),L$LUN      ;SEE IF UNIT TO BE TESTED
25 012416 005765 000002      TST C.UNIT(R5)
26 012422 100411      BMI LDNEXT          ;IF NOT, DON'T LOAD THIS UNIT
27 012424      ASSUME CT.AVL EQ BIT15
29 012424 033765 002212 000014      BIT TSTTYP,C.FLG(R5)      ;DO WE TEST THIS TYPE?
30 012432 001405      BEQ LDNEXT          ;IF NOT, BRANCH
34 012434 004737 016372      CALL LOADDM          ;LOAD THE DM PROGRAM
35 012440 001402      BEQ LDNEXT          ;IF ERROR, GO TO NEXT CONTROLLER
36 012442 005237 002176      INC URNING          ;IF NO ERROR, COUNT UNIT RUNNING
37 012446 062705 000054      LDNEXT: ADD #C.SIZE,R5      ;MOVE TO NEXT CONTROLLER TABLE
38 012452 005337 002200      DEC UCNT          ;CHECK IF MORE CONTROLLERS
39 012456 001351      BNE LDDM          ;LOAD NEXT
40 012460 005037 002220      CLR UFREEZ          ;CLEAR UNIT FREEZE FLAG
41 012464 012737 177777 003226      MOV #-1,FCTNUM ;INVALIDATE FCT BLOCK NUMBER (BLOCK IN MEMORY)
42
43     ;CHECK IF ANY CONTROLLERS LOADED
44
45 012472 005737 002176      TST URNING          ;ANY UNITS LOADED?
46
47     ;THE DM PROGRAMS ARE NOW IN CONTROL
48     ;RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
49
50 012476 000207      RETURN

```

```

2      ;CLOSEF
3
4      ;CLOSE DATA FILE FOR DM PROGRAMS
5
6      ;INPUTS:
7          FILOPN - ZERO IF FILE NOT OPEN
8      ;OUTPUTS:
9          NONE
10
11 012500 005737 002202  CLOSEF: TST FILOPN          ;SEE IF FILE CURRENTLY OPEN
12 012504 001403          BEQ 1$
13 012506          CLOSE          ; IF SO, CLOSE IT
14 012510 005037 002202          ;AND MARK AS SO          TRAP      C$CLOS
15 012514 000207          1$:      RETURN

```

```

1      ;RDREC
2      ;READ A RECORD FROM THE INPUT FILE. PLACE DATA INTO FREE MEMORY.
3
4      ;INPUTS:
5          R1 - FILE TYPE
6              1 - UDA52 FORMATTER
7              2 - UDA50 FORMATTER
8
9      ;OUTPUTS:
10         DATA FROM RECORD IN MEMORY
11         CARRY CLEAR IF NO ERROR, CARRY SET IF ERROR
12
13 RDREC:  PUSH <R0,R1,R2,R3,R4,R5>
14         MOV R0,-(SP)
15         MOV R1,-(SP)
16         MOV R2,-(SP)
17         MOV R3,-(SP)
18         MOV R4,-(SP)
19         MOV R5,-(SP)
20
21         TST FILOPN          ;SEE IF FILE ALREADY OPEN
22         BNE RDSTS          ;IF NOT, OPEN FILE NOW
23         OPEN #FNAME
24
25         INC FILOPN          ;AND MARK AS OPEN
26         CLR R5              ;CLEAR LOAD ADDRESS (SEARCH MODE)
27         BREAK              ;ALLOW PROGRAM TO BE INTERRUPTED
28
29         GETBYTE R4          ;GETBYTE CALLS DON'T SEEM TO BREAK ON CONTROL-C!
30
31         TST R4              ;IF ZERO
32         BEQ RDST            ;KEEP READING
33         CMP #1,R4           ;WHEN NOT ZERO
34         BNE RWRDE1          ;IT BETTER BE A ONE
35         GETBYTE R0          ;AND THE NEXT BYTE
36
37         ADD R0,R4
38         TST R0
39         BEQ RDDAT
40
41         TRAP CS$GETB
42         MOV R0,R4
43
44         TRAP CS$GETB
45
46         TRAP CS$GETB
47
48         TRAP CS$GETB
49
50         TRAP CS$GETB
51
52         TRAP CS$GETB
53
54         TRAP CS$GETB
55
56         TRAP CS$GETB
57
58         TRAP CS$GETB
59
60         TRAP CS$GETB
61
62         TRAP CS$GETB
63
64         TRAP CS$GETB
65
66         TRAP CS$GETB
67
68         TRAP CS$GETB
69
70         TRAP CS$GETB
71
72         TRAP CS$GETB
73
74         TRAP CS$GETB
75
76         TRAP CS$GETB
77
78         TRAP CS$GETB
79
80         TRAP CS$GETB
81
82         TRAP CS$GETB
83
84         TRAP CS$GETB
85
86         TRAP CS$GETB
87
88         TRAP CS$GETB
89
90         TRAP CS$GETB
91
92         TRAP CS$GETB
93
94         TRAP CS$GETB
95
96         TRAP CS$GETB
97
98         TRAP CS$GETB
99
100        TRAP CS$GETB
101
102        TRAP CS$GETB
103
104        TRAP CS$GETB
105
106        TRAP CS$GETB
107
108        TRAP CS$GETB
109
110        TRAP CS$GETB
111
112        TRAP CS$GETB
113
114        TRAP CS$GETB
115
116        TRAP CS$GETB
117
118        TRAP CS$GETB
119
120        TRAP CS$GETB
121
122        TRAP CS$GETB
123
124        TRAP CS$GETB
125
126        TRAP CS$GETB
127
128        TRAP CS$GETB
129
130        TRAP CS$GETB
131
132        TRAP CS$GETB
133
134        TRAP CS$GETB
135
136        TRAP CS$GETB
137
138        TRAP CS$GETB
139
140        TRAP CS$GETB
141
142        TRAP CS$GETB
143
144        TRAP CS$GETB
145
146        TRAP CS$GETB
147
148        TRAP CS$GETB
149
150        TRAP CS$GETB
151
152        TRAP CS$GETB
153
154        TRAP CS$GETB
155
156        TRAP CS$GETB
157
158        TRAP CS$GETB
159
160        TRAP CS$GETB
161
162        TRAP CS$GETB
163
164        TRAP CS$GETB
165
166        TRAP CS$GETB
167
168        TRAP CS$GETB
169
170        TRAP CS$GETB
171
172        TRAP CS$GETB
173
174        TRAP CS$GETB
175
176        TRAP CS$GETB
177
178        TRAP CS$GETB
179
180        TRAP CS$GETB
181
182        TRAP CS$GETB
183
184        TRAP CS$GETB
185
186        TRAP CS$GETB
187
188        TRAP CS$GETB
189
190        TRAP CS$GETB
191
192        TRAP CS$GETB
193
194        TRAP CS$GETB
195
196        TRAP CS$GETB
197
198        TRAP CS$GETB
199
200        TRAP CS$GETB
201
202        TRAP CS$GETB
203
204        TRAP CS$GETB
205
206        TRAP CS$GETB
207
208        TRAP CS$GETB
209
210        TRAP CS$GETB
211
212        TRAP CS$GETB
213
214        TRAP CS$GETB
215
216        TRAP CS$GETB
217
218        TRAP CS$GETB
219
220        TRAP CS$GETB
221
222        TRAP CS$GETB
223
224        TRAP CS$GETB
225
226        TRAP CS$GETB
227
228        TRAP CS$GETB
229
230        TRAP CS$GETB
231
232        TRAP CS$GETB
233
234        TRAP CS$GETB
235
236        TRAP CS$GETB
237
238        TRAP CS$GETB
239
240        TRAP CS$GETB
241
242        TRAP CS$GETB
243
244        TRAP CS$GETB
245
246        TRAP CS$GETB
247
248        TRAP CS$GETB
249
250        TRAP CS$GETB
251
252        TRAP CS$GETB
253
254        TRAP CS$GETB
255
256        TRAP CS$GETB
257
258        TRAP CS$GETB
259
260        TRAP CS$GETB
261
262        TRAP CS$GETB
263
264        TRAP CS$GETB
265
266        TRAP CS$GETB
267
268        TRAP CS$GETB
269
270        TRAP CS$GETB
271
272        TRAP CS$GETB
273
274        TRAP CS$GETB
275
276        TRAP CS$GETB
277
278        TRAP CS$GETB
279
280        TRAP CS$GETB
281
282        TRAP CS$GETB
283
284        TRAP CS$GETB
285
286        TRAP CS$GETB
287
288        TRAP CS$GETB
289
290        TRAP CS$GETB
291
292        TRAP CS$GETB
293
294        TRAP CS$GETB
295
296        TRAP CS$GETB
297
298        TRAP CS$GETB
299
300        TRAP CS$GETB
301
302        TRAP CS$GETB
303
304        TRAP CS$GETB
305
306        TRAP CS$GETB
307
308        TRAP CS$GETB
309
310        TRAP CS$GETB
311
312        TRAP CS$GETB
313
314        TRAP CS$GETB
315
316        TRAP CS$GETB
317
318        TRAP CS$GETB
319
320        TRAP CS$GETB
321
322        TRAP CS$GETB
323
324        TRAP CS$GETB
325
326        TRAP CS$GETB
327
328        TRAP CS$GETB
329
330        TRAP CS$GETB
331
332        TRAP CS$GETB
333
334        TRAP CS$GETB
335
336        TRAP CS$GETB
337
338        TRAP CS$GETB
339
340        TRAP CS$GETB
341
342        TRAP CS$GETB
343
344        TRAP CS$GETB
345
346        TRAP CS$GETB
347
348        TRAP CS$GETB
349
350        TRAP CS$GETB
351
352        TRAP CS$GETB
353
354        TRAP CS$GETB
355
356        TRAP CS$GETB
357
358        TRAP CS$GETB
359
360        TRAP CS$GETB
361
362        TRAP CS$GETB
363
364        TRAP CS$GETB
365
366        TRAP CS$GETB
367
368        TRAP CS$GETB
369
370        TRAP CS$GETB
371
372        TRAP CS$GETB
373
374        TRAP CS$GETB
375
376        TRAP CS$GETB
377
378        TRAP CS$GETB
379
380        TRAP CS$GETB
381
382        TRAP CS$GETB
383
384        TRAP CS$GETB
385
386        TRAP CS$GETB
387
388        TRAP CS$GETB
389
390        TRAP CS$GETB
391
392        TRAP CS$GETB
393
394        TRAP CS$GETB
395
396        TRAP CS$GETB
397
398        TRAP CS$GETB
399
400        TRAP CS$GETB
401
402        TRAP CS$GETB
403
404        TRAP CS$GETB
405
406        TRAP CS$GETB
407
408        TRAP CS$GETB
409
410        TRAP CS$GETB
411
412        TRAP CS$GETB
413
414        TRAP CS$GETB
415
416        TRAP CS$GETB
417
418        TRAP CS$GETB
419
420        TRAP CS$GETB
421
422        TRAP CS$GETB
423
424        TRAP CS$GETB
425
426        TRAP CS$GETB
427
428        TRAP CS$GETB
429
430        TRAP CS$GETB
431
432        TRAP CS$GETB
433
434        TRAP CS$GETB
435
436        TRAP CS$GETB
437
438        TRAP CS$GETB
439
440        TRAP CS$GETB
441
442        TRAP CS$GETB
443
444        TRAP CS$GETB
445
446        TRAP CS$GETB
447
448        TRAP CS$GETB
449
450        TRAP CS$GETB
451
452        TRAP CS$GETB
453
454        TRAP CS$GETB
455
456        TRAP CS$GETB
457
458        TRAP CS$GETB
459
460        TRAP CS$GETB
461
462        TRAP CS$GETB
463
464        TRAP CS$GETB
465
466        TRAP CS$GETB
467
468        TRAP CS$GETB
469
470        TRAP CS$GETB
471
472        TRAP CS$GETB
473
474        TRAP CS$GETB
475
476        TRAP CS$GETB
477
478        TRAP CS$GETB
479
480        TRAP CS$GETB
481
482        TRAP CS$GETB
483
484        TRAP CS$GETB
485
486        TRAP CS$GETB
487
488        TRAP CS$GETB
489
490        TRAP CS$GETB
491
492        TRAP CS$GETB
493
494        TRAP CS$GETB
495
496        TRAP CS$GETB
497
498        TRAP CS$GETB
499
500        TRAP CS$GETB
501
502        TRAP CS$GETB
503
504        TRAP CS$GETB
505
506        TRAP CS$GETB
507
508        TRAP CS$GETB
509
510        TRAP CS$GETB
511
512        TRAP CS$GETB
513
514        TRAP CS$GETB
515
516        TRAP CS$GETB
517
518        TRAP CS$GETB
519
520        TRAP CS$GETB
521
522        TRAP CS$GETB
523
524        TRAP CS$GETB
525
526        TRAP CS$GETB
527
528        TRAP CS$GETB
529
530        TRAP CS$GETB
531
532        TRAP CS$GETB
533
534        TRAP CS$GETB
535
536        TRAP CS$GETB
537
538        TRAP CS$GETB
539
540        TRAP CS$GETB
541
542        TRAP CS$GETB
543
544        TRAP CS$GETB
545
546        TRAP CS$GETB
547
548        TRAP CS$GETB
549
550        TRAP CS$GETB
551
552        TRAP CS$GETB
553
554        TRAP CS$GETB
555
556        TRAP CS$GETB
557
558        TRAP CS$GETB
559
560        TRAP CS$GETB
561
562        TRAP CS$GETB
563
564        TRAP CS$GETB
565
566        TRAP CS$GETB
567
568        TRAP CS$GETB
569
570        TRAP CS$GETB
571
572        TRAP CS$GETB
573
574        TRAP CS$GETB
575
576        TRAP CS$GETB
577
578        TRAP CS$GETB
579
580        TRAP CS$GETB
581
582        TRAP CS$GETB
583
584        TRAP CS$GETB
585
586        TRAP CS$GETB
587
588        TRAP CS$GETB
589
590        TRAP CS$GETB
591
592        TRAP CS$GETB
593
594        TRAP CS$GETB
595
596        TRAP CS$GETB
597
598        TRAP CS$GETB
599
600        TRAP CS$GETB
601
602        TRAP CS$GETB
603
604        TRAP CS$GETB
605
606        TRAP CS$GETB
607
608        TRAP CS$GETB
609
610        TRAP CS$GETB
611
612        TRAP CS$GETB
613
614        TRAP CS$GETB
615
616        TRAP CS$GETB
617
618        TRAP CS$GETB
619
620        TRAP CS$GETB
621
622        TRAP CS$GETB
623
624        TRAP CS$GETB
625
626        TRAP CS$GETB
627
628        TRAP CS$GETB
629
630        TRAP CS$GETB
631
632        TRAP CS$GETB
633
634        TRAP CS$GETB
635
636        TRAP CS$GETB
637
638        TRAP CS$GETB
639
640        TRAP CS$GETB
641
642        TRAP CS$GETB
643
644        TRAP CS$GETB
645
646        TRAP CS$GETB
647
648        TRAP CS$GETB
649
650        TRAP CS$GETB
651
652        TRAP CS$GETB
653
654        TRAP CS$GETB
655
656        TRAP CS$GETB
657
658        TRAP CS$GETB
659
660        TRAP CS$GETB
661
662        TRAP CS$GETB
663
664        TRAP CS$GETB
665
666        TRAP CS$GETB
667
668        TRAP CS$GETB
669
670        TRAP CS$GETB
671
672        TRAP CS$GETB
673
674        TRAP CS$GETB
675
676        TRAP CS$GETB
677
678        TRAP CS$GETB
679
680        TRAP CS$GETB
681
682        TRAP CS$GETB
683
684        TRAP CS$GETB
685
686        TRAP CS$GETB
687
688        TRAP CS$GETB
689
690        TRAP CS$GETB
691
692        TRAP CS$GETB
693
694        TRAP CS$GETB
695
696        TRAP CS$GETB
697
698        TRAP CS$GETB
699
700        TRAP CS$GETB
701
702        TRAP CS$GETB
703
704        TRAP CS$GETB
705
706        TRAP CS$GETB
707
708        TRAP CS$GETB
709
710        TRAP CS$GETB
711
712        TRAP CS$GETB
713
714        TRAP CS$GETB
715
716        TRAP CS$GETB
717
718        TRAP CS$GETB
719
720        TRAP CS$GETB
721
722        TRAP CS$GETB
723
724        TRAP CS$GETB
725
726        TRAP CS$GETB
727
728        TRAP CS$GETB
729
730        TRAP CS$GETB
731
732        TRAP CS$GETB
733
734        TRAP CS$GETB
735
736        TRAP CS$GETB
737
738        TRAP CS$GETB
739
740        TRAP CS$GETB
741
742        TRAP CS$GETB
743
744        TRAP CS$GETB
745
746        TRAP CS$GETB
747
748        TRAP CS$GETB
749
750        TRAP CS$GETB
751
752        TRAP CS$GETB
753
754        TRAP CS$GETB
755
756        TRAP CS$GETB
757
758        TRAP CS$GETB
759
760        TRAP CS$GETB
761
762        TRAP CS$GETB
763
764        TRAP CS$GETB
765
766        TRAP CS$GETB
767
768        TRAP CS$GETB
769
770        TRAP CS$GETB
771
772        TRAP CS$GETB
773
774        TRAP CS$GETB
775
776        TRAP CS$GETB
777
778        TRAP CS$GETB
779
780        TRAP CS$GETB
781
782        TRAP CS$GETB
783
784        TRAP CS$GETB
785
786        TRAP CS$GETB
787
788        TRAP CS$GETB
789
790        TRAP CS$GETB
791
792        TRAP CS$GETB
793
794        TRAP CS$GETB
795
796        TRAP CS$GETB
797
798        TRAP CS$GETB
799
800        TRAP CS$GETB
801
802        TRAP CS$GETB
803
804        TRAP CS$GETB
805
806        TRAP CS$GETB
807
808        TRAP CS$GETB
809
810        TRAP CS$GETB
811
812        TRAP CS$GETB
813
814        TRAP CS$GETB
815
816        TRAP CS$GETB
817
818        TRAP CS$GETB
819
820        TRAP CS$GETB
821
822        TRAP CS$GETB
823
824        TRAP CS$GETB
825
826        TRAP CS$GETB
827
828        TRAP CS$GETB
829
830        TRAP CS$GETB
831
832        TRAP CS$GETB
833
834        TRAP CS$GETB
835
836        TRAP CS$GETB
837
838        TRAP CS$GETB
839
840        TRAP CS$GETB
841
842        TRAP CS$GETB
843
844        TRAP CS$GETB
845
846        TRAP CS$GETB
847
848        TRAP CS$GETB
849
850        TRAP CS$GETB
851
852        TRAP CS$GETB
853
854        TRAP CS$GETB
855
856        TRAP CS$GETB
857
858        TRAP CS$GETB
859
860        TRAP CS$GETB
861
862        TRAP CS$GETB
863
864        TRAP CS$GETB
865
866        TRAP CS$GETB
867
868        TRAP CS$GETB
869
870        TRAP CS$GETB
871
872        TRAP CS$GETB
873
874        TRAP CS$GETB
875
876        TRAP CS$GETB
877
878        TRAP CS$GETB
879
880        TRAP CS$GETB
881
882        TRAP CS$GETB
883
884        TRAP CS$GETB
885
886        TRAP CS$GETB
887
888        TRAP CS$GETB
889
890        TRAP CS$GETB
891
892        TRAP CS$GETB
893
894        TRAP CS$GETB
895
896        TRAP CS$GETB
897
898        TRAP CS$GETB
899
900        TRAP CS$GETB
901
902        TRAP CS$GETB
903
904        TRAP CS$GETB
905
906        TRAP CS$GETB
907
908        TRAP CS$GETB
909
910        TRAP CS$GETB
911
912        TRAP CS$GETB
913
914        TRAP CS$GETB
915
916        TRAP CS$GETB
917
918        TRAP CS$GETB
919
920        TRAP CS$GETB
921
922        TRAP CS$GETB
923
924        TRAP CS$GETB
925
926        TRAP CS$GETB
927
928        TRAP CS$GETB
929
930        TRAP CS$GETB
931
932        TRAP CS$GETB
933
934        TRAP CS$GETB
935
936        TRAP CS$GETB
937
938        TRAP CS$GETB
939
940        TRAP CS$GETB
941
942        TRAP CS$GETB
943
944        TRAP CS$GETB
945
946        TRAP CS$GETB
947
948        TRAP CS$GETB
949
950        TRAP CS$GETB
951
952        TRAP CS$GETB
953
954        TRAP CS$GETB
955
956        TRAP CS$GETB
957
958        TRAP CS$GETB
959
960        TRAP CS$GETB
961
962        TRAP CS$GETB
963
964        TRAP CS$GETB
965
966        TRAP CS$GETB
967
968        TRAP CS$GETB
969
970        TRAP CS$GETB
971
972        TRAP CS$GETB
973
974        TRAP CS$GETB
975
976        TRAP CS$GETB
977
978        TRAP CS$GETB
979
980        TRAP CS$GETB
981
982        TRAP CS$GETB
983
984        TRAP CS$GETB
985
986        TRAP CS$GETB
987
988        TRAP CS$GETB
989
990        TRAP CS$GETB
991
992        TRAP CS$GETB
993
994        TRAP CS$GETB
995
996        TRAP CS$GETB
997
998        TRAP CS$GETB
999
1000       TRAP CS$GETB

```

1 012604 020001
2 012606 103416
3 012610 101120
4 012612 004737 013030
5 012616 013702 002224
6 012622 004737 013030
7 012626
 012626 104426
8 012630 060004
9 012632 105704
10 012634 001120
11 012636 013705 002146
12 012642 000744

CMP R0,R1
BLO RDDAT
BHI RDERR
CALL FWORD
MOV FDATA,R2
CALL FWORD
GETBYTE R0

ADD R0,R4
TSTB R4
BNE RWRDE1
MOV FFREE,R5
BR RDST

;CHECK IF TYPE OF FILE LOOKING FOR
;IF TOO SOON IN FILE, KEEP SEARCHING
;IF PAST TYPE, GIVE ERROR RETURN
;GET NEXT TWO WORDS

;GET CHECKSUM
 TRAP C\$GETB
;ADD TO COMPUTED SUM
;SEE IF THIS SUM IS ZERO
; IF NOT, REPORT CHECKSUM ERROR
;GET STORAGE ADDRESS
;SWITCH FROM SEARCH TO STORE MODE

1 012644 004737 013030
2 012650 013703 002224
3 012654 004737 013030
4 012660 162703 000006
5 012664 001443
6
7 012666 005705
8 012670 001425
9 012672 013701 002224
10 012676 060301
11 012700 162701 001000
12 012704 060501
13 012706 032701 000001
14 012712 001401
15 012714 005201
16 012716 163701 002146
17 012722 101403
18 012724 006001
19 012726 004737 011744
20 012732 013702 002224
21 012736 162702 001000
22 012742 060502
23 012744
24 012744 104426
25 012746 005705
26 012750 001401
27 012752 110022
28 012754 060004
29 012756 005303
30 012760 001371
31 012762 104426
32 012764 060004
33 012766 105704
34 012770 001671
35 012772 000441

RDDAT: CALL FWORD
MOV FDATA,R3
CALL FWORD
SUB #6,R3
BEQ RWORDT

TST R5
BEQ 3\$
MOV FDATA,R1
ADD R3,R1
SUB #DMFRST,R1
ADD R5,R1
BIT #1,R1
BEQ 1\$
INC R1
1\$: SUB FFREE,R1
BLOS 2\$
ROR R1
CALL ALOCM
2\$: MOV FDATA,R2
SUB #DMFRST,R2
ADD R5,R2
3\$: GETBYTE R0

TST R5
BEQ 4\$
MOVB R0,(R2)+
4\$: ADD R0,R4
DEC R3
BNE 3\$
GETBYTE R0

ADD R0,R4
TSTB R4
BEQ RDST
BR RWRDE1

;READ BYTE COUNT
;SAVE IN R3
;READ LOAD ADDRESS
;SUBTRACT BYTES ALREADY READ FROM BYTE COUNT
;IF RESULT IS ZERO, THIS IS A
;TRANSFER BLOCK
;IF IN SEARCH MODE,
; BYPASS TRANSFER ADDRESS COMPUTATION
;GET LOAD ADDRESS
;ADD BYTES IN RECORD

;COMPUTE STORAGE ADDRESS
;CHECK IF ODD BYTE ADDRESS
;IF SO,
; INCREMENT
;SEE IF MORE MEMORY NEEDS TO BE ALLOCATED

;REDUCE TO WORDS
;ALLOCATE THE MEMORY
;GET LOAD ADDRESS

;GET DATA BYTE
TRAP CSGETB

;IF IN SEARCH MODE,
; BYPASS DATA STORAGE
;STORE IN MEMORY
;UPDATE CHECKSUM
;COUNT THE BYTE
;GET THEM ALL
;GET CHECKSUM
TRAP CSGETB

;ADD
;IF CHECKSUM CORRECT,
; THEN GO READ NEXT RECORD
; ELSE REPORT ERROR

```

1 012774          RWORDT: GETBYTE R0
  012774 104426
2 012776 060004   ADD R0,R4
3 013000 105704   TSTB R4
4 013002 001035   BNE RWRDE1
5 013004 005705   TST R5
6 013006 001662   BEQ RDST
7 013010          POP <R5,R4,R3,R2,R1,R0>
  013010 012605
  013012 012604
  013014 012603
  013016 012602
  013020 012601
  013022 012600
8 013024 000241   CLC
9 013026 000207   RETURN
10
11 013030          FWORD: GETBYTE R0
  013030 104426
12 013032 060004   ADD R0,R4
13 013034 110037 002224  MOVB R0,FDATA
14 013040          GETBYTE R0
  013040 104426
15 013042 060004   ADD R0,R4
16 013044 110037 002225  MOVB R0,FDATA+1
17 013050 000207   RETURN

```

```

;READ CHECKSUM BYTE          TRAP    C$GETB
;ADD TO COMPUTED CHECKSUM
;CHECK LOW BYTE OF SUM
;BRANCH IF CHECKSUM ERROR
;IF IN SEARCH MODE,
;  KEEP ON SEARCHING

      MOV (SP)+,R5
      MOV (SP)+,R4
      MOV (SP)+,R3
      MOV (SP)+,R2
      MOV (SP)+,R1
      MOV (SP)+,R0

;READ A BYTE FROM FILE          TRAP    C$GETB
;UPDATE CHECKSUM ERROR
;START TO BUILD WORD
;READ ANOTHER BYTE FROM FILE
;UPDATE CHECKSUM          TRAP    C$GETB
;COMPLETE WORD

```

1 013052 004737 012500
2 013056 012605
013060 012604
013062 012603
013064 012602
013066 012601
013070 012600
3 013072 000261
4 013074 000207
5
6 013076
013076 104454
013100 000005
013102 000000
013104 011054
7 013106
013106 104444

RDERR: CALL CLOSEF
POP <R5,R4,R3,R2,R1,R0>

SEC
RETURN

RWRDE1: ERRSF 5,,ERR005

DOCLN

;CLOSE FILE AS POSITION IS LOST

MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0

;ERROR RETURN, FILE NOT FOUND

TRAP C\$ERSF
.WORD 5
.WORD 0
.WORD ERR005
TRAP C\$DCLN

```

2      ;RESPDM
3      ;
4      ;RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
5      ;HAVE TERMINATED.
6
7 013110 013705 002162      RESPDM: MOV TSTTAB,R5      ;GET CONTROLLER TABLE ADDRESS
8 013114 013737 002174 002200      MOV URUN,UCNT      ;SET COUNTER OF UNITS
9 013122      RESPCT: BREAK      ;ALLOW DRS TO SEE TERMINAL INPUT
10 013122 104422      TRAP      C$BRK
11 013124 016504 000016      MOV C.RING(R5),R4      ;GET HOST COMM AREA ADDRESS
12 013130 032765 000002 000014      BIT #CT.RN,C.FLG(R5)      ;CHECK IF PROGRAM RUNNING
13 013136 001502      BEQ RSPNXT      ;IF NOT, LOOK AT NEXT
14 013140 116537 000002 002074      MOVB C.UNIT(R5),L$LUN      ;STORE UNIT NUMBER UNDER TEST
15 013146 032765 000010 000014      BIT #CT.MSG,C.FLG(R5)      ;SEE IF INTERRUPT RECEIVED
16 013154 001150      BNE RSPIN      ;IF SO, LOOK AT PACKET
17 013156 032765 000004 000014      BIT #CT.CMD,C.FLG(R5)      ;SEE IF COMMAND HAS BEEN SENT
18 013166 000137 013734      BNE 1$      ;IF NOT, SEND ONE
19      JMP RSPOUT
20
21      ;CHECK IF UDA STILL RUNNING
22 1$:      MOV (R5),R3      ;GET ADDRESS OF UDAIP
23 013172 011503      MOV 2(R3),R1      ;LOOK AT UDASA REGISTER
24 013174 016301 000002      BEQ RSPTM      ;IF ZERO, UDA STILL RUNNING
25 013200 001405      ERRDF 30,,ERR030      ;REPORT UDA HAS FATAL ERROR
26 013202 104455      TRAP      C$ERDF
27 013204 000036      .WORD 30
28 013206 000000      .WORD 0
29 013210 011404      .WORD ERR030
30 013212 000465      BR RSPDRP      ;DROP CONTROLLER FROM TESTING
31
32      ;CHECK FOR TIMEOUT OF RESPONSE
33
34 RSPTM: TST C.TGT(R5)      ;SEE IF DUP PROGRAM TO BE TIMED
35 BEQ RSPNTO
36 TST KW.CSR      ;SEE IF A CLOCK ON SYSTEM
37 BEQ RSPNTO      ;DON'T TIME IF NO CLOCK
38 CMP KW.EL+2,C.TOH(R5)      ;COMPARE TO TIMEOUT COUNTER
39 BHI RSPTMO
40 BNE RSPNTO
41 CMP KW.EL,C.TO(R5)
42 BLO RSPNTO
43 RSPTMO: BIT #CT.STA,C.FLG(R5)      ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
44 BNE RSPTOE      ;SEE IF A GET DUST STATUS COMMAND OUTSTANDING
45 TST HC.CCT(R4)      ;REPORT ERROR IF SO
46 BMI RSPTOE      ;SEE IF UDA TOOK LAST COMMAND PACKET
47 MOV #CT.TM1,R0      ;REPORT ERROR IF NOT
48 BIT #CT.TM1,C.FLG(R5)      ;SEE IF FIRST TIMEOUT ALREADY HAPPENED
49 BEQ 1$
50 ASL R0      ;IF SO,
51 BIS #CT.STA,R0      ;SET SECOND TIME OUT FLAG
52 BIS R0,C.FLG(R5)      ;SET THE PROPER TIMEOUT BIT
53 MOV #OP.GDS,R0      ; AND STATUS REQUESTED BIT
54 CALL BLDCMD      ;BUILD GET DUST STATUS COMMAND
55 MOV #RG.OWN,HC.CCT(R4)      ;MARK COMMAND TO UDA
56 TST @ (R5)      ;TELL UDA COMMAND IS THERE
57 JMP RSPOU4

```

54 013344

RSPNTO:

```

1          ;SWITCH TO NEXT CONTROLLER
2
3 013344 005737 002220 RSPNXT: TST UFREEZ      ;FROZEN TO ONE UNIT?
4 013350 001264      BNE RESPCT      ;STAY THERE IF SO
5 013352 062705 000054 ADD #C.SIZE,R5      ;MOVE TO NEXT TABLE
6 013356 005337 002200 DEC UCNT          ;CHECK IF MORE CONTROLLERS
7 013362 001257      BNE RESPCT      ;LOOK AT NEXT CONTROLLER
8 013364 000651      BR RESPDM       ;LOOK AT FIRST CONTROLLER AGAIN
9
10         ;REMOVE A CONTROLLER FROM TESTING
11
12 013366 005065 000014 RSPDRP: CLR C.FLG(R5)    ;CLEAR PROGRAM RUNNING
13 013372 005037 002220 CLR UFREEZ
14 013376 010504      MOV R5,R4
15 013400 062704 000020 ADD #C.DRO,R4
16 013404 012702 000010 MOV #8.,R2
17 013410 012403 1$: MOV (R4)+,R3
18 013412 001420      BEQ 3$
19 013414 005763 000002 TST D.UNIT(R3)
20 013420      ASSUME DT.AVL EQ BIT15
21 013420 100003      BPL 2$
22 013422 005302      DEC R2
23 013424 001371      BNE 1$
24 013426 000412      BR 3$
25 013430 052763 100000 000002 2$: BIS #DT.AVL,D.UNIT(R3)
26 013436 005302      DEC R2
27 013440 001405      BEQ 3$
28 013442 005714      TST (R4)
29 013444 001403      BEQ 3$
30 013446 004737 016372 CALL LOADDM      ;START DM PROGRAM AGAIN
31 013452 001223      BNE RESPCT
32 013454 005337 002176 3$: DEC URNING      ;REDUCE RUNNING CONTROLLERS COUNT
33 013460 001331      BNE RSPNXT      ;IF ANY STILL RUNNING, LOOK AT THEM
34 013462 000207      RETURN          ;ELSE RETURN TO TEST SECTION
35
36 013464      RSPTOE: ERRDF 31,,ERR031 ;REPORT TIMEOUT ERROR
    013464 104455      TRAP          C$ERDF
    013466 000037      .WORD        31
    013470 000000      .WORD        0
    013472 011420      .WORD        ERR031
37 013474 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 013476 012700 000204      RSPIN: MOV #OP.END+OP.SSD,R0      ;GET SEND DATA END PACKET OPCODE
6 013502 032765 000020 000014 BIT #CT.REQ,C.FLG(R5)      ;LOOK IF SEND DATA OR RECEIVE DATA
7 013510 001402      BEQ RSPMWR
8 013512 012700 000205      MOV #OP.END+OP.RSD,R0      ;CHANGE TO RECEIVE DATA END PACKET OPCODE
9 013516 120064 000030      RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4) ;COMPARE TO OPCODE IN END PACKET
10 013522 001145      BNE RSPERR
11
12      ;LOOK AT STATUS CODE
13
14 013524 032764 000037 000032      BIT #ST.MSK,HC.MPK+P.STS(R4) ;CHECK FOR STATUS CODE ST.SUC (ZERO)
15 013532 001004      BNE RSPERW
16
17      ;CHECK FOR EXPECTED REFERENCE NUMBER
18
19 013534 026564 000052 000020      CMP C.REF(R5),HC.MPK+P.CRF(R4) ;CHECK IF CORRECT REF NUMBER
20 013542 001405      BEQ RSPPTW
21 013544      RSPERW: ERRDF 33,,ERR033
22 013544 104455      TRAP C$ERDF
23 013546 000041      .WORD 33
24 013550 000000      .WORD 0
25 013552 011450      .WORD ERR033
26 013554 000704      BR RSPDRP      ;DROP UNIT FROM TESTING
27
28      ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
29
30 013556 032765 000020 000014      RSPPTW: BIT #CT.REQ,C.FLG(R5) ;CHECK IF RESPONSE FROM DM PROGRAM
31 013564 001463      RSPOU: BEQ RSPOUT ;LOOK AT REQUEST NUMBER IF SO

```

```

1      ;MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
2
3 013566 016401 000306 RSPPT2: MOV HC.BF2(R4),R1      ;GET REQUEST NUMBER
4 013572 042701 007777 BIC #C<DU.TYP>,R1      ;CHECK TYPE
5 013576 001403 BEQ 1$      ;IF ZERO, ERROR
6 013600 020127 060000 CMP R1,#DU.SPC      ;CHECK IF IN EXPECTED RANGE
7 013604 101405 BLOS RSPPT3
8 013606 1$: ERRDF 32,,ERR032      ;BAD REQUEST NUMBER
   013606 104455 TRAP C$ERDF
   013610 000040 .WORD 32
   013612 000000 .WORD 0
   013614 011432 .WORD ERR032
9 013616 000663 BR RSPDRP      ;DROP UNIT FROM TESTING
10
11 013620 016403 000034 RSPPT3: MOV HC.MPK+P.BCNT(R4),R3      ;GET BYTE COUNT OF CHARACTERS RECEIVED IN R3
12 013624 162703 000002 SUB #2,R3      ;(FIRST TWO CHARACTERS ARE TYPE WORD)
13 013630 012700 000004 MOV #OP.SSD,R0      ;BUILD A SEND DATA COMMAND PACKET
14 013634 004737 016574 CALL BLDCMD      ; FOR ANSWER TO DM PROGRAM
15 013640 012700 000164 MOV #HC.BF1,R0      ;POINT TO BUFFER IN PACKET
16 013644 004737 016736 CALL CLRBUF      ; AND CLEAR BUFFER
17 013650 010402 MOV R4,R2      ;R2 POINTS TO SEND BUFFER
18 013652 062704 000122 ADD #HC.BSZ,R4      ;R4 POINTS TO CHARACTERS IN RECEIVE BUFFER
19 013656 042724 170000 BIC #DU.TYP,(R4)+      ;CLEAR TYPE FIELD IN BUFFER
20 013662 000301 SWAB R1      ;GET TYPE RIGHT JUSTIFIED
21 013664 006201 ASR R1      ;TIMES TWO
22 013666 006201 ASR R1
23 013670 006201 ASR R1
24 013672 010100 MOV R1,R0      ;COPY MESSAGE TYPE TO R1
25 013674 005001 CLR R1      ;R1 CONTAINS ZERO SEND BYTE COUNT
26 013676 004770 014162 CALL @RSPDSP-2(R0)      ;CALL REQUESTED ROUTINE
27 013702 001231 BNE RSPDRP      ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
28      ; Z SET IF UNIT TO CONTINUE RUNNING
29 013704 016504 000016 MOV C.RING(R5),R4      ;GET RING ADDRESS
30 013710 032701 000001 BIT #1,R1      ;LOOK AT CHARACTER COUNT TO SEND TO DUP PROGRAM
31 013714 001401 BEQ 1$      ;IF AN ODD COUNT
32 013716 005201 INC R1      ; INCREASE BY ONE
33 013720 010164 000120 1$: MOV R1,HC.CPK+P.BCNT(R4)      ;PUT CHARACTER COUNT IN COMMAND PACKET
34 013724 100003 BPL RSPOUT      ;IF NEGATIVE BYTE COUNT RETURNED
35 013726 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 013734 042765 000350 000014 RSPOUT: BIC #CT.MSG+CT.STA+CT.TM1+CT.TM2,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
40 013742 032765 000020 000014 BIT #CT.REQ,C.FLG(R5)      ;CHECK WHICH COMMAND TO SEND
41 013750 001014 BNE RSPOU2      ;BRANCH IF RESPONSE TO REQUEST
42
43 013752 012700 000005 MOV #OP.RSD,R0      ;BUILD RECEIVE DATA COMMAND
44 013756 004737 016574 CALL BLDCMD
45 013762 012700 000306 MOV #HC.BF2,R0      ;POINT TO MESSAGE BUFFER
46 013766 004737 016736 CALL CLRBUF      ; AND CLEAR IT
47 013772 052765 000020 000014 BIS #CT.REQ,C.FLG(R5)      ;SET REQUEST BIT
48 014000 000403 BR RSPOU3
49
50 014002 042765 000020 000014 RSPOU2: BIC #CT.REQ,C.FLG(R5)      ;CLEAR REQUEST BIT
51 014010 RSPOU3:
52 014010 004737 016660 CALL SNDCMD      ;SEND COMMAND TO UDA
53 014014 016500 000044 RSPOU4: MOV C.TOT(R5),R0      ;SET TIMEOUT

```

54	014020	010501			MOV R5,R1	
55	014022	062701	000040		ADD #C.TO,R1	;PUT TIME IN CONTROLLER TABLE
56	014026	004737	017172		CALL SETTO	
57	014032	000137	013344		JMP RSPNXT	;NOW WAIT FOR END PACKET
58	014036	122764	000201	000030	RSPERR: CMPB #OP.END+OP.GDS,HC.MPK+P.OPCD(R4)	;SEE IF GET DUST STATUS OPCCDE
59	014044	001237			BNE RSPERW	
60	014046	132764	000010	000037	BITB #DF.ACT,HC.MPK+P.DFLG(R4)	;IF DUST NO LONGER RUNNING
61	014054	001603			BEQ RSPTOE	;REPORT ERROR
62	014056	042765	000050	000014	BIC #CT.STA+CT.MSG,C.FLG(R5)	;CLEAR CONTROL BITS
63	014064	032765	000200	000014	BIT #CT.TM2,C.FLG(R5)	;IF AT SECOND TIMEOUT
64	014072	001413			BEQ 1\$	
65	014074	026465	000040	000046	CMP HC.MPK+P.DPI(R4),C.PRI(R5)	;COMPARE PROGRESS INDICATOR
66	014102	001004			BNE 2\$	
67	014104	026465	000042	000050	CMP HC.MPK+P.DPI+2(R4),C.PRI+2(R5)	;COMPARE PROGRESS INDICATOR
68	014112	001422			BEQ 4\$	;REPORT ERROR IF NOT CHANGED
69	014114	042765	000200	000014	BIC #CT.TM2,C.FLG(R5)	;CLEAR TIMEOUT 2 FLAG
70	014122	032765	000100	000014	BIT #CT.TM1,C.FLG(R5)	;IF AT FIRST TIMEOUT
71	014130	001406			BEQ 3\$	
72	014132	016465	000040	000046	MOV HC.MPK+P.DPI(R4),C.PRI(R5)	;GET COPY OF PROGRESS INDICATOR
73	014140	016465	000042	000050	MOV HC.MPK+P.DPI+2(R4),C.PRI+2(R5)	;GET COPY OF PROGRESS INDICATOR
74	014146	012764	140000	000006	MOV #RG.OWN+RG.FLG,HC.MCT(R4)	;GIVE MESSAGE BUFFER BACK TO UDA
75	014154	000137	013344		JMP RSPNXT	
76	014160	000137	013464		JMP RSPTOE	

1
2
3 014164 014200
4 014166 014252
5 014170 014424
6 014172 014514
7 014174 014524
8 014176 014534
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	R2 CONTAINS THIS ADDRESS
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	

```

1      ;MESSAGE TYPE 1
2
3      ;ANSWER QUESTION FOR DUP PROGRAM
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
12     ;OUTPUT:
13     R1 - COUNT OF CHARACTERS IN SEND BUFFER
14     Z SET TO CONTINUE RUNNING DUP PROGRAM
15     Z CLEAR TO STOP THE DUP PROGRAM
16
17 014200 004737 014666
18 014204 062700 000004
19 014210 014403
20 014212 001411
21 014214 020327 000007
22 014220 001410
23 014222 104455
24 014224 000144
25 014226 000000
26 014230 011512
27 014232 000244
28 014234 000207
29
30 014236 012700 003344
31 014242
32 014244 005201
33 014246 112022
34 014248 001375
35 014250 000207

```

QUEST: CALL GTDRV ;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
ERRDF 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
12     ;OUTPUT:
13     R1 - COUNT OF CHARACTERS IN SEND BUFFER
14     Z SET TO CONTINUE RUNNING DUP PROGRAM
15     Z CLEAR TO STOP THE DUP PROGRAM
16
17 DQUEST: CALL GTDRVT      ;GET DRIVE TABLE ADDRESS INTO R0
18         MOV -(R4),R3    ;GET QUESTION NUMBER
19         CMP R3,#DQUESZ
20         BHI DQUEX
21         ASL R3
22         JMP @DQUEJP(R3)
23
24 DQUEJP: .WORD DQUEX      ; 0 (NOT USED)
25         .WORD DQUNIT    ; 1 ENTER UNIT NUMBER TO FORMAT
26         .WORD DQUEX      ; 2 (NOT USED)
27         .WORD DQUEX      ; 3 (NOT USED)
28         .WORD DQRFMT     ; 4 USE EXISTING BAD SECTOR INFORMATION
29         .WORD DQRSTR     ; 5 DOWN-LINE LOAD BAD SECTOR BLOCK INFORMATION
30         .WORD DQCONT     ; 6 CONTINUE IF BAD BLOCK INFO INACCESSIBLE
31         DQUESZ=<<.-DQUEJP>/2>-1
32
33 ;ENTER UNIT NUMBER TO FORMAT
34 DQUNIT: PUSH R5
35
36         CLR R4
37         MOV (R0),R3      ;GET DRIVE NUMBER
38         ASSUME D.DRV EQ 0
39         MOV #10,R0      ;RADIX 10.
40 DQUNL1: CALL DIVIDE
41         PUSH R5
42
43         INC R1
44         TST R3
45         BNE DQUNL1
46         MOV R1,R0
47 DQUNL2: POP R5
48
49         ADD #'0,R5
50         MOVB R5,(R2)+
51         DEC R0
52         BNE DQUNL2
53         POP R5
54
55 DQUEX: SEZ
56        RETURN
57
58 DQRFMT: BIT #SO.FMT,MODE

```

54	014372	001410				BEQ DQNO
55	014374	112712	000131		DQYES:	MOVB #'Y,(R2)
56	014400	005201				INC R1
57	014402	000766				BR DQUEX
58						
59	014404	032737	000010	003230	DQRSTR:	BIT #SO.STR,MODE
60	014412	001370				BNE DQYES
61	014414				DQCONT:	
62	014414	112712	000116		DQNO:	MOVB #'N,(R2)
63	014420	005201				INC R1
64	014422	000756				BR DQUEX

1			:MESSAGE TYPE 3		
2			:PRINT INFORMATION FROM DUP PROGRAM		
3			:INPUT:		
4			R5 - POINTER TO CONTROLLER TABLE		
5			R4 - POINTER TO DATA IN RECEIVE BUFFER		
6			R3 - CHARACTER COUNT IN RECEIVE BUFFER		
7			R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)		
8			R1 - ZERO		
9			:OUTPUT:		
10			R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM		
11			Z SET TO CONTINUE RUNNING DUP PROGRAM		
12					
13					
14					
15	014424	016400	177776	INFO: MOV -2(R4),R0	:GET MESSAGE NUMBER
16	014430	001417		BEQ INFOX	:IF ZERO, IGNORE IT
17	014432	020027	000100	CMP R0,#100	:IF OCTAL 100
18	014436	001420		BEQ INFOE	: PRINT ERROR MESSAGE
19	014440	005737	002220	TST UFREEZ	
20	014444	001007		BNE INFOP	
21	014446	005237	002220	INC UFREEZ	
22	014452	004737	014666	CALL GTDRV	
23	014456	010002		MOV R0,R2	
24	014460	004737	014712	CALL HEADER	
25	014464	004737	014632	INFOP: CALL MSG	:PRINT THE MESSAGE
26	014470	012701	100000	INFOX: MOV #BIT15,R1	:RETURN A NEGATIVE BYTE COUNT
27	014474	000264		SEZ	
28	014476	000207		RETURN	:RETURN WITH Z SET
29					
30	014500			INFOE: ERRDF 101,,ERR101	:ANSWER WAS REJECTED BY DUP PROGRAM
	014500	104455			TRAP
	014502	000145			.WORD
	014504	000000			.WORD
	014506	011526			.WORD
31	014510	000244			C\$ERDF
32	014512	000207			101
					0
					ERR101
				CLZ	:RETURN WITH Z CLEAR TO STOP DUP PROGRAM
				RETURN	

```
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 014514 004737 014424 TERM: CALL INFO      ;PRINT THE MESSAGE
15 014520 000244          CLZ
16 014522 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 014524 004737 014424      ERRTRM: CALL INFO
15 014530 000244              CLZ
16 014532 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
12     ;OUTPUT:
13     Z SET TO SEND DATA TO PROGRAM
14 014534 023714 003226   SPECL:  CMP FCTNUM,(R4) ;SEE IF DESIRED BLOCK IS IN MEMORY
15 014540 001425         BEQ SPECLX      ; IF SO, SEND TO DUP PROGRAM
16 014542 002407         BLT SPECLR      ; IF LOWERED NUMBERED BLOCK IN MEMORY,
17                                     ; GO READ NEXT BLOCK
18 014544               SPECLC:
19
20 014544 104435         CLOSE      ;OTHERWISE, START READING FROM BEGINNING AGAIN
21 014546 012700 003254   OPEN #FNAME2
22                                     TRAP      C$CLOS
23 014552 104434         MOV #FNAME2,R0
24 014554 012737 177777 003226   TRAP      C$OPEN
25 014562 012703 001000   MOV #-1,FCTNUM
26 014566 012701 002226   SPECLR:  MOV #512,R3 ;GET BYTE COUNT IN A BLOCK
27                                     MOV #FCTBUF,R1 ;POINT TO STORAGE AREA
28                                     SPECLL:  GETBYTE (R1)+ ;READ THE FILE
29                                     TRAP      C$GETB
30 014572 104426         MOV      R0,(R1)+
31 014574 110021         BNCOMPLETE SPECLE ;PRINT ERROR IF NO MORE BYTES IN FILE
32                                     TRAP      C$SPECLE
33 014576 103005         DEC R3 ;COUNT THE BYTES
34 014600 005303         BNE SPECLL
35 014602 001373         INC FCTNUM ;KEEP COUNT OF BLOCK IN MEMORY
36 014604 005237 003226   BR SPECL
37 014610 000751
38
39 014612 005212         SPECLE:  INC (R2) ;TELL DUP PROGRAM DATA NOT AVAILABLE
40 014614 012762 002226 000002   SPECLX:  MOV #FCTBUF,2(R2) ;PUT ADDRESS OF DATA IN OUTPUT BUFFER
41 014622 012701 000006   MOV #6,R1 ;SEND 3 WORDS TO DUP PROGRAM
42 014626 000264         SEZ
43 014630 000207         RETURN ;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     ;
12     Z SET TO CONTINUE RUNNING DUP PROGRAM
13
14     MSG: PRINT #CR
15
16     1$:  MOVB (R4)+,R0
17           BEQ 2$
18           CMP R0,#12
19           BEQ 2$
20           PRINT R0
21
22           MOVB #CR,R0
23           CALL CPNT
24           ;PRINT CHARACTERS FROM DUP PROGRAM
25           ; DISCARDING LF AND NULL CHARACTERS
26
27           CALL CPNT
28           ;COUNT THE CHARACTERS
29
30     2$:  DEC R3
31           BGT 1$
32           RETURN

```

```

1      :GTDRVT
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     014666      GTDRVT: PUSH R5
12     014666      010546
13     014670      062705 000020
14     014674      012500
15     014676      016037 000002 002074
16     014704      100773
17     014706
18     014706      012605
19     014710      000207

      ADD #C.DRO,R5
      MOV (R5)+,R0
      MOV D.UNIT(R0),L$LUN
      ASSUME DT.AVL EQ BIT15
      BMI GTDRVL
      POP R5
      RETURN

      MOV R5,-(SP)
      MOV (SP)+,R5

```

```

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
10     ;OUTPUT:
11         R0 - POINTER TO DRIVE TABLE
12         PRINTED MESSAGE
13 014712 022737 000001 002012 HEADER: CMP #1,LSUNIT          ;IF MORE THAN ONE UNIT BEING TESTED
14 014720 001411                      BEQ 1$
15 014722                      PNTF MESSG,D.UNIT(R2),(R5),(R2)      ;PRINT UDA ADDRESS
16 014722 011246                      MOV (R2),-(SP)
17 014724 011546                      MOV (R5),-(SP)
18 014726 016246 000002                MOV D.UNIT(R2),-(SP)
19 014732 004137 016230                JSR R1,LPNTF
20 014736 004015                      .WORD MESSG
21 014740 000006                      .WORD PNT.CT
22 014742                      ASSUME C.UADR EQ 0
23 014742                      ASSUME D.DRV EQ 0
24 014742 000407                      BR 2$
25 014744 005737 003232 1$:          TST KW.CSR          ;IF NO CLOCK BEING USED
26 014750 001406                      BEQ 3$          ;BYPASS RUNTIME MESSAGE
27 014752                      PRINT #CR
28 014752 112700 000015                      MOV #CR,R0
29 014756 004737 016056                      CALL CPNT
30 014762 004737 020176 2$:          CALL RNTIME          ;PRINT RUNTIME IF A CLOCK IN USE
31 014766 000207 3$:          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
32     :OUTPUTS:
33     :   R2 AND R4 UPDATED TO END OF STRING AND PARAMETERS
34
35     OSTRNG: MOVB (R2)+,R1      ;GET CONTROL CHARACTER
36             BEQ OSTRE         ;EXIT IF NULL CHARACTER
37             MOV #ERRC,R0      ;GET POINTER TO CHARACTER TABLE
38             CMPB R1,(R0)      ;COMPARE CHARACTER WITH TABLE ENTRY
39             BEQ NCONF         ;BRANCH IF MATCH FOUND
40             TSTB (R0)+        ;INCREMENT POINTER
41             BNE NCONS         ;CONTINUE SEARCH IF NOT END OF TABLE
42             PNTF ERRME1       ;REPORT BAD CONTROL CHARACTER
43             JSR R1,LPNTF      ;
44             .WORD ERRME1      ;
45             .WORD PNT.CT      ;
46
47             BR OSTRE
48
49     NCONF:  SUB #ERRC,R0      ;GET INCREMENT INTO TABLE
50             ASL R0            ;DOUBLE TO WORD COUNT
51             CALL @ERRD(R0)    ;DISPATCH TO PRINT ROUTINE
52             BR OSTRNG        ;GET NEXT
53
54     GSTRE:  RETURN

```

34	014770	112201	
35	014772	001421	
36	014774	012700	015270
37	015000	120110	
38	015002	001407	
39	015004	105720	
40	015006	001374	
41	015010		
	015010	004137	016230
	015014	003726	
	015016	000000	
42	015020	000406	
43	015022	162700	015270
44	015026	006300	
45	015030	004770	015302
46	015034	000755	
47	015036	000207	

```

1      ;CONTROL CHARACTER WAS A QUOTE. PRINT ALL CHARACTERS TO THE NEXT QUOTE.
2
3 015040 112200      CON.QU: MOVB (R2)+,R0      ;GET CHARACTER
4 015042 120027 000042      CMPB R0,#'"      ;CHECK IF ENDING QUOTE
5 015046 001403      BEQ CON.QX      ;IF SO, GO GET NEXT CONTROL CHARACTER
6 015050      PRINT R0      ;PRINT THE CHARACTER
7 015050 004737 016056      BR CON.QU      CALL CPNT
8 015056 000771      CON.QX: RETURN      ;CONTINUE PRINTING
9 015056 000207
10
11      ;CONTROL CHARACTER WAS AN A. PRINT ASCII CHARACTERS FROM PARAMETERS.
12 015060 004737 015536      CON.A: CALL GETCNT      ;GET COUNT OF CHARACTERS
13 015064      CON.A1: PRINT (R4)+      ;PRINT THE CHARACTER
14 015064 112400      MOV (R4)+,R0      ;COUNT THE CHARACTERS
15 015066 004737 016056      CALL CPNT      ;PRINT UNTIL COUNT REACHES ZERO
16 015072 005301      DEC R1      ;CHECK IF R4 NOW ODD
17 015074 001373      BNE CON.A1      ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
18 015076 032704 000001      BIT #1,R4      ;NOW GET NEXT CONTROL CHARACTER
19 015102 001401      BEQ CON.A2
20 015104 005204      INC R4
21 015106 000207      CON.A2: RETURN
22
23      ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.
24 015110 012701 000012      CON.D: MOV #10.,R1      ;LOAD RADIX
25 015114 004737 015614      CALL PNTNUM      ;PRINT NUMBER
26 015120 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER
27
28      ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.
29 015122 012701 000020      CON.H: MOV #16.,R1      ;LOAD RADIX
30 015126 004737 015614      CALL PNTNUM      ;PRINT NUMBER
31 015132 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER

```

```

1      ;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.
2
3 015134 012701 000010  CON.O:  MOV #8.,R1      ;LOAD RADIX
4 015140 004737 015614      CALL PNINUM      ;PRINT NUMBER
5 015144 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER
6
7      ;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.
8
9 015146 004737 015536  CON.N:  CALL GETCNT      ;GET COUNT
10 015152      CON.N1: PRINT #CR      ;PRINT NEW LINE SEQUENCE
    015152 112700 000015      MOV #CR,R0
    015156 004737 016056      CALL CPNT
11 015162 005301      DEC R1      ;COUNT THE SEQUENCES
12 015164 001372      BNE CON.N1
13 015166 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER
14
15      ;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.
16
17 015170 004737 015536  CON.R:  CALL GETCNT      ;GET ROUTINE NUMBER
18 015174 020127 000010      CMP R1,#ERRRSZ      ;CHECK IF DEFINED ROUTINE NUMBER
19 015200 101004      BHI CON.R1
20 015202 060101      ADD R1,R1      ;DOUBLE COUNT TO GET WORD INDEX
21 015204 004771 015246      CALL @ERRRTB-2(R1)      ;CALL ROUTINE
22 015210 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER
23 015212      CON.R1: PNTF ERRME1      ;REPORT BAD MESSAGE STRING
    015212 004137 016230      JSR R1,LPNTF
    015216 003726      .WORD ERRME1
    015220 000000      .WORD PNT.CT
24 015222      POP R1      ;FIX THE STACK
    015222 012601      MOV (SP)+,R1
25 015224 000207      RETURN
26
27      ;CONTROL CHARACTER WAS AN S. PRINT SPACES.
28
29 015226 004737 015536  CON.S:  CALL GETCNT      ;GET COUNT
30 015232      CON.S1: PRINT '<#>'      ;PRINT A SPACE
    015232 112700 000040      MOV #',R0
    015236 004737 016056      CALL CPNT
31 015242 005301      DEC R1      ;COUNT THE SPACES
32 015244 001372      BNE CON.S1
33 015246 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER

```

```

1      ;ERROR ROUTINE DISPATCH TABLE
2
3 015250 015322      ERRRTB: .WORD CALRE      ;NOT USED
4 015252 015322      .WORD CALRE      ;NOT USED
5 015254 015322      .WORD CALRE      ;NOT USED
6 015256 015334      .WORD CALR4      ;PRINT BASIC LINE WITHOUT UDA ADDRESS
7 015260 015410      .WORD CALR5      ;PRINT BASIC LINE WITH UDA ADDRESS
8 015262 015466      .WORD CALR6      ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
9 015264 015502      .WORD CALR7      ;PRINT 'REPLACE UDA MODULE M7161 OR M7485'
10 015266 015520      .WORD CALR8      ;PRINT ' UDASA CONTAINS XXXXXX'
11      ; .WORD CALR9      ;REPRINT LAST NUMBER
12      000010      ERRRSZ=<.-ERRRTB>/2
13
14      ;BUILD TWO TABLES
15      : FIRST CONTAINING CONTROL CHARACTERS
16      : SECOND CONTAINING ROUTINE ADDRESSES
17
18      .MACRO BUILD
19      ENTRY ",CON.QU
20      ENTRY A,CON.A
21      ENTRY D,CON.D
22      ENTRY H,CON.H
23      ENTRY O,CON.O
24      ENTRY N,CON.N
25      ENTRY R,CON.R
26      ENTRY S,CON.S
27      .ENDM

```

1			;HERE IS FIRST TABLE	
2				
3			.MACRO ENTRY ARG1,ARG2	
4			.LIST	
5			.BYTE ''ARG1	
6			.NLIST	
7			.ENDM	
8				
9	015270		ERRC: BUILD	
	015270	042	.BYTE ''	
	015271	101	.BYTE 'A	
	015272	104	.BYTE 'D	
	015273	110	.BYTE 'H	
	015274	117	.BYTE 'O	
	015275	116	.BYTE 'N	
	015276	122	.BYTE 'R	
	015277	123	.BYTE 'S	
10	015300	000	.BYTE 0	
11			.EVEN	
12				
13			;HERE IS SECOND TABLE	
14				
15			.MACRO ENTRY ARG1,ARG2	
16			.LIST	
17			.WORD ARG2	
18			.NLIST	
19			.ENDM	
20				
21	015302		ERRD: BUILD	
	015302	015040	.WORD CON.QU	
	015304	015060	.WORD CON.A	
	015306	015110	.WORD CON.D	
	015310	015122	.WORD CON.H	
	015312	015134	.WORD CON.O	
	015314	015146	.WORD CON.N	
	015316	015170	.WORD CON.R	
	015320	015226	.WORD CON.S	

;FOLLOW WITH A NULL BYTE

1
2
3
4
5
015322
015322 004137 016230
015326 003726
015330 000000
015332 000207
;PRE-PROGRAMMED ROUTINES 1, 2 AND 3
;NOT USED - PRINTS ERROR MESSAGE
CALRE: PNTF ERRME1
RETURN

;PRINT ERROR MESSAGE

JSR R1,LPNTF
.WORD ERRME1
.WORD PNT.CT

```
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 015334 012746 004211
   015334 012746 004211
   015340 012746 004211
   015344 012746 004211
   015350 012746 004136
   015354 004137 016240
   015360 004212
   015362 000010
7 015364 004737 020176      CALL RNTIME
   015370      PRINT #CR
   015370 112700 000015
   015374 004737 016056
8 015400 012737 016156 003246      MOV #PX,PTYPE
9 015406 000207      RETURN
```

```
1 ;PRE-PROGRAMMED ROUTINE 5
2 ;PRINT BASIC LINE FOR HOST PROGRAM ERROR WITH UDA ADDRESS
3 ;THEN SWITCH TO EXTENDED FORMAT
4
5 CALR5: PNTB BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
6
7 015410 012746 004211
8 015414 012746 004211
9 015420 011546
015422 012746 004155
015426 012746 004136
015432 004137 016240
015436 004212
015440 000012
6 015442 004737 020176
7 015446
015446 112700 000015
015452 004737 016056
8 015456 012737 016156 003246
9 015464 000207
```

```
CALL RNTIME
PRINT #CR
```

```
MOV #PX,PTYPE
RETURN
```

```
MOV #BAS,-(SP)
MOV #BAS,-(SP)
MOV (R5),-(SP)
MOV #BASL2,-(SP)
MOV #BASNO,-(SP)
JSR R1,LPNTB
.WORD BASLN
.WORD PNT.CT
```

```
MOVB #CR,R0
CALL CPNT
```

```
1 ;PRE-PROGRAMMED ROUTINE 6
2 ;CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
3
4 015466 CALR6: PUSH R2 ;SAVE CURRENT STRING POINTER
5 015466 010246 ;MOV R2,-(SP)
6 015470 012402 ;GET NEW STRING POINTER
7 015472 004737 014770 ;OUTPUT USING THIS STRING
8 015476 012602 ;GET OLD POINTER BACK
9 015476 000207 ;MOV (SP)+,R2
10 015500 ;NOW CONTINUE THE OLD STRING
11 RETURN
```

```
1  
2  
3  
4 015502  
   015502 010246  
5 015504 012702 010533  
6 015510 004737 014770  
7 015514  
   015514 012602  
8 015516 000207  
  
;PRE-PROGRAMMED ROUTINE 7  
;PRINT "REPLACE UDA MODULE M7161"  
  
CALR7:  PUSH R2  
  
        MOV R2,-(SP)  
  
        MOV #XFRU,R2  
        CALL OSTRNG  
        POP R2  
  
        MOV (SP)+,R2  
  
        RETURN
```

```
1  
2  
3  
4 015520      ;PRE-PROGRAMMED ROUTINE 8  
   015520      ;PRINT "" UDASA CONTAINS XXXXXX"  
5 015522      CALR8:  PUSH R2  
   012702      MOV #XSA,R2  
6 015526      010502      CALL OSTRNG  
   004737      014770      POP R2  
7 015532      MOV (SP)+,R2  
   015532      012602  
8 015534      000207      RETURN
```

1
2
3
4

: REPRINT LAST NUMBER
: R4 -> TABLE
: CALR9: TST -(R4)
: 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      ;OUTPUTS:
10     ;      R1 - NUMBER READ OR A ONE
11     ;      R2 - POINTING TO CHARACTER AFTER NUMBER
12     ;
13 015536 GETCNT: PUSH R0
14 015536          MOV RO,-(SP)
15 015540 010046          CLR R1          ;START WITH ZERO COUNT
16 015542 121227 000060 GETCNX: CMPB (R2),#'0      ;CHECK IF CHARACTER A DIGIT
17 015546 103415          BLO GETCDN      ;BRANCH IF LOWER THAN ZERO
18 015550 121227 000071      CMPB (R2),#'9
19 015554 101012          BHI GETCDN      ;BRANCH IF HIGHER THAN NINE
20 015556 006301          ASL R1          ;MULTIPLY NUMBER BY 10
21 015560 010100          MOV R1,R0      ;SAVE 2N
22 015562 006301          ASL R1          ;COMPUTE 4N
23 015566 060001          ADD R0,R1      ;COMPUTE 8N
24 015570 112200          MOVB (R2)+,R0  ;8N + 2N = 10N
25 015572 162700 000060      SUB #'0,R0  ;GET DIGIT FROM STING
26 015576 060001          ADD R0,R1      ;GET RID OF ASCII
27 015600 000760          BR GETCNX      ;ADD TO NUMBER
28 015602 005701          GETCDN: TST R1  ;GO TO NEXT CHARACTER
29 015604 001001          BNE GETCXX      ;CHECK IF NUMBER IS ZERO
30 015606 005201          INC R1          ;IF ZERO, CHANGE
31 015610          GETCXX: POP R0          ; TO DEFAULT OF ONE
32 015612 012600          RETURN          MOV (SP)+,R0
32 015612 000207

```

1			:PNTNUM	
2			:PRINT A NUMBER	
3			:INPUTS:	
4			:R1 - RADIX OF NUMBER	
5			:R2 - ASCII STRING TO COUNT OF BITS IN NUMBER	
6			:R4 - POINTER TO NUMBER (LOW WORD)	
7			:OUTPUTS:	
8			:NUMBER IS PRINTED. LEADING ZEROS ARE PRINTED EXCEPT FOR	
9			:DECIMAL NUMBERS.	
10			:R0 - CONTENTS DESTROYED	
11				
12				
13				
14	015614	010100	PNTNUM: MOV R1,R0	:SAVE RADIX
15	015616	004737 015536	CALL GETCNT	:GET COUNT OF BITS
16	015622		PNTNUS: PUSH <R2,R3,R5>	
	015622	010246		MOV R2,-(SP)
	015624	010346		MOV R3,-(SP)
	015626	010546		MOV R5,-(SP)
17	015630	012403	MOV (R4)+,R3	:GET ONE PARAMETER WORD
18	015632	005005	CLR R5	:CLEAR STORAGE FOR OTHER
19	015634	020127 000020	CMP R1,#16.	:MORE THAN 16 BITS IN NUMBER?
20	015640	003401	BLE 1\$	
21	015642	012405	MOV (R4)+,R5	:YES, GET SECOND PARAMETER WORD
22	015644		1\$: PUSH R4	
	015644	010446		MOV R4,-(SP)
23	015646	010504	MOV R5,R4	:PUT HIGH WORD IN R4
24	015650	012702 000020	MOV #16.,R2	:COMPUTE BITS NOT WANTED
25	015654	160102	SUB R1,R2	:BY SUBTRACTING BITS TO USE
26	015656	002002	BGE 2\$	:FROM 16.
27	015660	062702 000020	ADD #16.,R2	:IF NEGATIVE, ADD 16 FOR FIRST WORD
28	015664	001414	2\$: BEQ 6\$	:IF ZERO, NO BITS NEED BE CLEARED
29	015666	012705 100000	MOV #BIT15,R5	:START MASK WITH SIGN BIT SET
30	015672	005302	3\$: DEC R2	:COUNT BITS IN MASK
31	015674	001402	BEQ 4\$	
32	015676	006205	ASR R5	:SHIFT MORE BITS TO RIGHT
33	015700	000774	BR 3\$	
34	015702	020127 000020	4\$: CMP R1,#16.	:MORE THAN 16 BITS IN NUMBER?
35	015706	003402	BLE 5\$	
36	015710	040504	BIC R5,R4	:YES, CLEAR IN HIGH WORD
37	015712	000401	BR 6\$	
38	015714	040503	5\$: BIC R5,R3	:NO, CLEAR IN LOW WORD
39	015716	004737 016334	6\$: CALL DIVIDE	:DIVIDE BY RADIX IN R0
40	015722		PUSH R5	:PUSH REMAINDER ON STACK
	015722	010546		MOV R5,-(SP)
41	015724	005202	INC R2	:COUNT DIGITS ON STACK
42	015726	005703	TST R3	:CHECK IF QUOTIENT IS ZERO
43	015730	001372	BNE 6\$	
44	015732	005704	TST R4	
45	015734	001370	BNE 6\$	

```

1 015736 020027 00C012      CMP R0,#10.
2 015742 001423      BEQ 10$
3 015744 010103      MOV R1,R3
4 015746 162700 000014      SUB #12.,R0
5 015752 003002      BGT 7$
6 015754 012700 000003      MOV #3,R0
7 015760 004737 016334      7$: CALL DIVIDE
8 015764 005705      TST R5
9 015766 001401      BEQ 8$
10 015770 005203      INC R3
11 015772 160203      8$: SUB R2,R3
12 015774 001406      BEQ 10$
13 015776      9$: PRINT #'0
    015776 112700 000060
    016002 004737 016056
14 016006 005303      DEC R3
15 016010 001372      BNE 9$
16
17 016012      10$: POP R5
    016012 012605
18 016014 062705 000060      ADD #'0,R5
19 016020 020527 000071      CMP R5,#'9
20 016024 003402      BLE 11$
21 016026 062705 000007      ADD #<'A-'9-1>,R5
22 016032      11$: PRINT R5
    016032 110500
    016034 004737 016056
23 016040 005302      DEC R2
24 016042 001363      BNE 10$
25 016044      POP <R4,R5,R3,R2>
    016044 012604
    016046 012605
    016050 012603
    016052 012602
26 016054 000207      RETURN

```

```

; IF RADIX IS DECIMAL
; JUST GO PRINT DIGITS ON STACK
; OTHERWISE COMPUTE NUMBER OF LEADING ZEROS
; DIVIDEND IS BITS IN NUMBER
; DIVISOR IS BITS PER DIGIT PRINTED
; (3 OR 4)

; IF REMAINDER NOT ZERO
; INCREMENT QUOTIENT

; SUBTRACT DIGITS ON STACK
; NO LEADING ZEROS IF ZERO
; PRINT A ZERO
                                MOVB #'0,R0
                                CALL CPNT

; REPEAT UNTIL COUNT REACHES ZERO

; GET CHARACTER FROM STACK
                                MOV (SP)+,R5
; CONVERT TO ASCII DIGIT
; IF GREATER THAN A 9
; CONVERT TO A OR HIGHER
; FOR HEX DIGIT
; PRINT THE CHARACTER
                                MOV R5,R0
                                CALL CPNT

; REPEAT FOR ALL DIGITS
; ON STACK

                                MOV (SP)+,R4
                                MOV (SP)+,R5
                                MOV (SP)+,R3
                                MOV (SP)+,R2

```

```

1      ;PRINT ONE CHARACTER
2      ;
3      ;CALL WITH MACRO PRINT
4
5 016056 110037 003250      CPNT:  MOV R0,ERRCHR
6 016062                                PUSH R1
7 016062 010146                                MOV R1,-(SP)
8 016064 012701 003664      MOV #ERRONE,R1
9 016070 120027 000015      CMPB R0,#CR
10 016074 001002      BNE 1$
11 016076 012701 003667      MOV #ERRNL,R1
12 016102 000177 165140      1$:  JMP @PTYPE
13 016106 012746 003250      PF:  PRINTF R1,#ERRCHR
14 016112 010146
15 016114 012746 000002
16 016120 010600
17 016122 104417
18 016124 062706 000006      MOV #ERRCHR,-(SP)
19 016130 000435                                MOV R1,-(SP)
20 016132 012746 003250      PB:  PRINTB R1,#ERRCHR      MOV #2,-(SP)
21 016136 010146                                MOV SP,R0
22 016140 012746 000002                                TRAP C$PNTF
23 016144 010600                                ADD #6,SP
24 016146 104414
25 016150 062706 000006      MOV #ERRCHR,-(SP)
26 016154 000423      PX:  PRINTX R1,#ERRCHR      MOV R1,-(SP)
27 016156 012746 003250      MOV #2,-(SP)
28 016162 010146                                MOV SP,R0
29 016164 012746 000002                                TRAP C$PNTB
30 016170 010600                                ADD #6,SP
31 016172 104415
32 016174 062706 000006      MOV #ERRCHR,-(SP)
33 016200 000411      PS:  PRINTS R1,#ERRCHR      MOV R1,-(SP)
34 016202 012746 003250      MOV #2,-(SP)
35 016206 010146                                MOV SP,R0
36 016210 012746 000002                                TRAP C$PNTX
37 016214 010600                                ADD #6,SP
38 016216 104416
39 016220 062706 000006      CPNTX: POP R1
40 016224 012601      RETURN
41 016226 000207      MOV (SP)+,R1

```

```

1          :PRINT FORMATTED MESSAGE
2
3          :CALL WITH MACRO PNT, PNTF, PNTB, PNTX, OR PNTS
4
5 016230 012737 016106 003246 LPNTF: MOV #PF,PTYPE
6 016236 000413                BR LPNT
7 016240 012737 016132 003246 LPNTB: MOV #PB,PTYPE
8 016246 000407                BR LPNT
9 016250 012737 016156 003246 LPNTX: MOV #PX,PTYPE
10 016256 000403               BR LPNT
11 016260 012737 016202 003246 LPNTS: MOV #PS,PTYPE
12 016266                LPNT:  PUSH <R2,R3,R4,R5>
13 016266 010246                MOV R2,-(SP)
14 016270 010346                MOV R3,-(SP)
15 016272 010446                MOV R4,-(SP)
16 016274 010546                MOV R5,-(SP)
17 016276 012102                MOV (R1)+,R2
18 016300 010604                MOV SP,R4
19 016302 062704 000012        ADD #10.,R4
20 016306                PUSH R1
21 016306 010146                CALL OSTRNG
22 016310 004737 014770        POP <R0,R5,R4,R3,R2,R1>
23 016314                :GET ADDRESS OF STRING
24 016314 012600                :COMPUTE ADDRESS OF ARGUMENTS
25 016316 012605                : WHICH ARE NOW ON STACK (IF ANY)
26 016320 012604                :SAVE RETURN ADDRESS
27 016322 012603                MOV R1,-(SP)
28 016324 012602                :PRINT THE FORMATTED MESSAGE
29 016326 012601                :RESTORE ALL REGISTERS
30 016330 062006                MOV (SP)+,R0
31 016332 000110                MOV (SP)+,R5
32                                MOV (SP)+,R4
33                                MOV (SP)+,R3
34                                MOV (SP)+,R2
35                                MOV (SP)+,R1
36                                ADD (R0)+,SP
37                                :ADJUST STACK POINTER OVER ARGUMENTS
38                                JMP @R0
39                                :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	016334			DIVIDE: PUSH R2	
	016334	010246			
17	016336	012702	000040	MOV #32.,R2	MOV R2,-(SP)
18	016342	005005		CLR R5	:SET UP SHIFT COUNT
19	016344	006303		1\$: ASL R3	:START WITH ZERO REMAINDER
20	016346	006104		ROL R4	:SHIFT LEFT INTO R5
21	016350	006105		ROL R5	
22	016352	020005		CMP R0,R5	:WILL DIVISOR GO INTO REMAINDER
23	016354	101002		BHI 2\$	:ONLY SUBTRACT IF IT WILL
24	016356	160005		SUB R0,R5	:SUBTRACT DIVISOR
25	016360	005203		INC R3	:PUT A ONE INTO QUOTIENT
26	016362	005302		2\$: DEC R2	:COUNT THE SHIFTS
27	016364	001367		BNE 1\$	
28	016366			POP R2	
	016366	012602			MOV (SP)+,R2
29	016370	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
14 016372 013701 002164 000044 LOADDM: MOV DMPROG,R1      ;GET STORAGE ADDRESS OF DM PROGRAM
15 016376 116165 000021      MOV B DMTMO(R1),C.TOT(R5) ;GET TIMEOUT VALUE
16 016404 105065 000045      CLRB C.TOT+1(R5)
17 016410 016504 000004      MOV C.VEC(R5),R4      ;GET VECTOR OF UDA
18 016414      AND CT.VEC,R4
19 016420 010501 177000      MOV R5,R1      ;GET INTERRUPT SERVICE LINK
20 016422 062701 000010      ADD #C.JSR,R1
21 016426      SETVEC R4,R1,#PRI07      ;SET UP INTERRUPT VECTOR
22 016432 012746 000340      MOV #PRI07,-(SP)
23 016434 010146      MOV R1,-(SP)
24 016436 010446      MOV R4,-(SP)
25 016438 012746 000003      MOV #3,-(SP)
26 016442 104437      TRAP C$SVEC
27 016444 062706 000010      ADD #10,SP
28
29      ;INITIALIZE UDA WITH SMALLEST
30      ;RING BUFFER AND INTERRUPTS ENABLED
31      ;BRANCH IF AN ERROR
32
33      CALL UDASINT
34      BEQ LOADER

```

1	016456	017701	163502		MOV @DMPROG,R1	:GET SIZE OF PROGRAM
2	016462	012700	000002		MOV #OP.ESP,R0	:BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET
3	016466	004737	016574		CALL BLD CMD	
4	016472	013764	002164	000124	MOV DMPROG,HC.CPK+P.UADR(R4)	:LOAD MAIN PROGRAM ADDRESS
5	016500	010164	000120		MOV R1,HC.CPK+P.BCNT(R4)	: AND SIZE
6	016504	013764	002164	000140	MOV DMPROG,HC.CPK+P.OVRL(R4)	:LOAD OVERLAY ADDRESS
7	016512	067764	163446	000140	ADD @DMPROG,HC.CPK+P.OVRL(R4)	
8	016520	004737	016660		CALL SND CMD	:SEND COMMAND TO UDA
9	016524	004737	017000		CALL WAITMS	:WAIT FOR MESSAGE RESPONSE
10	016530	001417			BEQ LOADER	:ABORT IF NO RESPONSE
11	016532	032764	000037	000032	BIT #ST.MSK,HC.MPK+P.STS(R4)	:CHECK FOR ERRORS
12	016540	001007			BNE LOADE1	
13	016542	042765	000024	000014	BIC #CT.CMD+CT.REQ,C.FLG(R5)	:CLEAR COMMAND OUTSTANDING FLAG
14	016550	052765	000002	000014	BIS #CT.RN,C.FLG(R5)	:SET DM PROGRAM RUNNING FLAG
15	016556	000207			RETURN	

```
1      ;UDA FAILED TO DOWNLINE LOAD DM PROGRAM
2
3 016560      104455
   016560      000042
   016562      000000
   016564      000000
   016566      011456
4 016570      000264
5 016572      000207

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

LOADER: SEZ                      ;SET Z TO INDICATE ERROR OCCURRED
        RETURN
```

1				:BLDCMD	
2				:BUILD A COMMAND IN COMMAND PACKET	
3				:INPUTS:	
4				R5 - CONTROLLER TABLE ADDRESS	
5				R0 - COMMAND CODE	
6				:OUTPUTS:	
7				R4 - ADDRESS OF HOST COMM AREA	
8				COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL OTHER FIELDS CLEARED.	
9				CMD REFERENCE NUMBER IN CONTROLLER TABLE INCREMENTED AND RESULT	
10				IN COMMAND PACKET.	
11				R0 - CONTENTS DESTROYED	
12					
13					
14					
15	016574			BLDCMD: PUSH <R1,R0>	
	016574	010146			MOV R1,-(SP)
	016576	010046			MOV R0,-(SP)
16	016600	016504	000016	MOV C.RING(R5),R4	:GET ADDRESS OF HOST COMM AREA
17	016604	010400		MOV R4,R0	:COPY TO R0
18	016606	062700	000100	ADD #HC.CEV,R0	:COMPUTE ADDRESS OF COMMAND ENVELOPE
19	016612	012720	000060	MOV #HC.PSZ,(R0)+	:LOAD PACKET LENGTH
20	016616	012701	001000	MOV #DUP,R1	:LOAD DIAG CIRCUIT IDENTIFIER
21	016622	022716	000031	CMP #OP.MWR,(SP)	:IF CODE IS MAINTENANCE WRITE
22	016626	001002		BNE BLDC0	: GET OTHER CIRCUIT IDENTIFIER
23	016630	012701	177777	MOV #DIAG,R1	
24	016634	010120		BLDC0: MOV R1,(R0)+	:PUT IDENTIFIER INTO PACKET
25	016636	012701	000030	MOV #<HC.PSZ>/2,R1	:GET WORDS TO CLEAR
26	016642	005020		BLDC1: CLR (R0)+	:CLEAR PACKET
27	016644	005301		DEC R1	
28	016646	001375		BNE 3LDC1	
29	016650			POP #C.CPK+P.OPCD(R4)	:PUT OPCODE IN PACKET
	016650	012664	000114		MOV (SP)+,HC.CPK+P.OPCD(R4)
30	016654			POP R1	:RESTORE R1
	016654	012601			MOV (SP)+,R1
31	016656	000207		RETURN	

```

1      ;SND CMD
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 016660      SND CMD: PUSH <R0,R1>
15 016660      MOV R0,-(SP)
16 016662      MOV R1,-(SP)
17 016664      MOV C.RING(R5),R4      ;LOAD R4 WITH HOST COMM AREA ADDRESS
18 016670      INC C.REF(R5)          ;INCREMENT CMD REFERENCE NUMBER
19 016674      MOV C.REF(R5),HC.CPK+P.CRF(R4) ;PUT IN PACKET
20 016702      MOV #RG.OWN+RG.FLG,HC.MCT(R4) ;MARK MESSAGE PACKET AVAILABLE
21 016710      MOV #RG.OWN,HC.CCT(R4)      ;MARK COMMAND TO UDA
22 016716      TST @R5                ;TELL UDA COMMAND IS THERE
23 016722      BIS #CT.CMD,C.FLG(R5)      ;MARK COMMAND ISSUED
24 016730      POP <R1,R0>
25 016732      MOV (SP)+,R1
26 016734      MOV (SP)+,R0
27
28      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     ;
11     ;OUTPUTS:
12     ;      DATA BUFFER CLEARED
13     ;      COMMAND PACKET POINTING TO BUFFER
14     ;      BYTE COUNT SET TO SIZE OF BUFFER
15     ;      R4 - ADDRESS OF DATA BUFFER
16     CLRBUF: PUSH <R0,R1>
17     ;
18     ;
19     ;
20     ;
21     ;
22     ;
23     ;
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   ;
1001   ;
1002   ;
1003   ;
1004   ;
1005   ;
1006   ;
1007   ;
1008   ;
1009   ;
1010   ;
1011   ;
1012   ;
1013   ;
1014   ;
1015   ;
1016   ;
1017   ;
1018   ;
1019   ;
1020   ;
1021   ;
1022   ;
1023   ;
1024   ;
1025   ;
1026   ;
1027   ;
1028   ;
1029   ;
1030   ;
1031   ;
1032   ;
1033   ;
1034   ;
1035   ;
1036   ;
1037   ;
1038   ;
1039   ;
1040   ;
1041   ;
1042   ;
1043   ;
1044   ;
1045   ;
1046   ;
1047   ;
1048   ;
1049   ;
1050   ;
1051   ;
1052   ;
1053   ;
1054   ;
1055   ;
1056   ;
1057   ;
1058   ;
1059   ;
1060   ;
1061   ;
1062   ;
1063   ;
1064   ;
1065   ;
1066   ;
1067   ;
1068   ;
1069   ;
1070   ;
1071   ;
1072   ;
1073   ;
1074   ;
1075   ;
1076   ;
1077   ;
1078   ;
1079   ;
1080   ;
1081   ;
1082   ;
1083   ;
1084   ;
1085   ;
1086   ;
1087   ;
1088   ;
1089   ;
1090   ;
1091   ;
1092   ;
1093   ;
1094   ;
1095   ;
1096   ;
1097   ;
1098   ;
1099   ;
1100   ;
1101   ;
1102   ;
1103   ;
1104   ;
1105   ;
1106   ;
1107   ;
1108   ;
1109   ;
1110   ;
1111   ;
1112   ;
1113   ;
1114   ;
1115   ;
1116   ;
1117   ;
1118   ;
1119   ;
1120   ;
1121   ;
1122   ;
1123   ;
1124   ;
1125   ;
1126   ;
1127   ;
1128   ;
1129   ;
1130   ;
1131   ;
1132   ;
1133   ;
1134   ;
1135   ;
1136   ;
1137   ;
1138   ;
1139   ;
1140   ;
1141   ;
1142   ;
1143   ;
1144   ;
1145   ;
1146   ;
1147   ;
1148   ;
1149   ;
1150   ;
1151   ;
1152   ;
1153   ;
1154   ;
1155   ;
1156   ;
1157   ;
1158   ;
1159   ;
1160   ;
1161   ;
1162   ;
1163   ;
1164   ;
1165   ;
1166   ;
1167   ;
1168   ;
1169   ;
1170   ;
1171   ;
1172   ;
1173   ;
1174   ;
1175   ;
1176   ;
1177   ;
1178   ;
1179   ;
1180   ;
1181   ;
1182   ;
1183   ;
1184   ;
1185   ;
1186   ;
1187   ;
1188   ;
1189   ;
1190   ;
1191   ;
1192   ;
1193   ;
1194   ;
1195   ;
1196   ;
1197   ;
1198   ;
1199   ;
1200   ;
1201   ;
1202   ;
1203   ;
1204   ;
1205   ;
1206   ;
1207   ;
1208   ;
1209   ;
1210   ;
1211   ;
1212   ;
1213   ;
1214   ;
1215   ;
1216   ;
1217   ;
1218   ;
1219   ;
1220   ;
1221   ;
1222   ;
1223   ;
1224   ;
1225   ;
1226   ;
1227   ;
1228   ;
1229   ;
1230   ;
1231   ;
1232   ;
1233   ;
1234   ;
1235   ;
1236   ;
1237   ;
1238   ;
1239   ;
1240   ;
1241   ;
1242   ;
1243   ;
1244   ;
1245   ;
1246   ;
1247   ;
1248   ;
1249   ;
1250   ;
1251   ;
1252   ;
1253   ;
1254   ;
1255   ;
1256   ;
1257   ;
1258   ;
1259   ;
1260   ;
1261   ;
1262   ;
1263   ;
1264   ;
1265   ;
1266   ;
1267   ;
1268   ;
1269   ;
1270   ;
1271   ;
1272   ;
1273   ;
1274   ;
1275   ;
1276   ;
1277   ;
1278   ;
1279   ;
1280   ;
1281   ;
1282   ;
1283   ;
1284   ;
1285   ;
1286   ;
1287   ;
1288   ;
1289   ;
1290   ;
1291   ;
1292   ;
1293   ;
1294   ;
1295   ;
1296   ;
1297   ;
1298   ;
1299   ;
1300   ;
1301   ;
1302   ;
1303   ;
1304   ;
1305   ;
1306   ;
1307   ;
1308   ;
1309   ;
1310   ;
1311   ;
1312   ;
1313   ;
1314   ;
1315   ;
1316   ;
1317   ;
1318   ;
1319   ;
1320   ;
1321   ;
1322   ;
1323   ;
1324   ;
1325   ;
1326   ;
1327   ;
1328   ;
1329   ;
1330   ;
1331   ;
1332   ;
1333   ;
1334   ;
1335   ;
1336   ;
1337   ;
1338   ;
1339   ;
1340   ;
1341   ;
1342   ;
1343   ;
1344   ;
1345   ;
1346   ;
1347   ;
1348   ;
1349   ;
1350   ;
1351   ;
1352   ;
1353   ;
1354   ;
1355   ;
1356   ;
1357   ;
1358   ;
1359   ;
1360   ;
1361   ;
1362   ;
1363   ;
1364   ;
1365   ;
1366   ;
1367   ;
1368   ;
1369   ;
1370   ;
1371   ;
1372   ;
1373   ;
1374   ;
1375   ;
1376   ;
1377   ;
1378   ;
1379   ;
1380   ;
1381   ;
1382   ;
1383   ;
1384   ;
1385   ;
1386   ;
1387   ;
1388   ;
1389   ;
1390   ;
1391   ;
1392   ;
1393   ;
1394   ;
1395   ;
1396   ;
1397   ;
1398   ;
1399   ;
1400   ;
1401   ;
1402   ;
1403   ;
1404   ;
1405   ;
1406   ;
1407   ;
1408   ;
1409   ;
1410   ;
1411   ;
1412   ;
1413   ;
1414   ;
1415   ;
1416   ;
1417   ;
1418   ;
1419   ;
1420   ;
1421   ;
1422   ;
1423   ;
1424   ;
1425   ;
1426   ;
1427   ;
1428   ;
1429   ;
1430   ;
1431   ;
1432   ;
1433   ;
1434   ;
1435   ;
1436   ;
1437   ;
1438   ;
1439   ;
1440   ;
1441   ;
1442   ;
1443   ;
1444   ;
1445   ;
1446   ;
1447   ;
1448   ;
1449   ;
1450   ;
1451   ;
1452   ;
1453   ;
1454   ;
1455   ;
1456   ;
1457   ;
1458   ;
1459   ;
1460   ;
1461   ;
1462   ;
1463   ;
1464   ;
1465   ;
1466   ;
1467   ;
1468   ;
1469   ;
1470   ;
1471   ;
1472   ;
1473   ;
1474   ;
1475   ;
1476   ;
1477   ;
1478   ;
1479   ;
1480   ;
1481   ;
1482   ;
1483   ;
1484   ;
1485   ;
1486   ;
1487   ;
1488   ;
1489   ;
1490   ;
1491   ;
1492   ;
1493   ;
1494   ;
1495   ;
1496   ;
1497   ;
1498   ;
1499   ;
1500   ;
1501   ;
1502   ;
1503   ;
1504   ;
1505   ;
1506   ;
1507   ;
1508   ;
1509   ;
1510   ;
1511   ;
1512   ;
1513   ;
1514   ;
1515   ;
1516   ;
1517   ;
1518   ;
1519   ;
1520   ;
1521   ;
1522   ;
1523   ;
1524   ;
1525   ;
1526   ;
1527   ;
1528   ;
1529   ;
1530   ;
1531   ;
1532   ;
1533   ;
1534   ;
1535   ;
1536   ;
1537   ;
1538   ;
1539   ;
1540   ;
1541   ;
1542   ;
1543   ;
1544   ;
1545   ;
1546   ;
1547   ;
1548   ;
1549   ;
1550   ;
1551   ;
1552   ;
1553   ;
1554   ;
1555   ;
1556   ;
1557   ;
1558   ;
1559   ;
1560   ;
1561   ;
1562   ;
1563   ;
1564   ;
1565   ;
1566   ;
1567   ;
1568   ;
1569   ;
1570   ;
1571   ;
1572   ;
1573   ;
1574   ;
1575   ;
1576   ;
1577   ;
1578   ;
1579   ;
1580   ;
1581   ;
1582   ;
1583   ;
1584   ;
1585   ;
1586   ;
1587   ;
1588   ;
1589   ;
1590   ;
1591   ;
1592   ;
1593   ;
1594   ;
1595   ;
1596   ;
1597   ;
1598   ;
1599   ;
1600   ;
1601   ;
1602   ;
1603   ;
1604   ;
1605   ;
1606   ;
1607   ;
1608   ;
1609   ;
1610   ;
1611   ;
1612   ;
1613   ;
1614   ;
1615   ;
1616   ;
1617   ;
1618   ;
1619   ;
1620   ;
1621   ;
1622   ;
1623   ;
1624   ;
1625   ;
1626   ;
1627   ;
1628   ;
1629   ;
1630   ;
1631   ;
1632   ;
1633   ;
1634   ;
1635   ;
1636   ;
1637   ;
1638   ;
1639   ;
1640   ;
1641   ;
1642   ;
1643   ;
1644   ;
1645   ;
1646   ;
1647   ;
1648   ;
1649   ;
1650   ;
1651   ;
1652   ;
1653   ;
1654   ;
1655   ;
1656   ;
1657   ;
1658   ;
1659   ;
1660   ;
1661   ;
1662   ;
1663   ;
1664   ;
1665   ;
1666   ;
1667   ;
1668   ;
1669   ;
1670   ;
1671   ;
1672   ;
1673   ;
1674   ;
1675   ;
1676   ;
1677   ;
1678   ;
1679   ;
1680   ;
1681   ;
1682   ;
1683   ;
1684   ;
1685   ;
1686   ;
1687   ;
1688   ;
1689   ;
1690   ;
1691   ;
1692   ;
1693   ;
1694   ;
1695   ;
1696   ;
1697   ;
1698   ;
1699   ;
1700   ;
1701   ;
1702   ;
1703   ;
1704   ;
1705   ;
1706   ;
1707   ;
1708   ;
1709   ;
1710   ;
1711   ;
1712   ;
1713   ;
1714   ;
1715   ;
1716   ;
1717   ;
1718   ;
1719   ;
1720   ;
1721   ;
1722   ;
1723   ;
1724   ;
1725   ;
1726   ;
1727   ;
1728   ;
1729   ;
1730   ;
1731   ;
1732   ;
1733   ;
1734   ;
1735   ;
1736   ;
1737   ;
1738   ;
1739   ;
1740   ;
1741   ;
1742   ;
1743   ;
1744   ;
1745   ;
1746   ;
1747   ;
1748   ;
1749   ;
1750   ;
1751   ;
1752   ;
1753   ;
1754   ;
1755   ;
1756   ;
1757   ;
1758   ;
1759   ;
1760   ;
1761   ;
1762   ;
1763   ;
1764   ;
1765   ;
1766   ;
1767   ;
1768   ;
1769   ;
1770   ;
1771   ;
1772   ;
1773   ;
1774   ;
1775   ;
1776   ;
1777   ;
1778   ;
1779   ;
1780   ;
1781   ;
1782   ;
1783   ;
1784   ;
1785   ;
1786   ;
1787   ;
1788   ;
1789   ;
1790   ;
1791   ;
1792   ;
1793   ;
1794   ;
1795   ;
1796   ;
1797   ;
1798   ;
1799   ;
1800   ;
1801   ;
1802   ;
1803   ;
1804   ;
1805   ;
1806   ;
1807   ;
1808   ;
1809   ;
1810   ;
1811   ;
1812   ;
1813   ;
1814   ;
1815   ;
1816   ;
1817   ;
1818   ;
1819   ;
1820   ;
1821   ;
1822   ;
1823   ;
1824   ;
1825   ;
1826   ;
1827   ;
1828   ;
1829   ;
1830   ;
1831   ;
1832   ;
1833   ;
1834   ;
1835   ;
1836   ;
1837   ;
1838   ;
1839   ;
1840   ;
1841   ;
1842   ;
1843   ;
1844   ;
1845   ;
1846   ;
1847   ;
1848   ;
1849   ;
1850   ;
1851   ;
1852   ;
1853   ;
1854   ;
1855   ;
1856   ;
1857   ;
1858   ;
1859   ;
1860   ;
1861   ;
1862   ;
1863   ;
1864   ;
1865   ;
1866   ;
1867   ;
1868   ;
1869   ;
1870   ;
1871   ;
1872   ;
1873   ;
1874   ;
1875   ;
1876   ;
1877   ;
1878   ;
1879   ;
1880   ;
1881   ;
1882   ;
1883   ;
1884   ;
1885   ;
1886   ;
1887   ;
1888   ;
1889   ;
1890   ;
18
```

```

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     WAITMS: PUSH <R0,R1>
12     017000      010046      000036      MOV #30,,R0      ;SET TIME OUT VALUE OF 30 SECONDS
13     017002      010146      MOV R5,R1      ;POINT TO TIME OUT COUNTER
14     017010      010501      ADD #C.TO,R1
15     017012      062701      000040      CALL SETTO
16     017016      004737      017172      MOV (R5),R0      ;GET ADDRESS OF UDAIP REGISTER
17     017022      011500      1$:      BIT #CT.MSG,C.FLG(R5)      ;LOOK IF INTERRUPT OCCURRED
18     017024      032765      000010 000014      BNE 3$      ;BRANCH IF SO
19     017032      001030      MOV 2(R0),R1      ;LOOK AT UDASA REGISTER
20     017034      016001      000002      BNE 4$      ;BRANCH IF ERROR CODE PRESENT
21     017040      001034      BREAK
22     017042      104422      TRAP      C$BRK
23     017044      005737      003232      TST KW.CSR      ;SEE IF A CLOCK ON SYSTEM
24     017050      001764      BEQ 1$
25     017052      023765      003244 000042      CMP KW.EL+2,C.TOH(R5)      ;CHECK IF TIMEOUT HAS HAPPENED
26     017060      101005      BHI 2$
27     017062      001357      BNE 1$
28     017064      023765      003242 000040      CMP KW.EL,C.TO(R5)
29     017072      103753      2$:      BLO 1$
30     017074      104455      ERRDF 36,,ERR036
31     017076      000044      TRAP      C$ERDF
32     017100      000000      .WORD 36
33     017102      011464      .WORD 0
34     017104      POP <R1,R0>      .WORD ERR036
35     017106      012601      MOV (SP)+,R1
36     017108      012600      MOV (SP)+,R0
37     017110      000264      SEZ
38     017112      000207      RETURN

```

```
1 017114 042765 00C010 000014 3$: BIC #CT.MSG,C.FLG(R5)
2 017122 012601 POP <R1,R0>
  017122 012600
  017124 012600
3 017126 000244 CLZ
4 017130 000207 RETURN
5 017132 4$: ERRDF 37,,ERR037
  017132 104455
  017134 000045
  017136 000000
  017140 011476
6 017142 POP <R1,R0>
  017142 012601
  017144 012600
7 017146 000264 SEZ
8 017150 000207 RETURN
```

;CLEAR MESSAGE RECEIVED FLAG

```
MOV (SP)+,R1
MOV (SP)+,R0
```

;GIVE NO ERROR RETURN

```
TRAP C$ERDF
.WORD 37
.WORD 0
.WORD ERR037
```

```
MOV (SP)+,R1
MOV (SP)+,R0
```

```
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     BGNSRV NXMI
11
12     017152 017152
13
14     017152 012737 177777 002222      MOV #-1,NXMAD
15
16     017160
17     017160
18     017160 000002      ENDSRV
19
20
21
22
23
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
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2
```

```

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 R0,UDASRV] INSTRUCTION FROM WITHIN
7      ;THE CONTROLLER TABLE. THE PC STORED IN R0 IS THE ADDRESS OF THE C.FLG
8      ;WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
9      ;OF R0 FOLLOWED BY THE INTERRUPTED PC AND PS.
10     ;
11     ;INPUTS:
12     ;      R0 - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
13     ;      STACK - SAVED CONTENTS OF R0
14     ;OUTPUTS:
15     ;      CT.CMD CLEARED AND CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
16     ;      R0 - RESTORED FROM STACK
17
18 017162 BGNSRV UDASRV
19 017162
20 017162 052710 000010      BIS #CT.MSG,(R0)      ;SET CT.MSG
21 017166 012600      POP R0      ;RESTORE R0
22 017170      ENDsrv
23 017170
24 017170 000002
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
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607

```

1				:SETTO	
2				:SET TIMEOUT COUNTER TO SOME NUMBER OF SECONDS FROM CURRENT TIME.	
3				:INPUTS:	
4				RO - NUMBER OF SECONDS FOR TIMEOUT	
5				R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT	
6				:OUTPUTS:	
7				RO - CONTENTS DESTROYED	
8				R1 - INCREMENTED BY 2	
9				:COMPUTE CLOCK TICKS TIL TIMEOUT	
10				SETTO: PUSH <R2,R3>	
11					
12					
13					
14	017172				
15	017172	010246			
16	017174	010346			
17	017176	005002			
18	017200	013703	003240	CLR R2	:CLEAR PRODUCT
19	017204	006200		MOV KW.HZ,R3	:GET MULTIPLICAND
20	017206	103001		ASR R0	:SHIFT MULTIPLIER TO RIGHT
21	017210	060302		BCC SET01	:IF A ONE BIT SHIFTED OUT
22	017212	006303		ADD R3,R2	: ADD MULTIPLICAND TO PRODUCT
23	017214	005700		ASL R3	:DOUBLE THE MULTIPLICAND
24	017216	001372		TST R0	
25				BNE SET00	:CONTINUE UNTIL MULTIPLIER IS ZERO
26					
27				:GET CURRENT TIME	
28	017220	013700	003242	SET02: MOV KW.EL,R0	:GET TIME
29	017224	013703	003244	MOV KW.EL+2,R3	
30	017230	020037	003242	CMP R0,KW.EL	:IF CHANGED DURING RETRIEVAL
31	017234	001371		BNE SET02	: GET IT AGAIN
32					
33				:ADD TIME TIL TIMEOUT	
34	017236	060200		ADD R2,R0	:ADD
35	017240	005503		ADC R3	
36					
37				:PUT RESULT IN STORAGE	
38	017242	010021		MOV R0,(R1)+	
39	017244	010311		MOV R3,(R1)	
40					
41	017246			POP <R3,R2>	
42	017246	012603			MOV (SP)+,R3
	017250	012602			MOV (SP)+,R2
	017252	000207		RETURN	

```

1      ;UDAINIT
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 HOST COMMUNICATION AREA WITH ALL ONES
19
20 017254 016502 000016  UDAINT: MOV C.RING(R5),R2                ;GET FIRST ADDRESS OF RING BUFFER
21 017260 012703 000006  MOV #<HC.RSZ*2+HC.ISZ>/2,R3          ;GET SIZE OF RING BUFFER
22 017264 012722 177777  UDAI1L: MOV #-1,(R2)+          ;WRITE ONES TO BUFFER
23 017270 005303          DEC R3                ;COUNT THE WORDS IN BUFFER
24 017272 003374          BGT UDAI1L            ;LOOP UNTIL ENTIRE BUFFER WRITTEN
25
26     ;DO THE INITIALIZATION
27
28 017274 004737 017470  CALL UDAIST                ;DO FIRST THREE STEPS
29 017300 103471          BCS UDAIEX              ;GET OUT IF UDA MICROCODE REPORTED FAILURE
30 017302 012364 000002  MOV (R3)+,2(R4)          ;WRITE NEXT WORD TO UDASA REGISTER
31 017306 012703 000310  MOV #200.,R3            ;GET TRY COUNTER
32 017312 016402 000002  UDAI1A: MOV 2(R4),R2      ;LOOK AT UDASA
33 017316 001410          BEQ UDAI1C
34 017320 100005          BPL UDAI1B
35 017322          ERRDF 24,,ERR024
36 017322 104455          TRAP C$ERDF
37 017324 000030          .WORD 24
38 017326 000000          .WORD 0
39 017330 011352          .WORD ERR024
40 017332 000454          BR UDAIEX
41 017334 005303  UDAI1B: DEC R3
42 017336 001365  UDAI1C: BNE UDAI1A
43 017340 010264 000002  MOV R2,2(R4)
44 017344 011402  MOV (R4),R2
45 017346 004737 020014  CALL UDARSP
46 017352 103444  BCS UDAIEX

```

```

;WRITE 0 TO UDASA (PURGE)
;READ FROM UDAIP (POLL)
;WAIT FOR STEP OR ERROR BIT
;GET OUT IF UDA MICROCODE REPORTED FAILURE

```

```

1      ;CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
2
3 017354 016502 000016      UDAI2:  MOV C.RING(R5),R2      ;GET FIRST ADDRESS OF RING BUFFER
4 017360 012703 000006      MOV #<HC.RSZ*2+HC.ISZ>/2,R3      ;GET SIZE OF RING BUFFER
5 017364 005722      UDAI2L: TST (R2)+      ;CHECK WORD IN BUFFER
6 017366 001003      BNE UDAI2E      ;GO TO ERROR REPORTER IF NOT ZERO
7 017370 005303      DEC R3      ;COUNT THE WORDS IN BUFFER
8 017372 003374      BGT UDAI2L      ;LOOP UNTIL ALL WORDS CHECKED
9 017374 000405      BR UDAI3
10
11 017376      UDAI2E: ERRDF 23,,ERR023      ;REPORT BUFFER NOT CLEARED
    017376 104455      TRAP C$ERDF
    017400 000027      .WORD 23
    017402 000000      .WORD 0
    017404 011266      .WORD ERR023
    017406 000426      BR UDAIEX
12
13
14      ;SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION
15
16 017410      UDAI3:
17 017410 016500 000006      MOV C.BST(R5),R0      ;GET BURST VALUE
18 017414 006300      ASL R0      ;SHIFT TO POSITION
19 017416 006300      ASL R0
20 017420 052700 000001      BIS #SA.GO,R0      ;SET THE GO BIT
21 017424 010064 000002      MOV R0,2(R4)      ;SEND TO UDA
22 017430 016501 000016      MOV C.RING(R5),R1
23 017434 010161 000004      MOV R1,HC.MSG(R1)
24 017440 062761 000020 000004      ADD #HC.MPK,HC.MSG(R1)
25 017446 010161 000010      MOV R1,HC.CMD(R1)
26 017452 062761 000104 000010      ADD #HC.CPK,HC.CMD(R1)
27 017460 000244      CLZ      ;CLEAR Z AS NO ERROR INDICATION
28 017462 000207      RETURN
29
30      ;ERROR RETURN
31
32 017464 000264      UDAIEX: SEZ
33 017466 000207      RETURN      ;SET Z TO INDICATE ERROR OCCURRED

```

```

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     017470      UDAIST: BREAK
13     017470      104422      TRAP      C$BRK
14     017472      PUSH R1
15     017472      010146      MOV R1,-(SP)
16     017474      016504      000004      MOV C.VEC(R5),R4
17     017500      AND CT.VEC,R4
18     017500      042704      177000      BIC #^C<CT.VEC>,R4
19     017504      006204      ASR R4
20     017506      006204      ASR R4
21     017510      052704      100000      BIS #SA.STP,R4
22     017514      010437      017706      MOV R4,UDAID1
23     017520      016537      000016      017712      MOV C.RING(R5),UDAID2
24     017526      062737      000004      017712      ADD #HC.MSG,UDAID2
25     ;SET STEP BIT IN DATA WORD
26     ;LOAD INTERRUPT VECTOR
27     ;LOAD MEMORY ADDRESS
28     ; OF FIRST RESPONSE RING
29
30     ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER
31
32     017534      016504      000000      MOV C.UADR(R5),R4
33     017540      005037      002222      ;GET ADDRESS OF UDAIP REGISTER
34     017544      CLR NXMAD      ;CLEAR MEMORY ERROR FLAG
35     017544      SETVEC #4,#NXMI,#PRI07      ;SET UP VECTOR 4
36
37     017544      012746      000340      MOV #PRI07,-(SP)
38     017550      012746      017152      MOV #NXMI,-(SP)
39     017554      012746      000004      MOV #4,-(SP)
40     017560      012746      000003      MOV #3,-(SP)
41     017564      104437      TRAP      C$SVEC
42     017566      062706      000010      ADD #10,SP
43
44     017572      005764      000002      TST 2(R4)
45     017576      005014      ;ACCESS UDASA REGISTER
46     017600      CLR (R4)      ;WRITE TO UDAIP
47     017600      012700      000004      CLRVEC #4
48     017604      104436      ;GIVE UP THE VECTOR
49     017606      005737      002222      MOV #4,R0
50     017612      001406      TRAP      C$CVEC
51     017614      TST NXMAD      ;SEE IF A MEMORY ERROR OCCURRED
52     017616      BEQ UDAISG
53     017614      ERRDF 20,,ERR020
54     017616      104455      TRAP      C$ERDF
55     017616      000024      .WORD    20
56     017620      000000      .WORD    0
57     017622      011154      .WORD    ERR020
58
59     017624      000261      SEC
60     017626      000424      BR UDAISE

```

```

1          ;SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
2
3 017630 012737 004000 020152 UDAISG: MOV #SA.S1,UDARSD          ;STORE RESPONSE MASK
4 017636 012703 017704          MOV #UDAIDT,R3                ;AND INDEX TO TABLE
5
6          ;WAIT FOR AND CHECK RESPONSE DATA
7
8 017642 004737 020014          UDAISL: CALL UDARSP            ;WAIT FOR STEP OR ERROR BITS
9 017646 103414                  BCS UDAISE                    ;EXIT IF ERROR
10 017650 004733                  CALL @ (R3)+                  ;CALL RESPONSE CHECKER FOR STEP
11 017652 103412                  BCS UDAISE                    ;GET OUT IF ERROR
12 017654 006337 020152          ASL UDARSD                    ;SHIFT TO NEXT STEP BIT
13 017660 032737 040000 020152  BIT #SA.S4,UDARSD            ;CHECK IF NOW AT STEP 4
14 017666 001003                  BNE UDAISX                    ;GET OUT IF SO
15 017670 012364 000002          MOV (R3)+,2(R4)                ;WRITE DATA TO UDASA REGISTER
16 017674 000762                  BR UDAISL                    ;STAY IN LOOP
17
18 017676 000241          UDAISX: CLC                          ;CLEAR CARRY FOR NO ERROR INDICATION
19 017700          UDAISE: POP R1
20 017702 012601          MOV (SP)+,R1
    000207          RETURN

```

```

1      ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION
2
3 017704 017722      UDAIDT: .WORD UDAIR1      ;FIRST WORD RESPONSE CHECK ROUTINE
4 017706 000000      UDAID1: .WORD 0          ;FIRST WORD TO SEND TO UDASA
5 017710 017730      ;.WORD UDAIR2          ;SECOND WORD RESPONSE CHECK ROUTINE
6 017712 000000      UDAID2: .WORD 0          ;SECOND WORD TO SEND TO UDASA
7 017714 017750      ;.WORD UDAIR3          ;THIRD WORD RESPONSE CHECK ROUTINE
8 017716 100000      UDAID3: .WORD SA.TST      ;THIRD WORD TO SEND TO UDASA
9 017720 017766      ;.WORD UDAIR4          ;FOURTH WORD RESPONSE CHECK ROUTINE
10
11      ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
12      ;CHECK FOR PROPER CONTROLLER TYPE
13
14 017722 012701 004400      UDAIR1: MOV #SA.S1+SA.D1,R1      ;SET STEP ONE BIT
15 017726 000422      BR UDAIRC      ;NOW COMPARE
16
17      ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
18      ;CHECK FOR ECHO OF INTI AND VECTOR
19
20 017730 013701 017706      UDAIR2: MOV UDAID1,R1      ;GET WORD SENT TO UDASA
21 017734 000301      SWAB R1      ;GET HIGH 8 BITS
22 017736 042701 177400      BIC #177400,R1
23 017742 052701 010000      BIS #SA.S2,R1      ;SET STEP 2 BIT
24 017746 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 017750 013701 017706      UDAIR3: MOV UDAID1,R1      ;GET WORD SENT TO UDASA
30 017754 042701 177400      BIC #177400,R1      ;JUST LOW 8 BITS
31 017760 052701 020000      BIS #SA.S3,R1      ;SET STEP 3 BIT
32 017764 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 017766 010201      UDAIR4: MOV R2,R1      ;GET RESPONSE FROM UDA
38 017770 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 017774 020102      UDAIRC: CMP R1,R2      ;COMPARE THE DATA
43 017776 001405      BEQ UDAIRX      ;EXIT IF COMPARED CORRECTLY
44 020000      ERRDF 25,,ERR025      ;REPORT ERROR
45 020000 104455      TRAP C$ERDF
46 020002 000031      .WORD 25
47 020004 000000      .WORD 0
48 020006 011366      .WORD ERR025
49 020010 000261
50 020012 000207
51      SEC
52      UDAIRX: RETURN

```

```

1      :UDARSP
2      :
3      :WAIT FOR UDA TO RESPOND WITH DATA IN UDASA REGISTER.
4      :EITHER STEP BIT FROM MASK IN LOCATION UDARSD OR ERROR BIT
5      :WILL CAUSE A TERMINATION.
6      :AN ERROR MESSAGE WILL BE PRINTED IF THE UDA DOES NOT RESPOND
7      :IN 10 SECONDS OR IF ERROR SETS.
8      :
9      :INPUTS:
10     :   UDASRD - MASK OF STEP BIT TO LOOK FOR
11     :   R5 - ADDRESS OF CONTROLLER TABLE
12     :   R4 - ADDRESS OF UDAIP REGISTER
13     :
14     :OUTPUTS:
15     :   ERROR MESSAGE IF TIME OUT ON RESPONSE OR ERROR BIT SETS
16     :   R2 - DATA FROM UDASA REGISTER
17     :   CARRY SET IF ERROR BIT SETS OR TIME OUT
18     UDARSP: PUSH R1
19     020014 010146 100000 020152      BIS #SA.ERR,UDARSD      ;SET ERROR BIT IN MASK WORD
20     020016 052737 000012      MOV #10.,R0      ;SET UP FOR 10 SECOND TIMEOUT
21     020024 012700 000012      MOV R5,R1      ;POINT TO COUNTER IN CONTROLLER TABLE
22     020030 010501      ADD #C.TO,R1
23     020032 062701 000040      CALL SETTO
24     020036 004737 017172      POP R1
25     020042 012601      MOV (SP)+,R1
26     020044 033764 020152 000002 UDARS1: BIT UDARSD,2(R4)      ;LOOK AT ERROR AND STEP BIT
27     020052 001024      BNE UDARS2      ;BRANCH IF EITHER SET
28     020054 104422      BREAK
29     020056 005737 003232      TST KW.CSR      TRAP C$BRK      ;SEE IF CLOCK ON SYSTEM
30     020062 001770      BEQ UDARS1
31     020064 023765 003244 000042      CMP KW.EL+2,C.TO(R5)      ;CHECK IF TIME OUT OCCURRED
32     020072 101005      BHI 1$
33     020074 001363      BNE UDARS1
34     020076 023765 003242 000040      CMP KW.EL,C.TO(R5)
35     020104 103757      BLO UDARS1
36     020106 016402 000002      1$: MOV 2(R4),R2      ;GET REGISTER CONTENTS
37     020112 000000      ERRDF 22,,ERR022      ;REPORT TIME OUT ERROR
38     020114 104455      TRAP C$ENDF
39     020116 000026      .WORD 22
40     020118 000000      .WORD 0
41     020120 011220      .WORD ERR022
42     020122 000407      BR UDARSE

```

```
1          ;CHECK IF ERROR BIT SET
2
3 020124 016402 000002      UDARS2: MOV 2(R4),R2
4 020130 100006              BPL UDARSX
5 020132                      ERRDF 21,,ERR021
6 020132 104455
7 020134 000025
8 020136 000000
9 020140 011166
10 020142 000261
11 020144 000207      UDARSE: SEC
12                                RETURN
13
14          ;NORMAL EXIT
15
16 020146 000241      UDARSX: CLC
17 020150 000207      UDARSX: RETURN
18
19          ;LOCATION FOR STEP BIT MASK
20
21 020152 000000      UDARSD: .WORD 0
22
23          ;GET REGISTER CONTENTS
24          ;EXIT IF ERROR NOT SET
25          ;REPORT ERROR INFO
26
27          TRAP      C$ERDF
28          .WORD    21
29          .WORD    0
30          .WORD    ERR021
31
32          ;CLEAR CARRY AS NO ERROR INDICATION
33
34          ;LOAD BY CALLING ROUTINE
```

```
1      ;CLOG
2      ;
3      ;COMPUTE LOGARITHMIC VALUE OF NUMBER TO BASE 2.
4      ;
5      ;INPUTS:
6      ;      R0 - LOGARITHM TO BE CONVERTED
7      ;OUTPUTS:
8      ;      R1 - VALUE OF 2 RAISED TO POWER OF INPUT NUMBER
9      ;
10     ;CLOG:  PUSH R0
11            CLR R1
12            SEC
13            ;SET UP ZERO START VALUE
14            ;WITH CARRY READY TO SHIFT IN
15            ;SHIFT TO LEFT
16            ; UNTIL R0
17            ; GOES NEGATIVE
            ROL R1
            DEC R0
            BPL CLOGLP
            POP R0
            RETURN
```

```
1      ;KW11I
2      ;
3      ;CLOCK INTERRUPT SERVICE ROUTINE
4
5 020154 BGNSRV KW11I
6 020154 062737 000001 003242      ADI -1,KW.EL      KW11I::
7 020162 005537 003244      ADC KW.EL+2      ;COUNT THE INTERRUPT
8 020166 012777 000105 163036      MOV #KWOUT.,@KW.CSR      ;RESTART THE CLOCK
9 020174      ENDSRV
   020174
   020174 000002      L10035:
                        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
14 020176 005737 003232      RNTIME: TST KW.CSR          ;CHECK IF A CLOCK PRESENT
15 020202 001465              BEQ RNTIMX          ;BRANCH IF NOT
16 020204 010046              PUSH <R0,R3,R4,R5>
17 020206 010346              MOV R0,-(SP)
18 020210 010446              MOV R3,-(SP)
19 020212 010546              MOV R4,-(SP)
20 020214 013703 003242      MOV KW.EL,R3          ;GET ELAPSED TIME
21 020220 013704 003244      MOV KW.EL+2,R4
22 020224 013700 003240      MOV KW.HZ,R0          ;GET SPEED OF CLOCK
23 020230 004737 016334      CALL DIVIDE          ;COMPUTE SECONDS OF ELAPSED TIME
24 020234 012700 000074      MOV #60,R0          ;NOW DIVIDE BY 60
25 020240 004737 016334      CALL DIVIDE          ;TO COMPUTE MINUTES
26 020244 010546              PUSH R5            ;SAVE REMAINDER AS SECONDS
27 020246 004737 016334      CALL DIVIDE          ;DIVIDE BY 60 AGAIN
28 020252 010346              PNT RNTIM,R3        ;PRINT HOURS
29 020254 004137 016266      MOV R3,-(SP)
30 020260 003672              JSR R1,LPNT
31 020262 000002              .WORD RNTIM
32 020264 020527 000011      .WORD PNT.CT
33 020270 003004              CMP R5,#9.          ;IF MINUTES 9 OR LESS
34 020272 112700 000060      BGT 1$              ;PRINT A LEADING ZERO
35 020276 004737 016056      PRINT #'0
36 020278 000000
37 020280 000000
38 020282 000000
39 020284 000000
40 020286 000000
41 020288 000000
42 020290 000000
43 020292 000000
44 020294 000000
45 020296 000000
46 020298 000000
47 020300 000000
48 020302 000000
49 020304 000000
50 020306 000000
51 020308 000000
52 020310 000000
53 020312 000000
54 020314 012605              POP R5              ;GET SECONDS
55 020316 020527 000011      CMP R5,#9.          ;IF 9 OR LESS
56 020320 003004              BGT 2$              ;PRINT A LEADING ZERO
57 020322 112700 000060      PRINT #'0
58 020324 004737 016056      POP <R5,R4,R3,R0>
59 020326 000000
60 020328 000000
61 020330 000000
62 020332 000000
63 020334 000000
64 020336 000000
65 020338 000000
66 020340 000000
67 020342 000000
68 020344 000000
69 020346 012605

```

020350	012604				MOV (SP)+,R4
020352	012603				MOV (SP)+,R3
020354	012600				MOV (SP)+,R0
35 020356			RNTIMX: PRINT '<' >	;PRINT A SPACE	
020356	112700	000040			MOVB #' ,R0
020362	004737	016056			CALL CPNT
36 020366	000207		RETURN		
37					
38 020370			ENDMOD		

```
1          .SBTTL  PROTECTION TABLE
2
3 020370      BGNMOD
4
5          :++
6          : THIS TABLE IS USED BY THE RUNTIME SERVICES
7          : TO PROTECT THE LOAD MEDIA.
8          :--
9
10 020370      BGNPROT
11 020370
12 020370 177777      -1          ;OFFSET INTO P-TABLE FOR CSR ADDRESS
13 020372 177777      -1          ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
14 020374 177777      -1          ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
15
16 020376      ENDPROT
17
```

.SBTTL INITIALIZE SECTION

++
THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
AT THE BEGINNING OF EACH PASS.
--

1							
2							
3							
4							
5							
6							
7							
8	020376			BGNINIT			
9	020376				LSINIT::		
10	020376	012700	000040	READEF #EF.START	;CHECK IF STARTED BY OPERATOR		
	020402	104447				MOV	#EF.START,RO
						TRAP	C\$REFG
11	020404			BCOMPLETE INIT0	; IF NOT,		
	020404	103415				BCS	INIT0
12	020406			READEF #EF.RESTART			
	020406	012700	000037			MOV	#EF.RESTART,RO
	020412	104447				TRAP	C\$REFG
13	020414			BCOMPLETE INIT1			
	020414	103415				BCS	INIT1
14	020416			READEF #EF.CONTINUE			
	020416	012700	000036			MOV	#EF.CONTINUE,RO
	020422	104447				TRAP	C\$REFG
15	020424			BCOMPLETE INIT1			
	020424	103411				BCS	INIT1
16	020426			READEF #EF.PWR			
	020426	012700	000034			MOV	#EF.PWR,RO
	020432	104447				TRAP	C\$REFG
17	020434			BCOMPLETE INIT1			
	020434	103405				BCS	INIT1
18	020436			INITQT: DOCLN	; ABORT PROGRAM		
	020436	104444				TRAP	C\$DCLN
19	020440			INIT0:			
21	020440	004737	012500	CALL CLOSEF	; MAKE SURE FILE IS CLOSED		
22	020444	005037	002214	CLR FNUM	; NO FILE READ		
24	020450	012700	000003	INIT1: MOV #SO.FMT,RO	;BUILD MODE WORD FROM SOFTWARE QUESTIONS		
25	020454	030037	002144	BIT RO,SFPTBL	;SEE IF REFORMAT		
26	020460	001011		BNE INIT2	; BRANCH IF SO		
27	020462	012700	000004	MOV #SO.CNS,RO	;SEE IF RECONSTRUCT		
28	020466	030037	002144	BIT RO,SFPTBL			
29	020472	001004		BNE INIT2	; BRANCH IF SO		
30	020474	006300		ASL RO	;SEE IF RESTORE		
31	020476			ASSUME SO.STR EQ SO.CNS*2			
32	020476	030037	002144	BIT RO,SFPTBL			
33	020502	001755		BEQ INITQT	;IF NOT, ABORT PROGRAM		
34	020504	010037	003230	INIT2: MOV RO,MODE	;SAVE MODE FLAGS		
35	020510	005737	003344	TST DATED	;SEE IF ALREADY ASKED FOR DATE		
36	020514	001002		BNE INIT3			
37	020516	004737	021430	CALL DATE	;IF NOT, GET IT NOW		
38	020522	004737	012064	INIT3: CALL RESET	;RESET ALL UNITS		
39	020526			MEMORY FFEE	;RESET START OF FREE MEMORY		
	020526	104431				TRAP	C\$MEM
	020530	010037	002146			MOV	RO,FFEE
40	020534	017737	161406	MOV @FFEE,FSIZE	;RESET SIZE OF FREE MEMORY		
41							
42				;INITIALIZE CLOCK			
43							

44		000105		KWOUT.=105		;DATA TO SEND TO KW11 TO START CLOCK
45	020542	005037	003232	CLR KW.CSR		;MARK CLOCK AS NOT ON SYSTEM
46	020546	005037	003242	CLR KW.EL		;CLEAR ELAPSED TIME
47	020552	005037	003244	CLR KW.EL+2		
48	020556			CLOCK L,R0		;SEE IF AN L CLOCK PRESENT
	020556	012700	000114			MOV #L,R0
	020562	104462				TRAP C\$CLCK
49	020564			BCOMplete KYES		BCS KYES
	020564	103413				
50	020566			CLOCK P,R0		;SEE IF A P CLOCK PRESENT
	020566	012700	000120			MOV #P,R0
	020572	104462				TRAP C\$CLCK
51	020574			BCOMplete KYES		BCS KYES
	020574	103407				
52	020576	005037	003232	CLR KW.CSR		;IF NEITHER, CLEAR CSR STORAGE WORD
53	020602			PNTF NOCLOCK		
	020602	004137	016230			JSR R1,LPNTF
	020606	004061				.WORD NOCLOCK
	020610	000000				.WORD PNT.CT
54	020612	000426		BR KNO		
55	020614	012037	003232	KYES: MOV (R0)+,KW.CSR		;STORE DATA RETURNED
56	020620	012037	003234	MOV (R0)+,KW.BRL		
57	020624	012037	003236	MOV (R0)+,KW.VEC		
58	020630	012037	003240	MOV (R0)+,KW.HZ		
59	020634			SETVEC KW.VEC,#KW111,KW.BRL		;SET THE VECTOR
	020634	013746	003234			MOV KW.BRL,-(SP)
	020640	012746	020154			MOV #KW111,-(SP)
	020644	013746	003236			MOV KW.VEC,-(SP)
	020650	012746	000003			MOV #3,-(SP)
	020654	104437				TRAP C\$SVEC
	020656	062706	000010			ADD #10,SP
60	020662	012777	000105	162342 KNO: MOV #KWOUT.,KW.CSR		;START THE CLOCK
61	020670					

```

1
2
3 020670 013737 002146 002156      MOV FFREE,CTABS      ;STORE START OF CONTROLLER TABLES
4 020676 005077 161254              CLR @CTABS          ;ZEROS MARKS END CONTROLLER TABLES
5 020702 005037 002160              CLR CTRLRS         ;CLEAR CONTROLLER COUNT
6 020706 012701 003474              MOV #IPADRS,R1        ; R1 -> IP ADDRESS
7 020712 012702 000010              MOV #8,R2             ; R2 IS A COUNTER
8 020716 005021 1$:                 CLR (R1)+            ; CLEAR ENTRY
9 020720 005302                      DEC R2                ; DONE?
10 020722 001375                     BNE 1$                ; IF NOT, BRANCH

11
12
13
14 020724 005002                      CLR R2              ;LOGICAL UNIT NUMBER IN R2
15 020726 010200                      INIT4: GPBARD R2,R0      ;GET POINTER TO A P-TABLE
16 020732 104442                      BNCOMPLETE NXTTAB      ;IGNORE IF NO TABLE RETURNED
17 020732 103110                      BCC NXTTAB
18
19
20 020734 013703 002156              MOV CTABS,R3          ;GET ADDRESS OF CONTROLLER TABLES
21 020740 005713                      INIT5: TST (R3)        ;CHECK IF ANY MORE TABLES
22 020742 001416                      BEQ NEWTAB          ;BUILD NEW TABLE IF FOUND ZERO WORD
23 020744 021013                      CMP (R0),(R3)        ;CHECK IF SAME UNIBUS ADDRESS
24 020746                      ASSUME C.UADR EQ 0
25 020746                      ASSUME HO.UBA EQ 0
26 020746 001463                      BEQ SAMTAB          ;CHECK TABLE IF ALREADY EXISTS
27 020750 016301 000004              MOV C.VEC(R3),R1      ;GET VECTOR FROM EXISTING CONTROLLER TABLE
28 020754 042701 177000              BIC #^C<CT.VEC>,R1
29 020760 026001 000002              CMP HO.VEC(R0),R1
30 020764 001002                      BNE 1$              ;SEE IF DIFFERENT VECTOR
31 020766 000137 021414              JMP SAMVEC          ;ERROR, CAN'T HAVE TWO UDA'S WITH SAME VECTOR
32 020772 062703 000054              1$: ADD #C.SIZE,R3  ;MOVE TO NEXT TABLE
33 020776 000760                      BR INIT5

```

1					
2					
3	021000	012703	000010	NEWTAB: MOV	#8.,R3
4	021004	012704	003474		MOV #IPADRS,R4
5	021010	005714		1\$: TST	(R4)
6	021012	001404			BEQ 2\$
7	021014	005724			TST (R4)+
8	021016	005303			DEC R3
9	021020	001373			BNE 1\$
10	021022	000401			BR 3\$
11	021024	011014		2\$: MOV	(R0),(R4)
12	021026	012701	000026	3\$: MOV	#C.SIZE/2,R1
13	021032	004737	011744		CALL ALOCM
14	021036	011021			MOV (R0),(R1)+
15	021040	010221			MOV R2,(R1)+
16	021042	016004	000004		MOV HO.BRL(R0),R4
17	021046	000304			SWAB R4
18	021050	006104			ROL R4
19	021052	056004	000002		BIS HO.VEC(R0),R4
20	021056	010421			MOV R4,(R1)+
21	021060	016021	000006		MOV HO.BST(R0),(R1)+
22	021064	012721	004037		MOV #4037,(R1)+
23	021070	012721	017162		MOV #UDASRV,(R1)+
24	021074	012703	000020		MOV #16.,R3
25					
26	021100	005021		INIT7: CLR	(R1)+
27	021102	005303			DEC R3
28	021104	001375			BNE INIT7
29	021106	005237	002160		INC CTRLRS
30	021112	005011			CLR (R1)
31	021114	000417			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 POINTERS 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 021116 016004 000004      SAMTAB: MOV HO.BRL(R0),R4      ;GET BR LEVEL FROM P-TABLE
4 021122 000304              SWAB R4                      ;SWAP TO HIGH BYTE
5 021124 006104              ROL R4                        ;SHIFT ONE MORE TO LEFT
6 021126 056004 000002      BIS HO.VEC(R0),R4            ;ADD VECTOR ADDRESS
7 021132 020463 000004      CMP R4,C.VEC(R3)             ;COMPARE WITH CONTROLLER TABLE
8 021136 001004              BNE 1$
9 021140 026063 000006 000006  CMP HO.BST(R0),C.BST(R3)    ;COMPARE BURST RATES
10 021146 001402              BEQ NXTTAB
11 021150 000137 021344      1$: JMP CTABER                ;FATAL ERROR IF NOT SAME
12
13          ;GET NEXT P-TABLE
14
15 021154 005202      NXTTAB: INC R2                      ;INCREMENT LOGICAL UNIT NUMBER
16 021156 023702 002012      CMP L$UNIT,R2               ;CHECK IF GOT ALL TABLES
17 021162 003261              BGT INIT4                   ;IF NOT, GO BACK FOR NEXT
18
19 021164 012701 000001      MOV #1,R1                    ;ALLOCATE SPACE FOR ZERO END WORD
20 021170 004737 011744      CALL ALOCM                   ;AFTER CONTROLLER TABLES

```

			;NOW BUILD DRIVE TABLES		
1					
2					
3	021174	005002			;LOGICAL UNIT NUMBER IN R2
4	021176		INIT8:	CLR R2 GPHARD R2,R0	;GET POINTER TO A P-TABLE
	021176	010200			
	021200	104442			MOV R2,R0
5	021202				TRAP C\$GPHRD
	021202	103040	BNCOMPLETE INIT14		;IF NOT AVAILABLE, GO GET NEXT BCC INIT14
6					
7			;FIND CONTROLLER TABLE		
8					
9	021204	013703	002156	MOV CTABS,R3	;GET ADDRESS OF CONTROLLER TABLES
10	021210	021013		INIT10: CMP (R0),(R3)	;CHECK IF SAME UNIBUS ADDRESS
11	021212	001403		BEQ INIT11	;BRANCH IF TABLE FOUND
12	021214	062703	000054	ADD #C.SIZE,R3	;MOVE TO NEXT TABLE
13	021220	000773		BR INIT10	

1				:BUILD DRIVE TABLE	
2					
3	021222	012701	000015	INIT11: MOV #D.SIZE/2,R1	:GET SIZE OF DRIVE TABLE
4	021226	004737	011744	CALL ALOCM	:ALLOCATE SPACE FROM FREE MEMORY
5				:	
6				R0 POINTS TO P-TABLE	
7				R1 POINTS TO DRIVE TABLE	
8				R3 POINTS TO CONTROLLER TABLE	
9	021232	010337	003302	R2 IS UNIT NUMBER	
10				MOV R3,TEMP	:SAVE CONTROLLER TABLE ADDRESS
11	021236	062703	000020		:IN CASE AN ERROR IS DETECTED
12	021242	012704	000010	ADD #C.DR0,R3	:BUILD POINTER TO C DR ENTRY IN CONTROLLER TABLE
13	021246	005713		MOV #8,R4	:GET MAX COUNT OF DRIVES ON ONE CONTROLLER
14	021250	001411		INIT12: TST (R3)	:CHECK IF ENTRY CONTAINS POINTER TO DRIVE TABLE
15	021252	026033	000010	BEQ INIT13	
16	021256	001002		CMP HO.LDR(R0),@ (R3)+	:CHECK DRIVE NUMBER IN DRIVE TABLE
17	021260	000137	021360	BNE 1\$	
18	021264	005304		JMP MLDRER	:IF SAME, TWO P-TABLES POINT TO SAME DRIVE
19	021266	001367		1\$: DEC R4	:COUNT DRIVES
20	021270	000137	021376	BNE INIT12	:IF EIGHT DRIVE TABLES EXIST,
21	021274	010113		JMP TOOMER	: THEN REPORT ERROR
22	021276	016021	000010	INIT13: MOV R1,(R3)	:LOAD DRIVE TABLE POINTER
23	021302	010221		MOV HO.LDR(R0),(R1)+	:LOAD DRIVE NUMBER
				MOV R2,(R1)+	:LOAD UNIT NUMBER

1					;GO TO NEXT DRIVE TABLE		
2							
3	021304	005202			INIT14: INC R2		;INCREMENT LOGICAL UNIT NUMBER
4	021306	023702	002012		CMP L\$UNIT,R2		;CHECK IF GOT ALL TABLES
5	021312	003331			BGT INIT8		;IF NOT, GET NEXT TABLE
6							
7					;SAVE CURRENT PARAMETERS TO FREE MEMORY		
8							
9	021314	013737	002146	002152	INIT15: MOV FFREE,FMEM		;SAVE START ADDRESS
10	021322	013737	002150	002154	MOV FSIZE,FMEMS		;SAVE SIZE
11							
12	021330				INITXX: SETPRI #PRI00		; SET RUNNING PRIORITY TO ZERO
	021330	012700	000000				MOV #PRI00,R0
	021334	104441					TRAP C\$SPRI
14	021336				CLOSE		;MAKE SURE DATA FILE IS CLOSED
	021336	104435					TRAP C\$CLOS
16	021340				EXIT INIT		
	021340	104432					TRAP C\$EXIT
	021342	000524					.WORD L10037-

1							
2	021344	010305		;DIFFERENT VECTORS, BR LEVELS OR BURST RATES FOR ONE CONTROLLER			
3	021346			CTABR: MOV R3,R5 ;GET CONTROLLER ADDRESS			
	021346	104454		ERRSF 1,,ERR001		TRAP	C\$ERSF
	021350	000001				.WORD	1
	021352	000000				.WORD	0
	021354	010770				.WORD	ERR001
4	021356			DOCLN		TRAP	C\$DCLN
	021356	104444					
5							
6				;TWO P-TABLES FOR SAME DRIVE			
7	021360	013705	003302	MLDRER: MOV TEMP,R5 ;GET CONTROLLER ADDRESS			
8	021364			ERRSF 2,,ERR002		TRAP	C\$ERSF
	021364	104454				.WORD	2
	021366	000002				.WORD	0
	021370	000000				.WORD	ERR002
	021372	011006					
9	021374			DOCLN		TRAP	C\$DCLN
	021374	104444					
10							
11				;MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER			
12							
13	021376	013705	003302	TOOMER: MOV TEMP,R5 ;GET CONTROLLER ADDRESS			
14	021402			ERRSF 3,,ERR003		TRAP	C\$ERSF
	021402	104454				.WORD	3
	021404	000003				.WORD	0
	021406	000000				.WORD	ERR003
	021410	011024					
15	021412			DOCLN		TRAP	C\$DCLN
	021412	104444					
16							
17				;TWO UDA'S USE THE SAME VECTOR			
18							
19	021414	010305		SAMVEC: MOV R3,R5 ;GET CONTROLLER ADDRESS			
20	021416			ERRSF 8,ERR008		TRAP	C\$ERSF
	021416	104454				.WORD	8
	021420	000010				.WORD	ERR008
	021422	011100				.WORD	0
	021424	000000					
21	021426			DOCLN		TRAP	C\$DCLN
	021426	104444					

```

1 021430          DATE:  GMANID DATEQ,DATEI,A,-1,1,11.,YES      ;GET DATE
  021430 104443
  021432 000406
  021434 003330
  021436 000152
  021440 003600
  021442 177777
  021444 000001
  021446 000013
  021450
2 021450 012705 003330      MOV #DATEI,R5      ;GET POINTER TO ANSWER
3 021454 121527 000060      CMPB (R5),#0
4 021460 103443      BLO DERR
5 021462 122527 000071      DAY:  CMPB (R5)+,#'9
6 021466 101040      BHI DERR
7 021470 121527 000055      CMPB (R5),#'-
8 021474 001406      BEQ DAS1
9 021476 121527 000060      CMPB (R5),#0
10 021502 103432      BLO DERR
11 021504 122527 000071      CMPB (R5)+,#'9
12 021510 101027      BHI DERR
13 021512 122527 000055      DAS1:  CMPB (R5)+,#'-
14 021516 001024      BNE DERR
15 021520 012704 000014      MOV #12.,R4      ;GET NUMBER OF MONTH
16 021524 012703 003405      MON1:  MOV #MONTHS,R3      ;GET POINTER TO MONTH NAMES
17 021530 005000      CLR R0
18 021532 121523      CMPB (R5),(R3)+
19 021534 001401      BEQ MON2
20 021536 005200      INC R0
21 021540 126523 000001      MON2:  CMPB 1(R5),(R3)+
22 021544 001401      BEQ MON3
23 021546 005200      INC R0
24 021550 126523 000002      MON3:  CMPB 2(R5),(R3)+
25 021554 001401      BEQ MON4
26 021556 005200      INC R0
27 021560 005700      MON4:  TST R0
28 021562 001407      BEQ MON5
29 021564 005304      DEC R4
30 021566 001360      BNE MON1
31 021570      DERR:  PNTF DATEX
  021570 004137 016230
  021574 010670
  021576 000000
32 021600 000713
33 021602 012701 003344      MON5:  BR DATE
34 021606 010403      MOV #DATEQ,R1      ;GET POINTER TO DATE FOR FORMATTER
35 021610 020327 000012      MOV R4,R3      ;GET COPY OF MONTH NUMBER
36 021614 103404      CMP R3,#10.      ; IF 10 OR GREATER
37 021616 112721 000061      BLO MON6
38 021622 162703 000012      MOVB #'1,(R1)+      ;PUT A '1' IN OUTPUT
39 021626 062703 000060      SUB #10.,R3
40 021632 110321      MON6:  ADD #'0,R3      ;CONVERT MONTH NUMBER TO ASCII
41 021634 112721 000055      MOV R3,(R1)+      ;PUT A NUMBER IN OUTPUT
42 021640 062704 003450      MOVB #'-,(R1)+      ;PUT A '-' IN OUTPUT
43      ADD #DAYS-1,R4      ;GET POINTER TO DAYS IN MONTH
44      ;INDEXED BY NUMBER OF MONTH
45 021644 012703 003330      MOV #DATEI,R3      ;GET POINTER TO DATE INPUT
46 021650 005000      CLR R0

```

10000\$:

```

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

```

```

CSGMAN
10000$
DATEI
T$CODE
DATEQ
-1
T$LOLIM
T$HILIM

```

```

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

```

46	021652	121327	00C055	DAY1:	CMPB (R3),#'-
47	021656	001413			BEQ DAY2
48	021660	111321			MOVB (R3),(R1)+ ;PUT DAY CHARACTER IN OUTPUT
49	021662	006300			ASL R0
50	021664	010002			MOV R0,R2
51	021666	006300			ASL R0
52	021670	006300			ASL R0
53	021672	060200			ADD R2,R0
54	021674	112302			MOVB (R3)+,R2
55	021676	162702	000060		SUB #'0,R2
56	021702	060200			ADD R2,R0
57	021704	000762			BR DAY1
58	021706	120014		DAY2:	CMPB R0,(R4)
59	021710	101327			BHI DERR
60	021712	005700			TST R0 ;SEE IF DATE IS ZERO
61	021714	001725			BEQ DERR ;ERROR IF SO
62	021716	062705	000003		ADD #3,R5
63	021722	121527	000055		CMPB (R5),#'- ;CHECK FOR '-' BETWEEN DAY
64	021726	001320			BNE DERR ; AND YEAR IN OUTPUT
65	021730	112521			MOVB (R5)+,(R1)+ ;PUT '-' IN OUTPUT
66	021732	010504			MOV R5,R4 ;GET COPY OF INPUT STRING POINTER
67	021734	005000			CLR R0
68	021736	005002			CLR R2
69	021740	121427	000060	YER1:	CMPB (R4),#'0
70	021744	103416			BLO YER2
71	021746	121427	000071		CMPB (R4),#'9
72	021752	101013			BHI YER2
73	021754	006300			ASL R0
74	021756	010003			MOV R0,R3
75	021760	006300			ASL R0
76	021762	006300			ASL R0
77	021764	060300			ADD R3,R0
78	021766	112403			MOVB (R4)+,R3
79	021770	162703	000060		SUB #'0,R3
80	021774	060300			ADD R3,R0
81	021776	005202			INC R2
82	022000	000757			BR YER1
83	022002	105714		YER2:	TSTB (R4)
84	022004	001271			BNE DERR
85	022006	020227	000002		CMP R2,#2
86	022012	001407			BEQ YER3
87	022014	020227	000004		CMP R2,#4
88	022020	001263			BNE DERR
89	022022	020027	003554		CMP R0,#1900.
90	022026	103660			BLO DERR
91	022030	000413			BR YER5
92	022032	012702	003465	YER3:	MOV #YEAR19,R2
93	022036	020027	000106		CMP R0,#70.
94	022042	103002			BHIS YER4
95	022044	012702	003470		MOV #YEAR20,R2
96	022050	105712		YER4:	TSTB (R2)
97	022052	001402			BEQ YER5
98	022054	112221			MOVB (R2)+,(R1)+
99	022056	000774			BR YER4
100	022060	112521		YER5:	MOVB (R5)+,(R1)+
101	022062	001376			BNE YER5
102	022064	000207			RETURN

103
104 022066
022066
022066 104411

ENDINIT

L10037: TRAP CSINIT

.SBTTL AUTODROP SECTION

:++
: THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
: THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
: SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
: DROPPED FROM TESTING.
:--

1
2
3
4
5
6
7
8
9
10 022070
022070

BGNAUTO

LSAUTO::

11
12 022070
022070
022070 104461

ENDAUTO

L10040:
TRAP CSAUTO

.SBTTL CLEANUP CODING SECTION

;++
: THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
: AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
:--

1									
2									
3									
4									
5									
6									
7									
8	022072			BGNCLN				L\$CLEAN::	
9	022072								
11	022072			CLOSE		;CLOSE DATA FILE			
	022072	104435							
13	022074	004737	012064	CALL RESET		;RESET ALL UDAS	TRAP	C\$CLOS	
14									
15	022100			ENDCLN					
	022100								
	022100	104412				L10041:	TRAP	C\$CLEAN	
16									
17	022102			ENDMOD					

1				.SBTTL	TEST 1: DUP PROGRAM DRIVER		
2							
3	022102			BGNMOD			
4							
5	022102			BGNTST			
6	022102	032737	000003	003230	BIT #SO.FMT,MODE	T1::	
7	022110	001157			BNE T1FMT		
8	022112				MANUAL		
9	022114	104450			BCOMPLETE T1G0	TRAP	C\$MANI
10	022116	103406			ERRSF 10,,ERR010	BCS	T1G0
	022116	104454				TRAP	C\$ERSF
	022120	000012				.WORD	10
	022122	000000				.WORD	0
	022124	011130				.WORD	ERR010
11	022126				EXIT TST		
	022126	104432				TRAP	C\$EXIT
	022130	000626				.WORD	L10042-.
12	022132	032737	000010	003230	T1G0: BIT #SO.STR,MODE		
13	022140	001432			BEQ T1CNS		
14	022142	023727	002012	000001	CMP L\$UNIT,#1		
15	022150	001406			BEQ T1RST		
16	022152				ERRSF 9,,ERR009		
	022152	104454				TRAP	C\$ERSF
	022154	000011				.WORD	9
	022156	000000				.WORD	0
	022160	011116				.WORD	ERR009
17	022162				EXIT TST		
	022162	104432				TRAP	C\$EXIT
	022164	000572				.WORD	L10042-.
18							
19	022166				T1RST: PNTF FILNAM		
	022166	004137	016230			JSR R1,LPNTF	
	022172	010707				.WORD	FILNAM
	022174	000000				.WORD	PNT.CT
20	022176				GMANID FILNAQ,FNAME2,A,-1,1,10.,NO	;GET FILE NAME	
	022176	104443				TRAP	C\$GMAN
	022200	000406				BR	10000\$
	022202	003254				.WORD	FNAME2
	022204	000142				.WORD	T\$CODE
	022206	003630				.WORD	FILNAQ
	022210	177777				.WORD	-1
	022212	000001				.WORD	T\$LOLIM
	022214	000012				.WORD	T\$HILIM
	022216					10000\$:	
22	022216				OPEN #FNAME2		
	022216	012700	003254			MOV	#FNAME2,R0
	022222	104434				TRAP	C\$OPEN
24	022224	000511			BR T1FMT		
25	022226	013705	002156		T1CNS: MOV CTABS,R5		
26	022232	010504			T1SER1: MOV R5,R4		
27	022234	062704	000020		ADD #C.DR0,R4		
28	022240	012703	000010		MOV #8,R3		
29	022244	011402			T1SER2: MOV (R4),R2	;GET DRIVE TABLE POINTER	
30	022246	001474			BEQ T1SERN		

```

31 022250          PNTF SERNUM,D.UNIT(R2),(R5),(R2)
    022250 011246
    022252 011546
    022254 016246 000002
    022260 004137 016230
    022264 004223
    022266 000006
32 022270          ASSUME C.UADR EQ 0
33 022270          ASSUME D.DRV EQ 0
34 022270 T1SER3: GMANID SERNO,TEMP,A,-1,1,20.,NO ;GET SERIAL NUMBER
    022270 104443
    022272 000406
    022274 003302
    022276 000142
    022300 003662
    022302 177777
    022304 000001
    022306 000024
    022310
35 022310 012701 003302
36 022314 005000
37 022316 105711
38 022320 001410
39 022322 005200
40 022324 121127 000060
41 022330 103416
42 022332 122127 000071
43 022336 101767
44 022340 000412
45 022342 020027 000024
46 022346 103422
47 022350 012701 003302
48 022354 012700 003360
49 022360 122120
50 022362 001776
51 022364 103413
52 022366
    022366 012746 003360
    022372 012746 010600
    022376 012746 000002
    022402 010600
    022404 104417
    022406 062706 000006
53 022412 000726
54 022414 062702 000004
55 022420 012701 003302
56 022424 112122
57 022426 001376
58 022430 005303
59 022432 001402
60 022434 005724
61 022436 000702
62 022440 062705 000054
63 022444 005715
64 022446 001271
65 022450 013737 002156 002162 T1FMT:
66 022456 013701 002160

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

          TRAP CS$GMAN
          BR 10001$
          .WORD TEMP
          .WORD T$CODE
          .WORD SERNO
          .WORD -1
          .WORD T$LOLIM
          .WORD T$HILIM

10001$:
          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
T1SER6: CMPB (R1)+,(R0)+
          BEQ T1SER6
          BLO T1SER8
T1SER7: PRINTF #SERNX,#HIGHEST

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

T1SER8: BR T1SER3
          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
          MOV CTABS,TSTTAB
          MOV CTRLRS,R1

          ;GET FIRST TABLE ADDRESS
          ;RUN DM PROGRAM ON ALL CONTROLLERS

```

```

67      ; FIND OUT WHICH CONTROLLER IS WHICH
68 022462 010137 002174      MOV      R1,URUN
69 022466 013737 002146 002164  MOV      FFREE,DMPROG
70 022474 012701 000001      MOV      #1,R1      ; MUST READ AT LEAST ON DMPROG
71 022500 004737 012006      CALL     TINIT      ; READ DMCODE
72 022504 013737 002174 002200  MOV      URUN,UCNT
73 022512 013705 002162      MOV      TSTTAB,R5      ; POINT TO CONTROLLER TABLE
74 022516 005037 002204      CLR      U50UNI      ; CLEAR NUMBER OF UDA50'S
75 022522 005765 000002      TSTCNT: TST     C,UNIT(R5)
76 022526 100425      BMI      2$      ; IF L'IT DROPPED, BRANCH
77 022530 004737 012254      CALL     GTDUST      ; GET DUST STATUS
78 022534 023764 002206 000034  CMP      U52EXT,HC.MPK+P.DEXT(R4) ; IS IT A 52?
79 022542 001007      BNE      1$      ; IF NOT, BRANCH
80 022544 052765 000400 000014  BIS      #CT.U52,C.FLG(R5)      ; SET FOR 52
81 022552 042765 001000 000014  BIC      #CT.U50,C.FLG(R5)
82 022560 000410      BR      2$
83 022562      1$:
84 022562 042765 000400 000014  BIC      #CT.U52,C.FLG(R5)      ; SET FOR 50
85 022570 052765 001000 000014  BIS      #CT.U50,C.FLG(R5)
86 022576 005237 002204      INC      U50UNI      ; KEEP TRACK IF WE NEED TO READ UDA50 FORMATTER
87 022602 005075 000000      2$:      CLR      @R5      ; INIT CONTROLLER
88 022606 062705 000054      ADD      #C.SIZE,R5      ; POINT TO NEXT CONTROLLER
89 022612 005337 002200      DEC      UCNT      ; CONTINUE UNTIL DONE
90 022616 001341      BNE      TSTCNT
91 022620 012737 000400 002212  ;
92 022626 013737 002160 002200  3$:      MOV      #CT.U52,TSTTYP      ; INIT CONTROLLER TYPE
93 022634 013705 002156      MOV      CTRLRS,UCNT      ; GET COMM AREA FOR EACH CONTROLLER
94 022640 013701 002164      MOV      CTABS,R5
95 022644 011137 002166      MOV      DMPROG,R1      ; GET DMSIZE
96 022650 066137 000004 002166  MOV      (R1),DMSIZE
97 022656 013737 002156 002162  ADD      4(R1),DMSIZE
98 022664 013701 002160      MOV      CTABS,TSTTAB      ; INIT PARAMETERS FOR RUNDM
99 022670 004737 012360      MOV      CTRLRS,R1      ; RUN ALL CONTROLLERS OF ONE TYPE AT ONCE
100 022674 001402      CALL     RUNDM
101 022676 004737 013110      BEQ     6$      ; NEXT CONTROLLER TYPE
102 022702 006337 002212      CALL     RESPDM      ; RESTORE FREE MEMORY
103 022706 013737 002152 002146  ASL      TSTTYP
104 022714 013737 002154 002150  MOV      FMEM,FFREE
105 022722 032737 001000 002212  MOV      FMEMS,FSIZE
106 022730 001410      BIT      #CT.U50,TSTTYP      ; ALL DONE?
107 022732 005737 002204      BEQ     7$      ; IF SO, EXIT
108 022736 001405      TST      U50UNI      ; DO WE CONTINUE FOR UDA50?
109 022740 012701 000002      BEQ     7$      ; IF NOT, EXIT
110 022744 004737 012006      MOV      #2,R1
111 022750 000726      CALL     TINIT      ; ELSE READ IN UDA50 FORMATTER
112 022752      BR      3$
113 022752 104432      7$:      EXIT     TST
114 022754 000002      TRAP     C$EXIT
115 022756      .WORD     L10042-.
116 022756 104401      L10042:      TRAP     C$TST
117 022760      ENDTST
118      ENDMOD

```

```
1      .SBTTL  HARDWARE PARAMETER CODING SECTION
2
3      022760      BGNMOD
4
5      ;++
6      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
9      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
10     ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11     ; WITH THE OPERATOR.
12     ;--
13
14     022760      BGNHRD
15     022760      000027
16     022762
17
18     022762      ;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:
19
20     022762      TABLE
21     022762      ;START A TEBLE DEFINITION
22     022762      ITEM HO.UBA      2      ; UNIGUS ADDRESS
23     022762      ITEM HO.VEC      2      ; UDA VECTOR
24     022762      ITEM HO.BRL      2      ; BR LEVEL
25     022762      ITEM HO.BST      2      ; BURST RATE
26     022762      ITEM HO.LDR      2      ; DRIVE NUMBER
27     022762      END
```

.WORD L10043-L\$HARD/2
L\$HARD::

1	022762				GPRMA	H.UBA,HO.UBA,O,160000,177774,YES		;BUS ADDRESS	
	022762	000031						.WORD	T\$CODE
	022764	023040						.WORD	H.UBA
	022766	160000						.WORD	T\$LOLIM
	022770	177774						.WORD	T\$HILIM
2	022772				GPRMA	H.VEC,HO.VEC,O,4,774,YES		; VECTOR	
	022772	001031						.WORD	T\$CODE
	022774	023066						.WORD	H.VEC
	022776	000004						.WORD	T\$LOLIM
	023000	000774						.WORD	T\$HILIM
3	023002				GPRMD	H.BRL,HO.BRL,D,-1,4.,7.,YES		; BR LEVEL	
	023002	002052						.WORD	T\$CODE
	023004	023075						.WORD	H.BRL
	023006	177777						.WORD	-1
	023010	000004						.WORD	T\$LOLIM
	023012	000007						.WORD	T\$HILIM
4	023014				GPRMD	H.BST,HO.BST,D,-1,0.,63.,YES		; BURST RATE	
	023014	003052						.WORD	T\$CODE
	023016	023106						.WORD	H.BST
	023020	177777						.WORD	-1
	023022	000000						.WORD	T\$LOLIM
	023024	000077						.WORD	T\$HILIM
5	023026				GPRMD	H.LDR,HO.LDR,D,-1,0.,255.,YES		; DRIVE SELECT NUMBER	
	023026	004052						.WORD	T\$CODE
	023030	023130						.WORD	H.LDR
	023032	177777						.WORD	-1
	023034	000000						.WORD	T\$LOLIM
	023036	000377						.WORD	T\$HILIM
6	023040				ENDHRD				
	023040							.EVEN	
								L10043:	
7									
8	023040	125	116	111	H.UBA:	.ASCIZ	\UNIBUS ADDRESS OF UDA\		
9	023066	126	105	103	H.VEC:	.ASCIZ	\VECTOR\		
10	023075	102	122	040	H.BRL:	.ASCIZ	\BR LEVEL\		
11	023106	125	116	111	H.BST:	.ASCIZ	\UNIBUS BURST RATE\		
12	023130	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     023146      BGNSFT
13     023146      000022
14     023150
15
16     023150      ;FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:
17     TABLE
18     023150      ITEM SO.BIT      2
19     000001      SO.FM1 = BIT0
20     000002      SO.FM2 = BIT1
21     000003      SO.FMT = SO.FM1+SO.FM2
22     000004      SO.CNS = BIT2
23     000010      SO.STR = BIT3
24
25     023150      ;START A TABLE DEFINITION
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
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2
```

1	023150				GPRML S.FMT,SO.BIT,SO.FM1,YES	;REFORMAT?			
	023150	000130					.WORD	T\$CODE	
	023152	023365					.WORD	S.FMT	
	023154	000001					.WORD	SO.FM1	
2	023156				XFERT SWEND				
	023156	017024					.WORD	T\$CODE	
3	023160				GPRML S.NRF,SO.BIT,SO.FM2,YES	;AGAIN - REFORMAT?			
	023160	000130					.WORD	T\$CODE	
	023162	023214					.WORD	S.NRF	
	023164	000002					.WORD	SO.FM2	
4	023166				XFERT SWEND				
	023166	013024					.WORD	T\$CODE	
5	023170				GPRML S.CNS,SO.BIT,SO.CNS,YES	;RECONSTRUCT			
	023170	000130					.WORD	T\$CODE	
	023172	023444					.WORD	S.CNS	
	023174	000004					.WORD	SO.CNS	
6	023176				XFERT SWEND				
	023176	007024					.WORD	T\$CODE	
7	023200				GPRML S.RST,SO.BIT,SO.STR,YES	;RESTORE?			
	023200	000130					.WORD	T\$CODE	
	023202	023507					.WORD	S.RST	
	023204	000010					.WORD	SO.STR	
8	023206				XFERT SWEND				
	023206	003024					.WORD	T\$CODE	
9	023210				DISPLAY S.NOF	;WARNING			
	023210	000003					.WORD	T\$CODE	
	023212	023630					.WORD	S.NOF	
10	023214				SWEND: ENDSFT				
	023214						.EVEN		
							L10044:		
11									
12	023214	015	012		S.NRF:	.BYTE 15,12			
13	023216	116	117	124		.ASCII\NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR\			
14	023320	015	012			.BYTE 15,12			
15	023322	111	116	106		.ASCII\INFORMATION ON THE DISKS.\			
16	023353	015	012			.BYTE 15,12			
17	023355	101	107	101		.ASCII\AGAIN - \			
18	023365	122	105	106	S.FMT:	.ASCII\REFORMAT USING EXISTING BAD SECTOR INFORMATION\			
19	023444	122	105	103	S.CNS:	.ASCII\RECONSTRUCT BAD SECTOR INFORMATION\			
20	023507	104	117	040	S.RST:	.ASCII\DO YOU HAVE A FILE ON THE SYSTEM LOAD DEVICE\			
21	023563	015	012			.BYTE 15,12			
22	023565	040	103	117		.ASCII\ CONTAINING BAD SECTOR INFORMATION\			
23	023630	131	117	125	S.NOF:	.ASCII\YOU CANNOT PROCEED WITHOUT SUCH A FILE.\			
24	023700	122	105	123		.ASCII\RESTART PROGRAM AND SELECT TO REFORMAT OR RECONSTRUCT DISK.\			
25	023774	000				.BYTE 0			
26						.EVEN			
27									
28					.DSABL AMA				
29	000000				.PSECT END				

```
1  
2  
3 000000  
4 000050  
5  
6  
7  
8 000120  
000120 000142'  
000122 000007  
000124  
9  
10 000124
```

.SBTTL PATCH AREA

\$PATCH::

.REPT 40.
.WORD 0
.ENDR

LASTAD

LSLAST::

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
6 000132 000154    .WORD 154
7 000134 000005    .WORD 5
8 000136 000077    .WORD 63
9 000140 000000    .WORD 0
10
11 000142          ENDPTAB
  000142
12
13 000142          ENDSETUP
14
15
16
17
18
19
20
21          000001    .END
```

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 29440 WORDS (115 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
A:ZUDECO,B:ZUDECO/C=[20,0]SVC34R.MLB/P:1,ZUDECO.DOC,ZUDECO.MAC

L10045: .WORD 0
 .WORD L10047-./2-1

: UNIBUS ADDRESS
: VECTOR ADDRESS
: BR LEVEL
: UNIBUS BURST RATE
: LOGICAL DRIVE NUMBER

L10047:

SPATCH	143-3#													
A1	1-5#	1-11	1-70	28-35	50-19	53-17	138-88							
A2	1-6#	1-12	1-151	28-36	50-22	53-20	138-99							
ADR	32-10#													
ALOCM	58-16#	59-14	68-19	129-13	130-20	132-4								
ASS	1-8#	1-10	1-11	1-12	1-16	1-70	1-151	27-37	28-34	28-35	28-36	48-7	50-16	50-19
	50-22	53-17	53-20	60-1	61-31	64-28	65-1	83-19	127-20	133-13	137-10	138-21	138-69	138-84
	138-88	138-94	138-99	138-108										
ASSEMB	28-8	28-8												
BAS	52-15#	93-5	93-5	93-5	94-5	94-5								
BASL2	52-13#	94-5												
BASL3	52-14#													
BASLN	52-17#	93-5	94-5											
BASNO	52-12#	93-5	94-5											
BIT0	32-10#	141-19												
BIT00	32-10	32-10#												
BIT01	32-10	32-10#												
BIT02	32-10	32-10#												
BIT03	32-10	32-10#												
BIT04	32-10	32-10#												
BIT05	32-10	32-10#												
BIT06	32-10	32-10#												
BIT07	32-10	32-10#												
BIT08	32-10	32-10#												
BIT09	32-10	32-10#												
BIT1	32-10#	43-23	141-20											
BIT10	32-10#													
BIT11	32-10#													
BIT12	32-10#													
BIT13	32-10#													
BIT14	32-10#													
BIT15	32-10#	43-15	44-12	64-27	72-20	80-26	85-15	100-29						
BIT2	32-10#	43-24	141-22											
BIT3	32-10#	43-25	141-23											
BIT4	32-10#	43-27												
BIT5	32-10#	43-30												
BIT6	32-10#	43-31												
BIT7	32-10#	43-33												
BIT8	32-10#	43-34												
BIT9	32-10#	43-35												
BLDC0	108-22	108-24#												
BLDC1	108-26#	108-28												
BLDCMD	63-18	71-50	74-14	74-44	106-3	108-15#								
BOE	32-10#													
C\$AU	28-8#													
C\$AUTO	28-8#	136-12												
C\$BRK	28-8#	61-12	66-19	71-9	111-21	118-12	121-27							
C\$BSEG	28-8#													
C\$BSUB	28-8#													
C\$CEFG	28-8#													
C\$CLCK	28-8#	127-48	127-50											
C\$CLEA	28-8#	137-15												
C\$CLOS	28-8#	65-13	83-20	133-14	137-11									
C\$CLP1	28-8#													
C\$CVEC	28-8#	61-22	118-30											
C\$DCLN	28-8#	57-8	61-30	62-21	70-7	127-18	134-4	134-9	134-15	134-21				

[illegible]

[illegible]

[illegible]

[illegible]

	55-128	113-10	114-18	124-5	125-38	126-3	126-10	127-8	133-16	136-10	137-8	137-17	138-3	138-5
	138-11	138-17	138-137	138-138	138-139	139-3	139-14	141-12	142-9	142-9	143-10	144-1	144-3	144-3
	144-11	144-13												
FSCLEA	28-8#	137-8	137-15											
FSDU	28-8#													
FSEND	28-8	28-8	28-8	28-8	28-8	28-8	28-8	28-8	28-8	28-8	28-8	28-8	28-8	28-8
	28-8	28-8	28-8#	28-26	31-16	32-3	55-16	55-20	55-24	55-28	55-32	55-36	55-40	55-44
	55-48	55-52	55-56	55-65	55-71	55-85	55-89	55-93	55-97	55-101	55-106	55-110	55-114	55-118
	55-122	55-126	55-130	113-14	114-21	124-9	125-38	126-3	133-16	135-104	136-12	137-15	137-17	138-3
	138-5	138-5	138-5	138-11	138-17	138-137	138-138	138-138	138-139	139-3	140-6	142-10	143-10	144-1
	144-3	144-11	144-13											
F\$HARD	28-8#	139-14	140-6	142-2	142-4	142-6	142-8							
F\$HW	28-8#	30-10	30-17											
F\$INIT	28-8#	127-8	135-104											
F\$JMP	28-8#	133-16	138-11	138-17	138-137									
F\$MOD	28-8#	28-26	31-16	32-3	125-38	126-3	137-17	138-3	138-139	139-3	143-10			
F\$MSG	28-8#	55-14	55-16	55-18	55-20	55-22	55-24	55-26	55-28	55-30	55-32	55-34	55-36	55-38
	55-40	55-42	55-44	55-46	55-48	55-50	55-52	55-54	55-56	55-58	55-65	55-67	55-71	55-73
	55-85	55-87	55-89	55-91	55-93	55-95	55-97	55-99	55-101	55-103	55-106	55-108	55-110	55-112
	55-114	55-116	55-118	55-120	55-122	55-124	55-126	55-128	55-130					
F\$PROT	28-8#	126-10	126-16											
F\$PWR	28-8#													
F\$RPT	28-8#													
F\$SEG	28-8#													
F\$SOFT	28-8#	141-12	142-2	142-4	142-6	142-8	142-10							
F\$SRV	28-8#	113-10	113-14	114-18	114-21	124-5	124-9							
F\$SUB	28-8#													
F\$SW	28-8#	31-10	31-14											
F\$TEST	28-8#	138-5	138-138											
FCTBUF	48-36#	83-25	83-34											
FCTNUM	48-37#	64-41*	83-14	83-23*	83-30*									
FDATA	48-35#	67-5	68-2	68-9	68-20	69-13*	69-16*							
FFREE	48-11#	55-74	58-16	58-20*	60-16*	60-20*	62-13	62-16	67-11	68-16	127-39*	127-40	128-3	133-9
	138-70	138-121*												
FILNAM	54-13#	138-19												
FILNAQ	51-4#	138-20												
FILOPN	48-26#	65-11	65-14*	66-14	66-17*									
FMEM	48-13#	60-16	133-9*	138-121										
FMEMS	48-14#	60-17	133-10*	138-122										
FMERR	57-7#	58-18												
FNAME	48-52#	66-16												
FNAME2	48-51#	83-21	138-20	138-22										
FNUM	48-31#	60-18	60-25*	127-22*										
FS	1-4#	1-8	1-10	1-16	27-37	28-34	48-7	50-16	60-1	61-31	64-28	65-1	83-19	127-20
FSIZE	133-13	137-10	138-21	138-69	138-84	138-94	138-108							
FWORD	48-12#	58-17*	60-17*	60-21*	62-17	127-40*	133-10	138-122*						
	67-4	67-6	68-1	68-3	69-11#									
G\$CNT0	28-8#													
G\$DELM	28-8#													
G\$DISP	28-8#	142-9												
G\$EXCP	28-8#													
G\$HILI	28-8#													
G\$LOLI	28-8#													
G\$NO	28-8#	138-20	138-34											
G\$OFFS	28-8#	135-1	138-20	138-34	140-1	140-2	140-3	140-4	140-5	142-1	142-3	142-5	142-7	
G\$OFFSI	28-8#	135-1	138-20	138-34	140-1	140-2	140-3	140-4	140-5	142-1	142-3	142-5	142-7	

[illegible]

ISMSG	138-3# 28-8# 55-40# 55-85# 55-114#	138-139 55-14# 55-42# 55-87# 55-116#	138-139# 55-16# 55-44# 55-89# 55-118#	139-3 55-18# 55-46# 55-91# 55-120#	139-3# 55-20# 55-48# 55-93# 55-122#	143-10 55-22# 55-50# 55-95# 55-124#	143-10# 55-24# 55-52# 55-97# 55-126#	55-26# 55-54# 55-99# 55-128#	55-28# 55-56# 55-101# 55-130#	55-30# 55-58# 55-103#	55-32# 55-65# 55-106#	55-34# 55-67# 55-108#	55-36# 55-71# 55-110#	55-38# 55-73# 55-112#
ISPROT	28-8#	126-10#												
ISPTAB	28-8#	144-3	144-3#	144-11	144-11#									
ISPR	28-8#													
ISRPT	28-8#													
ISSEG	28-8#	138-5												
ISSETU	28-8#	144-1	144-1#	144-3	144-13	144-13#								
ISSFT	141-12#	142-9	142-10#											
ISSRV	28-8#	113-10#	113-14#	114-18#	114-21#	124-5#	124-9#							
ISSUB	28-8#	138-5												
ISTST	28-8#	138-5	138-5#	138-11	138-17	138-137	138-138	138-138#	138-138#					
IBE	32-10#													
IDU	32-10#													
IER	32-10#													
INFO	75-5	80-15#	81-14	82-14										
INFOE	80-18	80-30#												
INFOP	80-20	80-25#												
INFOX	80-16	80-26#												
INIT0	127-11	127-19#												
INIT1	127-13	127-15	127-17	127-24#										
INIT10	131-10#	131-13												
INIT11	131-11	132-3#												
INIT12	132-13#	132-19												
INIT13	132-14	132-21#												
INIT14	131-5	133-3#												
INIT15	133-9#													
INIT2	127-26	127-29	127-34#											
INIT3	127-36	127-38#												
INIT4	128-15#	130-17												
INIT5	128-21#	128-33												
INIT7	129-26#	129-28												
INIT8	131-4#	133-5												
INITQT	127-18#	127-33												
INITXX	133-12#													
INMEM	48-32#													
IPADRS	49-34#	61-14	128-6	129-4										
ISR	32-10#													
IXE	32-10#													
JSJMP	28-8#													
KNO	127-54	127-61#												
KW.BRL	48-43#	127-56*	127-59											
KW.CSR	48-42#	71-32	86-19	111-22	121-28	124-8*	125-13	127-45*	127-52*	127-55*	127-60*			
KW.EL	48-46#	71-34	71-37	111-24	111-27	115-26	115-27	115-28	121-30	121-33	124-6*	124-7*	125-16	125-17
	127-46*	127-47*												
KW.HZ	48-45#	115-16	125-18	127-58*										
KW.VEC	48-44#	127-57*	127-59											
KW111	124-5#	127-59												
KWOUT.	124-8	127-44#	127-60											
KYES	127-49	127-51	127-55#											
LSACP	28-34#													
LSAPT	28-34#													
LSAUT	28-34#													

L\$AUTO	28-34	136-10#			
L\$CCP	28-34#				
L\$CLEA	28-34	137-8#			
L\$CO	28-34#				
L\$DEPO	28-34#				
L\$DESC	28-34	50-17#			
L\$DESP	28-34#				
L\$DEVP	28-34#				
L\$DISP	28-34	29-9#			
L\$DLY	28-34#				
L\$DTP	28-34#				
L\$DTYP	28-34#				
L\$DUT	28-34#				
L\$DVTY	28-34	50-12#			
L\$EF	28-34#				
L\$ENVI	28-34#				
L\$ETP	28-34#				
L\$EXP1	28-34#				
L\$EXP4	28-34#				
L\$EXP5	28-34#				
L\$HARD	28-34	139-14	139-14#		
L\$HIME	28-34#				
L\$HPCP	28-34#				
L\$HPTP	28-34#				
L\$HW	28-34	30-10	30-10#		
L\$ICP	28-34#				
L\$INIT	28-34	127-8#			
L\$LADP	28-34#				
L\$LAST	28-34	143-8#	144-13		
L\$LOAD	28-34#				
L\$LUN	28-34#	64-24*	71-13*	85-14*	
L\$MREV	28-34#				
L\$NAME	28-34#				
L\$PRIO	28-34#				
L\$PROT	28-34	126-10#			
L\$PRT	28-34#				
L\$REPP	28-34#				
L\$REV	28-34#				
L\$SOFT	28-34	141-12	141-12#		
L\$SPC	28-34#				
L\$SPCP	28-34#				
L\$SPTP	28-34#				
L\$STA	28-34#				
L\$SW	28-34	31-10	31-10#		
L\$TEST	28-34#				
L\$TIML	28-34#				
L\$UNIT	28-34#	86-13	130-16	133-4	138-14
L10000	30-10	30-17#			
L10001	31-10	31-14#			
L10002	55-16#				
L10003	55-20#				
L10004	55-24#				
L10005	55-28#				
L10006	55-32#				
L10007	55-36#				
L10010	55-40#				

[illegible]

MD.SEC	39-9#					
MD.SEQ	39-14#					
MD.SER	39-10#					
MD.SPD	39-15#					
MD.SSH	39-11#					
MD.SWP	39-21#					
MD.VOL	39-17#					
MD.WBN	39-12#					
MD.WBV	39-13#					
MESG	80-25	84-13#				
MESSG	52-9#	86-15				
MLDRER	132-17	134-7#				
MODE	48-38#	79-53	79-59	127-34*	138-6	138-12
MON1	135-17#	135-30				
MON2	135-19	135-21#				
MON3	135-22	135-24#				
MON4	135-25	135-27#				
MON5	135-28	135-33#				
MON6	135-36	135-39#				
MONTHS	49-7#	135-16				
MSCP	36-31#					
MSGPKL	55-149#	55-152				
MSGPKT	55-105	55-145#				
NCONF	87-38	87-43#				
NCONS	87-37#	87-40				
NEWTAB	128-22	129-3#				
NOCLOC	52-11#	127-53				
NULL	48-50#					
NXMAD	48-34#	61-10*	61-18	113-12*	118-26*	118-31
NXMI	61-11	113-10#	118-27			
NXTTAB	128-16	129-31	130-10	130-15#		
OSAPTS	28-8#	28-34				
OSAU	28-8#	28-34				
OSBGNR	28-8#	28-34				
OSBGNS	28-8#	28-32#	28-34			
OSDU	28-8#	28-34				
OSERRT	28-8#	28-34				
OSGNSW	28-8#	28-32#	28-34			
OSPOIN	28-8#	28-32	28-32#	28-32#	28-32#	28-34
OSSETU	28-8#	28-32#	28-34	143-8		
OP.ABO	38-3#					
OP.ACC	38-4#					
OP.AVA	38-22#					
OP.AVL	38-5#					
OP.CCD	38-6#					
OP.CMP	38-7#					
OP.DUP	38-23#					
OP.ELP	38-30#					
OP.END	38-20#	73-5	73-8	74-58		
OP.ERS	38-8#					
OP.ESP	38-29#	106-2				
OP.FLU	38-9#					
OP.GCS	38-10#					
OP.GDS	38-27#	71-49	74-58			
OP.GSS	38-28#	63-17				
OP.GUS	38-11#					

OP.MRD	38-18#					
OP.MWR	38-19#	108-21				
OP.ONL	38-12#					
OP.RD	38-13#					
OP.RLC	38-25#					
OP.RPL	38-14#					
OP.RSD	38-32#	73-8	74-43			
OP.SCC	38-15#					
OP.SEX	38-21#					
OP.SHC	38-24#					
OP.SSD	38-31#	73-5	74-13			
OP.SUC	38-16#					
OP.WR	38-17#					
OSTRE	87-35	87-42	87-47#			
OSTRNG	87-34#	87-46	95-6	96-6	97-6	103-17
P.BCNT	40-21#	41-9#	74-11	74-33*	106-5*	110-19*
P.BUFF	40-22#					
P.CMST	41-14#					
P.CNCL	41-48#					
P.CNTF	40-40#	41-46#				
P.CNTI	41-49#					
P.CPSP	40-34#					
P.CRF	40-17#	41-4#	73-19	109-17*		
P.CTMO	41-47#					
P.CYL	41-26#					
P.DEXT	41-52#	138-82				
P.DFLG	41-53#	74-60				
P.DMDT	40-50#					
P.DPI	41-54#	74-65	74-67	74-72	74-73	
P.DTO	41-55#					
P.ELGF	40-32#					
P.FBBK	41-10#					
P.FLGS	41-7#					
P.GRP	41-25#					
P.HSTI	40-31#	41-19#	41-35#			
P.HTMO	40-41#					
P.LBN	40-24#					
P.MEDI	41-21#	41-37#				
P.MLUN	41-17#	41-33#				
P.MOD	40-20#					
P.OPCD	40-19#	41-6#	73-9	74-58	108-29*	
P.OTRF	40-27#	41-13#				
P.OVRL	40-51#	106-6*	106-7*			
P.RBN	40-36#					
P.RBNS	41-28#					
P.RCTC	41-29#					
P.RCTS	41-27#					
P.RGID	40-46#					
P.RGOF	40-47#					
P.SHST	41-23#	41-39#				
P.SHUN	40-33#	41-22#	41-38#			
P.STS	41-8#	73-14	106-11			
P.TIME	40-43#					
P.TRCK	41-24#					
P.UADR	40-23#	106-4*	110-18*			
P.UNCL	41-40#					

[illegible]

[illegible]

[illegible]

TSARGC	28-34 28-34# 102-14# 138-52	28-34 28-34# 102-14# 138-52#	28-34 28-34# 102-16 138-52#	28-34 55-70 102-16 138-52#	28-34 55-70 102-16# 138-52#	28-34 102-12 102-16# 138-20	28-34 102-12 102-18 138-20	28-34 102-12 102-18 138-20	28-34 102-12# 102-18# 138-20#	28-34 102-12# 102-18# 138-20#	28-34# 102-14 102-18# 138-20#	28-34# 102-14 102-18# 138-34	28-34# 102-14 102-18# 138-34	
TSCODE	135-1 138-34 140-2# 140-5 142-2 142-4 142-5# 142-7 142-9	135-1 138-34# 140-2# 140-5 142-2 142-4 142-5# 142-7# 142-9#	135-1 138-34# 140-3 140-5 142-2 142-4 142-6 142-7# 142-9#	135-1# 138-34# 140-3 140-5# 142-2# 142-4 142-6 142-7# 142-8	135-1# 140-1 140-3 140-5# 142-2# 142-4 142-6 142-8	135-1# 140-1 140-3# 140-5# 142-2# 142-4# 142-6 142-8	138-20 140-1 140-3# 142-1 142-2# 142-4# 142-6 142-8	138-20 140-1# 140-3# 142-1 142-2# 142-4# 142-6 142-8	138-20 140-1# 140-4 142-1# 142-2# 142-3# 142-4# 142-6# 142-8	138-20# 140-1# 140-4 142-1# 142-2# 142-3# 142-4# 142-6# 142-8	138-20# 140-2 140-4# 142-1# 142-2# 142-3# 142-5 142-6# 142-8#	138-20# 140-2 140-4# 142-1# 142-2# 142-3# 142-5 142-6# 142-8#	138-34 140-2 140-4# 142-2 142-3# 142-5 142-7 142-8#	138-34 140-2# 140-4# 142-2 142-3# 142-5 142-7 142-8#
TSERRN	28-8# 73-21# 116-35# 134-8#	57-7 74-8 117-11 134-14	57-7# 74-8# 117-11# 134-14#	61-28 78-22 118-33 134-20	61-28# 78-22# 118-33# 134-20#	62-20 80-30 120-44 138-10	62-20# 80-30# 120-44# 138-10#	70-6 107-3 121-36 138-16	70-6# 107-3# 121-36# 138-16#	71-25 111-29 122-5 140-2#	71-25# 111-29# 122-5# 140-2#	72-36 112-5 134-3 140-3#	72-36# 112-5# 134-3# 140-3#	73-21 116-35 134-8 140-4#
TSEXCP	135-1 140-5	135-1# 140-5#	138-20 140-5#	138-20# 138-34	138-20# 138-34	138-34# 140-1	138-34# 140-1#	140-2 140-1#	140-2# 140-1#	140-2# 140-1#	140-3 140-3#	140-3# 140-4	140-4# 140-4#	
TSFLAG	133-16 138-137#	133-16 138-137#	133-16# 138-137#	133-16# 138-11	133-16# 138-11	138-11# 138-11	138-11# 138-11#	138-17 138-11#	138-17 138-11#	138-17# 138-17#	138-17# 138-17#	138-137 138-137#	138-137# 138-137#	
TSFREE	143-8	144-13#												
TSGMAN	28-8#	135-1	135-1#	135-1#	138-20	138-20#	138-20#	138-34	138-34#	138-34#				
TSHILI	135-1 140-5	135-1# 140-5#	138-20 140-5#	138-20# 138-34	138-20# 138-34	138-34# 140-1	138-34# 140-1#	140-2 140-1#	140-2# 140-1#	140-2# 140-1#	140-3 140-3#	140-4 140-4#	140-4# 140-4#	
TSLAST	28-8#	143-8#	144-1											
TSLOLI	135-1 140-5	135-1# 140-5#	138-20 140-5#	138-20# 138-34	138-20# 138-34	140-1 138-34#	140-1# 138-34#	140-2 140-1#	140-2# 140-1#	140-2# 140-1#	140-3 140-3#	140-4 140-4#	140-4# 140-4#	
TSLSYM	28-8 55-56 55-130	28-8# 55-65 113-14	30-17 55-71 114-21	31-14 55-85 124-9	55-16 55-89 135-104	55-20 55-93 136-12	55-24 55-97 137-15	55-28 55-101 138-138	55-32 55-106 140-6	55-36 55-110 142-10	55-40 55-114	55-44 55-118	55-48 55-122	55-52 55-126
TSLTNO	143-8#													
TSNEST	28-8# 31-14 55-16 55-24 55-32 55-40 55-48 55-56 55-71 55-89 55-97 55-106 55-114 55-122 55-130 114-21 125-38# 135-104 137-15 138-138 140-6 142-10#	28-26 31-14 55-16 55-24 55-32 55-40 55-48 55-56 55-71 55-89 55-97 55-106 55-114 55-122 55-130 114-21 126-3 135-104 137-15 138-138 140-6 143-10	28-26 31-14 55-16 55-24 55-32 55-40 55-48 55-56 55-71 55-89 55-97 55-106 55-114 55-122 55-130 114-21 126-3 135-104 137-15 138-138 140-6 143-10	28-26# 31-14# 55-16# 55-24# 55-32# 55-40# 55-48# 55-56# 55-71# 55-89# 55-97# 55-106# 55-114# 55-122# 55-130# 114-21# 126-3# 135-104# 137-15# 138-138# 140-6# 143-10#	30-10 31-16 55-18 55-26 55-34 55-42 55-50 55-58 55-73 55-91 55-99 55-108 55-116 55-124 113-10 124-5 136-10 137-17 138-139 141-12 143-10#	30-10 31-16 55-18 55-26 55-34 55-42 55-50 55-58 55-73 55-91 55-99 55-108 55-116 55-124 113-10 124-5 136-10 137-17 138-139 141-12 143-10#	30-10# 31-16# 55-18# 55-26# 55-34# 55-42# 55-50# 55-58# 55-73# 55-91# 55-99# 55-108# 55-116# 55-124# 113-10# 124-5# 136-10# 137-17# 138-139# 141-12# 143-10#	30-17 31-16# 55-20 55-28 55-36 55-44 55-52 55-65 55-85 55-93 55-101 55-110 55-118 55-126 113-14 124-9 136-12 137-17# 138-3 139-3 142-2	30-17 32-3 55-20 55-28 55-36 55-44 55-52 55-65 55-85 55-93 55-101 55-110 55-118 55-126 113-14 124-9 136-12 137-17# 138-3 139-3 142-4	30-17 32-3 55-20 55-28 55-36 55-44 55-52 55-65 55-85 55-93 55-101 55-110 55-118 55-126 113-14 124-9 136-12 137-17# 138-3 139-3 142-6	30-17# 32-3# 55-20# 55-28# 55-36# 55-44# 55-52# 55-65# 55-85# 55-93# 55-101# 55-110# 55-118# 55-126# 113-14# 124-9# 136-2# 137-8 138-5 139-14 142-8	31-10 55-14 55-22 55-30 55-38 55-46 55-54 55-67 55-87 55-95 55-103 55-112 55-120 55-128 114-18 125-38 137-8 138-5 139-14 142-10	31-10 55-14 55-22 55-30 55-38 55-46 55-54 55-67 55-87 55-95 55-103 55-112 55-120 55-128 114-18 125-38 137-8 138-5 139-14 142-10	31-10# 55-14# 55-22# 55-30# 55-38# 55-46# 55-54# 55-67# 55-87# 55-95# 55-103# 55-112# 55-120# 55-128# 114-18# 125-38 137-8# 138-5# 139-14# 142-10
TSNSO	28-26#	31-16	32-3#	125-38	126-3#	137-17	138-3#	138-139	139-3#	143-10				
TSNSI	30-10#	30-17	31-10#	31-14	55-14#	55-16	55-18#	55-20	55-22#	55-24	55-26#	55-28	55-30#	55-32

	55-34#	55-36	55-38#	55-40	55-42#	55-44	55-46#	55-48	55-50#	55-52	55-54#	55-56	55-58#	55-65
	55-67#	55-71	55-73#	55-85	55-87#	55-89	55-91#	55-93	55-95#	55-97	55-99#	55-101	55-103#	55-106
	55-108#	55-110	55-112#	55-114	55-116#	55-118	55-120#	55-122	55-124#	55-126	55-128#	55-130	113-10#	113-14
	114-18#	114-21	124-5#	124-9	126-10#	126-16	127-8#	135-104	136-10#	136-12	137-8#	137-15	138-5#	138-138
	139-14#	140-6	141-12#	142-2	142-4	142-6	142-8	142-10						
TSPCNT	144-1#	144-3	144-3	144-3#										
TSPTAB	144-3	144-3#												
TSPTHV	28-34	144-13#												
TSPINU	28-8#	144-3	144-3#	144-13	144-13									
TSSAVL	28-8#													
TSSSEGL	28-8#													
TSSIZE	143-8	144-13#												
TSSUBN	28-8#	138-5#												
TSTAGL	28-8#													
TSTAGN	28-8#	30-10	30-10	30-10#	31-10	31-10	31-10#	55-14	55-14	55-14#	55-18	55-18	55-18#	55-22
	55-22	55-22#	55-26	55-26	55-26#	55-30	55-30	55-30#	55-34	55-34	55-34#	55-38	55-38	55-38#
	55-42	55-42	55-42#	55-46	55-46	55-46#	55-50	55-50	55-50#	55-54	55-54	55-54#	55-58	55-58
	55-58#	55-67	55-67	55-67#	55-73	55-73	55-73#	55-87	55-87	55-87#	55-91	55-91	55-91#	55-95
	55-95	55-95#	55-99	55-99	55-99#	55-103	55-103	55-103#	55-108	55-108	55-108#	55-112	55-112	55-112#
	55-116	55-116	55-116#	55-120	55-120	55-120#	55-124	55-124	55-124#	55-128	55-128	55-128#	113-10	113-10
	113-10#	114-18	114-18	114-18#	124-5	124-5	124-5#	126-10	126-10	126-10#	127-8	127-8	127-8#	136-10
	136-10	136-10#	137-8	137-8	137-8#	138-5	138-5	138-5#	139-14	139-14	139-14#	141-12	141-12	141-12#
	144-1	144-1	144-1#	144-3	144-3	144-3	144-3	144-3#	144-3#	144-3#				
TSTEMP	29-9	29-9	29-9#	29-9#	30-17	30-17#	31-14	31-14#	31-16	31-16#	55-16	55-16#	55-20	55-20#
	55-24	55-24#	55-28	55-28#	55-32	55-32#	55-36	55-36#	55-40	55-40#	55-44	55-44#	55-48	55-48#
	55-52	55-52#	55-56	55-56#	55-65	55-65#	55-71	55-71#	55-85	55-85#	55-89	55-89#	55-93	55-93#
	55-97	55-97#	55-101	55-101#	55-106	55-106#	55-110	55-110#	55-114	55-114#	55-118	55-118#	55-122	55-122#
	55-126	55-126#	55-130	55-130#	113-14	113-14#	114-21	114-21#	124-9	124-9#	125-38	125-38#	126-16	126-16#
	133-16	133-16#	135-1	135-1	135-1	135-1#	135-1#	135-1#	135-104	135-104#	136-12	136-12#	137-15	137-15#
	137-17	137-17#	138-11	138-11#	138-17	138-17#	138-20	138-20	138-20	138-20#	138-20#	138-20#	138-34	138-34
	138-34	138-34#	138-34#	138-34#	138-137	138-137#	138-138	138-138#	138-139	138-139#	140-1	140-1	140-1	140-1#
	140-1#	140-1#	140-2	140-2	140-2	140-2#	140-2#	140-2#	140-3	140-3	140-3	140-3#	140-3#	140-3#
	140-4	140-4	140-4	140-4#	140-4#	140-4#	140-5	140-5	140-5	140-5#	140-5#	140-5#	140-6	140-6#
	142-1	142-1	142-1	142-1#	142-1#	142-1#	142-3	142-3	142-3	142-3#	142-3#	142-3#	142-5	142-5
	142-5	142-5#	142-5#	142-5#	142-7	142-7	142-7	142-7#	142-7#	142-7#	142-10	142-10#	143-10	143-10#
TSTEST	28-8#	138-5	138-5	138-5#	143-8									
TSTSTM	28-8#	55-16	55-20	55-24	55-28	55-32	55-36	55-40	55-44	55-48	55-52	55-56	55-65	55-70
	55-71	55-85	55-89	55-93	55-97	55-101	55-106	55-110	55-114	55-118	55-122	55-126	55-130	57-7
	57-8	61-11	61-12	61-22	61-28	61-30	62-20	62-21	63-9	65-13	66-16	66-19	66-21	66-26
	67-7	68-23	68-30	69-1	69-11	69-14	70-6	70-7	71-9	71-25	72-36	73-21	74-8	78-22
	80-30	83-20	83-21	83-26	102-12	102-14	102-16	102-18	105-20	107-3	111-21	111-29	112-5	116-35
	117-11	118-12	118-27	118-30	118-33	120-44	121-27	121-36	122-5	127-10	127-12	127-14	127-16	127-18
	127-39	127-48	127-50	127-59	128-15	131-4	133-12	133-14	133-16	134-3	134-4	134-8	134-9	134-14
	134-15	134-20	134-21	135-1	135-104	136-12	137-11	137-15	138-8	138-10	138-11	138-16	138-17	138-20
	138-22	138-34	138-52	138-137	138-138									
TSTSTS	28-8#	138-5#												
T1	29-9	138-5#												
T1CNS	138-13	138-25#												
T1FMT	138-7	138-24	138-65#											
T1GO	138-9	138-12#												
T1RST	138-15	138-19#												
T1SER1	138-26#	138-64												
T1SER2	138-29#	138-61												
T1SER3	138-34#	138-53												
T1SER4	138-37#	138-43												
T1SER5	138-38	138-45#												

[illegible]

UF.576	40-12#					
UF.CMR	40-3#					
UF.CMW	40-4#					
UF.INA	40-6#					
UF.RPL	40-5#					
UF.SCH	40-7#					
UF.SCL	40-8#					
UF.WBN	40-9#					
UF.WPH	40-10#					
UF.WPS	40-11#					
UFREEZ	48-33#	64-40*	72-3	72-13*	80-19	80-21*
URNING	48-24#	64-16*	64-36*	64-45	72-32*	
URUN	48-23#	64-15*	64-20	71-8	138-68*	138-76
WAITMS	63-20	106-9	111-11#			
X\$ALWA	28-8#					
X\$FALS	28-8#					
X\$OFFS	28-8#	142-2	142-4	142-6	142-8	
X\$TRUE	28-8#	142-2	142-4	142-6	142-8	
X1	53-5#	55-15				
X10	53-15#	55-47				
X100	53-46#	55-125				
X101	53-47#	55-129				
X11	53-16#	55-51				
X1A	53-1#	55-15				
X2	53-6#	55-19				
X20	53-23#	55-55				
X21	53-27#	55-64				
X22	53-29#	55-69				
X23A	53-31#	55-74				
X23B	53-35#	55-78				
X24	53-36#	55-88				
X25	53-38#	55-92				
X2A	53-2#	55-19				
X3	53-7#	55-23				
X30	53-40#	55-96				
X31	53-41#	55-100				
X32	53-42#	55-104				
X36	53-43#	55-117				
X37	53-45#	55-121				
X3A	53-3#	55-23				
X4	53-8#	55-27				
X5	53-10#	55-31				
X7	53-11#	55-35				
X8	53-12#	55-39				
X8A	53-4#	55-39				
X9	53-13#	55-43				
XFRU	54-8#	55-70	55-84	96-5		
XMSG1	54-1#	55-145				
XMSG2	54-2#	55-149				
XPKT1	54-3#	55-132				
XPKT2	54-6#	55-138				
XSA	54-7#	97-5				
YEAR19	49-31#	135-92				
YEAR20	49-32#	135-95				
YER1	135-69#	135-82				
YER2	135-70	135-72	135-83#			

VER3	135-86	135-92#	
VER4	135-94	135-96#	135-99
VER5	135-91	135-97	135-100# 135-101

.BR	45-31#													
AND	45-3#	55-61	63-6	105-17	118-15									
ASSUME	45-39#	64-27	72-20	79-36	85-15	86-16	86-17	127-31	128-24	128-25	138-32	138-33		
BCOMPL	127-11	127-13	127-15	127-17	127-49	127-51	138-9							
BGNAUT	136-10													
BGNCLN	137-8													
BGNHRD	139-14													
BGNHW	30-10													
BGNINI	127-8													
BGNMOD	28-26	32-3	126-3	138-3	139-3									
BGNMSG	55-14	55-18	55-22	55-26	55-30	55-34	55-38	55-42	55-46	55-50	55-54	55-58	55-67	55-73
	55-87	55-91	55-95	55-99	55-103	55-108	55-112	55-116	55-120	55-124	55-128			
BGNPRO	126-10													
BGNPTA	144-3													
BGNSET	144-1													
BGNSFT	141-12													
BGNSRV	113-10	114-18	124-5											
BGNSW	31-10													
BGNTST	138-5													
BNCOMP	83-27	128-16	131-5											
BREAK	61-12	66-19	71-9	111-21	118-12	121-27								
BUILD	90-18#	91-9	91-21											
CLOCK	127-48	127-50												
CLOSE	65-13	83-20	133-14	137-11										
CLRVEC	61-22	118-30												
DESCRI	50-17													
DEVTYP	50-12													
DISPAT	29-9													
DISPLA	142-9													
DOCLN	57-8	61-30	62-21	70-7	127-18	134-4	134-9	134-15	134-21					
END	33-29#	43-51	44-15	139-25	141-25									
ENDAUT	136-12													
ENDCLN	137-15													
ENDHRD	140-6													
ENDHW	30-17													
ENDINI	135-104													
ENDMOD	31-16	125-38	137-17	138-139	143-10									
ENDMSG	55-16	55-20	55-24	55-28	55-32	55-36	55-40	55-44	55-48	55-52	55-56	55-65	55-71	55-85
	55-89	55-93	55-97	55-101	55-106	55-110	55-114	55-118	55-122	55-126	55-130			
ENDPRO	126-16													
ENDPTA	144-11													
ENDSET	144-13													
ENDSFT	142-10													
ENDSRV	113-14	114-21	124-9											
ENDSW	31-14													
ENDTST	138-138													
ENTRY	91-3#	91-9	91-9	91-9	91-9	91-9	91-9	91-9	91-9	91-15#	91-21	91-21	91-21	91-21
	91-21	91-21	91-21	91-21										
EQUALS	32-10													
ERRDF	61-28	71-25	72-36	73-21	74-8	78-22	80-30	107-3	111-29	112-5	116-35	117-11	118-33	120-44
	121-36	122-5												
ERRSF	57-7	62-20	70-6	134-3	134-8	134-14	134-20	138-10	138-16					
EXIT	133-16	138-11	138-17	138-137										
GETBYT	66-21	66-26	67-7	68-23	68-30	69-1	69-11	69-14	83-26					
GMANID	135-1	138-20	138-34											
GPHARD	128-15	131-4												

GPRMA	140-1	140-2												
GPRMD	135-1	135-1#	138-20	138-20#	138-34	138-34#	140-3	140-4	140-5					
GPRML	142-1	142-3	142-5	142-7										
HEADER	28-34													
ITEM	33-24#	43-12	43-13	43-16	43-19	43-20	43-21	43-22	43-36	43-37	43-38	43-39	43-40	43-41
	43-42	43-43	43-44	43-45	43-46	43-47	43-48	43-49	44-9	44-10	44-13	139-20	139-21	139-22
	139-23	139-24	141-18											
LASTAD	143-8													
MS BYTE	28-34	28-34	28-34	28-34#										
MSCHEC	133-16	133-16#	138-11	138-11#	138-17	138-17#	138-137	138-137#						
MSCNTO	135-1	135-1#	138-20	138-20#	138-34	138-34#	140-1	140-1#	140-2	140-2#	140-3	140-3#	140-4	140-4#
	140-5	140-5#	142-1	142-1#	142-3	142-3#	142-5	142-5#	142-7	142-7#				
MSCOUN	55-70	55-70#	102-12	102-12#	102-14	102-14#	102-16	102-16#	102-18	102-18#	138-52	138-52#		
MSDATA	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	50-17#													
MSDECR	30-17	30-17#	31-14	31-14#	31-16	31-16#	55-16	55-16#	55-20	55-20#	55-24	55-24#	55-28	55-28#
	55-32	55-32#	55-36	55-36#	55-40	55-40#	55-44	55-44#	55-48	55-48#	55-52	55-52#	55-56	55-56#
	55-65	55-65#	55-71	55-71#	55-85	55-85#	55-89	55-89#	55-93	55-93#	55-97	55-97#	55-101	55-101#
	55-106	55-106#	55-110	55-110#	55-114	55-114#	55-118	55-118#	55-122	55-122#	55-126	55-126#	55-130	55-130#
	113-14	113-14#	114-21	114-21#	124-9	124-9#	125-38	125-38#	126-16	126-16#	135-104	135-104#	136-12	136-12#
	137-15	137-15#	137-17	137-17#	138-138	138-138#	138-139	138-139#	140-6	140-6#	142-10	142-10#	143-10	143-10#
	144-3	144-3#												
MSDEFA	135-1	135-1#	138-20	138-20#	138-34	138-34#	140-1	140-1#	140-2	140-2#	140-3	140-3#	140-4	140-4#
	140-5	140-5#	142-1	142-1#	142-3	142-3#	142-5	142-5#	142-7	142-7#				
MSENDE	30-17#	31-14#	31-16#	55-16#	55-20#	55-24#	55-28#	55-32#	55-36#	55-40#	55-44#	55-48#	55-52#	55-56#
	55-65#	55-71#	55-85#	55-89#	55-93#	55-97#	55-101#	55-106#	55-110#	55-114#	55-118#	55-122#	55-126#	55-130#
	113-14#	114-21#	124-9#	125-38#	135-104#	136-12#	137-15#	137-17#	138-138#	138-139#	140-6#	142-10#	143-10#	
MSERRI	57-7	57-7#	61-28	61-28#	62-20	62-20#	70-6	70-6#	71-25	71-25#	72-36	72-36#	73-21	73-21#
	74-8	74-8#	78-22	78-22#	80-30	80-30#	107-3	107-3#	111-29	111-29#	112-5	112-5#	116-35	116-35#
	117-11	117-11#	118-33	118-33#	120-44	120-44#	121-36	121-36#	122-5	122-5#	134-3	134-3#	134-8	134-8#
	134-14	134-14#	134-20	134-20#	138-10	138-10#	138-16	138-16#						
MSEXCP	135-1	135-1	135-1#	138-20	138-20	138-20#	138-34	138-34	138-34#	140-1	140-1	140-1#	140-2	140-2
	140-2#	140-3	140-3	140-3#	140-4	140-4#	140-5	140-5	140-5	140-5#				
MSEXIT	133-16	133-16#	138-11	138-11#	138-17	138-17#	138-137	138-137#						
MSEXSE	133-16#	138-11#	138-17#	138-137#										
MSEXTJ	133-16#	138-11#	138-17#	138-137#										
MSGEN	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	31-10	31-10	31-10#	31-10#	31-14	31-14#	50-12	50-12#	50-17	50-17#	55-14	55-14#	55-16	55-16#
	55-18	55-18#	55-20	55-20#	55-22	55-22#	55-24	55-24#	55-26	55-26#	55-28	55-28#	55-30	55-30#
	55-32	55-32#	55-34	55-34#	55-36	55-36#	55-38	55-38#	55-40	55-40#	55-42	55-42#	55-44	55-44#
	55-46	55-46#	55-48	55-48#	55-50	55-50#	55-52	55-52#	55-54	55-54#	55-56	55-56#	55-58	55-58#
	55-65	55-65#	55-67	55-67#	55-71	55-71#	55-73	55-73#	55-85	55-85#	55-87	55-87#	55-89	55-89#
	55-91	55-91#	55-93	55-93#	55-95	55-95#	55-97	55-97#	55-99	55-99#	55-101	55-101#	55-103	55-103#
	55-106	55-106#	55-108	55-108#	55-110	55-110#	55-112	55-112#	55-114	55-114#	55-116	55-116#	55-118	55-118#
	55-120	55-120#	55-122	55-122#	55-124	55-124#	55-126	55-126#	55-128	55-128#	55-130	55-130#	113-10	113-10#
	113-14	113-14#	114-18	114-18#	114-21	114-21#	124-5	124-5#	124-9	124-9#	126-10	126-10#	127-8	127-8#
	135-1	135-1#	135-104	135-104#	136-10	136-10#	136-12	136-12#	137-8	137-8#	137-15	137-15#	138-5	138-5#
	138-20	138-20#	138-34	138-34#	138-138	138-138#	139-14	139-14#	140-6	140-6#	141-12	141-12#	142-10	142-10#
	143-8	143-8#	144-3	144-3#	144-11	144-11#								

MSGENB	135-1	135-1#	138-20	138-20#	138-34	138-34#								
MSGETS	30-17	30-17#	31-14	31-14#	31-16	31-16#	55-16	55-16#	55-20	55-20#	55-24	55-24#	55-28	55-28#
	55-32	55-32#	55-36	55-36#	55-40	55-40#	55-44	55-44#	55-48	55-48#	55-52	55-52#	55-56	55-56#
	55-65	55-65#	55-71	55-71#	55-85	55-85#	55-89	55-89#	55-93	55-93#	55-97	55-97#	55-101	55-101#
	55-106	55-106#	55-110	55-110#	55-114	55-114#	55-118	55-118#	55-122	55-122#	55-126	55-126#	55-130	55-130#
	113-14	113-14#	114-21	114-21#	124-9	124-9#	125-38	125-38#	126-16	126-16#	135-104	135-104#	136-12	136-12#
	137-15	137-15#	137-17	137-17#	138-138	138-138#	138-139	138-139#	140-6	140-6#	142-2	142-2#	142-4	142-4#
	142-6	142-6#	142-8	142-8#	142-10	142-10#	143-10	143-10#						
MSGETT	133-16#	138-11#	138-17#	138-137#	142-2	142-2#	142-4	142-4#	142-6	142-6#	142-8	142-8#		
MSGNGB	28-26#	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	31-10#	32-3#	50-12	50-12#	50-17	50-17#	55-14	55-14#	55-18	55-18#	55-22	55-22#	55-26	55-26#
	55-30	55-30#	55-34	55-34#	55-38	55-38#	55-42	55-42#	55-46	55-46#	55-50	55-50#	55-54	55-54#
	55-58	55-58#	55-67	55-67#	55-73	55-73#	55-87	55-87#	55-91	55-91#	55-95	55-95#	55-99	55-99#
	55-103	55-103#	55-108	55-108#	55-112	55-112#	55-116	55-116#	55-120	55-120#	55-124	55-124#	55-128	55-128#
	113-10	113-10#	114-18	114-18#	124-5	124-5#	126-3#	126-10	126-10#	127-8	127-8#	136-10	136-10#	137-8
MSGNIN	137-8#	138-3#	139-3#	139-14	139-14#	141-12	141-12#	143-8	143-8#					
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34	28-34
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#	28-34#
	50-12	50-12	50-12#	50-12#	50-17	50-17#	50-17#	50-17#	55-16	55-16#	55-20	55-20#	55-24	55-24#
	55-28	55-28#	55-32	55-32#	55-36	55-36#	55-40	55-40#	55-44	55-44#	55-48	55-48#	55-52	55-52#
	55-56	55-56#	55-65	55-65#	55-70	55-70#	55-70	55-70#	55-70	55-70#	55-70#	55-70#	55-70#	55-71
	55-71#	55-85	55-85#	55-89	55-89#	55-93	55-93#	55-97	55-97#	55-101	55-101#	55-106	55-106#	55-110
	55-110#	55-114	55-114#	55-118	55-118#	55-122	55-122#	55-126	55-126#	55-130	55-130#	57-7	57-7	57-7
	57-7	57-7#	57-7#	57-7#	57-7#	57-7#	57-8	57-8#	61-11	61-11	61-11	61-11	61-11	61-11
	61-11#	61-11#	61-11#	61-11#	61-11#	61-11#	61-12	61-12#	61-22	61-22	61-22#	61-22#	61-28	61-28
	61-28	61-28	61-28#	61-28#	61-28#	61-28#	61-30	61-30#	62-20	62-20	62-20	62-20	62-20	62-20#
	62-20#	62-20#	62-20#	62-20#	62-21	62-21#	63-9	63-9	63-9	63-9	63-9	63-9	63-9#	63-9#
	63-9#	63-9#	63-9#	63-9#	65-13	65-13#	66-16	66-16	66-16#	66-16#	66-19	66-19#	66-21	66-21
	66-21#	66-21#	66-26	66-26#	66-26#	67-7	67-7#	67-7#	68-23	68-23#	68-23#	68-30	68-30#	68-30#
	69-1	69-1#	69-1#	69-11	69-11#	69-11#	69-14	69-14#	69-14#	70-6	70-6	70-6	70-6	70-6#
	70-6#	70-6#	70-6#	70-6#	70-7	70-7#	71-9	71-9#	71-25	71-25	71-25	71-25	71-25#	71-25#
	71-25#	71-25#	71-25#	72-36	72-36	72-36	72-36	72-36#	72-36#	72-36#	72-36#	72-36#	73-21	73-21
	73-21	73-21	73-21#	73-21#	73-21#	73-21#	74-8	74-8	74-8	74-8	74-8	74-8#	74-8#	74-8#
	74-8#	74-8#	78-22	78-22	78-22	78-22	78-22#	78-22#	78-22#	78-22#	78-22#	80-30	80-30	80-30
	80-30	80-30#	80-30#	80-30#	80-30#	80-30#	83-20	83-20#	83-21	83-21	83-21#	83-21#	83-26	83-26
	83-26#	83-26#	83-27	83-27#	102-12	102-12	102-12	102-12	102-12	102-12	102-12#	102-12#	102-12#	102-12#
	102-12#	102-14	102-14	102-14	102-14	102-14	102-14	102-14#	102-14#	102-14#	102-14#	102-14#	102-16	102-16
	102-16	102-16	102-16	102-16	102-16#	102-16#	102-16#	102-16#	102-16#	102-16#	102-18	102-18	102-18	102-18
	102-18	102-18#	102-18#	102-18#	102-18#	102-18#	105-20	105-20	105-20	105-20	105-20	105-20	105-20#	105-20#
	105-20#	105-20#	105-20#	105-20#	107-3	107-3	107-3	107-3	107-3#	107-3#	107-3#	107-3#	107-3#	111-21
	111-21#	111-29	111-29	111-29	111-29	111-29#	111-29#	111-29#	111-29#	111-29#	112-5	112-5	112-5	112-5
	112-5#	112-5#	112-5#	112-5#	112-5#	113-14	113-14#	114-21	114-21#	116-35	116-35	116-35	116-35	116-35#
	116-35#	116-35#	116-35#	116-35#	117-11	117-11	117-11	117-11	117-11#	117-11#	117-11#	117-11#	117-11#	118-12
	118-12#	118-27	118-27	118-27	118-27	118-27	118-27	118-27#	118-27#	118-27#	118-27#	118-27#	118-27#	118-30
	118-30	118-30#	118-30#	118-33	118-33	118-33	118-33	118-33#	118-33#	118-33#	118-33#	118-33#	120-44	120-44
	120-44	120-44	120-44#	120-44#	120-44#	120-44#	120-44#	121-27	121-27#	121-36	121-36	121-36	121-36	121-36#

MSMCLO	28-8	28-8#												
MSPOP	30-17	30-17#	31-14	31-14#	31-16	31-16#	55-16	55-16#	55-20	55-20#	55-24	55-24#	55-28	55-28#
	55-32	55-32#	55-36	55-36#	55-40	55-40#	55-44	55-44#	55-48	55-48#	55-52	55-52#	55-56	55-56#
	55-65	55-65#	55-71	55-71#	55-85	55-85#	55-89	55-89#	55-93	55-93#	55-97	55-97#	55-101	55-101#
	55-106	55-106#	55-110	55-110#	55-114	55-114#	55-118	55-118#	55-122	55-122#	55-126	55-126#	55-130	55-130#
	113-14	113-14#	114-21	114-21#	124-9	124-9#	125-38	125-38#	126-16	126-16#	135-104	135-104#	136-12	136-12#
	137-15	137-15#	137-17	137-17#	138-138	138-138#	138-139	138-139#	140-6	140-6#	142-10	142-10#	143-10	143-10#
MSPRIN	55-70	55-70#	102-12	102-12#	102-14	102-14#	102-16	102-16#	102-18	102-18#	138-52	138-52#		
MSPUSH	28-26	28-26#	30-10	30-10#	31-10	31-10#	32-3	32-3#	55-14	55-14#	55-18	55-18#	55-22	55-22#
	55-26	55-26#	55-30	55-30#	55-34	55-34#	55-38	55-38#	55-42	55-42#	55-46	55-46#	55-50	55-50#
	55-54	55-54#	55-58	55-58#	55-67	55-67#	55-73	55-73#	55-87	55-87#	55-91	55-91#	55-95	55-95#
	55-99	55-99#	55-103	55-103#	55-108	55-108#	55-112	55-112#	55-116	55-116#	55-120	55-120#	55-124	55-124#
	55-128	55-128#	113-10	113-10#	114-18	114-18#	124-5	124-5#	126-3	126-3#	126-10	126-10#	127-8	127-8#
MSPUT	136-10	136-10#	137-8	137-8#	138-3	138-3#	138-5	138-5#	139-3	139-3#	139-14	139-14#	141-12	141-12#
	55-70	55-70#	55-70#	61-11	61-11	61-11	61-11	61-11#	63-9	63-9	63-9	63-9	63-9#	102-12
	102-12	102-12	102-12#	102-14	102-14	102-14	102-14#	102-16	102-16	102-16	102-16#	102-18	102-18	102-18
	102-18#	105-20	105-20	105-20	105-20	105-20#	118-27	118-27	118-27	118-27	118-27#	127-59	127-59	127-59
	127-59	127-59#	138-52	138-52	138-52	138-52#								
MSPUT1	55-70	55-70	55-70#	55-70#	61-11	61-11	61-11	61-11	61-11#	61-11#	61-11#	61-11#	63-9	63-9
	63-9	63-9	63-9#	63-9#	63-9#	63-9#	102-12	102-12	102-12	102-12#	102-12#	102-12#	102-14	102-14
	102-14	102-14#	102-14#	102-14#	102-16	102-16	102-16	102-16#	102-16#	102-16#	102-18	102-18	102-18	102-18#
	102-18#	102-18#	105-20	105-20	105-20	105-20	105-20#	105-20#	105-20#	105-20#	105-20#	118-27	118-27	118-27
	118-27#	118-27#	118-27#	118-27#	127-59	127-59	127-59	127-59	127-59#	127-59#	127-59#	127-59#	138-52	138-52
	138-52	138-52#	138-52#	138-52#										
MSRADI	135-1	135-1#	138-20	138-20#	138-34	138-34#	140-1	140-1#	140-2	140-2#	140-3	140-3#	140-4	140-4#
	140-5	140-5#	142-1	142-1#	142-3	142-3#	142-5	142-5#	142-7	142-7#				
MSRBRO	66-21	66-21#	66-26	66-26#	67-7	67-7#	68-23	68-23#	68-30	68-30#	69-1	69-1#	69-11	69-11#
	69-14	69-14#	83-26	83-26#										
MSRNRO	127-39	127-39#	127-48	127-48#	127-50	127-50#	128-15	128-15#	131-4	131-4#				
MSETS	28-26	28-26#	30-10	30-10#	31-10	31-10#	32-3	32-3#	55-14	55-14#	55-18	55-18#	55-22	55-22#
	55-26	55-26#	55-30	55-30#	55-34	55-34#	55-38	55-38#	55-42	55-42#	55-46	55-46#	55-50	55-50#
	55-54	55-54#	55-58	55-58#	55-67	55-67#	55-73	55-73#	55-87	55-87#	55-91	55-91#	55-95	55-95#
	55-99	55-99#	55-103	55-103#	55-108	55-108#	55-112	55-112#	55-116	55-116#	55-120	55-120#	55-124	55-124#
	55-128	55-128#	113-10	113-10#	114-18	114-18#	124-5	124-5#	126-3	126-3#	126-10	126-10#	127-8	127-8#
MSSVC	136-10	136-10#	137-8	137-8#	138-3	138-3#	138-5	138-5#	139-3	139-3#	139-14	139-14#	141-12	141-12#
	55-16	55-16#	55-20	55-20#	55-24	55-24#	55-28	55-28#	55-32	55-32#	55-36	55-36#	55-40	55-40#
	55-44	55-44#	55-48	55-48#	55-52	55-52#	55-56	55-56#	55-65	55-65#	55-70	55-70#	55-71	55-71#
	55-85	55-85#	55-89	55-89#	55-93	55-93#	55-97	55-97#	55-101	55-101#	55-106	55-106#	55-110	55-110#
	55-114	55-114#	55-118	55-118#	55-122	55-122#	55-126	55-126#	55-130	55-130#	57-7	57-8	57-8#	61-11
	61-11#	61-12	61-12#	61-22	61-22#	61-28	61-30	61-30#	62-20	62-21	62-21#	63-9	63-9#	65-13
	65-13#	66-16	66-16#	66-19	66-19#	66-21	66-21#	66-26	66-26#	67-7	67-7#	68-23	68-23#	68-30
	68-30#	69-1	69-1#	69-11	69-11#	69-14	69-14#	70-6	70-7	70-7#	71-9	71-9#	71-25	72-36
	73-21	74-8	78-22	80-30	83-20	83-20#	83-21	83-21#	83-26	83-26#	102-12	102-12#	102-14	102-14#
	102-16	102-16#	102-18	102-18#	105-20	105-20#	107-3	111-21	111-21#	111-29	112-5	116-35	117-11	118-12
	118-12#	118-27	118-27#	118-30	118-30#	118-33	120-44	121-27	121-27#	121-36	122-5	127-10	127-10#	127-12
	127-12#	127-14	127-14#	127-16	127-16#	127-18	127-18#	127-39	127-39#	127-48	127-48#	127-50	127-50#	127-59
	127-59#	128-15	128-15#	131-4	131-4#	133-12	133-12#	133-14	133-14#	133-16	133-16#	134-3	134-4	134-4#
	134-8	134-9	134-9#	134-14	134-15	134-15#	134-20	134-21	134-21#	135-1	135-1#	135-104	135-104#	136-12
	136-12#	137-11	137-11#	137-15	137-15#	138-8	138-8#	138-10	138-11	138-11#	138-16	138-17	138-17#	138-20
	138-20#	138-22	138-22#	138-34	138-34#	138-52	138-52#	138-137	138-137#	138-138	138-138#			
MSTLAB	55-16#	55-20#	55-24#	55-28#	55-32#	55-36#	55-40#	55-44#	55-48#	55-52#	55-56#	55-65#	55-70#	55-71#
	55-85#	55-89#	55-93#	55-97#	55-101#	55-106#	55-110#	55-114#	55-118#	55-122#	55-126#	55-130#	57-7#	57-8#
	61-11#	61-12#	61-22#	61-28#	61-30#	62-20#	62-21#	63-9#	65-13#	66-16#	66-19#	66-21#	66-26#	67-7#
	68-23#	68-30#	69-1#	69-11#	69-14#	70-6#	70-7#	71-9#	71-25#	72-36#	73-21#	74-8#	78-22#	80-30#
	83-20#	83-21#	83-26#	102-12#	102-14#	102-16#	102-18#	105-20#	107-3#	111-21#	111-29#	112-5#	116-35#	117-11#
	118-12#	118-27#	118-30#	118-33#	120-44#	121-27#	121-36#	122-5#	127-10#	127-12#	127-14#	127-16#	127-18#	127-39#

[illegible]