

RABO

PDP-11 UDA DRV FMTR
CZUDEBO

AH-S837B-MC
FICHE 1 OF 1

SEP 1982
COPYRIGHT © 81-82
MADE IN USA



The main body of the document is a large, dense grid of data. Each cell in the grid contains a small table or set of data points, likely representing a detailed technical specification or a data matrix. The text is too small to be legible in this scan, but the layout is consistent across the entire page.

.REM ~

IDENTIFICATION

PRODUCT CODE: AC-S836B-MC
PRODUCT NAME: CZUDEBO UDA DISK FORMATTER
PRODUCT DATE: 14-APR-82
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: DALE KECK

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

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

COPYRIGHT (C) 1981,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

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 | 22 |
| 4.0 PERFORMANCE AND PROGRESS REPORTS | 25 |
| 5.0 TEST SUMMARIES | 26 |

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/RA80 spare HDA from RM80 format to RA80 format).
3. Restore - Format the disk using bad sector information obtained from a disk file on the XXDP+ system load device. This method is provided for use by manufacturing. No files are provided, nor any method of obtaining the files, at this time.

The format operation is performed by a Diagnostic Utilities and Protocol (DUP) program loaded into the 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, lets 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 RESTORE will result in an error. Also, a date of 1-JAN-70 will be written on the disk.

1.2 SYSTEM REQUIREMENTS

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

- PDP-11 Unibus processor
- 28K words of memory (minimum)
- Console terminal
- XXDP+ load media containing this program
- One or more UDA-50 subsystems

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

| | |
|---------|---|
| DISPLAY | Type a list of all device information |
| FLAGS | Type the state of all flags (see section 2.3) |
| ZFLAGS | Clear all flags (see section 2.3) |

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

2.2 SWITCHES -----

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

| 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:DDDD | Execute DDDD passes (DDDD = 1 to 64000) |
| /FLAGS:FLGS | Set specified flags. Flags are described in section 2.3. |
| /EOP:DDDD | Report end of pass message after every DDDD passes only. (DDDD = 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 be 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 |
|------|---|
| HCE | 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 |
| JAM | 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 (O) 172150 ?

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

VECTOR (O) 154 ?

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

BR LEVEL (D) 5 ?

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

UNIBUS BURST RATE (D) 63 ?

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

DRIVE NUMBER (D) 0 ?

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

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 UDA AT xxxxxx DRIVE xxx
(A) ?

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

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

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

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

2.7 EXTENDED P-TABLE DIALOGUE -----

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
# UNITS (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 ZUDEBO'
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 (O) 172150 ?

VECTOR (O) 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 Revectored LBNS
  2 Primary revectored LBNS
  0 Secondary/tertiary revectored 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 Revectored LBNS
  5 Primary revectored LBNS
  1 Secondary/tertiary revectored 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 Revectored LBNS
  2 Primary revectored LBNS
  0 Secondary/tertiary revectored 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 CUMLLATIVE 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 xxxxx 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 PRGGRAM 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.

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 UNIBUS
OR REPLACE UDA MODULE M7161

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:xx
UDA RESIDENT DIAGNOSTICS DETECTED FAILURE
UDASA CONTAINS 105154
REPLACE UDA MODULE M7162

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 is the Unibus interface board.
M7162 is the SDI interface board.

00022 CZUDE DVC FTL ERR 00022 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION
STEP BIT EXPECTED 004000
UDASA CONTAIN^r 000000
REPLACE UDA MOD M7161

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

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 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

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.

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

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

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

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

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

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

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

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

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

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

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

00037 CZUDE DVC FTL ERR 00037 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM
UDASA CONTAINS 100004
REPLACE UDA MODULE M7161

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.

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 problem.

Drive init timeout

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

Illegal response to start-up question

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

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 Revectored LBNS
  5 Primary revectored LBNS
  0 Secondary/tertiary revectored 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.

1
25
26
27
28
29
30
31
32
33
34

.SBTTL PROGRAM HEADER

BGNMOD

..++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--

POINTER BGNSW, BGNSFT, BGNSETUP

HEADER CZUDE,B,P,0,1,PRI07 ;FIELD SERVICE

002000
002000 103
002001 132
002002 125
002003 104
002004 105
002005 000
002006 000
002007 000
002010
002010 102
002011
002011 120
002012
002012 000001
002014
002014 000000
002016
002016 021064
002020
002020 021252
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 /B/
LSDEPO::
.ASCII /P/
LSUNIT::
.WORD T\$PTHV
L\$TIML::
.WORD 0
L\$HPCP::
.WORD L\$HARD
L\$SPCP::
.WORD L\$SOFT
L\$HPT?::
.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 003442
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 000000
002072
002072 000000
002074
002074 000000
002076
002076 003464
002100
002100 104035
002102
002102 000000
002104
002104 017036
002106
002106 020454
002110
002110 020452
002112
002112 017030
002114
002114 000000
002116
002116 000000
002120
002120 000000

LSEF::
 .WORD 0
 .WORD 0
LSSPC::
 .WORD 0
LSDEVP::
 .WORD LSDVTYP
LSREPP::
 .WORD 0
LSEXP4::
 .WORD 0
LSEXP5::
 .WORD 0
LSAUT::
 .WORD 0
LSDUT::
 .WORD 0
LSLUN::
 .WORD 0
LSDESP::
 .WORD LSDESC
LSLOAD::
 EMT ESLOAD
LSETP::
 .WORD 0
LSICP::
 .WORD LSINIT
LSCCP::
 .WORD LSCLEAN
LSACP::
 .WORD LSAUTO
LSPRT::
 .WORD LSPROT
LSTEST::
 .WORD 0
LSDLY::
 .WORD 0
LSHIME::
 .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 020462

.WORD 1
LSDISPATCH: :
.WORD T1

1
2
3
4
5
6
7
8
9

.SBTTL DEFAULT HARDWARE P-TABLE

:++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
: AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
:--

| | | | | | | | | |
|----|--------|--------|-------|--------|--|--|----------|----------------------|
| 10 | 002126 | | BGNHW | DFPTBL | | | | |
| | 002126 | 000005 | | | | | .WORD | L10000-L\$HW/2 |
| | 002130 | | | | | | L\$HW:: | |
| | 002130 | | | | | | DFPTBL:: | |
| 11 | | | | | | | | |
| 12 | 002130 | 172150 | .WORD | 172150 | | | : | UNIBUS ADDRESS |
| 13 | 002132 | 000154 | .WORD | 154 | | | : | VECTOR ADDRESS |
| 14 | 002134 | 000005 | .WORD | 5. | | | : | BR LEVEL |
| 15 | 002136 | 000077 | .WORD | 63. | | | : | UNIBUS BURST RATE |
| 16 | 002140 | 000000 | .WORD | 0. | | | : | LOGICAL DRIVE NUMBER |
| 17 | 002142 | | ENDHW | | | | | |
| | 002142 | | | | | | | L10000: |

```
1          .SBTTL  SOFTWARE P-TABLE
2
3          :++
4          : THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
5          : PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
6          : SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
7          : AT RUN TIME.
8          :--
9
10         002142          BGNSW  SFPTBL
11         002142 000001
12         002144
13         002144          L$$W: .WORD  L10001-L$$W/2
14         002146          SFPTBL:
15         002146
16         002146          :OFFSE
17         002146          : 0.  YES/NO  USE
18         002146          ANSWERS
19         002146          L10001:
20         002146          ENDMOD
```


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

.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

EQUALS

: BIT DIFINITIONS

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

| | |
|--------|--------------|
| 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      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

11          000015      CR=      15
```

;VALUE TO PASS TO PRINT MACRO TO END LINE

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

;MACRO DEFINITIONS FOR GLOBAL EQUATES

;THESE MACROS ARE USED TO DEFINE INDEXES INTO A TABLE

;CALLING SEQUENCE MUST BE

```
TABLE  
ITEM NAME BYTES  
ITEM NAME BYTES  
ITEM NAME BYTES  
END SIZE
```

;TABLE DEFINES THAT A TABLE IS ABOUT TO BE DEFINED AND END TERMINATES THE DEFINITION.
;ANY NUMBER OF ITEM LINES CAN APPEAR. NAME IS THE NAME OF THE SYMBOL BEING EQUATED TO
;THE INDEX. THE INDEX ALWAYS STARTS AT ZERO. BYTES SPECIFIES THE SIZE OF THE VALUE TO BE
;STORED AT THAT INDEX IN BYTES. THE SIZE ARGUMENT TO THE END STATEMENT IS OPTIONAL, IT
;BE EQUATED TO THE SIZE OF THE TABLE IN BYTES. THE SYMBOL TINDEX IS USED TO KEEP TRACK
;OF THE INDEX VALUE AND WILL BE EQUAL TO THE SIZE OF THE TABLE AFTER THE END STATEMENT.

```
.MACRO TABLE  
TINDEX=0
```

```
.ENDM
```

```
.MACRO ITEM NAME BYTES  
NAME=TINDEX  
TINDEX=TINDEX+BYTES
```

```
.ENDM
```

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

```
.ENDM
```

```

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

```
1          ;UDASA REGISTER STEP THREE READ BITS
2
3          000177      SA.VCE= 000177      ;INTERRUPT VECTOR ECHO
4          000200      SA.INE= 000200      ;INTERRUPT ENABLE ECHO
5          000400      SA.NVE= 000400      ;VECTOR NOT PROGRAMMABLE
6          ;          003000      ;RESERVED
7
8          ;UDASA REGISTER STEP THREE WRITE BITS
9
10         ;          077777      ;HIGH ORDER MESSAGE RING BYTE ADDRESS
11         100000      SA.TST= 100000      ;PURGE POLE TEST ENABLE
12
13         ;UDASA REGISTER STEP FOUR READ BITS
14
15         000377      SA.MCV= 000377      ;UDA MICROCODE VERSION
16         ;          003400      ;RESERVED
17
18         ;UDASA REGISTER STEP FOUR WRITE BITS
19
20         000001      SA.GO= 000001      ;GO BIT TO START UDA FIRMWARE
21         000002      SA.LFC= 000002      ;LAST FAILURE CODE REQUEST
22         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         000002      OP.ESP= 2          ;DUP EXECUTE SUPPLIED PROGRAM
29         000003      OP.ELP= 3          ;DUP EXECUTE LOCAL PROGRAM
30         000004      OP.SD= 4           ;DUP SEND STUD DATA
31         000005      OP.RSD= 5          ;DUP RECEIVE STUD DATA
32
33         ;NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
34         ;PACKET FLAG TO THE COMMAND OPCODE. FOR EXAMPLE, A READ COMMAND'S END PACKET
35         ;CONTAINS THE VALUE OP.RD+OP.END IN ITS OPCODE FIELD. THE INVALID COMMAND END
36         ;PACKET CONTAINS JUST THE END PACKET FLAG (I.E., OP.END) IN ITS OPCODE FIELD.
37         ;THE SERIOUS EXCEPTION END PACKET CONTAINS THE SUM OF THE END PACKET FLAG
38         ;PLUS THE SERIOUS EXCEPTION OPCODE SHOWN ABOVE (I.E., OP.SEX+OP.END) IN ITS
39         ;OPCODE FIELD.
40
41         ;COMMAND OPCODE BITS 3 THROUGH 5 INDICATE THE COMMAND CLASS, WHICH IS ENCODED
42         ;AS FOLLOWS:
43         ; 000 IMMEDIATE COMMANDS
44         ; 001 SEQUENTIAL COMMANDS
45         ; 010 NON-SEQUENTIAL COMMANDS THAT DO NOT INCLUDE A BUFFER DESCRIPTOR
46         ; 100 NON-SEQUENTIAL COMMANDS THAT DO INCLUDE A BUFFER DESCRIPTOR
    
```



```
1          ;COMMAND MODIFIERS
2
3          ;      = 020000
4          040000 MD.CMP= 040000
5          100000 MD.EXP= 100000
6          010000 MD.ERR= 010000
7          004000 MD.SCH= 004000
8          002000 MD.SCL= 002000
9          000100 MD.SEC= 000100
10         000400 MD.SER= 000400
11         000200 MD.SSH= 000200
12         000100 MD.WBN= 000100
13         000400 MD.WBV= 000400
14         000020 MD.SEQ= 000020
15         000001 MD.SPD= 000001
16         000001 MD.FEU= 000001
17         000002 MD.VOL= 000002
18         000001 MD.NXU= 000001
19         000001 MD.RIP= 000001
20         000002 MD.IMF= 000002
21         000004 MD.SWP= 000004
22         000010 MD.CWB= 000010
23         000001 MD.PRI= 000001
24
25         ;END PACKET FLAGS
26
27         000200 EF.BBR= 000200
28         000100 EF.BBU= 000100
29         000040 EF.LOG= 000040
30         000020 EF.SEX= 000020
31
32         ;CONTROLLER FLAGS
33
34         000200 CF.ATN= 000200
35         000100 CF.MSC= 000100
36         000040 CF.OTH= 000040
37         000020 CF.THS= 000020
38         000002 CF.SHD= 000002
39         000001 CF.576= 000001
```

```
;CLEAR SERIOUS EXCEPTION
;COMPARE
;EXPRESS REQUEST
;FORCE ERROR
;SUPPRESS CACHING (HIGH SPEED)
;SUPPRESS CACHING (LOW SPEED)
;SUPPRESS ERROR CORRECTION
;SUPPRESS ERROR RECOVERY
;SUPPRESS SHADOWING
;WRITE-BACK (NON-VOLATILE)
;WRITE BACK (VOLATILE)
;WRITE SHADOW SET ONE UNIT AT A TIME
;SPIN-DOWN
;FLUSH ENTIRE UNIT
;VOLATILE ONLY
;NEXT UNIT
;ALLOW SELF DESTRUCTION
;IGNORE MEDIA FORMAT ERROR
;SET WRITE PROTECT
;CLEAR WRITE-BACK DATA LOST
;PRIMARY REPLACEMENT BLOCK
```

```
;BAD BLOCK REPORTED
;BAD BLOCK UNREPORTED
;ERROR LOG GENERATED
;SERIOUS EXCEPTION
```

```
;ENABLE ATTENTION MESSAGES
;ENABLE MISCELLANEOUS ERROR LOG MESSAGES
;ENABLE OTHER HOST'S ERROR LOG MESSAGES
;ENABLE THIS HOST'S ERROR LOG MESSAGES
;SHADOWING
;576 BYTE SECTORS
```

GLOBAL EQUATES SECTION

```

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         ;
17         000000 P.CRF= 0. ;GENERIC COMMAND PACKET OFFSETS:
18         000004 P.UNIT= 4. ;COMMAND REFERENCE NUMBER
19         000010 P.OPCD= 8. ;UNIT NUMBER
20         000012 P.MOD= 10. ;OPCODE
21         000014 P.BCNT= 12. ;MODIFIERS
22         000020 P.BUFF= 16. ;BYTE COUNT
23         000020 P.UADR= 16. ;BUFFER DESCRIPTOR
24         000034 P.LBN= 28. ;UNIBUS ADDRESS OF BUFFER DESCRIPTOR
25         ; ;LOGICAL BLOCK NUMBER
26
27         000014 P.OTRF= 12. ;ABORT AND GET COMMAND STATUS COMMAND PACKET OFFSETS:
28         ; ;OUTSTANDING REFERENCE NUMBER
29
30         ;
31         000016 P.UNFL= 14. ;ONLINE AND SET UNIT CHARACTERISTICS COMMAND PACKET OFFSETS:
32         000020 P.HSTI= 16. ;UNIT FLAGS
33         000034 P.ELGF= 28. ;HOST IDENTIFIER / RESERVED
34         000040 P.SHUN= 32. ;ERROR LOG FLAGS
35         000042 P.CPSP= 34. ;SHADOW UNIT
36         ; ;COPY SPEED
37         000014 P.RBN= 12. ;REPLACE COMMAND PACKET OFFSETS:
38         ; ;REPLACEMENT BLOCK NUMBER
39
40         ;
41         000014 P.VRSN= 12. ;SET CONTROLLER CHARACTERISTICS COMMAND PACKET OFFSETS:
42         000016 P.CNTF= 14. ;MSCP VERSION
43         000020 P.HTMO= 16. ;CONTROLLER FLAGS
44         000022 P.USEF= 18. ;HOST TIMEOUT
45         000024 P.TIME= 20. ;USE FRACTION
46         ; ;QUAD-WORD TIME AND DATE
47
48         ;
49         000034 P.RGID= 28. ;MAINTENANCE READ AND MAINTENANCE WRITE COMMAND PACKET OFFSETS:
50         000040 P.RGOF= 32. ;REGION ID
51         ; ;REGION OFFSET
52
53         ;
54         000024 P.DMDT= 20. ;EXECUTE SUPPLIED PROGRAM COMMAND PACKET OFFSETS:
55         000034 P.OVRL= 28. ;DMDT TERMINAL ADDRESS (MAINT WRITE ONLY)
56         ; ;BUFFER DESCRIPTOR FOR OPERLAYS

```

| | | | |
|----|--------|---------------------|--|
| 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
24         000002      DF.LCL= 002        ;SUPPLIED PROGRAM COMMAND
25
26         000001      DF.SA= 001         ;SET IF THIS DUST HAS A LOCAL LOAD MEDIA FOR LOADING
27
28
29
30         ;DUP MESSAGE TYPES
31
32         010000      DU.QUE = 10000     ;QUESTION
33         020000      DU.DFL = 20000     ;DEFAULT QUESTION
34         030000      DU.INF = 30000     ;INFORMATION
35         040000      DU.TER = 40000     ;TERMINATOR
36         050000      DU.FTL = 50000     ;FATAL ERROR
37         060000      DU.SPC = 60000     ;SPECIAL
38
39         170000      DU.TYP= 170000     ;MESSAGE TYPE FIELD
40
41         ;DM PROGRAM HEADER DEFINITIONS
42
43         000000      DMTRLN= 0          ;OFFSET TO SIZE OF PROGRAM NEEDING DOWNLINE LOAD
44         000004      DMOVRL= 4          ;OFFSET TO SIZE OF OVERLAY
45         000021      DMTMO= 21          ;TIMEOUT VALUE IN SECONDS (ONE BYTE)
46         000040      DMMAIN= 40         ;OFFSET TO FIRST WORD OF MAIN PROGRAM
47         001000      DMFRST= 1000       ;ADDRESS IN DM FILE CONTAINING FIRST BYTE OF HEADER
    
```

```

1          ;CONTROLLER TABLE DEFINITIONS
2          ;
3          ;ONE TABLE WILL BE SET UP BY INITIALIZE SECTION FOR EACH UDA SELECTED
4          ;FOR TESTING. TABLES ARE CONTIGUOUS. THE END OF THE TABLES IS
5          ;MARKED BY A WORD OF ZEROS.
6          ;
7          ;THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF CTABS.
8          ;THE NUMBER OF TABLES IS CONTAINED IN CTRLRS.
9
10         TABLE          ;START A TABLE DEFINITION
11
12         ITEM C.UADR      2          ;UNIBUS ADDRESS OF UDAIP REGISTER
13         ITEM C.UNIT      2
14         000077          ; LOGICAL UNIT NUMBER (FIRST)
15         100000          ; SET WHEN NOT AVAILABLE FOR TESTING
16         ITEM C.VEC      2
17         000777          ; VECTOR ADDRESS
18         007000          ; BR LEVEL
19         ITEM C.BST      2          ; BURST LEVEL
20         ITEM C.JSR      2          ; INTERRUPT SERVICE ROUTINE FOR CONTROLLER
21         ITEM C.JAD      2          ; THESE TWO WORDS LOADED WITH [JSR R0,UDASRV]
22         ITEM C.FLG      2          ; FLAGS
23         000002          ; DM PROGRAM RUNNING
24         000004          ; COMMAND ISSUED, WAITING FOR RESPONSE
25         000010          ; MESSAGE RESPONSE RECEIVED
26
27         000020          ; WHENEVER THIS BIT IS SET, CT.CMD IS CLEARED
28
29
30         CT.STA= BITS    5          ; BUFFER HAS BEEN GIVEN TO UDA FOR REQUEST
31         CT.TM1= BIT6    1          ; SET WHENEVER READ STUD DATA COMMAND
32
33
34         CT.TM2= BIT7    1          ; GIVEN TO UDA
35         ITEM C.RING      2          ; GET DUST STATUS COMMAND HAS BEEN SENT
36         ITEM C.DR0      2          ; ONE TIMEOUT PERIOD HAS EXPIRED BETWEEN SEND OR
37         ITEM C.DR1      2          ; RECEIVE DATA RESPONSE
38         ITEM C.DR2      2          ; SECOND TIMEOUT HAS EXPIRED
39         ITEM C.DR3      2          ; RING BUFFER ADDRESS
40         ITEM C.DR4      2          ; POINTER TO DRIVE TABLES
41         ITEM C.DR5      2          ; IF ZERO, NO DRIVE TABLE EXISTS
42         ITEM C.DR6      2
43         ITEM C.DR7      2
44         ITEM C.TO      2          ; TIMEOUT COUNTER
45         ITEM C.TOH      2          ; (TWO WORDS)
46         ITEM C.TOT      2          ; DUP PROGRAM TIMEOUT VALUE IN SECONDS
47         ITEM C.PRI      4          ; DUP PROGRAM PROGRESS INDICATOR
48         ITEM C.REF      2          ; COMMAND REFERENCE NUMBER
49         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.UINT= 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
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

```
:PRINT CHARACTER  
: ARGUMENT MUST BE SOURCE STATEMENT TO MOVE CHARACTER TO PRINT (MOV ARG,R0)  
: EX: 'PRINT R1' WILL PRINT THE CHARACTER IN R1  
: SPECIAL CASE: 'PRINT #CR' WILL PRINT END OF LINE SEQUENCE  
: THE PRINTING IS DONE AT THE MODE OF THE LAST PRINT LINE CALL  
: IE., PNTX, PNTB, PNTX, PNTS  
  
.MACRO PRINT ARG1  
  .IF DIF <ARG1>,R0  
    .LIST  
    .NLIST  
    .ENDC  
    .LIST  
    .NLIST  
  .ENDM  
:PROCESSING MACRO FOR NEXT SET OF FORMATTED MESSAGE MACROS  
.MACRO PNT... RTN,ADR,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
  PNT.CT=0  
  .IRP AA,<ARG8,ARG7,ARG6,ARG5,ARG4,ARG3,ARG2,ARG1>  
    .IF NB,<AA>  
      .LIST  
      .NLIST  
      PNT.CT=PNT.CT+2  
    .ENDC  
  .ENDM  
  .LIST  
  .NLIST  
  .ENDM  
  .LIST  
  .NLIST  
  .ENDM
```

MOV B ARG1,R0

CALL CPNT

MOV AA,-(SP)

JSR R1,RTN
.WORD ADR
.WORD PNT.CT

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

```
:PRINT FORMATTED MESSAGE MACROS  
: USE THESE MACROS TO PRINT A FORMATTED MESSAGE  
: FIRST ARGUMENT MUST BE ADDRESS OF FIRST CHARACTER OF MESSAGE STRING  
: TO BE PUT INTO WORD (.WORD ARG)  
: UP TO 8 SOURCE STATEMENTS MAY FOLLOW TO SPECIFY PARAMETERS TO BE  
: USED BY THE FORMAT  
  
.MACRO PNTF ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
PNT... LPNTF ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
.ENDM  
.MACRO PNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
PNT... LPNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
.ENDM  
.MACRO PNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
PNT... LPNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
.ENDM  
.MACRO PNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
PNT... LPNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
.ENDM  
.MACRO PNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
PNT... LPNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8  
.ENDM
```

```

1      .SBTTL GLOBAL DATA SECTION
2
3      :++
4      : THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
5      : IN MORE THAN ONE TEST.
6      :--
7
8 002146 FFREE:: .BLKW 1 ;FIRST FREE WORD IN MEMORY
9 002150 FSIZE:: .BLKW 1 ;SIZE OF FREE MEMORY IN WORDS
10 002152 FMEM: .BLKW 1 ;COPY OF FFREE AT END OF INIT SECTION
11 002154 FMEMS: .BLKW 1 ;COPY OF FSIZE AT END OF INIT SECTION
12 002156 CTABS:: .BLKW 1 ;START OF CONTROLLER TABLE STORAGE
13 002160 CTRLRS: .BLKW 1 ;COUNT OF UDA CONTROLLERS IN PTABLES
14 002162 TSTTAB: .BLKW 1 ;POINTER TO FIRST CONTROLLER TABLE UNDER TEST
15      .GLOBL UDAFM
16 002164 000000G DMPROG: .WORD UDAFM ;START ADDRESS OF DM PROGRAM
17 002166 DMEND: .BLKW 1 ;END ADDRESS OF DM PROGRAM(FIRST FREE MEMORY ADR)
18 002170 DMENDS: .BLKW 1 ;FREE MEMORY SIZE FROM END OF DM PROGRAM
19 002172 URUN: .BLKW 1 ;NUMBER OF UNITS TO RUN AT ONE TIME
20 002174 URNING: .BLKW 1 ;NUMBER OF UNITS STILL RUNNING
21 002176 UCNT: .BLKW 1 ;COUNTER OF UNITS UNDER TEST
22 002200 UFREEZ: .BLKW 1 ;FREEZE ON UNIT WHEN NOT ZERO
23 002202 NXMAD: .BLKW 1 ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
24 002204 000000 FDATA: .WORD 0
25 002206 FCTBUF: .BLKB 512. ;STORAGE FOR FCT BLOCK
26 003206 FCTNUM: .BLKW 1 ;FCT BLOCK NUMBER
27 003210 MODE: .BLKW 1 ;MODE WORD, SAME BIT DEFS AS SO.BIT
28
29      :CLOCK CONTROL
30
31 003212 000000 KW.CSR: .WORD 0 ;CSR OF CLOCK
32 003214 KW.BRL: .BLKW 1 ;BR LEVEL
33 003216 KW.VEC: .BLKW 1 ;VECTOR
34 003220 KW.HZ: .BLKW 1 ;HERTZ (50. OR 60.)
35 003222 KW.EL: .BLKW 2 ;ELAPSED TIME
36
37 003226 014526 PTYPE: .WORD PF ;PRINT TYPE
38 003230 000 000 ERRCHR: .BYTE 0,0 ;FIRST BYTE LOADED WITH OUTPUT CHARACTER
39 003232 000000 NULL: .WORD 0 ;USED TO PRINT A NULL CHARACTER
40 003234 000000 FNAME: .WORD 0 ;SPACE FOR DATA FILE NAME
41 003236 .BLKB 10.
    
```

```

1 003250          TEMP: .BLKB 22.          ;USED TO GET ANSWER FROM GMANID CALL
2 003276      061    055    112 DATE1: .ASCIZ\1-JAN-70\    ;DEFAULT DATE
3 003307          .BLKB 3
4 003312      000000    DATE0: .WORD 0 ;DATE STRING IN FORMATTER FORMAT
5 003314          .BLKB 10.          ;(FIRST WORD ZERO SAYS NO DATE HERE YET)
6 003326      061    070    064 HIGHEST: .ASCIZ\18446744073709551615\    ;HIGHEST DISK SERIAL NUMBER
7 003353      104    105    103 MONTHS: .ASCII\DEC\    ;NAME OF MONTHS
8 003356      116    117    126 .ASCII\NOV\
9 003361      117    103    124 .ASCII\OCT\
10 003364     123    105    120 .ASCII\SEP\
11 003367     101    125    107 .ASCII\AUG\
12 003372     112    125    114 .ASCII\JUL\
13 003375     112    125    116 .ASCII\JUN\
14 003400     115    101    131 .ASCII\MAY\
15 003403     101    120    122 .ASCII\APR\
16 003406     115    101    122 .ASCII\MAR\
17 003411     106    105    102 .ASCII\FEB\
18 003414     112    101    116 .ASCII\JAN\
19 003417     037          DAYS: .BYTE 31.          ;NUMBER OF DAYS IN EACH MONTH
20 003420     035          .BYTE 29.
21 003421     037          .BYTE 31.
22 003422     036          .BYTE 30.
23 003423     037          .BYTE 31.
24 003424     036          .BYTE 30.
25 003425     037          .BYTE 31.
26 003426     037          .BYTE 31.
27 003427     036          .BYTE 30.
28 003430     037          .BYTE 31.
29 003431     036          .BYTE 30.
30 003432     037          .BYTE 31.
31 003433     061    071    000 YEAR19: .ASCIZ\19\
32 003436     062    060    000 YEAR20: .ASCIZ\20\
33          .EVEN
    
```

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

.SBTTL GLOBAL TEXT SECTION

:++
: THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: MORE THAN ONE TEST.
:--

:
: NAMES OF DEVICES SUPPORTED BY PROGRAM

:
: DEVTYP <UDA-50 CONTROLLER>

003442
003442
003442 125 104 101

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

:
: TEST DESCRIPTION

:
: DESCRIPT <CZUDEBP UDA-50 DISK DRV FORMATTER>

003464
003464
003464 103 132 125

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

1
2 ;UNFORMATTED MESSAGES
3 003526 105 116 124 DATEQ: .ASCIZ\ENTER DATE AS DD-MMM-YY\
4 003556 040 106 117 FILNAQ: .ASCIZ\ FOR DISK TO BE FORMATTED\
5 003610 040 000 SERNO: .ASCIZ\ \

```

1          ; FORMAT STATEMENTS USED IN PRINT CALLS
2
3 003612    045    124    000  ERRONE: .ASCIZ\%T\
4 003615    045    116    000  ERRNL: .ASCIZ\%N\
5 003620    042    040    040  RNTIM: .ASCIZ\'\' RUNTIME 'D16'':\
6 003643    104    071    042  RNTIM1: .ASCIZ\D9'':\
7 003651    104    071    000  RNTIM2: .ASCIZ\D9\
8 003654    042    040    040  ERRME1: .ASCIZ\'\' * * * ERROR PROCESSING MESSAGE STRING * * *\
9 003743    116    042    125  MESSG: .ASCIZ\N'UNIT 'D6'' UDA AT 'D16'' DRIVE 'D9S\
10 004007   042    116    117  NOCLOCK: .ASCIZ\NO LINE CLOCK AVAILABLE FOR TIMING EVENTS'\
11
12 004064   042    110    117  BASNO: .ASCIZ\'HOST PROGRAM'\
13 004103   042    040    040  BASL2: .ASCIZ\'\' UDA AT 'D16\
14 004122   042    040    040  BASL3: .ASCIZ\'\' DRIVE 'D9\
15 004137   000                    BAS: .BYTE 0 ;NULL TO PRINT NOTHING
16
17 004140   122    066    122  BASLN: .ASCIZ\R6R6R6R6\ ;USED TO PRINT BASIC LINE OF ERROR MESSAGE
18 004151   116    042    123  SERNUM: .ASCIZ\N'SERIAL NUMBER FOR UNIT 'D6'' UDA AT 'D16'' DRIVE 'D9\
    
```

| | | | | | | |
|----|--------|-----|-----|-----|-------|--|
| 1 | 004236 | | | | X1A: | |
| 2 | 004236 | | | | X2A: | |
| 3 | 004236 | | | | X3A: | |
| 4 | 004236 | 042 | 111 | 040 | X8A: | .ASCIZ\''I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS'\ |
| 5 | 004335 | 122 | 065 | 122 | X1: | .ASCIZ\R5R6'UDA HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE'\ |
| 6 | 004431 | 122 | 065 | 122 | X2: | .ASCIZ\R5R6'TWO UNITS SELECT THE SAME DRIVE'\ |
| 7 | 004500 | 122 | 065 | 122 | X3: | .ASCIZ\R5R6'MORE THAN EIGHT DRIVES SELECTED ON THIS UDA'\ |
| 8 | 004563 | 122 | 064 | 042 | X4: | .ASCII\R4'NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED'\ |
| 9 | 004654 | 042 | 120 | 114 | | .ASCIZ\''PLEASE START PROGRAM OVER AND TEST FEWER UNITS AT A TIME'\ |
| 10 | 004750 | 122 | 065 | 122 | X8: | .ASCIZ\R5R6'TWO UDA'S USE THE SAME VECTOR'\ |
| 11 | 005015 | 122 | 064 | 042 | X9: | .ASCII\R4'ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE.''\ |
| 12 | 005120 | 042 | 120 | 114 | | .ASCIZ\''PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK.''\ |
| 13 | 005207 | 122 | 064 | 042 | X10: | .ASCIZ\R4'THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE.''\ |
| 14 | 005306 | 122 | 065 | 042 | X20: | .ASCII\R5'MEMORY ERROR TRYING TO READ UDA REGISTERS'\ |
| 15 | 005364 | 042 | 103 | 110 | | .ASCII\''CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7161'\ |
| 16 | 005452 | 042 | 117 | 122 | | .ASCII\''OR UNIBUS'\ |
| 17 | 005466 | 042 | 117 | 122 | | .ASCIZ\''OR 'R7'\ |
| 18 | 005476 | 122 | 065 | 042 | X21: | .ASCII\R5'UDA RESIDENT DIAGNOSTICS DETECTED FAILURE'NR8\ |
| 19 | 005556 | 042 | 122 | 105 | | .ASCIZ\''REPLACE UDA MODULE M716'D2N\ |
| 20 | 005613 | 122 | 065 | 042 | X22: | .ASCII\R5'STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION'\ |
| 21 | 005714 | 042 | 123 | 124 | | .ASCIZ\''STEP BIT EXPECTED 'D16NR8R7\ |
| 22 | 005751 | 122 | 065 | 042 | X23A: | .ASCII\R5'UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION'\ |
| 23 | 006063 | 104 | 071 | 042 | | .ASCII\D9'' WORDS WERE TO BE CLEARED STARTING AT ADDRESS 'D16N\ |
| 24 | 006151 | 042 | 106 | 111 | | .ASCII\''FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):'\ |
| 25 | 006226 | 123 | 066 | 042 | | .ASCIZ\S6'ADDRESS'S4'CONTENTS'\ |
| 26 | 006257 | 123 | 067 | 117 | X23B: | .ASCIZ\S7016S5016N\ |
| 27 | 006273 | 122 | 065 | 042 | X24: | .ASCII\R5'UDASA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION'\ |
| 28 | 006406 | 042 | 120 | 125 | | .ASCIZ\''PURGE/POLE DIAGNOSTICS WERE REQUESTED'NR8R7\ |
| 29 | 006463 | 122 | 065 | 042 | X25: | .ASCII\R5'UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION'\ |
| 30 | 006577 | 042 | 040 | 040 | | .ASCIZ\'' UDASA EXPECTED 'D16NR8R7\ |
| 31 | 006634 | 122 | 065 | 042 | X30: | .ASCIZ\R5'UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE RUNNING DM PROGRAM'NR8\ |
| 32 | 006747 | 122 | 065 | 042 | X31: | .ASCIZ\R5'DUP PROGRAM IS HUNG'\ |
| 33 | 007000 | 122 | 065 | 042 | X32: | .ASCIZ\R5'MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER'\ |
| 34 | 007111 | 122 | 065 | 042 | X36: | .ASCII\R5'NO INTERRUPT RECEIVED FROM UDA FOR 30 SECONDS'\ |
| 35 | 007173 | 042 | 127 | 110 | | .ASCIZ\''WHILE LOADING DM PROGRAM'\ |
| 36 | 007227 | 122 | 065 | 042 | X37: | .ASCIZ\R5'UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM'NR8R7\ |
| 37 | 007344 | 122 | 065 | 042 | X100: | .ASCIZ\R5'DUP PROGRAM ASKED UNEXPECTED QUESTION ('D12')'\ |
| 38 | 007427 | 122 | 065 | 042 | X101: | .ASCIZ\R5'DUP PROGRAM REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION'\ |

| | | | | | | |
|----|--------|-----|-----|-----|---------|---|
| 1 | 007532 | 042 | 115 | 105 | XMSG1: | .ASCIZ\MESSAGE BUFFER CONTAINS:'N\ |
| 2 | 007566 | 123 | 063 | 117 | XMSG2: | .ASCIZ\S3016S1016S1016S1016S1016S1016S1016N\ |
| 3 | 007633 | 122 | 065 | 042 | XPKT1: | .ASCII\R5'RESPONSE PACKET FROM UDA DOES NOT CONTAIN EXPECTED DATA'N\ |
| 4 | 007727 | 042 | 105 | 111 | | .ASCII\EITHER UDA RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED CORRECTLY'N\ |
| 5 | 010037 | 123 | 063 | 042 | | .ASCIZ\S3'COMMAND PACKET SENT'S6'RESPONSE PACKET RECEIVED'N\ |
| 6 | 010124 | 123 | 066 | 117 | XPKT2: | .ASCIZ\S6016S1016S14016S1016N\ |
| 7 | 010153 | 042 | 040 | 040 | XSA: | .ASCIZ\ UDASA CONTAINS '016N\ |
| 8 | 010204 | 042 | 122 | 105 | XFRU: | .ASCIZ\REPLACE UDA MODULE M7161'N\ |
| 9 | | | | | | |
| 10 | | | | | | |
| 11 | 010240 | 045 | 101 | 111 | SERNX: | .ASCIZ\XAINPUT ERROR. ANSWER WITH DECIMAL NUMBER LO= 0 HI= %T\ |
| 12 | 010330 | 042 | 111 | 116 | DATEX: | .ASCIZ\INPUT ERROR.'N\ |
| 13 | 010347 | 042 | 116 | 101 | FILNAM: | .ASCIZ\NAME OF FILE CONTAINING BAD SECTOR INFORMATION'N\ |
| 14 | | | | | | .EVEN |


```

1      .SBTTL GLOBAL ERROR REPORT SECTION
2
3      ;++
4      ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
5      ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PRINTB
6      ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
7      ;--
8      177777      SVCINS= -1      : LIST INSTRUCTIONS, SHIFTED RIGHT
9      177777      SVCTST= -1     : LIST TEST TAGS, SHIFTED RIGHT
10     177777      SVCSUB= -1     : LIST SUBTEST TAGS, SHIFTED RIGHT
11     177777      SVCGBL= -1    : LIST GLOBAL TAGS, SHIFTED RIGHT
12     177777      SVCTAG= -1    : LIST OTHER TAGS, SHIFTED RIGHT
13
14     010430      BGNMSG ERR001
15     010430      PNTB X1,#X1A
16     010430      012746      004236      MOV #X1A,-(SP)
17     010434      004137      014660      JSR R1,LPNTB
18     010440      004335      .WORD X1
19     010442      000002      .WORD PNT.CT
20     010444      ENDMSG
21
22     010446      BGNMSG ERR002
23     010446      PNTB X2,#X2A
24     010446      012746      004236      MOV #X2A,-(SP)
25     010452      004137      014660      JSR R1,LPNTB
26     010456      004431      .WORD X2
27     010460      000002      .WORD PNT.CT
28     010462      ENDMSG
29
30     010464      BGNMSG ERR003
31     010464      PNTB X3,#X3A
32     010464      012746      004236      MOV #X3A,-(SP)
33     010470      004137      014660      JSR R1,LPNTB
34     010474      004500      .WORD X3
35     010476      000002      .WORD PNT.CT
36     010500      ENDMSG
37
38     010502      BGNMSG ERR004
39     010502      PNTB X4
40     010502      004137      014660      JSR R1,LPNTB
41     010506      004563      .WORD X4
42     010510      000000      .WORD PNT.CT
43     010512      ENDMSG
44
45     010514      BGNMSG ERR009
46     010514      PNTB X9
47     010514      004137      014660      JSR R1,LPNTB
48     010520      005015      .WORD X9
49     010522      000000      .WORD PNT.CT
50     010524      ENDMSG
51
52     010526      BGNMSG ERR010
53     010526      PNTB X10
54     010526      004137      014660      JSR R1,LPNTB
55     010532      005207      .WORD X10
56     010534      000000      .WORD PNT.CT
57     010536      ENDMSG
    
```

| | | | | | | |
|----|--------|--------|---------------|---------|--------------------|------------------|
| 37 | | | | | | |
| 38 | 010540 | | | BGNMSG | ERR008 | |
| 39 | 010540 | | | | PNTB X8,#X8A | |
| | 010540 | 012746 | 004236 | | | MOV #X8A,-(SP) |
| | 010544 | 004137 | 014660 | | | JSR R1,LPNTB |
| | 010550 | 004750 | | | | .WORD X8 |
| | 010552 | 000002 | | | | .WORD PNT.CT |
| 40 | 010554 | | | ENDMSG | | |
| 41 | | | | | | |
| 42 | 010556 | | | BGNMSG | ERR020 | |
| 43 | 010556 | | | | PNTB X20 | |
| | 010556 | 004137 | 014660 | | | JSR R1,LPNTB |
| | 010562 | 005306 | | | | .WORD X20 |
| | 010564 | 000000 | | | | .WORD PNT.CT |
| 44 | 010566 | | | ENDMSG | | |
| 45 | | | | | | |
| 46 | 010570 | | | BGNMSG | ERR021 | |
| 47 | 010570 | 010201 | | | MOV R2,R1 | |
| 48 | 010572 | 000301 | | | SWAB R1 | |
| 49 | 010574 | | | | AND 2,R1 | |
| | 010574 | 042701 | 177775 | | | BIC #^C<2>,R1 |
| 50 | 010600 | 006201 | | | ASR R1 | |
| 51 | 010602 | 005201 | | | INC R1 | |
| 52 | 010604 | | | | PNTB X21,R2,R1 | |
| | 010604 | 010146 | | | | MOV R1,-(SP) |
| | 010606 | 010246 | | | | MOV R2,-(SP) |
| | 010610 | 004137 | 014660 | | | JSR R1,LPNTB |
| | 010614 | 005476 | | | | .WORD X21 |
| | 010616 | 000004 | | | | .WORD PNT.CT |
| 53 | 010620 | | | ENDMSG | | |
| 54 | | | | | | |
| 55 | 010622 | | | BGNMSG | ERR022 | |
| 56 | 010622 | 042737 | 100000 016572 | | BIC #SA.ERR,UDARSD | |
| 57 | 010630 | | | | PNTB X22,UDARSD,R2 | |
| | 010630 | 010246 | | | | MOV R2,-(SP) |
| | 010632 | 013746 | 016572 | | | MOV UDARSD,-(SP) |
| | 010636 | 004137 | 014660 | | | JSR R1,LPNTB |
| | 010642 | 005613 | | | | .WORD X22 |
| | 010644 | 000004 | | | | .WORD PNT.CT |
| 58 | 010646 | | | | PRINTX #XFRU | |
| 59 | 010666 | | | ENDMSG | | |
| 60 | | | | | | |
| 61 | 010670 | | | BGNMSG | ERR023 | |
| 62 | 010670 | | | | PNTB X23A,R1,FFREE | |
| | 010670 | 013746 | 002146 | | | MOV FFREE,-(SP) |
| | 010674 | 010146 | | | | MOV R1,-(SP) |
| | 010676 | 004137 | 014660 | | | JSR R1,LPNTB |
| | 010702 | 005751 | | | | .WORD X23A |
| | 010704 | 000004 | | | | .WORD PNT.CT |
| 63 | 010706 | 005742 | | | TST -(R2) | |
| 64 | 010710 | 005712 | | ERR23A: | TST (R2) | |
| 65 | 010712 | 001410 | | | BEG ERR23B | |
| 66 | 010714 | | | | PNTB X23B,R2,(R2) | |
| | 010714 | 011246 | | | | MOV (R2),-(SP) |
| | 010716 | 010246 | | | | MOV R2,-(SP) |
| | 010720 | 004137 | 014660 | | | JSR R1,LPNTB |
| | 010724 | 006257 | | | | .WORD X23B |

| | | | | | |
|-----|--------|--------|--------|-------------------|--------------|
| | 010726 | 000004 | | | .WORD PNT.CT |
| 67 | 010730 | 005304 | | | |
| 68 | 010732 | 001403 | | DEC R4 | |
| 69 | 010734 | 005722 | | BEQ ERR23C | |
| 70 | 010736 | 005303 | | ERR23B: TST (R2)+ | |
| 71 | 010740 | 001363 | | DEC R3 | |
| 72 | 010742 | | | BNE ERR23A | |
| | 010742 | 004137 | 014660 | ERR23C: PNTB XFRU | |
| | 010746 | 010204 | | | JSR R1,LPNTB |
| | 010750 | 000000 | | | .WORD XFRU |
| 73 | 010752 | | | ENDMSG | .WORD PNT.CT |
| 74 | | | | | |
| 75 | 010754 | | | BGNMSG ERR024 | |
| 76 | 010754 | | | PNTB X24,R2 | |
| | 010754 | 010246 | | | MOV R2,-(SP) |
| | 010756 | 004137 | 014660 | | JSR R1,LPNTB |
| | 010762 | 006273 | | | .WORD X24 |
| | 010764 | 000002 | | | .WORD PNT.CT |
| 77 | 010766 | | | ENDMSG | |
| 78 | | | | | |
| 79 | 010770 | | | BGNMSG ERR025 | |
| 80 | 010770 | | | PNTB X25,R1,R2 | |
| | 010770 | 010246 | | | MOV R2,-(SP) |
| | 010772 | 010146 | | | MOV R1,-(SP) |
| | 010774 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011000 | 006463 | | | .WORD X25 |
| | 011002 | 000004 | | | .WORD PNT.CT |
| 81 | 011004 | | | ENDMSG | |
| 82 | | | | | |
| 83 | 011006 | | | BGNMSG ERR030 | |
| 84 | 011006 | | | PNTB X30,R1 | |
| | 011006 | 010146 | | | MOV R1,-(SP) |
| | 011010 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011014 | 006634 | | | .WORD X30 |
| | 011016 | 000002 | | | .WORD PNT.CT |
| 85 | 011020 | | | ENDMSG | |
| 86 | | | | | |
| 87 | 011022 | | | BGNMSG ERR031 | |
| 88 | 011022 | | | PNTB X31 | |
| | 011022 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011026 | 006747 | | | .WORD X31 |
| | 011030 | 000000 | | | .WORD PNT.CT |
| 89 | 011032 | | | ENDMSG | |
| 90 | | | | | |
| 91 | 011034 | | | BGNMSG ERR032 | |
| 92 | 011034 | | | PNTB X32 | |
| | 011034 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011040 | 007000 | | | .WORD X32 |
| | 011042 | 000000 | | | .WORD PNT.CT |
| 93 | 011044 | 004737 | 011234 | CALL MSGPKT | |
| 94 | 011050 | | | ENDMSG | |
| 95 | | | | | |
| 96 | 011052 | | | BGNMSG ERR033 | |
| 97 | 011052 | 004737 | 011142 | CALL PNTPKT | |
| 98 | 011056 | | | ENDMSG | |
| 99 | | | | | |
| 100 | 011060 | | | BGNMSG ERR034 | |

| | | | | | |
|-----|--------|--------|--------|--|-----------------|
| 101 | 011060 | 004737 | 011142 | CALL PNTPKT | |
| 102 | 011064 | | | ENDMSG | |
| 103 | | | | | |
| 104 | 011066 | | | BGNMSG ERR036 | |
| 105 | 011066 | | | PNTB X36 | |
| | 011066 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011072 | 007111 | | | .WORD X36 |
| | 011074 | 000000 | | | .WORD PNT.CT |
| 106 | 011076 | | | ENDMSG | |
| 107 | | | | | |
| 108 | 011100 | | | BGNMSG ERR037 | |
| 109 | 011100 | | | PNTB X37,R1 | |
| | 011100 | 010146 | | | MOV R1,-(SP) |
| | 011102 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011106 | 007227 | | | .WORD X37 |
| | 011110 | 000002 | | | .WORD PNT.CT |
| 110 | 011112 | | | ENDMSG | |
| 111 | | | | | |
| 112 | 011114 | | | BGNMSG ERR100 | |
| 113 | 011114 | | | PNTB X100,(R4) | |
| | 011114 | 011446 | | | MOV (R4),-(SP) |
| | 011116 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011122 | 007344 | | | .WORD X100 |
| | 011124 | 000002 | | | .WORD PNT.CT |
| 114 | 011126 | | | ENDMSG | |
| 115 | | | | | |
| 116 | 011130 | | | BGNMSG ERR101 | |
| 117 | 011130 | | | PNTB X101 | |
| | 011130 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011134 | 007427 | | | .WORD X101 |
| | 011136 | 000000 | | | .WORD PNT.CT |
| 118 | 011140 | | | ENDMSG | |
| 119 | | | | | |
| 120 | 011142 | | | PNTPKT: PNTB XPKT1 | |
| | 011142 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011146 | 007633 | | | .WORD XPKT1 |
| | 011150 | 000000 | | | .WORD PNT.CT |
| 121 | 011152 | 010401 | | MOV R4,R1 | |
| 122 | 011154 | 062701 | 000104 | ADD #HC.CPK,R1 | |
| 123 | 011160 | 010402 | | MOV R4,R2 | |
| 124 | 011162 | 062702 | 000020 | ADD #HC.MPK,R2 | |
| 125 | 011166 | 012703 | 000014 | MOV #12,R3 | |
| 126 | 011172 | | | PNTPKL: PNTB XPKT2,2(R1),(R1),2(R2),(R2) | |
| | 011172 | 011246 | | | MOV (R2),-(SP) |
| | 011174 | 016246 | 000002 | | MOV 2(R2),-(SP) |
| | 011200 | 011146 | | | MOV (R1),-(SP) |
| | 011202 | 016146 | 000002 | | MOV 2(R1),-(SP) |
| | 011206 | 004137 | 014660 | | JSR R1,LPNTB |
| | 011212 | 010124 | | | .WORD XPKT2 |
| | 011214 | 000010 | | | .WORD PNT.CT |
| 127 | 011216 | 062701 | 000004 | ADD #4,R1 | |
| 128 | 011222 | 062702 | 000004 | ADD #4,R2 | |
| 129 | 011226 | 005303 | | DEC R3 | |
| 130 | 011230 | 001360 | | BNE PNTPKL | |
| 131 | 011232 | 000207 | | RETURN | |
| 132 | | | | | |
| 133 | 011234 | | | MSGPKT: PNTB XMSG1 | |

011234 004137 014660
011240 007532
011242 000000
134 011244 016504 000016
135 011250 062704 000306
136 011254 012703 000005
137 011260
011260 016446 000014
011264 016446 000012
011270 016446 000010
011274 016446 000006
011300 016446 000004
011304 016446 000002
011310 011446
011312 004137 014660
011316 007566
011320 000016
138 011322 062704 000016
139 011326 005303
140 011330 001353
141 011332 000207

MOV C.RING(R5),R4
ADD #HC.BF2,R4
MOV #5,R3
MSGPKL: PNTB XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)

JSR R1,LPNTB
.WORD XMSG1
.WORD PNT.CT

MOV 12.(R4),-(SP)
MOV 10.(R4),-(SP)
MOV 8.(R4),-(SP)
MOV 6(R4),-(SP)
MOV 4(R4),-(SP)
MOV 2(R4),-(SP)
MOV (R4),-(SP)
JSR R1,LPNTB
.WORD XMSG2
.WORD PNT.CT

ADD #14.,R4
DEC R3
BNE MSGPKL
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

011334
011334 104454
011336 000004
011340 000000
011342 010502
011344
011344 104444

FMERR: ERRSF 4,,ERR004

DOCLN

:ABORT

TRAP C\$ERSF
.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     011346      ALOCM:  PUSH FFREE                ;SAVE FFREE AT ENTRY
17     011346 013746 002146      SUB R1,FSIZE                ;REDUCE SIZE OF FREE MEMORY
18     011352 160137 002150      BLT FMERR                 ;REPORT ERROR IF NOT ENOUGH MEMORY
19     011356 002766      ADD R1,R1                ;CHANGE WORDS TO BYTES
20     011360 060101      ADD R1,FFREE             ;CALCULATE NEW START OF FREE MEMORY
21     011362 060137 002146      POP R1                    ;GET START OF ALLOCATED MEMORY
22     011366 012601      RETURN                       MOV (SP)+,R1
23     011370 000207
    
```



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

```

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 011410 010137 002172  RUNDM:  MOV R1,URUN          ;SAVE NUMBER OF UNITS TO RUN
16 011414 005037 002174          CLR URNING        ;CLEAR NUMBER OF UNITS RUNNING
17
18     ;LOAD DM PROGRAM INTO EACH CONTROLLER
19
20 011420 013737 002172 002176  MOV URUN,UCNT      ;SET COUNTER OF UNITS
21 011426 013705 002162          MOV TSTTAB,R5     ;GET FIRST CONTROLLER TABLE
22 011432          LDDM:
23 011432 005065 000014          CLR C.FLG(R5)    ;CLEAR ALL FLAGS
24 011436 116537 000002 002074  MOVB C.UNIT(R5),L$SLUN ;SEE IF UNIT TO BE TESTED
25 011444 005765 000002          TST C.UNIT(R5)
26 011450 100407          BMI LDNEXT          ;IF NOT, DON'T LOAD THIS UNIT
27 011452          ASSUME CT.AVL EQ BIT15
28 011452 004737 011372          CALL HCOMM       ;ALLOCATE SPACE FOR HOST COMM AREA
29 011456 004737 015012          CALL LOADDM     ;LOAD THE DM PROGRAM
30 011462 001402          BEQ LDNEXT          ;IF ERROR, GO TO NEXT CONTROLLER
31 011464 005237 002174          INC URNING     ;IF NO ERROR, COUNT UNIT RUNNING
32 011470 062705 000054  LDNEXT: ADD #C.SIZE,R5 ;MOVE TO NEXT CONTROLLER TABLE
33 011474 005337 002176          DEC UCNT       ;CHECK IF MORE CONTROLLERS
34 011500 001354          BNE LDDM        ;LOAD NEXT
35 011502 005037 002200          CLR UFREEZ     ;CLEAR UNIT FREEZE FLAG
36 011506 012737 177777 003206  MOV #-1,FCTNUM ;INVALIDATE FCT BLOCK NUMBER (BLOCK IN MEMORY)
37
38     ;CHECK IF ANY CONTROLLERS LOADED
39
40 011514 005737 002174          TST URNING     ;ANY UNITS LOADED?
41
42     ;THE DM PROGRAMS ARE NOW IN CONTROL
43     ;RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
44
45 011520 000207          RETURN
    
```

```

1      ;RESPDM
2      ;
3      ;RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
4      ;HAVE TERMINATED.
5
6 011522 013705 002162      RESPDM: MOV TSTTAB,R5      ;GET CONTROLLER TABLE ADDRESS
7 011526 013737 002172 002176      MOV URUN,UCNT      ;SET COUNTER OF UNITS
8 011534      RESPCT: BREAK      ;ALLOW DRS TO SEE TERMINAL INPUT
9 011534 104422      TRAP      CSBRK
9 011536 016504 000016      MOV C.RING(R5),R4      ;GET HOST COMM AREA ADDRESS
10 011542 032765 000002 000014      BIT #CT.RN,C.FLG(R5)      ;CHECK IF PROGRAM RUNNING
11 011550 001502      BEQ RSPNXT      ;IF NOT, LOOK AT NEXT
12 011552 116537 000002 002074      MOVB C.UNIT(R5),L$LUN      ;STORE UNIT NUMBER UNDER TEST
13 011560 032765 000010 000014      BIT #CT.MSG,C.FLG(R5)      ;SEE IF INTERRUPT RECEIVED
14 011566 001150      BNE RSPIN      ;IF SO, LOOK AT PACKET
15 011570 032765 000004 000014      BIT #CT.CMD,C.FLG(R5)      ;SEE IF COMMAND HAS BEEN SENT
16 011576 001002      BNE 1$      ;IF NOT, SEND ONE
17 011600 000137 012346      JMP RSPOUT
18
19      ;CHECK IF UDA STILL RUNNING
20
21 011604 011503      1$: MOV (R5),R3      ;GET ADDRESS OF UDAIP
22 011606 016301 000002      MOV 2(R3),R1      ;LOOK AT UDASA REGISTER
23 011612 001405      BEQ RSPTM      ;IF ZERO, UDA STILL RUNNING
24 011614      ERRDF 30,,ERR030      ;REPORT UDA HAS FATAL ERROR
24 011614 104455      TRAP      CSERDF
24 011616 000036      .WORD      30
24 011620 000000      .WORD      0
24 011622 011006      .WORD      ERR030
25 011624 000465      BR RSPDRP      ;DROP CONTROLLER FROM TESTING
26
27      ;CHECK FOR TIMEOUT OF RESPONSE
28
29 011626 005765 000044      RSPTM: TST C.TOT(R5)      ;SEE IF DUP PROGRAM TO BE TIMED
30 011632 001451      BEQ RSPNTO
31 011634 005737 003212      TST KW.CSR      ;SEE IF A CLOCK ON SYSTEM
32 011640 001446      BEQ RSPNTO      ;DON'T TIME IF NO CLOCK
33 011642 023765 003224 000042      CMP KW.EL+2,C.TOH(R5)      ;COMPARE TO TIMEOUT COUNTER
34 011650 101005      BHI RSPTMO
35 011652 001041      BNE RSPNTO
36 011654 023765 003222 000040      CMP KW.EL,C.TO(R5)
37 011662 103435      BLO RSPNTO      ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
38 011664 032765 000040 000014      RSPTMO: BIT #CT.STA,C.FLG(R5)      ;SEE IF A GET DUST STATUS COMMAND OUTSTANDING
39 011672 001101      BNE RSPTOE      ;REPORT ERROR IF SO
40 011674 005764 000012      TST HC.CCT(R4)      ;SEE IF UDA TOOK LAST COMMAND PACKET
41 011700 100476      BMI RSPTOE      ;REPORT ERROR IF NOT
42 011702 012700 000100      MOV #CT.TM1,R0      ;SEE IF FIRST TIMEOUT ALREADY HAPPENED
43 011706 032765 000100 000014      BIT #CT.TM1,C.FLG(R5)
44 011714 001401      BEQ 1$
45 011716 006300      ASL R0      ;IF SO,
46 011720 052700 000040      1$: BIS #CT.STA,R0      ;SET SECOND TIME OUT FLAG
47 011724 050065 000014      BIS R0,C.FLG(R5)      ;SET THE PROPER TIMEOUT BIT
48 011730 012700 000001      MOV #OP.GDS,R0      ; AND STATUS REQUESTED BIT
49 011734 004737 015214      CALL BLDCMD      ;BUILD GET DUST STATUS COMMAND
50 011740 012764 100000 000012      MOV #RG.OWN,HC.CCT(R4)      ;MARK COMMAND TO UDA
51 011746 005775 000000      TST @ (R5)      ;TELL UDA COMMAND IS THERE
52 011752 000137 012426      JMP RSPOU4
    
```

53 011756

RSPNTO:

```

1          ;SWITCH TO NEXT CONTROLLER
2
3 011756 005737 002200 RSPNXT: TST UFREEZ      ;FROZEN TO ONE UNIT?
4 011762 001264          BNE RESPCT      ;STAY THERE IF SO
5 011764 062705 000054 ADD #C.SIZE,R5      ;MOVE TO NEXT TABLE
6 011770 005337 002176 DEC UCNT            ;CHECK IF MORE CONTROLLERS
7 011774 001257          BNE RESPCT      ;LOOK AT NEXT CONTROLLER
8 011776 000651          BR RESPDM       ;LOOK AT FIRST CONTROLLER AGAIN
9
10         ;REMOVE A CONTROLLER FROM TESTING
11
12 012000 005065 000014 RSPDRP: CLR C.FLG(R5)  ;CLEAR PROGRAM RUNNING
13 012004 005037 002200 CLR UFREEZ
14 012010 010504          MOV R5,R4
15 012012 062704 000020 ADD #C.DRO,R4
16 012016 012702 000010 MOV #8.,R2
17 012022 012403 1$:    MOV (R4)+,R3
18 012024 001420          BEQ 3$
19 012026 005763 000002 TST D.UNIT(R3)
20 012032          ASSUME DT.AVL EQ BIT15
21 012032 100003          BPL 2$
22 012034 005302          DEC R2
23 012036 001371          BNE 1$
24 012040 000412          BR 3$
25 012042 052763 100000 000002 2$:  BIS #DT.AVL,D.UNIT(R3)
26 012050 005302          DEC R2
27 012052 001405          BEQ 3$
28 012054 005714          TST (R4)
29 012056 001403          BEQ 3$
30 012060 004737 015012 CALL LOADDM        ;START DM PROGRAM AGAIN
31 012064 001223          BNE RESPCT
32 012066 005337 002174 3$:    DEC URNING      ;REDUCE RUNNING CONTROLLERS COUNT
33 012072 001331          BNE RSPNXT      ;IF ANY STILL RUNNING, LOOK AT THEM
34 012074 000207          RETURN      ;ELSE RETURN TO TEST SECTION
35
36 012076          RSPTOE: ERRDF 31,,ERR031 ;REPORT TIMEOUT ERROR
    012076 104455          TRAP          C$ERDF
    012100 000037          .WORD        31
    012102 000000          .WORD        0
    012104 011022          .WORD        ERR031
37 012106 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 012110 012700 000204          RSPIN: MOV #OP.END+OP.SSD,R0          ;GET SEND DATA END PACKET OPCODE
6 012114 032765 000020 000014  BIT #CT.REQ,C.FLG(R5)          ;LOOK IF SEND DATA OR RECEIVE DATA
7 012122 001402          BEQ RSPMWR
8 012124 012700 000205          MOV #OP.END+OP.RSD,R0          ;CHANGE TO RECEIVE DATA END PACKET OPCODE
9 012130 120064 000030          RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)      ;COMPARE TO OPCODE IN END PACKET
10 012134 001145          BNE RSPERR
11
12          ;LOOK AT STATUS CODE
13
14 012136 032764 000037 000032  BIT #ST.MSK,HC.MPK+P.STS(R4)      ;CHECK FOR STATUS CODE ST.SUC (ZERO)
15 012144 001004          BNE RSPERW
16
17          ;CHECK FOR EXPECTED REFERENCE NUMBER
18
19 012146 026564 000052 000020  CMP C.REF(R5),HC.MPK+P.CRF(R4)    ;CHECK IF CORRECT REF NUMBER
20 012154 001405          BEQ RSPPTW
21 012156          RSPERW: ERRDF 33,,ERR033
22 012156 104455          TRAP C$ERDF
23 012160 000041          .WORD 33
24 012162 000000          .WORD 0
25 012164 011052          .WORD ERR033
26 012166 000704          BR RSPDRP          ;DROP UNIT FROM TESTING
27
28          ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
29
30 012170 032765 000020 000014  RSPPTW: BIT #CT.REQ,C.FLG(R5)    ;CHECK IF RESPONSE FROM DM PROGRAM
31 012176 001463          RSPDU: BEQ RSPDU          ;LOOK AT REQUEST NUMBER IF SO
    
```

```

1          ;MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
2
3 012200 016401 000306 RSPPT2: MOV HC.BF2(R4),R1          ;GET REQUEST NUMBER
4 012204 042701 007777 BIC #*C<DU.TYP>,R1          ;CHECK TYPE
5 012210 001403 BEQ 1$          ;IF ZERO, ERROR
6 012212 020127 060000 CMP R1,#DU.SPC          ;CHECK IF IN EXPECTED RANGE
7 012216 101405 BLOS RSPPT3
8 012220 1$: ERRDF 32,,ERR032          ;BAD REQUEST NUMBER
          TRAP C$ERDF
          .WORD 32
          .WORD 0
          .WORD ERR032
9 012230 000663 BR RSPDRP          ;DROP UNIT FROM TESTING
10
11 012232 016403 000034 RSPPT3: MOV HC.MPK+P.BCNT(R4),R3      ;GET BYTE COUNT OF CHARACTERS RECEIVED IN R3
12 012236 162703 000002 SUB #2,R3          ;(FIRST TWO CHARACTERS ARE TYPE WORD)
13 012242 012700 000004 MOV #OP.SSD,R0      ;BUILD A SEND DATA COMMAND PACKET
14 012246 004737 015214 CALL BLDCMD        ; FOR ANSWER TO DM PROGRAM
15 012252 012700 000164 MOV #HC.BF1,R0      ;POINT TO BUFFER IN PACKET
16 012256 004737 015356 CALL CLRBUF        ; AND CLEAR BUFFER
17 012262 010402 MOV R4,R2          ;R2 POINTS TO SEND BUFFER
18 012264 062704 000122 ADD #HC.BSZ,R4      ;R4 POINTS TO CHARACTERS IN RECEIVE BUFFER
19 012270 042724 170000 BIC #DU.TYP,(R4)+ ;CLEAR TYPE FIELD IN BUFFER
20 012274 000301 SWAB R1          ;GET TYPE RIGHT JUSTIFIED
21 012276 006201 ASR R1          ;TIMES TWO
22 012300 006201 ASR R1
23 012302 006201 ASR R1
24 012304 010100 MOV R1,R0          ;COPY MESSAGE TYPE TO R0
25 012306 005001 CLR R1          ;R1 CONTAINS ZERO SEND BYTE COUNT
26 012310 004770 012574 CALL @RSPDSP-2(R0) ;CALL REQUESTED ROUTINE
27 012314 001231 BNE RSPDRP        ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
28          ; Z SET IF UNIT TO CONTINUE RUNNING
29 012316 016504 000016 MOV C.RING(R5),R4   ;GET RING ADDRESS
30 012322 032701 000001 BIT #1,R1          ;LOOK AT CHARACTER COUNT TO SEND TO DUP PROGRAM
31 012326 001401 BEQ 1$          ;IF AN ODD COUNT
32 012330 005201 INC R1          ; INCREASE BY ONE
33 012332 010164 000120 1$: MOV R1,HC.CPK+P.BCNT(R4) ;PUT CHARACTER COUNT IN COMMAND PACKET
34 012336 100003 BPL R$POUT        ;IF NEGATIVE BYTE COUNT RETURNED
35 012340 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 012346 042765 000350 000014 R$POUT: BIC #CT.MSG+CT.STA+CT.TM1+CT.TM2,C.FLG(R5) ;CLEAR MESSAGE RECEIVED FLAG
40 012354 032765 000020 000014 BIT #CT.REQ,C.FLG(R5) ;CHECK WHICH COMMAND TO SEND
41 012362 001014 BNE R$POU2        ;BRANCH IF RESPONSE TO REQUEST
42
43 012364 012700 000005 MOV #OP.RSD,R0      ;BUILD RECEIVE DATA COMMAND
44 012370 004737 015214 CALL BLDCMD
45 012374 012700 000306 MOV #HC.BF2,R0      ;POINT TO MESSAGE BUFFER
46 012400 004737 015356 CALL CLRBUF        ; AND CLEAR IT
47 012404 052765 000020 000014 BIS #CT.REQ,C.FLG(R5) ;SET REQUEST BIT
48 012412 000403 BR R$POU3
49
50 012414 042765 000020 000014 R$POU2: BIC #CT.REQ,C.FLG(R5) ;CLEAR REQUEST BIT
51 012422 R$POU3:
52 012422 004737 015300 CALL SND CMD       ;SEND COMMAND TO UDA
53 012426 016500 000044 R$POU4: MOV C.TOT(R5),R0 ;SET TIMEOUT

```

| | | | | | | | |
|----|--------|--------|--------|--------|---------|---------------------------------------|----------------------------------|
| 54 | 012432 | 010501 | | | | MOV R5,R1 | |
| 55 | 012434 | 062701 | 000040 | | | ADD #C.TO,R1 | ;PUT TIME IN CONTROLLER TABLE |
| 56 | 012440 | 004737 | 015612 | | | CALL SETTO | |
| 57 | 012444 | 000137 | 011756 | | | JMP RSPNXT | ;NOW WAIT FOR END PACKET |
| 58 | 012450 | 122764 | 000201 | 000030 | RSPERR: | CMPB #OP.END+OP.GDS,HC.MPK+P.OPCD(R4) | ;SEE IF GET DUST STATUS OPCODE |
| 59 | 012456 | 001237 | | | | BNE RSPERW | |
| 60 | 012460 | 132764 | 000010 | 000037 | | BITB #DF.ACT,HC.MPK+P.DFLG(R4) | ;IF DUST NO LONGER RUNNING |
| 61 | 012466 | 001603 | | | | BEQ RSPTOE | ; REPORT ERROR |
| 62 | 012470 | 042765 | 000050 | 000014 | | BIC #CT.STA+CT.MSG,C.FLG(R5) | ;CLEAR CONTROL BITS |
| 63 | 012476 | 032765 | 000200 | 000014 | | BIT #CT.TM2,C.FLG(R5) | ;IF AT SECOND TIMEOUT |
| 64 | 012504 | 001413 | | | | BEQ 1\$ | |
| 65 | 012506 | 026465 | 000040 | 000046 | | CMP HC.MPK+P.DPI(R4),C.PRI(R5) | ;COMPARE PROGRESS INDICATOR |
| 66 | 012514 | 001004 | | | | BNE 2\$ | |
| 67 | 012516 | 026465 | 000042 | 000050 | | CMP HC.MPK+P.DPI+2(R4),C.PRI+2(R5) | ;COMPARE PROGRESS INDICATOR |
| 68 | 012524 | 001422 | | | | BEQ 4\$ | ;REPORT ERROR IF NOT CHANGED |
| 69 | 012526 | 042765 | 000200 | 000014 | 2\$: | BIC #CT.TM2,C.FLG(R5) | ;CLEAR TIMEOUT 2 FLAG |
| 70 | 012534 | 032765 | 000100 | 000014 | 1\$: | BIT #CT.TM1,C.FLG(R5) | ;IF AT FIRST TIMEOUT |
| 71 | 012542 | 001406 | | | | BEQ 3\$ | |
| 72 | 012544 | 016465 | 000040 | 000046 | | MOV HC.MPK+P.DPI(R4),C.PRI(R5) | ;GET COPY OF PROGRESS INDICATOR |
| 73 | 012552 | 016465 | 000042 | 000050 | | MOV HC.MPK+P.DPI+2(R4),C.PRI+2(R5) | ;GET COPY OF PROGRESS INDICATOR |
| 74 | 012560 | 012764 | 140000 | 000006 | 3\$: | MOV #RG.OWN+RG.FLG,HC.MCT(R4) | ;GIVE MESSAGE BUFFER BACK TO UDA |
| 75 | 012566 | 000137 | 011756 | | | JMP RSPNXT | |
| 76 | 012572 | 000137 | 012076 | | 4\$: | JMP RSPTOE | |

1
2
3 012576 012612
4 012600 012664
5 012602 013036
6 012604 013126
7 012606 013136
8 012610 013146
9 000006

:RESPONSE REQUEST DISPATCH TABLE

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

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

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

:NORMAL DUP RECEIVE DATA BUFFER DESCRIPTION

:BYTE OFFSET FROM
:START OF BUFFER

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

USED TO SELECT ROUTINE
R4 CONTAINS THIS ADDRESS

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

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

;BYTE OFFSET FROM
;START OF BUFFER

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

R2 CONTAINS THIS ADDRESS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 012612 004737 013300
17 012616 062700 000004
18 012622 014403
19 012624 001411
20 012626 020327 000007
21 012632 001410
22 012634
012634 104455
012636 000144
012640 000000
012642 011114
23 012644 000244
24 012646 000207
25
26 012650 012700 003312
27 012654
28 012654 005201
29 012656 112022
30 012660 001375
31 012662 000207

```

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

        CLZ          ;CLEAR Z TO STOP DUP PROGRAM
        RETURN

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

```

TRAP  C$ERDF
.WORD 100
.WORD 0
.WORD ERR100
    
```

```

1      ;MESSAGE TYPE 2
2      ;
3      ;ANSWER QUESTION FOR DUP PROGRAM WITH DEFAULT ANSWER
4      ;
5      ;INPUT:
6      ;
7      ;   R5 - ADDRESS OF CONTROLLER TABLE
8      ;   R4 - POINTER TO DATA IN RECEIVE BUFFER
9      ;   R3 - CHARACTER COUNT IN RECEIVE BUFFER
10     ;   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
11     ;   R1 - ZERO
12     ;
13     ;OUTPUT:
14     ;   R1 - COUNT OF CHARACTERS IN SEND BUFFER
15     ;   Z SET TO CONTINUE RUNNING DUP PROGRAM
16     ;   Z CLEAR TO STOP THE DUP PROGRAM
17
16 012664 004737 013300 DQUEST: CALL GTDRVT      ;GET DRIVE TABLE ADDRESS INTO R0
17 012670 014403      MOV -(R4),R3      ;GET QUESTION NUMBER
18 012672 020327 000006      CMP R3,#DQUESZ
19 012676 101035      BHI DQUEX
20 012700 006303      ASL R3
21 012702 000173 012706      JMP @DQUEJP(R3)
22 012706 012772 DQUEJP: .WORD DQUEX      ; 0 (NOT USED)
23 012710 012724      .WORD DQUNIT      ; 1 ENTER UNIT NUMBER TO FORMAT
24 012712 012772      .WORD DQUEX      ; 2 (NOT USED)
25 012714 012772      .WORD DQUEX      ; 3 (NOT USED)
26 012716 012776      .WORD DQRFMT      ; 4 USE EXISTING BAD SECTOR INFORMATION
27 012720 013016      .WORD DQRSTR      ; 5 DOWN-LINE LOAD BAD SECTOR BLOCK INFORMATION
28 012722 013026      .WORD DQCONT      ; 6 CONTINUE IF BAD BLOCK INFO INACCESSIBLE
29      000006      DQUESZ=<<.-DQUEJP>/2>-1
30
31     ;ENTER UNIT NUMBER TO FORMAT
32
33 012724 DQUNIT: PUSH R5
34 012724 010546      CLR R4
35 012726 005004      MOV (R0),R3      ;GET DRIVE NUMBER
36 012730 011003      ASSUME D.DRV EQ 0
37 012732 012700 000012      MOV #10.,R0      ;RADIX 10.
38 012736 004737 014754 DQUNL1: CALL DIVIDE
39 012742 010546      PUSH R5
40 012744 005201      INC R1
41 012746 005703      TST R3
42 012750 001372      BNE DQUNL1
43 012752 010100      MOV R1,R0
44 012754 DQUNL2: POP R5
45 012754 012605      MOV (SP)+,R5
46 012756 062705 000060      ADD #0,R5
47 012762 110522      MOV R5,(R2)+
48 012764 005300      DEC R0
49 012766 001372      BNE DQUNL2
50 012770 012605      POP R5
51 012770 012605      MOV (SP)+,R5
52 012772 000264 DQUEX: SEZ
53 012774 000207      RETURN
54
55 012776 032737 000003 003210 DQRFMT: BIT #SO.FMT,MODE
    
```

```
54 013004 001410
55 013006 112712 000131      DQYES: BEQ DQNO
56 013012 005201              MOVB #'Y,(R2)
57 013014 000766              INC R1
58                               BR DQUEX
59 013016 032737 000010 003210 DQRSTR: BIT #SO.STR,MODE
60 013024 001370              BNE DQYES
61 013026                      DQCONT:
62 013026 112712 000116      DQNO:  MOVB #'N,(R2)
63 013032 005201              INC R1
64 013034 000756              BR DQUEX
```

```

1      :MESSAGE TYPE 3
2
3      :PRINT INFORMATION FROM DUP PROGRAM
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         R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM
14         Z SET TO CONTINUE RUNNING DUP PROGRAM
15 013036 016400 177776      INFO:  MOV -2(R4),R0      ;GET MESSAGE NUMBER
16 013042 001417            BEQ INFOX          ;IF ZERO, IGNORE IT
17 013044 020027 000100     CMP RO,#100        ;IF OCTAL 100
18 013050 001420            BEQ INFOE          ; PRINT ERROR MESSAGE
19 013052 005737 002200     TST UFREEZ
20 013056 001007            BNE INFOP
21 013060 005237 002200     INC UFREEZ
22 013064 004737 013300     CALL GTDRVT
23 013070 010002            MOV RO,R2
24 013072 004737 013324     CALL HEADER
25 013076 004737 013244     INFOP: CALL MMSG      ;PRINT THE MESSAGE
26 013102 012701 100000     INFOX: MOV #BIT15,R1 ;RETURN A NEGATIVE BYTE COUNT
27 013106 000264            SEZ
28 013110 000207            RETURN          ;RETURN WITH Z SET
29
30 013112            INFOE: ERRDF 101,,ERR101 ;ANSWER WAS REJECTED BY DUP PROGRAM
31 013112 104455            TRAP          CSERDF
32 013114 000145            .WORD          101
33 013116 000000            .WORD          0
34 013120 011130            .WORD          ERR101
35 013122 000244            CLZ          ;RETURN WITH Z CLEAR TO STOP DUP PROGRAM
36 013124 000207            RETURN
    
```

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

```
:MESSAGE TYPE 4  
:  
:TERMINATION MESSAGE  
:  
:INPUT:  
:   R5 - POINTER TO CONTROLLER TABLE  
:   R4 - POINTER TO DATA IN RECEIVE BUFFER  
:   R3 - CHARACTER COUNT IN RECEIVE BUFFER  
:   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)  
:   R1 - ZERO  
:  
:OUTPUT:  
:   Z CLEAR TO TERMINATE DUP PROGRAM  
:  
TERM:  CALL INFO      ;PRINT THE MESSAGE  
       CLZ  
       RETURN          ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM
```

013126 004737 013036
013132 000244
013134 000207

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

013136 004737 013036
013142 000244
013144 000207

```
:MESSAGE TYPE 5  
:ERROR TERMINATION MESSAGE  
:INPUT:  
:   R5 - POINTER TO CONTROLLER TABLE  
:   R4 - POINTER TO DATA IN RECEIVE BUFFER  
:   R3 - CHARACTER COUNT IN RECEIVE BUFFER  
:   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)  
:   R1 - ZERO  
:OUTPUT:  
:   Z CLEAR TO TERMINATE DUP PROGRAM  
ERRTRM: CALL INFO  
        CLZ  
        RETURN  
:RETURN Z CLEAR TO TERMINATE DUP PROGRAM
```

```

1      :MESSAGE TYPE 6
2
3      :SPECIAL TYPE - READ FCT BLOCK FROM FILE
4
5      :INPUT:
6          R5 - POINTER TO CONTROLLER TABLE
7          R4 - POINTER TO DATA IN RECEIVE BUFFER
8          R3 - CHARACTER COUNT IN RECEIVE BUFFER
9          R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10         R1 - ZERO
11      :OUTPUT:
12         Z SET TO SEND DATA TO PROGRAM
13
14 013146 023714 003206   SPECL:  CMP FCTNUM,(R4) ;SEE IF DESIRED BLOCK IS IN MEMORY
15 013152 001425         BEQ SPECLX      ; IF SO, SEND TO DUP PROGRAM
16 013154 002407         BLT SPECLR      ; IF LOWERED NUMBERED BLOCK IN MEMORY,
17                                     ; GO READ NEXT BLOCK
18 013156         SPECLC: CLOSE      ;OTHERWISE, START READING FROM BEGINNING AGAIN
19 013156 104435         OPEN #FNAME
20 013160 012700 003234         TRAP      C$CLOS
21 013164 104434         MOV #FNAME,R0
22 013166 012737 177777 003206   TRAP      C$OPEN
23 013174 012703 001000   SPECLR:  MOV #-1,FCTNUM
24 013200 012701 002206   SPECLR:  MOV #512,R3 ;GET BYTE COUNT IN A BLOCK
25 013204         MOV #FCTBUF,R1 ;POINT TO STORAGE AREA
26 013206 104426         SPECLL:  GETBYTE (R1)+ ;READ THE FILE
27 013210 110021         TRAP      C$GETB
28 013210 103005         MOV#B RO,(R1)+
29 013212 005303         BCC      SPECLE
30 013214 001373         BNCOMPLETE SPECLE ;PRINT ERROR IF NO MORE BYTES IN FILE
31 013216 005237 003206   DEC R3 ;COUNT THE BYTES
32 013222 000751         BNE SPECLL
33 013224 005212         INC FCTNUM ;KEEP COUNT OF BLOCK IN MEMORY
34 013226 012762 002206 000002   BR SPECL
35 013234 012701 000006   SPECLE:  INC (R2) ;TELL DUP PROGRAM DATA NOT AVAILABLE
36 013240 000264         SPECLX:  MOV #FCTBUF,2(R2) ;PUT ADDRESS OF DATA IN OUTPUT BUFFER
37 013242 000207         MOV #6,R1 ;SEND 3 WORDS TO DUP PROGRAM
38                                     SEZ
39                                     RETURN ;RETURN WITH Z SET TO SEND DATA TO DUP PROGRAM

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13 013244
    013244 112700 000015
    013250 004737 014476
14 013254 112400
15 013256 001405
16 013260 020027 000012
17 013264 001402
18 013266
    013266 004737 014476
19 013272 005303
20 013274 003367
21 013276 000207

:PRINT A MESSAGE IN THE RECEIVE BUFFER FROM THE DUP PROGRAM
:
:INPUT:
:   R4 - POINTER TO DATA IN RECEIVE BUFFER
:   R3 - CHARACTER COUNT IN RECEIVE BUFFER
:OUTPUT:
:   R4 - POINTER TO CHARACTER AFTER MESSAGE IN RECEIVE BUFFER
:   R3 - ZERO
:   R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM
:   R0 - CONTENTS DESTROYED
:   Z SET TO CONTINUE RUNNING DUP PROGRAM
MSG:  PRINT #CR
                                           MOVB #CR,R0
                                           CALL CPNT
1$:   MOVB (R4)+,R0                       ;PRINT CHARACTERS FROM DUP PROGRAM
      BEQ 2$                               ; DISCARDING LF AND NULL CHARACTERS
      CMP R0,#12
      BEQ 2$
      PRINT R0
2$:   DEC R3                               ;COUNT THE CHARACTERS
      BGT 1$
      RETURN
    CALL CPNT

```

```
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     013300      GTDRVT: PUSH R5
12     013300      010546      MOV R5,-(SP)
13     013302      062705      000020      ADD #C.DRO,R5
14     013306      012500      GTDRVL: MOV (R5)+,R0
15     013310      016037      000002      002074      MOV D.UNIT(R0),L$LUN
16     013316      100773      ASSUME DT.AVL EQ BIT15
17     013320      012605      BMI GTDRVL
18     013322      000207      POP R5
19                                     MOV (SP)+,R5
20                                     RETURN
```

```

1      :HEADER
2      :
3      :PRINT A HEADER IN FRONT OF EACH MESSAGE FROM DUP PROGRAM.
4      :A UDA ADDRESS IS PRINTED IF MORE THAN ONE JDA IS IN HARDWARE P-TABLE.
5      :A RUNTIME IS PRINTED IF A CLOCK IS BEING USED TO TIME PROGRAM EXECUTION.
6      :
7      :INPUT:
8      :      R5 - POINTER TO CONTROLLER TABLE
9      :OUTPUT:
10     :      R0 - POINTER TO DRIVE TABLE
11     :      PRINTED MESSAGE
12     :
13     013324 022737 000001 002012 HEADER: CMP #1,LSUNIT          ;IF MORE THAN ONE UNIT BEING TESTED
14     013332 001411 BEQ 1$
15     013334 PNTF MESSG,D.UNIT(R2),(R5),(R2)          ;PRINT UDA ADDRESS
16     013334 011246 MOV (R2),-(SP)
17     013336 011546 MOV (R5),-(SP)
18     013340 016246 000002 MOV D.UNIT(R2),-(SP)
19     013344 004137 014650 JSR R1,LPNTF
20     013350 003743 .WORD MESSG
21     013352 000006 .WORD P'T.CT
22     013354 ASSUME C.UADR EQ 0
23     013354 ASSUME D.DRV EQ 0
24     013354 000407 BR 2$
25     013356 005737 003212 1$: TST KW.CSR          ;IF NO CLOCK BEING USED
26     013362 001406 BEQ 3$          ;BYPASS RUNTIME MESSAGE
27     013364 PRINT #CR
28     013364 112700 000015 MOV B #CR,R0
29     013370 004737 014476 CALL CPNT
30     013374 004737 016636 2$: CALL RNTIME          ;PRINT RUNTIME IF A CLOCK IN USE
31     013400 000207 3$: 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     NCONS:  BR OSTRE
48             SUB #ERRC,R0     :GET INCREMENT INTO TABLE
49             ASL R0           :DOUBLE TO WORD COUNT
50             CALL @ERRD(R0)   :DISPATCH TO PRINT ROUTINE
51             BR OSTRNG       :GET NEXT
52
53     OSTRE:  RETURN

```

```

34 013402 112201
35 013404 001421
36 013406 012700 013704
37 013412 120110
38 013414 001407
39 013416 105720
40 013420 001374
41 013422
   013422 004137 014650
   013426 003654
   013430 000000
42 013432 000406
43 013434 162700 013704
44 013440 006300
45 013442 004770 013716
46 013446 000755
47 013450 000207

```

```

1      ;CONTROL CHARACTER WAS A QUOTE. PRINT ALL CHARACTERS TO THE NEXT QUOTE.
2
3 013452 112200      CON.QU: MOVB (R2)+,R0      ;GET CHARACTER
4 013454 120027 000042  CMPB RO,#'"      ;CHECK IF ENDING QUOTE
5 013460 001403      BEQ CON.QX      ;IF SO, GO GET NEXT CONTROL CHARACTER
6 013462      PRINT R0      ;PRINT THE CHARACTER
7 013466 004737 014476  BR CON.QU      CALL CPNT
8 013470 000771      CON.QX: RETURN      ;CONTINUE PRINTING
9
10     ;CONTROL CHARACTER WAS AN A. PRINT ASCII CHARACTERS FROM PARAMETERS.
11
12 013472 004737 014156  CON.A: CALL GETCNT      ;GET COUNT OF CHARACTERS
13 013476      CON.A1: PRINT (R4)+      ;PRINT THE CHARACTER
14 013476 112400      MOVB (R4)+,R0
15 013500 004737 014476  CALL CPNT
16 013504 005301      DEC R1      ;COUNT THE CHARACTERS
17 013506 001373      BNE CON.A1      ;PRINT UNTIL COUNT REACHES ZERO
18 013510 032704 000001  BIT #1,R4      ;CHECK IF R4 NOW ODD
19 013514 001401      BEQ CON.A2
20 013516 005204      INC R4      ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
21 013520 000207      CON.A2: RETURN      ;NOW GET NEXT CONTROL CHARACTER
22
23     ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.
24
25 013522 012701 000012  CON.D: MOV #10.,R1      ;LOAD RADIX
26 013526 004737 014234  CALL PNTNUM      ;PRINT NUMBER
27 013532 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER
28
29     ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.
30
31 013534 012701 000020  CON.H: MOV #16.,R1      ;LOAD RADIX
32 013540 004737 014234  CALL PNTNUM      ;PRINT NUMBER
33 013544 000207      RETURN      ;NOW GET NEXT CONTROL CHARACTER
    
```

```

1          ;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.
2
3 013546 012701 000010  CON.O:  MOV #8.,R1          ;LOAD RADIX
4 013552 004737 014234  CALL PNTNUM          ;PRINT NUMBER
5 013556 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
6
7          ;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.
8
9 013560 004737 014156  CON.N:  CALL GETCNT          ;GET COUNT
10 013564          CON.N1: PRINT #CR          ;PRINT NEW LINE SEQUENCE
    013564 112700 000015          ;COUNT THE SEQUENCES          MOVB #CR,R0
    013570 004737 014476          CALL CPNT
11 013574 005301          DEC R1
12 013576 001372          BNE CON.N1
13 013600 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
14
15          ;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.
16
17 013602 004737 014156  CON.R:  CALL GETCNT          ;GET ROUTINE NUMBER
18 013606 020127 000011  CMP R1,#ERRRSZ          ;CHECK IF DEFINED ROUTINE NUMBER
19 013612 101004          BHI CON.R1
20 013614 060101          ADD R1,R1          ;DOUBLE COUNT TO GET WORD INDEX
21 013616 004771 013660  CALL @ERRRTB-2(R1)      ;CALL ROUTINE
22 013622 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
23 013624          CON.R1: PNTF ERRME1          ;REPORT BAD MESSAGE STRING
    013624 004137 014650          JSR R1,LPNTF
    013630 003654          .WORD ERRME1
    013632 000000          .WORD PNT.CT
24 013634          POP R1          ;FIX THE STACK
    013634 012601          RETURN          MOV (SP)+,R1
25 013636 000207
26
27          ;CONTROL CHARACTER WAS AN S. PRINT SPACES.
28
29 013640 004737 014156  CON.S:  CALL GETCNT          ;GET COUNT
30 013644          CON.S1: PRINT '<#>'          ;PRINT A SPACE
    013644 112700 000040          ;COUNT THE SPACES          MOVB #' ',R0
    013650 004737 014476          CALL CPNT
31 013654 005301          DEC R1
32 013656 001372          BNE CON.S1
33 013660 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
    
```



```
1          ;ERROR ROUTINE DISPATCH TABLE
2
3 013662 013736      ERRRTB: .WORD CALRE          ;NOT USED
4 013664 013736      .WORD CALRE          ;NOT USED
5 013666 013736      .WORD CALRE          ;NOT USED
6 013670 013750      .WORD CALR4         ;PRINT BASIC LINE WITHOUT UDA ADDRESS
7 013672 014024      .WORD CALR5         ;PRINT BASIC LINE WITH UDA ADDRESS
8 013674 014102      .WORD CALR6         ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
9 013676 014116      .WORD CALR7         ;PRINT 'REPLACE UDA MODULE M7161'
10 013700 014134     .WORD CALR8         ;PRINT ' UDASA CONTAINS 'XXXXXX''
11 013702 014152     .WORD CALR9         ;REPRINT LAST NUMBER
12          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
2
3
4
5
6
7
8
9 013704
013704 042
013705 101
013706 104
013707 110
013710 117
013711 116
013712 122
013713 123
10 013714 000
11
12
13
14
15
16
17
18
19
20
21 013716
013716 013452
013720 013472
013722 013522
013724 013534
013726 013546
013730 013560
013732 013602
013734 013640

;HERE IS FIRST TABLE

.MACRO ENTRY ARG1,ARG2
.LIST
.BYTE ''ARG1
.NLIST
.ENDM

ERRC: BUILD
.BYTE ''
.BYTE 'A
.BYTE 'D
.BYTE 'H
.BYTE 'O
.BYTE 'N
.BYTE 'R
.BYTE 'S
.BYTE 0
.EVEN

;FOLLOW WITH A NULL BYTE

;HERE IS SECOND TABLE

.MACRO ENTRY ARG1,ARG2
.LIST
.WORD ARG2
.NLIST
.ENDM

ERRD: BUILD
.WORD CON.QU
.WORD CON.A
.WORD CON.D
.WORD CON.H
.WORD CON.O
.WORD CON.N
.WORD CON.R
.WORD CON.S

1
2
3
4
5

013736
013736 004137 014650
013742 003654
013744 000000
013746 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
   013750 012746 004137      MOV #BAS,-(SP)
   013750 012746 004137      MOV #BAS,-(SP)
   013754 012746 004137      MOV #BAS,-(SP)
   013760 012746 004137      MOV #BASNO,-(SP)
   013764 012746 004064      JSR R1,LPNTB
   013770 004137 014660      .WORD BASLN
   013774 004140              .WORD PNT.CT
   013776 000010
6 014000 004737 016636      CALL RNTIME
7 014004              PRINT #CR
   014004 112700 000015      MOV #CR,R0
   014010 004737 014476      CALL CPNT
8 014014 012737 014576 003226  MOV #PX,PType
9 014022 000207              RETURN
```

```
1  
2  
3  
4  
5 014024  
014024 012746 004137  
014030 012746 004137  
014034 011546  
014036 012746 004103  
014042 012746 004064  
014046 004137 014660  
014052 004140  
014054 000012  
6 014056 004737 016636  
7 014062  
014062 112700 000015  
014066 004737 014476  
8 014072 012737 014576 003226  
9 014100 000207
```

```
:PRE-PROGRAMMED ROUTINE 5  
:PRINT BASIC LINE FOR HOST PROGRAM ERROR WITH UDA ADDRESS  
:THEN SWITCH TO EXTENDED FORMAT
```

```
CALRS: PNTB BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
```

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

```
CALL RNTIME  
PRINT #CR
```

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

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

```
1 ;PRE-PROGRAMMED ROUTINE 6
2 ;CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
3
4 014102 CALR6: PUSH R2 ;SAVE CURRENT STRING POINTER
5 014102 010246 ;MOV R2,-(SP)
6 014104 012402 ;GET NEW STRING POINTER
7 014106 004737 013402 ;CALL OSTRNG ;OUTPUT USING THIS STRING
8 014112 012602 ;POP R2 ;GET OLD POINTER BACK
9 014114 000207 ;RETURN ;NOW CONTINUE THE OLD STRING
```

```
1      ;PRE-PROGRAMMED ROUTINE 7
2      ;PRINT 'REPLACE UDA MODULE M7161''
3
4 014116      CALR7:  PUSH R2
5 014116 010246      MOV R2,-(SP)
6 014120 012702 010204      MOV #XFRU,R2
7 014124 004737 013402      CALL OSTRNG
8 014130      POP R2
9 014132      MOV (SP)+,R2
10      RETURN
```

```
1      ;PRE-PROGRAMMED ROUTINE 8
2      ;PRINT  "" UDASA CONTAINS  XXXXXX""
3
4 014134 CALR8:  PUSH R2
   014134 010246
5 014136 012702 010153      MOV #XSA,R2
6 014142 004737 013402      CALL OSTRNG
7 014146      POP R2
   014146 012602
8 014150 000207      RETURN
                                MOV R2,-(SP)
                                MOV (SP)+,R2
```


1
2
3 014152 005744
4 014154 000207

: 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
10     :OUTPUTS:
11     :      R1 - NUMBER READ OR A ONE
12     :      R2 - POINTING TO CHARACTER AFTER NUMBER
13
14     GETCNT: PUSH R0
15
16     GETCNX: CLR R1
17             CMPB (R2),#'0
18             BLO GETCDN
19             CMPB (R2),#'9
20             BHI GETCDN
21             ASL R1
22             MOV R1,R0
23             ASL R1
24             ASL R1
25             ADD R0,R1
26             MOVB (R2)+,R0
27             SUB #'0,R0
28             ADD R0,R1
29             BR GETCNX
30
31     GETCDN: TST R1
32             BNE GETCXX
33             INC R1
34
35     GETCXX: POP R0
36
37             RETURN
38
39             MOV RO,-(SP)
40             ;START WITH ZERO COUNT
41             ;CHECK IF CHARACTER A DIGIT
42             ;BRANCH IF LOWER THAN ZERO
43             ;BRANCH IF HIGHER THAN NINE
44             ;MULTIPLY NUMBER BY 10
45             ;SAVE 2N
46             ;COMPUTE 4N
47             ;COMPUTE 8N
48             ;8N + 2N = 10N
49             ;GET DIGIT FROM STING
50             ;GET RID OF ASCII
51             ;ADD TO NUMBER
52             ;GO TO NEXT CHARACTER
53             ;CHECK IF NUMBER IS ZERO
54             ;IF ZERO, CHANGE
55             ;TO DEFAULT OF ONE
56             MOV (SP)+,R0

```

```

13 014156 010046
14 014156 005001
15 014162 121227 000060
16 014166 103415
17 014170 121227 000071
18 014174 101012
19 014176 006301
20 014200 010100
21 014202 006301
22 014204 006301
23 014206 060001
24 014210 112200
25 014212 162700 000060
26 014216 060001
27 014220 000760
28 014222 005701
29 014224 001001
30 014226 005201
31 014230
32 014230 012600
32 014232 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 | 014234 | 010100 | | PNTNUM: MOV R1,R0 | :SAVE RADIX |
| 15 | 014236 | 004737 | 014156 | CALL GETCNT | :GET COUNT OF BITS |
| 16 | 014242 | | | PNTNUS: PUSH <R2,R3,R5> | |
| | 014242 | 010246 | | | MOV R2,-(SP) |
| | 014244 | 010346 | | | MOV R3,-(SP) |
| | 014246 | 010546 | | | MOV R5,-(SP) |
| 17 | 014250 | 012403 | | MOV (R4)+,R3 | :GET ONE PARAMETER WORD |
| 18 | 014252 | 005005 | | CLR R5 | :CLEAR STORAGE FOR OTHER |
| 19 | 014254 | 020127 | 000020 | CMP R1,#16. | :MORE THAN 16 BITS IN NUMBER? |
| 20 | 014260 | 003401 | | BLE 1\$ | |
| 21 | 014262 | 012405 | | MOV (R4)+,R5 | :YES, GET SECOND PARAMETER WORD |
| 22 | 014264 | | | 1\$: PUSH R4 | |
| | 014264 | 010446 | | | MOV R4,-(SP) |
| 23 | 014266 | 010504 | | MOV R5,R4 | :PUT HIGH WORD IN R4 |
| 24 | 014270 | 012702 | 000020 | MOV #16.,R2 | :COMPUTE BITS NOT WANTED |
| 25 | 014274 | 160102 | | SUB R1,R2 | :BY SUBTRACTING BITS TO USE |
| 26 | 014276 | 002002 | | BGE 2\$ | :FROM 16. |
| 27 | 014300 | 062702 | 000020 | ADD #16.,R2 | :IF NEGATIVE, ADD 16 FOR FIRST WORD |
| 28 | 014304 | 001414 | | 2\$: BEQ 6\$ | :IF ZERO, NO BITS NEED BE CLEARED |
| 29 | 014306 | 012705 | 100000 | MOV #BIT15,R5 | :START MASK WITH SIGN BIT SET |
| 30 | 014312 | 005302 | | 3\$: DEC R2 | :COUNT BITS IN MASK |
| 31 | 014314 | 001402 | | BEQ 4\$ | |
| 32 | 014316 | 006205 | | ASR R5 | :SHIFT MORE BITS TO RIGHT |
| 33 | 014320 | 000774 | | BR 3\$ | |
| 34 | 014322 | 020127 | 000020 | 4\$: CMP R1,#16. | :MORE THAN 16 BITS IN NUMBER? |
| 35 | 014326 | 003402 | | BLE 5\$ | |
| 36 | 014330 | 040504 | | BIC R5,R4 | :YES, CLEAR IN HIGH WORD |
| 37 | 014332 | 000401 | | BR 6\$ | |
| 38 | 014334 | 040503 | | 5\$: BIC R5,R3 | :NO, CLEAR IN LOW WORD |
| 39 | 014336 | 004737 | 014754 | 6\$: CALL DIVIDE | :DIVIDE BY RADIX IN R0 |
| 40 | 014342 | | | PUSH R5 | :PUSH REMAINDER ON STACK |
| | 014342 | 010546 | | | MOV R5,-(SP) |
| 41 | 014344 | 005202 | | INC R2 | :COUNT DIGITS ON STACK |
| 42 | 014346 | 005703 | | TST R3 | :CHECK IF QUOTIENT IS ZERO |
| 43 | 014350 | 001372 | | BNE 6\$ | |
| 44 | 014352 | 005704 | | TST R4 | |
| 45 | 014354 | 001370 | | BNE 6\$ | |

| | | | | | | | |
|----|--------|--------|--------|-------|--------------------|--|--|
| 1 | 014356 | 020027 | 000012 | | CMP R0,#10. | | :IF RADIX IS DECIMAL |
| 2 | 014362 | 001423 | | | BEQ 10\$ | | : JUST GO PRINT DIGITS ON STACK |
| 3 | 014364 | 010103 | | | MOV R1,R3 | | :OTHERWISE COMPUTE NUMBER OF LEADING ZEROS |
| 4 | 014366 | 162700 | 000014 | | SUB #12.,R0 | | :DIVIDEND IS BITS IN NUMBER |
| 5 | 014372 | 003002 | | | BGT 7\$ | | :DIVISOR IS BITS PER DIGIT PRINTED |
| 6 | 014374 | 012700 | 000003 | | MOV #3,R0 | | : (3 OR 4) |
| 7 | 014400 | 004737 | 014754 | 7\$: | CALL DIVIDE | | |
| 8 | 014404 | 005705 | | | TST R5 | | :IF REMAINDER NOT ZERO |
| 9 | 014406 | 001401 | | | BEQ 8\$ | | :INCREMENT QUOTIENT |
| 10 | 014410 | 005203 | | | INC R3 | | |
| 11 | 014412 | 160203 | | 8\$: | SUB R2,R3 | | :SUBTRACT DIGITS ON STACK |
| 12 | 014414 | 001406 | | | BEQ 10\$ | | :NO LEADING ZEROS IF ZERO |
| 13 | 014416 | | | 9\$: | PRINT #'0 | | :PRINT A ZERO |
| | 014416 | 112700 | 000060 | | | | MOVB #'0,R0 |
| | 014422 | 004737 | 014476 | | | | CALL CPNT |
| 14 | 014426 | 005303 | | | DEC R3 | | |
| 15 | 014430 | 001372 | | | BNE 9\$ | | :REPEAT UNTIL COUNT REACHES ZERO |
| 16 | | | | | | | |
| 17 | 014432 | | | 10\$: | POP R5 | | :GET CHARACTER FROM STACK |
| | 014432 | 012605 | | | | | MOV (SP)+,R5 |
| 18 | 014434 | 062705 | 000060 | | ADD #'0,R5 | | :CONVERT TO ASCII DIGIT |
| 19 | 014440 | 020527 | 000071 | | CMP R5,#'9 | | :IF GREATER THAN A 9 |
| 20 | 014444 | 003402 | | | BLE 11\$ | | : CONVERT TO A OR HIGHER |
| 21 | 014446 | 062705 | 000007 | | ADD #'<'A-'9-1>,R5 | | : FOR HEX DIGIT |
| 22 | 014452 | | | 11\$: | PRINT R5 | | :PRINT THE CHARACTER |
| | 014452 | 110500 | | | | | MOVB R5,R0 |
| | 014454 | 004737 | 014476 | | | | CALL CPNT |
| 23 | 014460 | 005302 | | | DEC R2 | | :REPEAT FOR ALL DIGITS |
| 24 | 014462 | 001363 | | | BNE 10\$ | | : ON STACK |
| 25 | 014464 | | | | POP <R4,R5,R3,R2> | | |
| | 014464 | 012604 | | | | | MOV (SP)+,R4 |
| | 014466 | 012605 | | | | | MOV (SP)+,R5 |
| | 014470 | 012603 | | | | | MOV (SP)+,R3 |
| | 014472 | 012602 | | | | | MOV (SP)+,R2 |
| 26 | 014474 | 000207 | | | RETURN | | |

| | | | | | |
|----|--------|--------|------------------------|-----------------------|-------------------|
| 1 | | | :PRINT ONE CHARACTER | | |
| 2 | | | :CALL WITH MACRO PRINT | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | 014476 | 110037 | 003230 | CPNT: MOV B R0,ERRCHR | |
| 6 | 014502 | | | PUSH R1 | |
| | 014502 | 010146 | | | MOV R1,-(SP) |
| 7 | 014504 | 012701 | 003612 | MOV #ERRONE,R1 | |
| 8 | 014510 | 120027 | 000015 | CMPB R0,#CR | |
| 9 | 014514 | 001002 | | BNE 1\$ | |
| 10 | 014516 | 012701 | 003615 | MOV #ERRNL,R1 | |
| 11 | 014522 | 000177 | 166500 | 1\$: JMP @PTYPE | |
| 12 | 014526 | | | PF: PRINTF R1,#ERRCHR | |
| | 014526 | 012746 | 003230 | | MOV #ERRCHR,-(SP) |
| | 014532 | 010146 | | | MOV R1,-(SP) |
| | 014534 | 012746 | 000002 | | MOV #2,-(SP) |
| | 014540 | 010600 | | | MOV SP,R0 |
| | 014542 | 104417 | | | TRAP C\$PNTF |
| | 014544 | 062706 | 000006 | | ADD #6,SP |
| 13 | 014550 | 000435 | | PB: BR CPNTX | |
| 14 | 014552 | | | PRINTB R1,#ERRCHR | |
| | 014552 | 012746 | 003230 | | MOV #ERRCHR,-(SP) |
| | 014556 | 010146 | | | MOV R1,-(SP) |
| | 014560 | 012746 | 000002 | | MOV #2,-(SP) |
| | 014564 | 010600 | | | MOV SP,R0 |
| | 014566 | 104414 | | | TRAP C\$PNTB |
| | 014570 | 062706 | 000006 | | ADD #6,SP |
| 15 | 014574 | 000423 | | PX: BR CPNTX | |
| 16 | 014576 | | | PRINTX R1,#ERRCHR | |
| | 014576 | 012746 | 003230 | | MOV #ERRCHR,-(SP) |
| | 014602 | 010146 | | | MOV R1,-(SP) |
| | 014604 | 012746 | 000002 | | MOV #2,-(SP) |
| | 014610 | 010600 | | | MOV SP,R0 |
| | 014612 | 104415 | | | TRAP C\$PNTX |
| | 014614 | 062706 | 000006 | | ADD #6,SP |
| 17 | 014620 | 000411 | | PS: BR CPNTX | |
| 18 | 014622 | | | PRINTS R1,#ERRCHR | |
| | 014622 | 012746 | 003230 | | MOV #ERRCHR,-(SP) |
| | 014626 | 010146 | | | MOV R1,-(SP) |
| | 014630 | 012746 | 000002 | | MOV #2,-(SP) |
| | 014634 | 010600 | | | MOV SP,R0 |
| | 014636 | 104416 | | | TRAP C\$PNTS |
| | 014640 | 062706 | 000006 | | ADD #6,SP |
| 19 | 014644 | | | CPNTX: POP R1 | |
| | 014644 | 012601 | | | MOV (SP)+,R1 |
| 20 | 014646 | 000207 | | RETURN | |

```

1          ;PRINT FORMATTED MESSAGE
2          ;
3          ;CALL WITH MACRO PNT, PNTF, PNTB, PNTX, OR PNTS
4
5 014650 012737 014526 003226 LPNTF: MOV #PF,PTYPE
6 014656 000413                BR LPNT
7 014660 012737 014552 003226 LPNTB: MOV #PB,PTYPE
8 014666 000407                BR LPNT
9 014670 012737 014576 003226 LPNTX: MOV #PX,PTYPE
10 014676 000403               BR LPNT
11 014700 012737 014622 003226 LPNTS: MOV #PS,PTYPE
12 014706                LPNT:  PUSH <R2,R3,R4,R5>
13 014706 010246                MOV R2,-(SP)
14 014710 010346                MOV R3,-(SP)
15 014712 010446                MOV R4,-(SP)
16 014714 010546                MOV R5,-(SP)
17 014716 012102               MOV (R1)+,R2
18 014720 010604               MOV SP,R4
19 014722 062704 000012        ADD #10.,R4
20 014726                PUSH R1
21 014726 010146
22 014730 004737 013402        CALL OSTRNG
23 014734                POP <R0,R5,R4,R3,R2,R1>
24 014734 012600
25 014736 012605
26 014740 012604
27 014742 012603
28 014744 012602
29 014746 012601
30 014750 062006               ADD (R0)+,SP
31 014752 000110               JMP @R0

```

```

;GET ADDRESS OF STRING
;COMPUTE ADDRESS OF ARGUMENTS
; WHICH ARE NOW ON STACK (IF ANY)
;SAVE RETURN ADDRESS
MOV R1,-(SP)
;PRINT THE FORMATTED MESSAGE
;RESTORE ALL REGISTERS
MOV (SP)+,R0
MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
;ADJUST STACK POINTER OVER ARGUMENTS
;RETURN

```

```

1      ;DIVIDE
2      ;
3      ;DIVIDE A 32 BIT UNSIGNED NUMBER BY A 16 BIT UNSIGNED NUMBER.
4      ;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
5      ;WILL NOT CHECK FOR DIVIDE BY ZERO.
6      ;
7      ;INPUTS:
8      ;       R3 - LOW 16 BITS OF DIVIDEND
9      ;       R4 - HIGH 16 BITS OF DIVIDEND
10     ;       R0 - DIVISOR
11     ;OUTPUTS:
12     ;       R3 - LOW 16 BITS OF QUOTIENT
13     ;       R4 - HIGH 16 BITS OF QUOTIENT
14     ;       R5 - REMAINDER
15     ;
16     014754 010246 000040  DIVIDE: PUSH R2
17     014756 012702 000040  MOV #32.,R2
18     014762 005005 000040  CLR R5
19     014764 006303 000040  1$: ASL R3
20     014766 006104 000040  ROL R4
21     014770 006105 000040  ROL R5
22     014772 020005 000040  CMP R0,R5
23     014774 101002 000040  BHI 2$
24     014776 160005 000040  SUB R0,R5
25     015000 005203 000040  INC R3
26     015002 005302 000040  2$: DEC R2
27     015004 001367 000040  BNE 1$
28     015006 012602 000040  POP R2
29     015010 000207 000040  RETURN
    
```


| | | | | | | |
|----|--------|--------|--------|--------|-------------------------------|--|
| 1 | 015076 | 017701 | 165062 | | MOV @DMPROG,R1 | :GET SIZE OF PROGRAM |
| 2 | 015102 | 012700 | 000002 | | LOADB: MOV #OP.ESP,R0 | :BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET |
| 3 | 015106 | 004737 | 015214 | | CALL BLDCMD | |
| 4 | 015112 | 013764 | 002164 | 000124 | MOV DMPROG,HC.CPK+P.UADR(R4) | :LOAD MAIN PROGRAM ADDRESS |
| 5 | 015120 | 010164 | 000120 | | MOV R1,HC.CPK+P.BCNT(R4) | : AND SIZE |
| 6 | 015124 | 013764 | 002164 | 000140 | MOV DMPROG,HC.CPK+P.OVRL(R4) | :LOAD OVERLAY ADDRESS |
| 7 | 015132 | 067764 | 165026 | 000140 | ADD @DMPROG,HC.CPK+P.OVRL(R4) | |
| 8 | 015140 | 004737 | 015300 | | CALL SNDCMD | :SEND COMMAND TO UDA |
| 9 | 015144 | 004737 | 015420 | | CALL WAITMS | :WAIT FOR MESSAGE RESPONSE |
| 10 | 015150 | 001417 | | | BEQ LOADER | :ABORT IF NO RESPONSE |
| 11 | 015152 | 032764 | 000037 | 000032 | BIT #ST.MSK,HC.MPK+P.STS(R4) | :CHECK FOR ERRORS |
| 12 | 015160 | 001007 | | | BNE LOADE1 | |
| 13 | 015162 | 042765 | 000024 | 000014 | BIC #CT.CMD+CT.REQ,C.FLG(R5) | :CLEAR COMMAND OUTSTANDING FLAG |
| 14 | 015170 | 052765 | 000002 | 000014 | BIS #CT.RN,C.FLG(R5) | :SET DM PROGRAM RUNNING FLAG |
| 15 | 015176 | 000207 | | | RETURN | |

1
2
3 015200
015200 104455
015202 000042
015204 000000
015206 011060
4 015210 000264
5 015212 000207

;UDA FAILED TO DOWNLINE LOAD DM PROGRAM
LOADE1: ERRDF 34,,ERR034

LOADER: SEZ
RETURN

TRAP C\$ERDF
.WORD 34
.WORD 0
.WORD ERRO34

;SET Z TO INDICATE ERROR OCCURRED

```

1      :BLDCMD
2      :
3      :BUILD A COMMAND IN COMMAND PACKET
4      :
5      :INPUTS:
6      :       R5 - CONTROLLER TABLE ADDRESS
7      :       R0 - COMMAND CODE
8      :
9      :OUTPUTS:
10     :       R4 - ADDRESS OF HOST COMM AREA
11     :       COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL OTHER FIELDS CLEARED.
12     :       CMD REFERENCE NUMBER IN CONTROLLER TABLE INCREMENTED AND RESULT
13     :       IN COMMAND PACKET.
14     :       R0 - CONTENTS DESTROYED
15     BLDCMD: PUSH <R1,R0>
16     015214 010146
17     015214 010046
18     015220 016504 000016
19     015222 010400
20     015226 062700 000100
21     015232 012720 000060
22     015236 012701 001000
23     015242 022716 000031
24     015246 001002
25     015250 012701 177777
26     015254 010120
27     015256 012701 000030
28     015262 005020
29     015264 005301
30     015266 001375
31     015270
32     015270 012664 000114
33     015274
34     015274 012601
35     015276 000207
36
37     MOV C.RING(R5),R4
38     MOV R4,R0
39     ADD #HC.CEV,R0
40     MOV #HC.PSZ,(R0)+
41     MOV #DUP,R1
42     CMP #OP.MWR,(SP)
43     BNE BLDC0
44     MOV #DIAG,R1
45     MOV R1,(R0)+
46     MOV #<HC.PSZ>/2,R1
47     CLR (R0)+
48     DEC R1
49     BNE BLDC1
50     POP HC.CPK+P.OPCD(R4)
51     POP R1
52     RETURN
53
54     MOV R1,-(SP)
55     MOV R0,-(SP)
56     :GET ADDRESS OF HOST COMM AREA
57     :COPY TO R0
58     :COMPUTE ADDRESS OF COMMAND ENVELOPE
59     :LOAD PACKET LENGTH
60     :LOAD DIAG CIRCUIT IDENTIFIER
61     :IF CODE IS MAINTENANCE WRITE
62     : GET OTHER CIRCUIT IDENTIFIER
63
64     :PUT IDENTIFIER INTO PACKET
65     :GET WORDS TO CLEAR
66     :CLEAR PACKET
67
68     :PUT OPCODE IN PACKET
69     MOV (SP)+,HC.CPK+P.OPCD(R4)
70
71     :RESTORE R1
72     MOV (SP)+,R1

```

```

1      :SNDCMD
2
3      :SEND A COMMAND TO THE UDA.
4      :MARK BOTH PACKETS AVAILABLE TO THE
5      :UDA. SET COMMAND ISSUED BIT IN CONTROLLER TABLE AND INITIALIZE
6      :TIMEOUT COUNTER.
7
8      :INPUTS:
9      :   R5 - CONTROLLER TABLE ADDRESS
10     :OUTPUTS:
11     :   R4 - ADDRESS OF HOST COMM AREA
12
13
14     SNDCMD: PUSH <R0,R1>
15     015300      010046      000016      MOV C.RING(R5),R4      :LOAD R4 WITH HOST COMM AREA ADDRESS
16     015300      010146      000052      INC C.REF(R5)          :INCREMENT CMD REFERENCE NUMBER
17     015304      016504      000052      000104      MOV C.REF(R5),HC.CPK+P.CRF(R4) :PUT IN PACKET
18     015310      005265      000006      MOV #RG.OWN+RG.FLG,HC.MCT(R4) :MARK MESSAGE PACKET AVAILABLE
19     015314      016564      000012      MOV #RG.OWN,HC.CCT(R4)      :MARK COMMAND TO UDA
20     015322      012764      140000      000006      TST @ (R5)             :TELL UDA COMMAND IS THERE
21     015330      012764      100000      000012      BIS #CT.CMD,C.FLG(R5)     :MARK COMMAND ISSUED
22     015336      005775      000000      POP <R1,R0>
23     015354      000207      RETURN
24     015350      012601      MOV (SP)+,R1
25     015352      012600      MOV (SP)+,R0
    
```

```

1       ;CLRBUF
2
3       ;CLEAR THE SPECIFIED DATA BUFFER IN THE HOST COMM AREA
4       ;AND LOAD BUFFER DESCRIPTOR IN COMMAND PACKET TO THE BUFFER
5
6       ;INPUTS:
7       ;     R5 - CONTROLLER TABLE ADDRESS
8       ;     R4 - ADDRESS OF HOST COMM AREA
9       ;     R0 - OFFSET INTO HOST COMM AREA TO DATA BUFFER
10      ;OUTPUTS:
11      ;     DATA BUFFER CLEARED
12      ;     COMMAND PACKET POINTING TO BUFFER
13      ;     BYTE COUNT SET TO SIZE OF BUFFER
14      ;     R4 - ADDRESS OF DATA BUFFER
15
16      CLRBUF: PUSH <R0,R1>
17
18
19
20
21
22      CLRBFL: CLR (R0)+
23             DEC R1
24             BNE CLRBFL
25             POP <R1,R0>
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
2
```

```

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     015420      WAITMS: PUSH <R0,R1>
12     015420      010046
13     015422      010146
14     015424      012700      000036
15     015430      010501
16     015432      062701      000040
17     015436      004737      015612
18     015442      011500
19     015444      032765      000010      000014
20     015452      001030
21     015454      016001      000002
22     015460      001034
23     015462
24     015462      104422
25     015464      005737      003212
26     015470      001764
27     015472      023765      003224      000042
28     015500      101005
29     015502      001357
30     015504      023765      003222      000040
31     015512      103753
32     015514
33     015514      104455
34     015516      000044
35     015520      000000
36     015522      011066
37     015524
38     015524      012601
39     015526      012600
40     015530      000264
41     015532      000207

      :MOV #30.,R0
      :MOV R5,R1
      :ADD #C.TO,R1
      :CALL SETTO
      :MOV (R5),R0
      :BIT #CT.MSG,C.FLG(R5)
      :BNE 3$
      :MOV 2(R0),R1
      :BNE 4$
      :BREAK

      :TST KW.CSR
      :BEQ 1$
      :CMP KW.EL+2,C.TOH(R5)
      :BHI 2$
      :BNE 1$
      :CMP KW.EL,C.TO(R5)
      :BLO 1$
      :ERRDF 36,,ERR036

      :MOV R0,-(SP)
      :MOV R1,-(SP)
      :SET TIME OUT VALUE OF 30 SECONDS
      :POINT TO TIME OUT COUNTER

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

      TRAP      C$BRK
      :SEE IF A CLOCK ON SYSTEM

      :CHECK IF TIMEOUT HAS HAPPENED

      TRAP      C$ERDF
      .WORD      36
      .WORD      0
      .WORD      ERR036

      POP <R1,R0>

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

      SEZ
      RETURN
    
```

```
1 015534 042765 000010 000014 3$: BIC #CT.MSG,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
2 015542 012601 POP <R1,R0>
  015544 012600
3 015546 000244 CLZ
4 015550 000207 RETURN      ;GIVE NO ERROR RETURN
5 015552 104455 ERRDF 37,,ERR037
  015554 000045 TRAP C$ERDF
  015556 000000 .WORD 37
  015560 011100 .WORD 0
6 015562 POP <R1,R0>      .WORD ERR037
  015564 012601 MOV (SP)+,R1
  015566 000264 MOV (SP)+,R0
8 015570 000207 SEZ
  RETURN
```

```
1          :NXMI
2          :
3          :NON-EXISTANT MEMORY SERVICE ROUTINE
4          :
5          :INPUTS:
6          :      NXMAD SET TO ZERO
7          :OUTPUTS:
8          :      NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
9
10         BGNSRV NXMI
11
12         015572 012737 177777 002202          MOV #-1,NXMAD
13
14         015600          ENDSRV
          015600          L10030:
          015600 000002          RTI
```



```
1      ;UDASRV
2      ;
3      ;UDA INTERRUPT SERVICE ROUTINE. MARKS UDA CONTROLLER TABLE THAT AN
4      ;INTERRUPT HAS BEEN RECEIVED.
5      ;
6      ;THIS ROUTINE IS CALLED BY A [JSR RO,UDASRV] INSTRUCTION FROM WITHIN
7      ;THE CONTROLLER TABLE. THE PC STORED IN RO IS THE ADDRESS OF THE C.FLG
8      ;WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
9      ;OF RO FOLLOWED BY THE INTERRUPTED PC AND PS.
10     ;
11     ;INPUTS:
12     ;   RO - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
13     ;   STACK - SAVED CONTENTS OF RO
14     ;OUTPUTS:
15     ;   CT.CMD CLEARED AND CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
16     ;   RO - RESTORED FROM STACK
17
18 015602 BGNSRV UDASRV
19 015602         BIS #CT.MSG,(RO)           ;SET CT.MSG
20 015606         POP RO                     ;RESTORE RO
21 015610         ENDSRV
22 015610         UDASRV::
23 015610         MOV (SP)+,RO
24 015610         L10031:
25 015610         RTI
26 015610         052710 000010
27 015610         012600
28 015610         000002
```

```

1      ;SETTO
2
3      ;SET TIMEOUT COUNTER TO SOME NUMBER OF SECONDS FROM CURRENT TIME.
4
5      ;INPUTS:
6      ;      R0 - NUMBER OF SECONDS FOR TIMEOUT
7      ;      R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT
8
9      ;OUTPUTS:
10     ;      R0 - CONTENTS DESTROYED
11     ;      R1 - INCREMENTED BY 2
12
13     ;COMPUTE CLOCK TICKS TIL TIMEOUT
14     SETTO:  PUSH <R2,R3>
15     015612 010246
16     015614 010346
17     015616 005002
18     015620 013703 003220
19     015624 006200
20     015626 103001
21     015630 060302
22     015632 006303
23     015634 005700
24     015636 001372
25
26     ;CLEAR PRODUCT
27     ;GET MULTIPLICAND
28     ;SHIFT MULTIPLIER TO RIGHT
29     ;IF A ONE BIT SHIFTED OUT
30     ; ADD MULTIPLICAND TO PRODUCT
31     ;DOUBLE THE MULTIPLICAND
32
33     ;CONTINUE UNTIL MULTIPLIER IS ZERO
34
35     ;GET CURRENT TIME
36
37     SETO2:  MOV KW.EL,R0
38     ;GET TIME
39     MOV KW.EL+2,R3
40     ;IF CHANGED DURING RETRIEVAL
41     CMP R0,KW.EL
42     ; GET IT AGAIN
43     BNE SETO2
44
45     ;ADD TIME TIL TIMEOUT
46
47     ADD R2,R0
48     ;ADD
49     ADC R3
50
51     ;PUT RESULT IN STORAGE
52
53     MOV R0,(R1)+
54     MOV R3,(R1)
55
56     POP <R3,R2>
57
58     MOV (SP)+,R3
59     MOV (SP)+,R2
60
61     RETURN

```

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

```

:UDAIN
:FUNCTIONAL DESCRIPTION:
SUBROUTINE TO INITIALIZE A UDA AND BRING IT ON-LINE.
ALL STEPS ARE CHECKED. AN ERROR MESSAGE IS REPORTED IF ANY ERROR
DETECTED.
:INPUTS:
R5 - ADDRESS OF CONTROLLER TABLE.
:IMPLICIT INPUTS:
C.RING(R5) - ADDRESS GIVEN TO UDA AS START OF RING BUFFER.
LENGTH OF RING STRUCTURE IS ONE ENTRY EACH.
:OUTPUTS:
CONDITION Z - SET IF ANY ERROR REPORTED. CLEAR IF NO ERROR.
R4 - ADDRESS OF UDAIP REGISTER IN UDA
R5 - UNCHANGED.
:FILL HOST COMMUNICATION AREA WITH ALL ONES
UDAIN: MOV C.RING(R5),R2 ;GET FIRST ADDRESS OF RING BUFFER
MOV #<HC.RSZ*2+HC.ISZ>/2,R3 ;GET SIZE OF RING BUFFER
UDAI1L: MOV #-1,(R2)+ ;WRITE ONES TO BUFFER
DEC R3 ;COUNT THE WORDS IN BUFFER
BGT UDAI1L ;LOOP UNTIL ENTIRE BUFFER WRITTEN
:DO THE INITIALIZATION
CALL UDAIST ;DO FIRST THREE STEPS
BCS UDAIEX ;GET OUT IF UDA MICROCODE REPORTED FAILURE
MOV (R3)+,2(R4) ;WRITE NEXT WORD TO UDASA REGISTER
MOV #200,,R3 ;GET TRY COUNTER
UDAI1A: MOV 2(R4),R2 ;LOOK AT UDASA
BEQ UDAI1C
BPL UDAI1B
ERRDF 24,,ERR024
TRAP CSERDF
.WORD 24
.WORD 0
.WORD ERR024
BR UDAIEX
UDAI1B: DEC R3
BNE UDAI1A
UDAI1C: MOV R2,2(R4) ;WRITE 0 TO UDASA (PURGE)
MOV (R4),R2 ;READ FROM UDAIP (POLL)
CALL UDARSP ;WAIT FOR STEP OR ERROR BIT
BCS UDAIEX ;GET OUT IF UDA MICROCODE REPORTED FAILURE
    
```

```

015674 016502 000016
015700 012703 000006
015704 012722 177777
015710 005303
015712 003374
015714 004737 016110
015720 103471
015722 012364 000002
015726 012703 000310
015732 016402 000002
015736 001410
015740 100005
015742
015742 104455
015744 000030
015746 000000
015750 010754
015752 000454
015754 005303
015756 001365
015760 010264 000002
015764 011402
015766 004737 016434
015772 103444
    
```

```

1          ;CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
2
3 015774   016502   000016   UDAI2:  MOV C.RING(R5),R2          ;GET FIRST ADDRESS OF RING BUFFER
4 016000   012703   000006   MOV #<HC.RSZ*2+HC.ISZ>/2,R3      ;GET SIZE OF RING BUFFER
5 016004   005722                                     UDAI2L: TST (R2)+                ;CHECK WORD IN BUFFER
6 016006   001003   BNE UDAI2E                       ;GO TO ERROR REPORTER IF NOT ZERO
7 016010   005303   DEC R3                            ;COUNT THE WORDS IN BUFFER
8 016012   003374   BGT UDAI2L                       ;LOOP UNTIL ALL WORDS CHECKED
9 016014   000405   BR UDAI3
10
11 016016   UDAI2E: ERRDF 23,,ERR023 ;REPORT BUFFER NOT CLEARED
    016016   104455                                     TRAP      C$ERDF
    016020   000027                                     .WORD    23
    016022   000000                                     .WORD    0
    016024   010670                                     .WORD    ERR023
12 016026   000426   BR UDAIEX
13
14          ;SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION
15
16 016030   UDAI3:
17 016030   016500   000006   MOV C.BST(R5),R0                ;GET BURST VALUE
18 016034   006300   ASL R0                          ;SHIFT TO POSITION
19 016036   006300   ASL R0
20 016040   052700   000001   BIS #SA.GO,R0                  ;SET THE GO BIT
21 016044   010064   000002   MOV R0,2(R4)                   ;SEND TO UDA
22 016050   016501   000016   MOV C.RING(R5),R1
23 016054   010161   000004   MOV R1,HC.MSG(R1)
24 016060   062761   000020   000004   ADD #HC.MPK,HC.MSG(R1)
25 016066   010161   000010   MOV R1,HC.CMD(R1)
26 016072   062761   000104   000010   ADD #HC.CPK,HC.CMD(R1)
27 016100   000244   CLZ                              ;CLEAR Z AS NO ERROR INDICATION
28 016102   000207   RETURN
29
30          ;ERROR RETURN
31
32 016104   000264   UDAIEX: SEZ                      ;SET Z TO INDICATE ERROR OCCURRED
33 016106   000207   RETURN
    
```

```

1      ;UDAIST
2
3      ;START THE INITIALIZATION PROCESS ON THE SELECTED UDA.
4      ;STOP BEFORE WRITING THE THIRD WORD SO UDA DOES NOT
5      ;ATTEMPT ANY UNIBUS TRANSFERS.
6
7      ;INPUTS:
8      ;      R5 - ADDRESS OF CONTROLLER TABLE
9
10     ;LOAD TABLE OF DATA TO SEND TO UDASA REGISTER
11
12     UDAIST: BREAK
13     016110 104422          TRAP    C$BRK
14     016112 010146          PUSH R1
15     016114 016504 000004   MOV C.VEC(R5),R4
16     016120 042704 177000   AND CT.VEC,R4
17     016124 006204          ASR R4
18     016126 006204          ASR R4
19     016130 052704 100000   BIS #SA.STP,R4
20     016134 010437 016326   MOV R4,UDAID1
21     016140 016537 000016 016332  MOV C.RING(R5),UDAID2
22     016146 062737 000004 016332  ADD #HC.MSG,UDAID2
23
24     ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER
25     016154 016504 000000   MOV C.UADR(R5),R4
26     016160 005037 002202   CLR NXMAD
27     016164          SETVEC #4,#NXMI,#PRI07
28     016164 012746 000340   MOV #PRI07,-(SP)
29     016170 012746 015572   MOV #NXMI,-(SP)
30     016174 012746 000004   MOV #4,-(SP)
31     016200 012746 000003   MOV #3,-(SP)
32     016204 104437          TRAP   C$SVEC
33     016206 062706 000010   ADD #10,SP
34     016212 005764 000002   TST 2(R4)
35     016216 005014          CLR (R4)
36     016220          CLRVEC #4
37     016220 012700 000004   TST NXMAD
38     016224 104436          BEQ UDAISG
39     016226 005737 002202   ERRDF 20,,ERR020
40     016232 001406          TRAP   C$ERDF
41     016234          .WORD 20
42     016234 104455          .WORD 0
43     016236 000024          .WORD 0
44     016240 000000          .WORD 0
45     016242 010556          .WORD 0
46     016244 000261          SEC
47     016246 000424          BR UDAISE
    
```

```

1          ;SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
2
3 016250 012737 004000 016572 UDAISG: MOV #SA.S1,UDARSD      ;STORE RESPONSE MASK
4 016256 012703 016324          MOV #UDAIDT,R3          ;AND INDEX TO TABLE
5
6          ;WAIT FOR AND CHECK RESPONSE DATA
7
8 016262 004737 016434          UDAISL: CALL UDARSP          ;WAIT FOR STEP OR ERROR BITS
9 016266 103414                  BCS UDAISE          ;EXIT IF ERROR
10 016270 004733                  CALL @R3+          ;CALL RESPONSE CHECKER FOR STEP
11 015272 103412                  BCS UDAISE          ;GET OUT IF ERROR
12 016274 006337 016572          ASL UDARSD          ;SHIFT TO NEXT STEP BIT
13 016300 032737 040000 016572  BIT #SA.S4,UDARSD      ;CHECK IF NOW AT STEP 4
14 016306 001003                  BNE UDAISX          ;GET OUT IF SO
15 016310 012364 000002          MOV (R3)+,2(R4)     ;WRITE DATA TO UDASA REGISTER
16 016314 000762                  BR UDAISL          ;STAY IN LOOP
17
18 016316 000241          UDAISX: CLC          ;CLEAR CARRY FOR NO ERROR INDICATION
19 016320          UDAISE: POP R1
20 016322 012601          RETURN
    000207          MOV (SP)+,R1
    
```

```
1          ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION
2
3 016324 016342 UDA!DT: .WORD UDAIR1          ;FIRST WORD RESPONSE CHECK ROUTINE
4 016326 000000 UDAID1: .WORD 0          ;FIRST WORD TO SEND TO UDASA
5 016330 016350 UDAID2: .WORD UDAIR2          ;SECOND WORD RESPONSE CHECK ROUTINE
6 016332 000000 UDAID3: .WORD 0          ;SECOND WORD TO SEND TO UDASA
7 016334 016370 UDAID4: .WORD UDAIR3          ;THIRD WORD RESPONSE CHECK ROUTINE
8 016336 100000 UDAID5: .WORD SA.TST          ;THIRD WORD TO SEND TO UDASA
9 016340 016406 UDAID6: .WORD UDAIR4          ;FOURTH WORD RESPONSE CHECK ROUTINE
10
11         ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
12         ;CHECK FOR PROPER CONTROLLER TYPE
13
14 016342 012701 004400 UDAIR1: MOV #SA.S1+SA.DI,R1          ;SET STEP ONE BIT
15 016346 000422          BR UDAIRC          ;NOW COMPARE
16
17         ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
18         ;CHECK FOR ECHO OF INTI AND VECTOR
19
20 016350 013701 016326 UDAIR2: MOV UDAID1,R1          ;GET WORD SENT TO UDASA
21 016354 000301          SWAB R1          ;GET HIGH 8 BITS
22 016356 042701 177400          BIC #177400,R1
23 016362 052701 010000          BIS #SA.S2,R1          ;SET STEP 2 BIT
24 016366 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 016370 013701 016326 UDAIR3: MOV UDAID1,R1          ;GET WORD SENT TO UDASA
30 016374 042701 177400          BIC #177400,R1          ;JUST LOW 8 BITS
31 016400 052701 020000          BIS #SA.S3,R1          ;SET STEP 3 BIT
32 016404 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 016406 010201 UDAIR4: MOV R2,R1          ;GET RESPONSE FROM UDA
38 016410 042701 137400          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 016414 020102 UDAIRC: CMP R1,R2          ;COMPARE THE DATA
43 016416 001405          BEQ UDAIRX          ;EXIT IF COMPARED CORRECTLY
44 016420          ERARDF 25,,ERR025          ;REPORT ERROR
45         TRAP CSERDF
46         .WORD 25
47         .WORD 0
48         .WORD ERR025
49         SEC
50 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:     ;OUTPUTS:
14:     ;   ERROR MESSAGE IF TIME OUT ON RESPONSE OR ERROR BIT SETS
15:     ;   R2 - DATA FROM UDASA REGISTER
16:     ;   CARRY SET IF ERROR BIT SETS OR TIME OUT
17:     ;
18:     UDARSP: PUSH R1
19:     016434 010146          ;   MOV R1,-(SP)
20:     016436 052737 100000 016572  ;   BIS #SA.ERR,UDARSD      ;SET ERROR BIT IN MASK WORD
21:     016444 012700 000012          ;   MOV #10.,R0            ;SET UP FOR 10 SECOND TIMEOUT
22:     016450 010501          ;   MOV R5,R1              ;POINT TO COUNTER IN CONTROLLER TABLE
23:     016452 062701 000040          ;   ADD #C.TO,R1
24:     016456 004737 015612          ;   CALL SETTO
25:     016462          ;   POP R1
26:     016462 012601          ;   MOV (SP)+,R1
27:     016464 033764 016572 000002 UDARS1: BIT UDARSD,2(R4)  ;LOOK AT ERROR AND STEP BIT
28:     016472 001024          ;   BNE UDARS2             ;BRANCH IF EITHER SET
29:     016474          ;   BREAK
30:     016474 104422          ;   TST KW.CSR             ;SEE IF CLOCK ON SYSTEM
31:     016476 005737 003212          ;   BEQ UDARS1
32:     016502 001770          ;   CMP KW.EL+2,C.TO(R5)  ;CHECK IF TIME OUT OCCURRED
33:     016504 023765 003224 000042  ;   BHI 1$
34:     016512 101005          ;   BNE UDARS1
35:     016514 001363          ;   CMP KW.EL,C.TO(R5)
36:     016516 023765 003222 000040  ;   BLO UDARS1
37:     016524 103757          ;   MOV 2(R4),R2           ;GET REGISTER CONTENTS
38:     016526 016402 000002 1$:   ;   ERRDF 22,,ERR02       ;REPORT TIME OUT ERROR
39:     016532          ;
40:     016532 104455          ;   TRAP C$BRK
41:     016534 000026          ;   .WORD 22
42:     016536 000000          ;   .WORD 0
43:     016540 010622          ;   .WORD ERR02
44:     016542 000407          ;   BR UDARSE

```



```
1          ;CHECK IF ERROR BIT SET
2
3 016544 016402 000002      UDARS2: MOV 2(R4),R2          ;GET REGISTER CONTENTS
4 016550 100006          BPL UDARSX          ;EXIT IF ERROR NOT SET
5 016552          ERRDF 21,,ERR021          ;REPORT ERROR INFO
   016552 104455
   016554 000025
   016556 000000
   016560 010570
6 016562 000261          UDARSE: SEC
7 016564 000207          RETURN
8
9          ;NORMAL EXIT
10
11 016566 000241          UDARSX: CLC          ;CLEAR CARRY AS NO ERROR INDICATION
12 016570 000207          RETURN
13
14          ;LOCATION FOR STEP BIT MASK
15
16 016572 000000          UDARSD: .WORD 0          ;LOAD BY CALLING ROUTINE
```

```
TRAP  CSERDF
.WORD 21
.WORD 0
.WORD ERR021
```

```

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     016574      CLOG:  PUSH R0
11     016574      010046      ;
12     016576      005001      ;
13     016600      000261      ;   CLR R1      ;SET UP ZERO START VALUE      MOV R0,-(SP)
14     016602      006101      ;   SEC      ;WITH CARRY READY TO SHIFT IN
15     016604      005300      CLOGLP: ROL R1      ;SHIFT TO LEFT
16     016606      100375      ;   DEC R0      ; UNTIL R0
17     016610      012600      ;   BPL CLOGLP  ; GOES NEGATIVE
18     016612      000207      ;   POP R0      ;
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 ;

```

```
1          ;KW11I
2          ;
3          ;CLOCK INTERRUPT SERVICE ROUTINE
4
5 016614    BGNSRV KW11I
6 016614    062737 000001 003222          ADD #1,KW.EL          KW11I::
7 016622    005537 003224          ADC KW.EL+2          ;COUNT THE INTERRUPT
8 016626    012777 000105 164356        MOV #KWOUT.,@KW.CSR    ;RESTART THE CLOCK
9 016634    ENDSRV
016634
016634    000002          L10032:
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     016636 005737 003212 RNTIME: TST KW.CSR           ;CHECK IF A CLOCK PRESENT
14     016642 001465        BEQ RNTIMX           ;BRANCH IF NOT
15     016644        PUSH <R0,R3,R4,R5>
16     016644 010046        MOV R0,-(SP)
17     016646 010346        MOV R3,-(SP)
18     016650 010446        MOV R4,-(SP)
19     016652 010546        MOV R5,-(SP)
20     016654 013703 003222 MOV KW.EL,R3           ;GET ELAPSED TIME
21     016660 013704 003224 MOV KW.EL+2,R4
22     016664 013700 003220 MOV KW.HZ,R0           ;GET SPEED OF CLOCK
23     016670 004737 014754 CALL DIVIDE           ;COMPUTE SECONDS OF ELAPSED TIME
24     016674 012700 000074 MOV #60,R0           ;NOW DIVIDE BY 60
25     016700 004737 014754 CALL DIVIDE           ; TO COMPUTE MINUTES
26     016704        PUSH R5           ;SAVE REMAINDER AS SECONDS
27     016704 010546        MOV R5,-(SP)
28     016706 004737 014754 CALL DIVIDE           ;DIVIDE BY 60 AGAIN
29     016712        PNT RNTIM,R3       ;PRINT HOURS
30     016712 010346        MOV R3,-(SP)
31     016714 004137 014706 JSR R1,LPNT
32     016720 003620        .WORD RNTIM
33     016722 000002        .WORD PNT.CT
34     016724 020527 000011 CMP R5,#9           ;IF MINUTES 9 OR LESS
35     016730 003004        BGT 1$
36     016732        PRINT #'0         ;PRINT A LEADING ZERO
37     016732 112700 000060        MOVB #'0,R0
38     016736 004737 014476        CALL CPNT
39     016742 1$: PNT RNTIM1,R5       ;NOW PRINT MINUTES
40     016742 010546        MOV R5,-(SP)
41     016744 004137 014706 JSR R1,LPNT
42     016750 003643        .WORD RNTIM1
43     016752 000002        .WORD PNT.CT
44     016754        POP R5           ;GET SECONDS
45     016754 012605        MOV (SP)+,R5
46     016756 020527 000011 CMP R5,#9           ;IF 9 OR LESS
47     016762 003004        BGT 2$
48     016764        PRINT #'0         ;PRINT A LEADING ZERO
49     016764 112700 000060        MOVB #'0,R0
50     016770 004737 014476        CALL CPNT
51     016774 2$: PNT RNTIM2,R5       ;NOW PRINT SECONDS
52     016774 010546        MOV R5,-(SP)
53     016776 004137 014706 JSR R1,LPNT
54     017002 003651        .WORD RNTIM2
55     017004 000002        .WORD PNT.CT
56     017006        POP <R5,R4,R3,R0> ;HOURS IN R3
57     017006 012605        MOV (SP)+,R5

```

```

017010 012604
017012 012603
017014 012600
35 017016
017016 112700 000040
017022 004737 014476
36 017026 000207
37
38 017030

RNTIMX: PRINT '<#>'
;PRINT A SPACE

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

MOVB #',R0
CALL CPNT

RETURN

ENDMOD

```

```
1          .SBTTL  PROTECTION TABLE
2
3 017030          BGNMOD
4
5          :++
6          : THIS TABLE IS USED BY THE RUNTIME SERVICES
7          : TO PROTECT THE LOAD MEDIA.
8          :--
9
10 017030          BGNPROT
11 017030
12 017030 177777          -1          :OFFSET INTO P-TABLE FOR CSR ADDRESS
13 017032 177777          -1          :OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
14 017034 177777          -1          :OFFSET INTO P-TABLE FOR DRIVE NUMBER
15
16 017036          ENDPROT
17
```

```

1      .SBTTL INITIALIZE SECTION
2
3
4      :++
5      : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
6      : AT THE BEGINNING OF EACH PASS.
7      :--
8      017036      BGNINIT
9      017036
10     017036      READEF #EF.START          ;CHECK IF STARTED BY OPERATOR
11     017036      012700 000040          MOV #EF.START,RO
12     017042      104447          TRAP CSREFG
13     017044      BCOMPLETE INIT1        ; IF NOT,
14     017044      103415          BCS INIT1
15     017046      READEF #EF.RESTART
16     017046      012700 000037          MOV #EF.RESTART,RO
17     017052      104447          TRAP CSREFG
18     017054      BCOMPLETE INIT1
19     017054      103411          BCS INIT1
20     017056      READEF #EF.CONTINUE
21     017056      012700 000036          MOV #EF.CONTINUE,RO
22     017062      104447          TRAP CSREFG
23     017064      BCOMPLETE INIT1
24     017064      103405          BCS INIT1
25     017066      READEF #EF.PWR
26     017066      012700 000034          MOV #EF.PWR,RO
27     017072      104447          TRAP CSREFG
28     017074      BCOMPLETE INIT1
29     017074      103401          BCS INIT1
30     017076      INITQT: DOCLN          ; ABORT PROGRAM
31     017076      104444          TRAP CSDCLN
32     017100      012700 000003      INIT1: MOV #SO.FMT,RO ;BUILD MODE WORD FROM SOFTWARE QUESTIONS
33     017104      030037 002144      BIT RO,SFPTBL ;SEE IF REFORMAT
34     017110      001011      BNE INIT2 ; BRANCH IF SO
35     017112      012700 000004      MOV #SO.CNS,RO ;SEE IF RECONSTRUCT
36     017116      030037 002144      BIT RO,SFPTBL
37     017122      001004      BNE INIT2 ; BRANCH IF SO
38     017124      006300      ASL RO ;SEE IF RESTORE
39     017126      030037 002144      ASSUME SO.STR EQ SO.CNS*2
40     017132      001761      BEQ INITQT ;IF NOT, ABORT PROGRAM
41     017134      010037 003210      INIT2: MOV RO,MODE ;SAVE MODE FLAGS
42     017140      005737 003312      TST DATEO ;SEE IF ALREADY ASKED FOR DATE
43     017144      001002      BNE INIT3
44     017146      004737 020012      CALL DATE ;IF NOT, GET IT NOW
45     017152      017152 104433      INIT3: BRESET ;RESET ALL UNITS
46     017154      017154 104431      MEMORY FFREE ;RESET START OF FREE MEMORY
47     017156      010037 002146      TRAP CSRESET
48     017162      017737 162760 002150      MOV @FFREE,FSIZE ;RESET SIZE OF FREE MEMORY
49     TRAP CSMEM
50     MOV RO,FFREE
51
52     ;INITIALIZE CLOCK
53
54     KWOUT.=105 ;DATA TO SEND TO KW11 TO START CLOCK
55     CLR KW.CSR ;MARK CLOCK AS NOT ON SYSTEM

```

| | | | | | | | | |
|----|--------|--------|--------|--------|-----------------------------|--|-------------------------------------|------------------|
| 41 | 017174 | 005037 | 003222 | | CLR KW.EL | | ;CLEAR ELAPSED TIME | |
| 42 | 017200 | 005037 | 003224 | | CLR KW.EL+2 | | | |
| 43 | 017204 | | | | CLOCK L,RO | | ;SEE IF AN L CLOCK PRESENT | |
| | 017204 | 012700 | 000114 | | | | | MOV #L,RO |
| | 017210 | 104462 | | | | | | TRAP C\$CLCK |
| 44 | 017212 | | | | BCOMplete KYES | | | |
| | 017212 | 103413 | | | | | | BCS KYES |
| 45 | 017214 | | | | CLOCK P,RO | | ;SEE IF A P CLOCK PRESENT | |
| | 017214 | 012700 | 000120 | | | | | MOV #P,RO |
| | 017220 | 104462 | | | | | | TRAP C\$CLCK |
| 46 | 017222 | | | | BCOMplete KYES | | | |
| | 017222 | 103407 | | | | | | BCS KYES |
| 47 | 017224 | 005037 | 003212 | | CLR KW.CSR | | ;IF NEITHER, CLEAR CSR STORAGE WORD | |
| 48 | 017230 | | | | PNTF NOCLOCK | | | |
| | 017230 | 004137 | 014650 | | | | | JSR R1,LPNTF |
| | 017234 | 004007 | | | | | | .WORD NOCLOCK |
| | 017236 | 000000 | | | | | | .WORD PNT.CT |
| 49 | 017240 | 000426 | | | BR KNO | | | |
| 50 | 017242 | 012037 | 003212 | KYES: | MOV (RO)+,KW.CSR | | ;STORE DATA RETURNED | |
| 51 | 017246 | 012037 | 003214 | | MOV (RO)+,KW.BRL | | | |
| 52 | 017252 | 012037 | 003216 | | MOV (RO)+,KW.VEC | | | |
| 53 | 017256 | 012037 | 003220 | | MOV (RO)+,KW.HZ | | | |
| 54 | 017262 | | | | SETVEC KW.VEC,#KW111,KW.BRL | | ;SET THE VECTOR | |
| | 017262 | 013746 | 003214 | | | | | MOV KW.BRL,-(SP) |
| | 017266 | 012746 | 016614 | | | | | MOV #KW111,-(SP) |
| | 017272 | 013746 | 003216 | | | | | MOV KW.VEC,-(SP) |
| | 017276 | 012746 | 000003 | | | | | MOV #3,-(SP) |
| | 017302 | 104437 | | | | | | TRAP C\$SVEC |
| | 017304 | 062706 | 000010 | | | | | ADD #10,SP |
| 55 | 017310 | 012777 | 000105 | 163674 | MOV #KWOUT.,@KW.CSR | | ;START THE CLOCK | |
| 56 | 017316 | | | KNO: | | | | |


```

1          ;INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
2
3 017316 013737 002146 002156          MOV FFREE,CTABS          ;STORE START OF CONTROLLER TABLES
4 017324 005077 162626          CLR @CTABS          ;ZEROS MARKS END CONTROLLER TABLES
5 017330 005037 002160          CLR CTRLRS          ;CLEAR CONTROLLER COUNT
6
7          ;GET A P-TABLE FROM DRS
8
9 017334 005002          CLR R2          ;LOGICAL UNIT NUMBER IN R2
10 017336          INIT4: GPHARD R2,R0          ;GET POINTER TO A P-TABLE
    017336 010200          ;
    017340 104442          ;
11 017342          BNCOMPLETE NXTTAB          ;IGNORE IF NO TABLE RETURNED
    017342 103075          ;
    ;
12          ;SEE IF A CONTROLLER TABLE ALREADY EXISTS FOR CONTROLLER IN P-TABLE
13
14
15 017344 013703 002156          MOV CTABS,R3          ;GET ADDRESS OF CONTROLLER TABLES
16 017350 005713          INITS: TST (R3)          ;CHECK IF ANY MORE TABLES
17 017352 001416          BEQ NEWTAB          ;BUILD NEW TABLE IF FOUND ZERO WORD
18 017354 021013          CMP (R0),(R3)          ;CHECK IF SAME UNIBUS ADDRESS
19 017356          ;
20 017356          ;
    ;
21 017356 001450          BEQ SAMTAB          ;CHECK TABLE IF ALREADY EXISTS
22 017360 016301 000004          MOV C.VEC(R3),R1          ;GET VECTOR FROM EXISTING CONTROLLER TABLE
23 017364 042701 177000          BIC #^C<CT.VEC>,R1
24 017370 026001 ^00002          CMP HO.VEC(R0),R1
25 017374 001002          BNE 1$          ;SEE IF DIFFERENT VECTOR
26 017376 000137 017776          JMP SAMVEC          ;ERROR, CAN'T HAVE TWO UDA'S WITH SAME VECTOR
27 017402 062703 000054          1$: ADD #C.SIZE,R3          ;MOVE TO NEXT TABLE
28 017406 000760          BR INITS
    
```

| | | | | | |
|----|--------|--------|--------|--|--|
| 1 | | | | | |
| 2 | | | | | |
| 3 | 017410 | 012701 | 000026 | | |
| 4 | 017414 | 004737 | 011346 | | |
| 5 | 017420 | 011021 | | | |
| 6 | 017422 | 010221 | | | |
| 7 | 017424 | 016004 | 000004 | | |
| 8 | 017430 | 000304 | | | |
| 9 | 017432 | 006104 | | | |
| 10 | 017434 | 056004 | 000002 | | |
| 11 | 017440 | 010421 | | | |
| 12 | 017442 | 016021 | 000006 | | |
| 13 | 017446 | 012721 | 004037 | | |
| 14 | 017452 | 012721 | 015602 | | |
| 15 | 017456 | 012703 | 000020 | | |
| 16 | | | | | |
| 17 | 017462 | 005021 | | | |
| 18 | 017464 | 005303 | | | |
| 19 | 017466 | 001375 | | | |
| 20 | 017470 | 005237 | 002160 | | |
| 21 | 017474 | 005011 | | | |
| 22 | 017476 | 000417 | | | |


```

;BUILD A CONTROLLER TABLE
NEWTAB: MOV #C.SIZE/2,R1          ;GET WORDS IN CONTROLLER TABLE
        CALL ALOCM                ;ALLOCATE SPACE FOR IT
        MOV (R0),(R1)+            ;STORE UNIBUS ADDRESS
        MOV R2,(R1)+             ;UNIT NUMBER
        MOV HO.BRL(R0),R4        ;GET BR LEVEL
        SWAB R4                   ;SWAP TO HIGH BYTE
        ROL R4                    ;SHIFT ONE MORE TO LEFT
        BIS HO.VEC(R0),R4        ;ADD VECTOR ADDRESS
        MOV R4,(R1)+             ; TO TABLE
        MOV HO.BST(R0),(R1)+
        MOV #4037,(R1)+
        MOV #UDASRV,(R1)+
        MOV #16.,R3              ;PUT [JSR R0,UDASRV]
                                   ; INTO TABLE
                                   ;CLEAR POINTERS TO DRIVE TABLES,
                                   ; TIMEOUT COUNTER, FLAGS, REF. NUMBER

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

;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 017500 016004 000004      SAMTAB: MOV HO.BRL(R0),R4      ;GET BR LEVEL FROM P-TABLE
4 017504 000304              SWAB R4                ;SWAP TO HIGH BYTE
5 017506 006104              ROL R4                  ;SHIFT ONE MORE TO LEFT
6 017510 056004 000002      BIS HO.VEC(R0),R4        ;ADD VECTOR ADDRESS
7 017514 020463 000004      CMP R4,C.VEC(R3)        ;COMPARE WITH CONTROLLER TABLE
8 017520 001004              BNE 1$
9 017522 026063 000006 000006  CMP HO.BST(R0),C.BST(R3) ;COMPARE BURST RATES
10 017530 001402              BEQ NXTTAB
11 017532 000137 017726      1$: JMP CTABER        ;FATAL ERROR IF NOT SAME
12
13          ;GET NEXT P-TABLE
14
15 017536 005202              NXTTAB: INC R2          ;INCREMENT LOGICAL UNIT NUMBER
16 017540 023702 002012      CMP L$UNIT,R2        ;CHECK IF GOT ALL TABLES
17 017544 003274              BGT INIT4            ;IF NOT, GO BACK FOR NEXT
18
19 017546 012701 000001      MOV #1,R1          ;ALLOCATE SPACE FOR ZERO END WORD
20 017552 004737 011346      CALL ALOCM        ;AFTER CONTROLLER TABLES
```

```
1  
2  
3 017556 005002  
4 017560 010200  
   017562 104442  
5 017564 103040  
   017564 103040  
6  
7  
8  
9 017566 013703 002156  
10 017572 021013  
11 017574 001403  
12 017576 062703 000054  
13 017602 000773  
  
;NOW BUILD DRIVE TABLES  
  
INIT8: CLR R2  
        GPHARD R2,R0  
  
        BNCOMplete INIT14  
  
;FIND CONTROLLER TABLE  
  
INIT10: MOV CTABS,R3  
        CMP (R0),(R3)  
        BEQ INIT11  
        ADD #C.SIZE,R3  
        BR INIT10  
  
;LOGICAL UNIT NUMBER IN R2  
;GET POINTER TO A P-TABLE  
  
        MOV R2,R0  
        TRAP C$GPHRD  
;IF NOT AVAILABLE, GO GET NEXT  
        BCC INIT14  
  
;GET ADDRESS OF CONTROLLER TABLES  
;CHECK IF SAME UNIBUS ADDRESS  
;BRANCH IF TABLE FOUND  
;MOVE TO NEXT TABLE
```

```

1          :BUILD DRIVE TABLE
2
3 017604 012701 000015  INIT11: MOV #D.SIZE/2,R1      ;GET SIZE OF DRIVE TABLE
4 017610 004737 011346      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          : R2 IS UNIT NUMBER
9 017614 010337 003250      MOV R3,TEMP                ;SAVE CONTROLLER TABLE ADDRESS
10         :
11 017620 062703 000020      ADD #C.DR0,R3              ;IN CASE AN ERROR IS DETECTED
12 017624 012704 000010      MOV #8,R4                  ;BUILD POINTER TO C.DR ENTRY IN CONTROLLER TABLE
13 017630 005713              INIT12: TST (R3)            ;CHECK IF ENTRY CONTAINS POINTER TO DRIVE TABLE
14 017632 0C1411              BEQ INIT13
15 017634 026033 000010      CMP HO.LDR(R0),@ (R3)+    ;CHECK DRIVE NUMBER IN DRIVE TABLE
16 017640 001002              BNE 1$
17 017642 000137 017742      JMP MLDRER                 ;IF SAME, TWO P-TABLES POINT TO SAME DRIVE
18 017646 005304              1$: DEC R4                  ;COUNT DRIVES
19 017650 001367              BNE INIT12                 ;IF EIGHT DRIVE TABLES EXIST,
20 017652 000137 017760      JMP TOOMER                 ; THEN REPORT ERROR
21 017656 010113              INIT13: MOV R1,(R3)        ;LOAD DRIVE TABLE POINTER
22 017660 016021 000010      MOV HO.LDR(R0),(R1)+    ;LOAD DRIVE NUMBER
23 017664 010221              MOV R2,(R1)+              ;LOAD UNIT NUMBER
    
```

```
1          ;GO TO NEXT DRIVE TABLE
2
3 017666 005202          INIT14: INC R2          ;INCREMENT LOGICAL UNIT NUMBER
4 017670 023702 002012      CMP L$UNIT,R2          ;CHECK IF GOT ALL TABLES
5 017674 003331          BGT INIT8          ;IF NOT, GET NEXT TABLE
6
7          ;SAVE CURRENT PARAMETERS TO FREE MEMORY
8
9 017676 013737 002146 002152 INIT15: MOV FFREE,FMEM      ;SAVE START ADDRESS
10 017704 013737 002150 002154      MOV FSIZE,FMEMS      ;SAVE SIZE
11
12 017712          INITXX: SETPRI #PRI00          ; SET RUNNING PRIORITY TO ZERO
13 017712 012700 000000      MOV #PRI00,R0
14 017716 104441          TRAP C$SPRI
15 017720          CLOSE          ;MAKE SURE DATA FILE IS CLOSED
16 017720 104435          TRAP C$CLOS
17 017722          EXIT INIT
18 017722 104432          TRAP C$EXIT
19 017724 000524          .WORD L10034-
```

```

1          ;DIFFERENT VECTORS, BR LEVELS OR BURST RATES FOR ONE CONTROLLER
2 017726 010305 CTABER: MOV R3,R5 ;GET CONTROLLER ADDRESS
3 017730          ERRSF 1,,ERR001
   017730 104454          TRAP CSERSF
   017732 000001          .WORD 1
   017734 000000          .WORD 0
   017736 010430          .WORD ERR001
4 017740          DOCLN
   017740 104444          TRAP CSDCLN
5
6          ;TWO P-TABLES FOR SAME DRIVE
7 017742 013705 003250 MLDRE: MOV TEMP,R5 ;GET CONTROLLER ADDRESS
8 017746          ERRSF 2,,ERR002
   017746 104454          TRAP CSERSF
   017750 000002          .WORD 2
   017752 000000          .WORD 0
   017754 010446          .WORD ERR002
9 017756          DOCLN
   017756 104444          TRAP CSDCLN
10
11         ;MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER
12
13 017760 013705 003250 TOOMER: MOV TEMP,R5 ;GET CONTROLLER ADDRESS
14 017764          ERRSF 3,,ERR003
   017764 104454          TRAP CSERSF
   017766 000003          .WORD 3
   017770 000000          .WORD 0
   017772 010464          .WORD ERR003
15 017774          DOCLN
   017774 104444          TRAP CSDCLN
16
17         ;TWO UDA'S USE THE SAME VECTOR
18
19 017776 010305 SAMVEC: MOV R3,R5 ;GET CONTROLLER ADDRESS
20 020000          ERRSF 8,ERR008
   020000 104454          TRAP CSERSF
   020002 000010          .WORD 8
   020004 010540          .WORD ERR008
   020006 000000          .WORD 0
21 020010          DOCLN
   020010 104444          TRAP CSDCLN
    
```

```

1 020012 104443 DATE: GMANID DATEQ,DATEI,A,-1,1,11.,YES ;GET DATE
020012 000406 TRAP CSGMAN
020014 003276 BR 10000$
020016 000152 .WORD DATEI
020020 003526 .WORD T$CODE
020022 177777 .WORD DATEQ
020024 000001 .WORD -1
020026 000013 .WORD T$LOLIM
020030 000013 .WORD T$HILIM
020032 10000$:

2 020032 012705 003276 MOV #DATEI,R5 ;GET POINTER TO ANSWER
3 020036 121527 000060 CMPB (R5),#0
4 020042 103443 BLO DERR
5 020044 122527 000071 DAY: CMPB (R5)+,#9
6 020050 101040 BHI DERR
7 020052 121527 000055 CMPB (R5),#-
8 020056 001406 BEQ DAS1
9 020060 121527 000060 CMPB (R5),#0
10 020064 103432 BLO DERR
11 020066 122527 000071 CMPB (R5)+,#9
12 020072 101027 BHI DERR
13 020074 122527 000055 DAS1: CMPB (R5)+,#-
14 020100 001024 BNE DERR
15 020102 012704 000014 MOV #12.,R4 ;GET NUMBER OF MONTH
16 020106 012703 003353 MON1: MOV #MONTHS,R3 ;GET POINTER TO MONTH NAMES
17 020112 005000 CLR R0
18 020114 121523 CMPB (R5),(R3)+
19 020116 001401 BEQ MON2
20 020120 005200 INC R0
21 020122 126523 000001 MON2: CMPB 1(R5),(R3,
22 020126 001401 BEQ MON3
23 020130 005200 INC R0
24 020132 126523 000002 MON3: CMPB 2(R5),(R3)+
25 020136 001401 BEQ MON4
26 020140 005200 INC R0
27 020142 005700 MON4: TST R0
28 020144 001407 BEQ MON5
29 020146 005304 DEC R4
30 020150 001360 BNE MON1
31 020152 004137 014650 DERR: PNTF DATEX
020152 010330 JSR R1,LPNTF
020160 000000 .WORD DATEX
020162 000713 .WORD PNT.CT
32 020164 012701 003312 MON5: BR DATE
33 020170 010403 MOV #DATEQ,R1 ;GET POINTER TO DATE FOR FORMATTER
34 020172 020327 000012 MOV R4,R3 ;GET COPY OF MONTH NUMBER
35 020176 103404 CMP R3,#10. ; IF 10 OR GREATER
36 020200 112721 000061 BLO MON6
37 020204 162703 000012 MOV #1,(R1)+ ;PUT A '1' IN OUTPUT
38 020210 062703 000060 MON6: SUB #10.,R3
39 020214 110321 ADD #0,R3 ;CONVERT MONTH NUMBER TO ASCII
40 020216 112721 000055 MOV R3,(R1)+ ;PUT A NUMBER IN OUTPUT
41 020222 062704 003416 MOV #'-',(R1)+ ;PUT A '-' IN OUTPUT
42 020226 012703 003276 ADD #DAYS-1,R4 ;GET POINTER TO DAYS IN MONTH
43 ;INDEXED BY NUMBER OF MONTH
44 020232 005000 MOV #DATEI,R3 ;GET POINTER TO DATE INPUT
45 CLR R0
  
```



```

46 020234 121327 000055      DAY1:  CMPB (R3),#'-'
47 020240 001413              BEQ DAY2
48 020242 111321              MOVB (R3),(R1)+ ;PUT DAY CHARACTER IN OUTPUT
49 020244 006300              ASL R0
50 020246 010002              MOV R0,R2
51 020250 006300              ASL R0
52 020252 006300              ASL R0
53 020254 060200              ADD R2,R0
54 020256 112302              MOVB (R3)+,R2
55 020260 162702 000060      SUB #'0,R2
56 020264 060200              ADD R2,R0
57 020266 000762              BR DAY1
58 020270 120014              DAY2:  CMPB R0,(R4)
59 020272 101327              BHI DERR
60 020274 005700              TST R0 ;SEE IF DATE IS ZERO
61 020276 001725              BEQ DERR ;ERROR IF SO
62 020300 062705 000003      ADD #3,R5
63 020304 121527 000055      CMPB (R5),#'-' ;CHECK FOR '-' BETWEEN DAY
64 020310 001320              BNE DERR ; AND YEAR IN OUTPUT
65 020312 112521              MOVB (R5)+,(R1)+ ;PUT '-' IN OUTPUT
66 020314 010504              MOV R5,R4 ;GET COPY OF INPUT STRING POINTER
67 020316 005000              CLR R0
68 020320 005002              CLR R2
69 020322 121427 000060      YER1:  CMPB (R4),#'0
70 020326 103416              BLO YER2
71 020330 121427 000071      CMPB (R4),#'9
72 020334 101013              BHI YER2
73 020336 006300              ASL R0
74 020340 010003              MOV R0,R3
75 020342 006300              ASL R0
76 020344 006300              ASL R0
77 020346 060300              ADD R3,R0
78 020350 112403              MOVB (R4)+,R3
79 020352 162703 000060      SUB #'0,R3
80 020356 060300              ADD R3,R0
81 020360 005202              INC R2
82 020362 000757              BR YER1
83 020364 105714              YER2:  TSTB (R4)
84 020366 001271              BNE DERR
85 020370 020227 000002      CMP R2,#2
86 020374 001407              BEQ YER3
87 020376 020227 000004      CMP R2,#4
88 020402 001263              BNE DERR
89 020404 020027 003554      CMP R0,#1900.
90 020410 103660              BLO DERR
91 020412 000413              BR YER5
92 020414 012702 003433      YER3:  MOV #YEAR19,R2
93 020420 020027 000106      CMP R0,#70.
94 020424 103002              BHIS YER4
95 020426 012702 003436      MOV #YEAR20,R2
96 020432 105712              YER4:  TSTB (R2)
97 020434 001402              BEQ YER5
98 020436 112221              MOVB (R2)+,(R1)+
99 020440 000774              BR YER4
100 020442 112521              YER5:  MOVB (R5)+,(R1)+
101 020444 001376              BNE YER5
102 020446 000207              RETURN
    
```

103
104 020450
020450
020450 104411

ENDINIT

L10034: TRAP CSINIT

1
2
3
4
5
6
7
8
9

.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.
:--

10 020452
020452

BGNAUTO

LSAUTO::

11

12 020452
020452 104461

ENDAUTO

L10035: TRAP CSAUTO

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

.SBTTL CLEANUP CODING SECTION

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

020454
020454

BGNCLN

L\$CLEAN::

020454
020454

104435

CLOSE

;CLOSE DATA FILE

TRAP CSCLOS

020456
020456

104433

BRESET

;RESET ALL UDAS

TRAP CSRESET

020460
020460
020460

104412

ENDCLN

L10036:

TRAP CSCLEAN

020462

ENDMOD

```

1          .SBTTL TEST 1: DUP PROGRAM DRIVER
2
3 020462      BGNMOD
4
5 020462      BGNTST
6 020462      032737 000003 003210      BIT #SO.FMT,MODE
7 020470      001157      BNE T1FMT
8 020472      104450      MANUAL
9 020474      103406      BCOMPLETE T1GO
10 020476      104454      ERRSF 10,,ERR010
11 020506      000012      EXIT TST
12 020512      000000      T1GO: BIT #SO.STR,MODE
13 020520      010526      BEQ T1CNS
14 020522      000350      CMP L$UNIT,#1
15 020530      000010 003210 T1GO: BEQ T1RST
16 020532      000001      ERRSF 9,,ERR009
17 020542      104454      EXIT TST
18 020544      000011      TRAP C$EXIT
19 020546      000000      .WORD L10037-.
20 020556      010514      JSR R1,LPNTF
21 020576      000012      .WORD FILNAM
22 020602      000000      .WORD PNT.CT
23 020604      004137 014650      GMANID FILNAQ,FNAME,A,-1,1,10.,NO ;GET FILE NAME
24 020606      000406      TRAP C$GMAN
25 020612      000234      BR 10000$
26 020614      000142      .WORD FNAME
27 020620      000356      .WORD T$CODE
28 020622      177777      .WORD FILNAQ
29 020624      000001      .WORD -1
30 020626      000012      .WORD T$LOLIM
31 020628      000012      .WORD T$HILIM
32 020630      012700 003234      OPEN #FNAME
33 020632      104434      MOV #FNAME,R0
34 020634      000511      TRAP C$OPEN
35 020636      013705 002156      BR T1FMT
36 020638      010504      T1CNS: MOV CTABS,R5
37 020640      062704 000020      T1SER1: MOV R5,R4
38 020642      012703 000010      ADD #C.DR0,R4
39 020644      011402      MOV #8,R3
40 020646      001474      T1SER2: MOV (R4),R2 ;GET DRIVE TABLE POINTER
41 020648      001474      BEQ T1SERN
    
```

```
29 020630          PNTF SERNUM,D.UNIT(R2),(R5),(R2)
   020630 011246
   020632 011546
   020634 016246 000002
   020640 004137 014650
   020644 004151
   020646 000006
30 020650          ASSUME C.UADR EQ 0
31 020650          ASSUME D.DRV EQ 0
32 020650 T1SER3: GMANID SERNO,TEMP,A,-1,1,20.,NO ;GET SERIAL NUMBER
   020650 104443
   020652 000406
   020654 003250
   020656 000142
   020660 003610
   020662 177777
   020664 000001
   020666 000024
   020670
33 020670 012701 003250
   020674 005000
34 020674 005000
35 020676 105711
36 020700 001410
37 020702 005200
38 020704 121127 000060
39 020710 103416
40 020712 122127 000071
41 020716 101767
42 020720 000412
43 020722 020027 000024
44 020726 103422
45 020730 012701 003250
46 020734 012700 003326
47 020740 122120
48 020742 001776
49 020744 103413
50 020746
   020746 012746 003326
   020752 012746 010240
   020756 012746 000002
   020762 010600
   020764 104417
   020766 062706 000006
51 020772 000726
52 020774 062702 000004
53 021000 012701 003250
54 021004 112122
55 021006 001376
56 021010 005303
57 021012 001402
58 021014 005724
59 021016 000702
60 021020 062705 000054
61 021024 005715
62 021026 001271
63 021030 013737 002156 002162 T1FMT:
64 021036 013701 002160

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

TRAP C$GMAN
BR 10001$
.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

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

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

65 021042 004737 011410
 66 021046 001402
 67 021050 004737 011522
 68 021054
 021054 104432
 021056 000002
 69 021060
 021060
 021060 104401
 70 021062

6\$:

CALL RUNDM
 BEQ 6\$
 CALL RESPDM
 EXIT TST

 ENDTST

 ENDMOD

: AT ONCE

TRAP C\$EXIT
 .WORD L10037-

 L10037: TRAP C\$ETST

```
1          .SBTTL  HARDWARE PARAMETER CODING SECTION
2
3 021062          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 021062          BGNHRD
15 021062          000027
16 021064
17
18 021064          ;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:
19
20 021064          TABLE                                ;START A TEBLE DEFINITION
21 021064          ITEM HO.UBA                2          : UNIBUS ADDRESS
22 021064          ITEM HO.VEC                2          : UDA VECTOR
23 021064          ITEM HO.BRL                2          : BR LEVEL
24 021064          ITEM HO.BST                2          : BURST RATE
25 021064          ITEM HO.LDR                2          : DRIVE NUMBER
                END
```

.WORD L10040-L\$HARD/2
L\$HARD::

| | | | | | | | | | | |
|----|--------|--------|-----|-----|--------|----------------------------------|-------------------------|-----------------------|----------|--|
| 1 | 021064 | | | | GPRMA | H.UBA,HO.UBA,0,160000,177774,YES | | :BUS ADDRESS | | |
| | 021064 | 000031 | | | | | | .WORD | T\$CODE | |
| | 021066 | 021142 | | | | | | .WORD | H.UBA | |
| | 021070 | 160000 | | | | | | .WORD | T\$LOLIM | |
| | 021072 | 177774 | | | | | | .WORD | T\$HILIM | |
| 2 | 021074 | | | | GPRMA | H.VEC,HO.VEC,0,4,774,YES | | : VECTOR | | |
| | 021074 | 001031 | | | | | | .WORD | T\$CODE | |
| | 021076 | 021170 | | | | | | .WORD | H.VEC | |
| | 021100 | 000004 | | | | | | .WORD | T\$LOLIM | |
| | 021102 | 000774 | | | | | | .WORD | T\$HILIM | |
| 3 | 021104 | | | | GPRMD | H.BRL,HO.BRL,D,-1,4.,7.,YES | | : BR LEVEL | | |
| | 021104 | 002052 | | | | | | .WORD | T\$CODE | |
| | 021106 | 021177 | | | | | | .WORD | H.BRL | |
| | 021110 | 177777 | | | | | | .WORD | -1 | |
| | 021112 | 000004 | | | | | | .WORD | T\$LOLIM | |
| | 021114 | 000007 | | | | | | .WORD | T\$HILIM | |
| 4 | 021116 | | | | GPRMD | H.BST,HO.BST,D,-1,0.,63.,YES | | : BURST RATE | | |
| | 021116 | 003052 | | | | | | .WORD | T\$CODE | |
| | 021120 | 021210 | | | | | | .WORD | H.BST | |
| | 021122 | 177777 | | | | | | .WORD | -1 | |
| | 021124 | 000000 | | | | | | .WORD | T\$LOLIM | |
| | 021126 | 000077 | | | | | | .WORD | T\$HILIM | |
| 5 | 021130 | | | | GPRMD | H.LDR,HO.LDR,D,-1,0.,255.,YES | | : DRIVE SELECT NUMBER | | |
| | 021130 | 004052 | | | | | | .WORD | T\$CODE | |
| | 021132 | 021232 | | | | | | .WORD | H.LDR | |
| | 021134 | 177777 | | | | | | .WORD | -1 | |
| | 021136 | 000000 | | | | | | .WORD | T\$LOLIM | |
| | 021140 | 000377 | | | | | | .WORD | T\$HILIM | |
| 6 | 021142 | | | | ENDHRD | | | | | |
| | 021142 | | | | | | | | .EVEN | |
| | | | | | | | | | L10040: | |
| 7 | | | | | | | | | | |
| 8 | 021142 | 125 | 116 | 111 | H.UBA: | .ASCIZ | \UNIBUS ADDRESS OF UDA\ | | | |
| 9 | 021170 | 126 | 105 | 103 | H.VEC: | .ASCIZ | \VECTOR\ | | | |
| 10 | 021177 | 102 | 122 | 040 | H.BRL: | .ASCIZ | \BR LEVEL\ | | | |
| 11 | 021210 | 125 | 116 | 111 | H.BST: | .ASCIZ | \UNIBUS BURST RATE\ | | | |
| 12 | 021232 | 104 | 122 | 111 | H.LDR: | .ASCIZ | \DRIVE NUMBER\ | | | |
| 13 | | | | | | .EVEN | | | | |

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

.SBTTL SOFTWARE PARAMETER CODING SECTION

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

BGNSFT

021250
021250 000022
021252

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

:FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:

TABLE ;START A TABLE DEFINITION
ITEM SO.BIT 2 ;YES/NO ANSWERS
SO.FM1 = BIT0 ; REFORMAT MODE
SO.FM2 = BIT1 ; (AGAIN)
SO.FMT = SO.FM1+SO.FM2
SO.CNS = BIT2 ; RECONSTRUCT MODE
SO.STR = BIT3 ; RESTORE MODE

021252
021252 000001
000002
000003
000004
000010

END

021252

| | | | | | | | | |
|----|--------|--------|-----|--|--|---------|---------|--|
| 1 | 021252 | | | GPRML S.FMT,SO.BIT,SO.FM1,YES ;REFORMAT? | | | | |
| | 021252 | 000130 | | | | .WORD | T\$CODE | |
| | 021254 | 021467 | | | | .WORD | S.FMT | |
| | 021256 | 000001 | | | | .WORD | SO.FM1 | |
| 2 | 021260 | | | XFERT SWEND | | | | |
| | 021260 | 017024 | | | | .WORD | T\$CODE | |
| 3 | 021262 | | | GPRML S.NRF,SO.BIT,SO.FM2,YES ;AGAIN - REFORMAT? | | | | |
| | 021262 | 000130 | | | | .WORD | T\$CODE | |
| | 021264 | 021316 | | | | .WORD | S.NRF | |
| | 021266 | 000002 | | | | .WORD | SO.FM2 | |
| 4 | 021270 | | | XFERT SWEND | | | | |
| | 021270 | 013024 | | | | .WORD | T\$CODE | |
| 5 | 021272 | | | GPRML S.CNS,SO.BIT,SO.CNS,YES ;RECONSTRUCT | | | | |
| | 021272 | 000130 | | | | .WORD | T\$CODE | |
| | 021274 | 021546 | | | | .WORD | S.CNS | |
| | 021276 | 000004 | | | | .WORD | SO.CNS | |
| 6 | 021300 | | | XFERT SWEND | | | | |
| | 021300 | 007024 | | | | .WORD | T\$CODE | |
| 7 | 021302 | | | GPRML S.RST,SO.BIT,SO.STR,YES ;RESTORE? | | | | |
| | 021302 | 000130 | | | | .WORD | T\$CODE | |
| | 021304 | 021611 | | | | .WORD | S.RST | |
| | 021306 | 000010 | | | | .WORD | SO.STR | |
| 8 | 021310 | | | XFERT SWEND | | | | |
| | 021310 | 003024 | | | | .WORD | T\$CODE | |
| 9 | 021312 | | | DISPLAY S.NOF ;WARNING | | | | |
| | 021312 | 000003 | | | | .WORD | T\$CODE | |
| | 021314 | 021732 | | | | .WORD | S.NOF | |
| 10 | 021316 | | | SWEND: ENDSFT | | | | |
| | 021316 | | | | | .EVEN | | |
| | | | | | | L10041: | | |
| 11 | | | | | | | | |
| 12 | 021316 | 015 | 012 | S.NRF: .BYTE 15,12 | | | | |
| 13 | 021320 | 116 | 117 | 124 .ASCII\NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR\ | | | | |
| 14 | 021422 | 015 | 012 | .BYTE 15,12 | | | | |
| 15 | 021424 | 111 | 116 | 106 .ASCII\INFORMATION ON THE DISKS.\ | | | | |
| 16 | 021455 | 015 | 012 | .BYTE 15,12 | | | | |
| 17 | 021457 | 101 | 107 | 101 .ASCII\AGAIN - \ | | | | |
| 18 | 021467 | 122 | 105 | 106 S.FMT: .ASCII\REFORMAT USING EXISTING BAD SECTOR INFORMATION\ | | | | |
| 19 | 021546 | 122 | 105 | 103 S.CNS: .ASCII\RECONSTRUCT BAD SECTOR INFORMATION\ | | | | |
| 20 | 021611 | 104 | 117 | 040 S.RST: .ASCII\DO YOU HAVE A FILE ON THE SYSTEM LOAD DEVICE\ | | | | |
| 21 | 021665 | 015 | 012 | .BYTE 15,12 | | | | |
| 22 | 021667 | 040 | 103 | 117 .ASCII\ CONTAINING BAD SECTOR INFORMATION\ | | | | |
| 23 | 021732 | 131 | 117 | 125 S.NOF: .ASCII\YOU CANNOT PROCEED WITHOUT SUCH A FILE.\ | | | | |
| 24 | 022002 | 122 | 105 | 123 .ASCII\RESTART PROGRAM AND SELECT TO REFORMAT OR RECONSTRUCT DISK.\ | | | | |
| 25 | 022076 | 000 | | .BYTE 0 | | | | |
| 26 | | | | .EVEN | | | | |
| 27 | | | | | | | | |
| 28 | | | | .DSABL AMA | | | | |
| 29 | 000000 | | | .PSECT END | | | | |

1
2
3 000000
4
5
6
7
8 000120

000120 000142'
000122 000007
000124
9
10 000124

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

LSLAST::
ENDMOD

.EVEN
.WORD TSFREE
.WORD TSSIZE

```
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
```

L10042: .WORD 0
.WORD L10044-. /2-1

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

L10044:

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 29184 WORDS (114 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
ZUDEBO,A:ZUDEBO/C=[20,0]S:C34R.MLB/P:1,ZUDEBO.DOC,ZUDEBO.MAC

| | | | | | | | | | | | | | | | | | | | | |
|---------|--------|--------|--------|--------|--------|--------|--------|-------|--------|-------|--------|--------|--------|--------|--|--|--|--|--|--|
| SPATCH | 132-3# | | | | | | | | | | | | | | | | | | | |
| ADR | 31-10# | | | | | | | | | | | | | | | | | | | |
| ALOCM | 57-16# | 58-14 | 118-4 | 119-20 | 121-4 | | | | | | | | | | | | | | | |
| ASSEMB | 27-8 | 27-8 | | | | | | | | | | | | | | | | | | |
| BAS | 51-15# | 82-5 | 82-5 | 82-5 | 83-5 | 83-5 | | | | | | | | | | | | | | |
| BASL2 | 51-13# | 83-5 | | | | | | | | | | | | | | | | | | |
| BASL3 | 51-14# | | | | | | | | | | | | | | | | | | | |
| BASLN | 51-17# | 82-5 | 83-5 | | | | | | | | | | | | | | | | | |
| BASNO | 51-12# | 82-5 | 83-5 | | | | | | | | | | | | | | | | | |
| BIT0 | 31-10# | 130-19 | | | | | | | | | | | | | | | | | | |
| BIT00 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT01 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT02 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT03 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT04 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT05 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT06 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT07 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT08 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT09 | 31-10 | 31-10# | | | | | | | | | | | | | | | | | | |
| BIT1 | 31-10# | 42-23 | 130-20 | | | | | | | | | | | | | | | | | |
| BIT10 | 31-10# | | | | | | | | | | | | | | | | | | | |
| BIT11 | 31-10# | | | | | | | | | | | | | | | | | | | |
| BIT12 | 31-10# | | | | | | | | | | | | | | | | | | | |
| BIT13 | 31-10# | | | | | | | | | | | | | | | | | | | |
| BIT14 | 31-10# | | | | | | | | | | | | | | | | | | | |
| BIT15 | 31-10# | 42-15 | 43-12 | 59-27 | 61-20 | 69-26 | 74-15 | 89-29 | | | | | | | | | | | | |
| BIT2 | 31-10# | 42-24 | 130-22 | | | | | | | | | | | | | | | | | |
| BIT3 | 31-10# | 42-25 | 130-23 | | | | | | | | | | | | | | | | | |
| BIT4 | 31-10# | 42-27 | | | | | | | | | | | | | | | | | | |
| BIT5 | 31-10# | 42-30 | | | | | | | | | | | | | | | | | | |
| BIT6 | 31-10# | 42-31 | | | | | | | | | | | | | | | | | | |
| BIT7 | 31-10# | 42-33 | | | | | | | | | | | | | | | | | | |
| BIT8 | 31-10# | | | | | | | | | | | | | | | | | | | |
| BIT9 | 31-10# | | | | | | | | | | | | | | | | | | | |
| BLDC0 | 97-22 | 97-24# | | | | | | | | | | | | | | | | | | |
| BLDC1 | 97-26# | 97-28 | | | | | | | | | | | | | | | | | | |
| BLDCMD | 60-49 | 63-14 | 63-44 | 95-3 | 97-15# | | | | | | | | | | | | | | | |
| BOE | 31-10# | | | | | | | | | | | | | | | | | | | |
| CSAU | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSAUTO | 27-8# | 125-12 | | | | | | | | | | | | | | | | | | |
| CSBRK | 27-8# | 60-8 | 100-21 | 107-12 | 110-27 | | | | | | | | | | | | | | | |
| CSBSEG | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSBSUB | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSCEFG | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSCLCK | 27-8# | 116-43 | 116-45 | | | | | | | | | | | | | | | | | |
| CSCLEA | 27-8# | 126-13 | | | | | | | | | | | | | | | | | | |
| CSCLOS | 27-8# | 72-18 | 122-13 | 126-10 | | | | | | | | | | | | | | | | |
| CSCLP1 | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSVEC | 27-8# | 107-30 | | | | | | | | | | | | | | | | | | |
| CSDECLN | 27-8# | 56-8 | 116-18 | 123-4 | 123-9 | 123-15 | 123-21 | | | | | | | | | | | | | |
| CSDODU | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSDRPT | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSDU | 27-8# | | | | | | | | | | | | | | | | | | | |
| CSEDIT | 27-8# | 27-34 | | | | | | | | | | | | | | | | | | |
| CSERDF | 27-8# | 60-24 | 61-36 | 62-21 | 63-8 | 67-22 | 69-30 | 96-3 | 100-29 | 101-5 | 105-35 | 106-11 | 107-33 | 109-44 | | | | | | |

| | | | | | | | | | | | | | | |
|---------|---------|--------|--------|--------|--------|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| EF.CON | 31-10# | 116-14 | | | | | | | | | | | | |
| EF.LOG | 38-29# | | | | | | | | | | | | | |
| EF.NEW | 31-10# | | | | | | | | | | | | | |
| EF.PWR | 31-10# | 116-16 | | | | | | | | | | | | |
| EF.RES | 31-10# | 116-12 | | | | | | | | | | | | |
| EF.SEX | 38-30# | | | | | | | | | | | | | |
| EF.STA | 31-10# | 116-10 | | | | | | | | | | | | |
| ERRO01 | 54-14# | 123-3 | | | | | | | | | | | | |
| ERRO02 | 54-18# | 123-8 | | | | | | | | | | | | |
| ERRO03 | 54-22# | 123-14 | | | | | | | | | | | | |
| ERRO04 | 54-26# | 56-7 | | | | | | | | | | | | |
| ERRO08 | 54-38# | 123-20 | | | | | | | | | | | | |
| ERRO09 | 54-30# | 127-16 | | | | | | | | | | | | |
| ERRO10 | 54-34# | 127-10 | | | | | | | | | | | | |
| ERRO20 | 54-42# | 107-33 | | | | | | | | | | | | |
| ERRO21 | 54-46# | 111-5 | | | | | | | | | | | | |
| ERRO22 | 54-55# | 110-36 | | | | | | | | | | | | |
| ERRO23 | 54-61# | 106-11 | | | | | | | | | | | | |
| ERRO24 | 54-75# | 105-35 | | | | | | | | | | | | |
| ERRO25 | 54-79# | 109-44 | | | | | | | | | | | | |
| ERRO30 | 54-83# | 60-24 | | | | | | | | | | | | |
| ERRO31 | 54-87# | 61-36 | | | | | | | | | | | | |
| ERRO32 | 54-91# | 63-8 | | | | | | | | | | | | |
| ERRO33 | 54-96# | 62-21 | | | | | | | | | | | | |
| ERRO34 | 54-100# | 96-3 | | | | | | | | | | | | |
| ERRO36 | 54-104# | 100-29 | | | | | | | | | | | | |
| ERRO37 | 54-108# | 101-5 | | | | | | | | | | | | |
| ERR100 | 54-112# | 67-22 | | | | | | | | | | | | |
| ERR101 | 54-116# | 69-30 | | | | | | | | | | | | |
| ERR23A | 54-64# | 54-71 | | | | | | | | | | | | |
| ERR23B | 54-65 | 54-69# | | | | | | | | | | | | |
| ERR23C | 54-68 | 54-72# | | | | | | | | | | | | |
| ERRC | 76-36 | 76-43 | 80-9# | | | | | | | | | | | |
| ERRCHR | 47-38# | 91-5* | 91-12 | 91-14 | 91-16 | 91-18 | | | | | | | | |
| ERRD | 76-45 | 80-21# | | | | | | | | | | | | |
| ERRME1 | 51-8# | 76-41 | 78-23 | 81-4 | | | | | | | | | | |
| ERRNL | 51-4# | 91-10 | | | | | | | | | | | | |
| ERRONE | 51-3# | 91-7 | | | | | | | | | | | | |
| ERRRSZ | 78-18 | 79-12# | | | | | | | | | | | | |
| ERRRTB | 78-21 | 79-3# | 79-12 | | | | | | | | | | | |
| ERRTRM | 64-7 | 71-14# | | | | | | | | | | | | |
| EVL | 31-10# | | | | | | | | | | | | | |
| FSAU | 27-8# | | | | | | | | | | | | | |
| FSAUTO | 27-8# | 125-10 | 125-12 | | | | | | | | | | | |
| F\$BGN | 27-8# | 27-26 | 30-16 | 31-3 | 54-14 | 54-18 | 54-22 | 54-26 | 54-30 | 54-34 | 54-38 | 54-42 | 54-46 | 54-55 |
| | 54-61 | 54-75 | 54-79 | 54-83 | 54-87 | 54-91 | 54-96 | 54-100 | 54-104 | 54-108 | 54-112 | 54-116 | 102-10 | 103-18 |
| | 113-5 | 114-38 | 115-3 | 115-10 | 116-8 | 122-14 | 125-10 | 126-8 | 126-15 | 127-3 | 127-5 | 127-11 | 127-17 | 127-68 |
| | 127-69 | 127-70 | 128-3 | 128-14 | 130-12 | 131-9 | 131-9 | 132-10 | 133-1 | 133-3 | 133-3 | 133-11 | 133-13 | |
| F\$CLEA | 27-8# | 126-8 | 126-13 | | | | | | | | | | | |
| F\$DU | 27-8# | | | | | | | | | | | | | |
| F\$END | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 | 27-8 |
| | 27-8 | 27-8 | 27-8# | 27-26 | 30-16 | 31-3 | 54-16 | 54-20 | 54-24 | 54-28 | 54-32 | 54-36 | 54-40 | 54-44 |
| | 54-53 | 54-59 | 54-73 | 54-77 | 54-81 | 54-85 | 54-89 | 54-94 | 54-98 | 54-102 | 54-106 | 54-110 | 54-114 | 54-118 |
| | 102-14 | 103-21 | 113-9 | 114-38 | 115-3 | 122-14 | 124-104 | 125-12 | 126-13 | 126-15 | 127-3 | 127-5 | 127-5 | 127-5 |
| | 127-11 | 127-17 | 127-68 | 127-69 | 127-69 | 127-70 | 128-3 | 129-6 | 131-10 | 132-10 | 133-1 | 133-3 | 133-11 | 133-13 |
| F\$HARD | 27-8# | 128-14 | 129-6 | 131-2 | 131-4 | 131-6 | 131-8 | | | | | | | |

| | | | | | |
|--------|---------|----------|---------|---------|--------|
| LSHPTP | 27-34# | | | | |
| LSHw | 27-34 | 29-10 | 29-10# | | |
| LSICP | 27-34# | | | | |
| LSINIT | 27-34 | 116-8# | | | |
| LSLADP | 27-34# | | | | |
| LSLAST | 27-34 | 132-8# | 133-13 | | |
| LSLOAD | 27-34# | | | | |
| LSLUN | 27-34# | 59-24* | 60-12* | 74-14* | |
| LSMREV | 27-34# | | | | |
| LSNAME | 27-34# | | | | |
| LSPRIO | 27-34# | | | | |
| LSPROT | 27-34 | 115-10# | | | |
| LSPRT | 27-34# | | | | |
| LSREPP | 27-34# | | | | |
| LSREV | 27-34# | | | | |
| LSSOFT | 27-34 | 130-12 | 130-12# | | |
| LSSPC | 27-34# | | | | |
| LSSPCP | 27-34# | | | | |
| LSSPTP | 27-34# | | | | |
| LSSTA | 27-34# | | | | |
| LSSW | 27-34 | 30-10 | 30-10# | | |
| LSTEST | 27-34# | | | | |
| LSTIML | 27-34# | | | | |
| LSUNIT | 27-34# | 75-13 | 119-16 | 122-4 | 127-14 |
| L10000 | 29-10 | 29-17# | | | |
| L10001 | 30-10 | 30-14# | | | |
| L10002 | 54-16# | | | | |
| L10003 | 54-20# | | | | |
| L10004 | 54-24# | | | | |
| L10005 | 54-28# | | | | |
| L10006 | 54-32# | | | | |
| L10007 | 54-36# | | | | |
| L10010 | 54-40# | | | | |
| L10011 | 54-44# | | | | |
| L10012 | 54-53# | | | | |
| L10013 | 54-59# | | | | |
| L10014 | 54-73# | | | | |
| L10015 | 54-77# | | | | |
| L10016 | 54-81# | | | | |
| L10017 | 54-85# | | | | |
| L10020 | 54-89# | | | | |
| L10021 | 54-94# | | | | |
| L10022 | 54-98# | | | | |
| L10023 | 54-102# | | | | |
| L10024 | 54-106# | | | | |
| L10025 | 54-110# | | | | |
| L10026 | 54-114# | | | | |
| L10027 | 54-118# | | | | |
| L10030 | 102-14# | | | | |
| L10031 | 103-21# | | | | |
| L10032 | 113-9# | | | | |
| L10034 | 122-14 | 124-104# | | | |
| L10035 | 125-12# | | | | |
| L10036 | 126-13# | | | | |
| L10037 | 127-11 | 127-17 | 127-68 | 127-69# | |
| L10040 | 128-14 | 129-6# | | | |

| | | | | | | |
|--------|---------|---------|---------|---------|--------|--------|
| NOCLOC | 51-10# | 116-48 | | | | |
| NULL | 47-39# | | | | | |
| NXMAD | 47-23# | 102-12* | 107-26* | 107-31 | | |
| NXMI | 102-10# | 107-27 | | | | |
| NXTTAB | 117-11 | 118-22 | 119-10 | 119-15# | | |
| OSAPTS | 27-8# | 27-34 | | | | |
| OSAU | 27-8# | 27-34 | | | | |
| OSBGNR | 27-8# | 27-34 | | | | |
| OSBGNS | 27-8# | 27-32# | 27-34 | | | |
| OSDU | 27-8# | 27-34 | | | | |
| OSERRT | 27-8# | 27-34 | | | | |
| OSGNSW | 27-8# | 27-32# | 27-34 | | | |
| OSPOIN | 27-8# | 27-32 | 27-32# | 27-32# | 27-32# | 27-34 |
| OSSETU | 27-8# | 27-32# | 27-34 | 132-8 | | |
| OP.ABO | 37-3# | | | | | |
| OP.ACC | 37-4# | | | | | |
| OP.AVA | 37-22# | | | | | |
| OP.AVL | 37-5# | | | | | |
| OP.CCD | 37-6# | | | | | |
| OP.CMP | 37-7# | | | | | |
| OP.DUP | 37-23# | | | | | |
| OP.ELP | 37-29# | | | | | |
| OP.END | 37-20# | 62-5 | 62-8 | 63-58 | | |
| OP.ERS | 37-8# | | | | | |
| OP.ESP | 37-28# | 95-2 | | | | |
| OP.FLU | 37-9# | | | | | |
| OP.GCS | 37-10# | | | | | |
| OP.GDS | 37-27# | 60-48 | 63-58 | | | |
| OP.GUS | 37-11# | | | | | |
| OP.MRD | 37-18# | | | | | |
| OP.MWR | 37-19# | 97-21 | | | | |
| OP.ONL | 37-12# | | | | | |
| OP.RD | 37-13# | | | | | |
| OP.RLC | 37-25# | | | | | |
| OP.RPL | 37-14# | | | | | |
| OP.RSD | 37-31# | 62-8 | 63-43 | | | |
| OP.SCC | 37-15# | | | | | |
| OP.SEX | 37-21# | | | | | |
| OP.SHC | 37-24# | | | | | |
| OP.SSD | 37-30# | 62-5 | 63-13 | | | |
| OP.SUC | 37-16# | | | | | |
| OP.WR | 37-17# | | | | | |
| OSTRE | 76-35 | 76-42 | 76-47# | | | |
| OSTRNG | 76-34# | 76-46 | 84-6 | 85-6 | 86-6 | 92-17 |
| P.BCNT | 39-21# | 40-9# | 63-11 | 63-33* | 95-5* | 99-19* |
| P.BUFF | 39-22# | | | | | |
| P.CMST | 40-14# | | | | | |
| P.CNCL | 40-48# | | | | | |
| P.CNTF | 39-40# | 40-46# | | | | |
| P.CNTI | 40-49# | | | | | |
| P.CPSP | 39-34# | | | | | |
| P.CRF | 39-17# | 40-4# | 62-19 | 98-17* | | |
| P.CTMO | 40-47# | | | | | |
| P.CYL | 40-26# | | | | | |
| P.DEXT | 40-52# | | | | | |
| P.DFLG | 40-53# | 63-60 | | | | |

| | | | | | | | | | | | | | | |
|--------|---------|---------|---------|---------|---------|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| 109-44 | 109-44 | 109-44 | 109-44 | 109-44 | 109-44 | 109-44 | 109-44 | 109-44 | 109-44 | 109-44 | 110-27 | 110-27 | 110-27 | 110-36 |
| 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 110-36 | 111-5 | 111-5 | 111-5 |
| 111-5 | 111-5 | 111-5 | 111-5 | 111-5 | 111-5 | 111-5 | 111-5 | 111-5 | 111-5 | 113-9 | 113-9 | 113-9 | 116-10 | 116-10 |
| 116-10 | 116-10 | 116-10 | 116-10 | 116-11 | 116-11 | 116-11 | 116-12 | 116-12 | 116-12 | 116-12 | 116-12 | 116-12 | 116-12 | 116-13 |
| 116-13 | 116-13 | 116-14 | 116-14 | 116-14 | 116-14 | 116-14 | 116-14 | 116-15 | 116-15 | 116-15 | 116-15 | 116-16 | 116-16 | 116-16 |
| 116-16 | 116-16 | 116-16 | 116-17 | 116-17 | 116-17 | 116-18 | 116-18 | 116-18 | 116-18 | 116-33 | 116-33 | 116-33 | 116-34 | 116-34 |
| 116-34 | 116-34 | 116-34 | 116-34 | 116-43 | 116-43 | 116-43 | 116-43 | 116-43 | 116-43 | 116-43 | 116-44 | 116-44 | 116-44 | 116-45 |
| 116-45 | 116-45 | 116-45 | 116-45 | 116-45 | 116-46 | 116-46 | 116-46 | 116-46 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 |
| 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 117-10 | 117-10 |
| 117-10 | 117-10 | 117-10 | 117-10 | 117-11 | 117-11 | 117-11 | 120-4 | 120-4 | 120-4 | 120-4 | 120-4 | 120-4 | 120-4 | 120-5 |
| 120-5 | 120-5 | 122-12 | 122-12 | 122-12 | 122-12 | 122-12 | 122-12 | 122-12 | 122-12 | 122-13 | 122-13 | 122-13 | 122-14 | 122-14 |
| 122-14 | 122-14 | 122-14 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 | 123-3 |
| 123-3 | 123-4 | 123-4 | 123-4 | 123-8 | 123-8 | 123-8 | 123-8 | 123-8 | 123-8 | 123-8 | 123-8 | 123-8 | 123-8 | 123-8 |
| 123-8 | 123-8 | 123-9 | 123-9 | 123-9 | 123-14 | 123-14 | 123-14 | 123-14 | 123-14 | 123-14 | 123-14 | 123-14 | 123-14 | 123-14 |
| 123-14 | 123-14 | 123-14 | 123-15 | 123-15 | 123-15 | 123-20 | 123-20 | 123-20 | 123-20 | 123-20 | 123-20 | 123-20 | 123-20 | 123-20 |
| 123-20 | 123-20 | 123-20 | 123-20 | 123-21 | 123-21 | 123-21 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 |
| 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 | 124-1 |
| 124-1 | 124-1 | 124-1 | 124-104 | 124-104 | 124-104 | 125-12 | 125-12 | 125-12 | 126-10 | 126-10 | 126-10 | 126-10 | 126-11 | 126-11 |
| 126-11 | 126-13 | 126-13 | 126-13 | 127-8 | 127-8 | 127-8 | 127-9 | 127-9 | 127-9 | 127-9 | 127-10 | 127-10 | 127-10 | 127-10 |
| 127-10 | 127-10 | 127-10 | 127-10 | 127-10 | 127-10 | 127-10 | 127-10 | 127-10 | 127-11 | 127-11 | 127-11 | 127-11 | 127-11 | 127-11 |
| 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-16 | 127-17 | 127-17 |
| 127-17 | 127-17 | 127-17 | 127-17 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 |
| 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 |
| 127-21 | 127-21 | 127-21 | 127-21 | 127-21 | 127-21 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 |
| 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 | 127-32 |
| 127-32 | 127-32 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 |
| 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-50 | 127-68 | 127-68 | 127-68 | 127-68 | 127-68 | 127-68 | 127-68 | 127-69 | 127-69 |
| 127-69 | 128-14 | 128-14 | 128-14 | 129-1 | 129-1 | 129-1 | 129-1 | 129-1 | 129-1 | 129-1 | 129-1 | 129-1 | 129-1 | 129-1 |
| 129-1 | 129-1 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 | 129-2 |
| 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 | 129-3 |
| 129-3 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 |
| 129-4 | 129-4 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 |
| 129-5 | 129-5 | 129-5 | 129-6 | 129-6 | 129-6 | 130-12 | 130-12 | 130-12 | 131-1 | 131-1 | 131-1 | 131-1 | 131-1 | 131-1 |
| 131-1 | 131-1 | 131-1 | 131-1 | 131-2 | 131-2 | 131-2 | 131-3 | 131-3 | 131-3 | 131-3 | 131-3 | 131-3 | 131-3 | 131-3 |
| 131-3 | 131-3 | 131-4 | 131-4 | 131-4 | 131-5 | 131-5 | 131-5 | 131-5 | 131-5 | 131-5 | 131-5 | 131-5 | 131-5 | 131-5 |
| 131-6 | 131-6 | 131-6 | 131-7 | 131-7 | 131-7 | 131-7 | 131-7 | 131-7 | 131-7 | 131-7 | 131-7 | 131-7 | 131-8 | 131-8 |
| 131-8 | 131-9 | 131-9 | 131-9 | 131-9 | 131-9 | 131-9 | 131-10 | 131-10 | 131-10 | 131-10 | 132-8 | 132-8 | 132-8 | 132-8 |
| 132-8 | 132-8 | 132-8 | 132-8 | 132-8 | 133-3 | 133-3 | 133-3 | 133-3 | 133-3 | 133-3 | 133-3 | 133-3 | 133-3 | 133-3 |
| SVCSUB | 27-8# | 27-12# | 54-10# | 55-3# | | | | | | | | | | |
| SVCTAG | 27-8# | 27-14# | 29-17 | 30-14 | 54-12# | 54-16 | 54-16 | 54-16 | 54-20 | 54-20 | 54-20 | 54-24 | 54-24 | 54-24 |
| | 54-28 | 54-28 | 54-28 | 54-32 | 54-32 | 54-32 | 54-36 | 54-36 | 54-36 | 54-40 | 54-40 | 54-40 | 54-44 | 54-44 |
| | 54-44 | 54-53 | 54-53 | 54-53 | 54-59 | 54-59 | 54-59 | 54-73 | 54-73 | 54-73 | 54-77 | 54-77 | 54-77 | 54-81 |
| | 54-81 | 54-81 | 54-85 | 54-85 | 54-85 | 54-89 | 54-89 | 54-89 | 54-94 | 54-94 | 54-94 | 54-98 | 54-98 | 54-98 |
| | 54-102 | 54-102 | 54-102 | 54-106 | 54-106 | 54-106 | 54-110 | 54-110 | 54-110 | 54-114 | 54-114 | 54-114 | 54-118 | 54-118 |
| | 54-118 | 55-5# | 102-14 | 103-21 | 113-9 | 124-1 | 124-104 | 125-12 | 126-13 | 127-20 | 127-32 | 127-69 | 129-6 | 131-10 |
| | 133-3 | 133-11 | | | | | | | | | | | | |
| SVCTST | 27-8# | 27-11# | 54-9# | 55-2# | 127-5 | | | | | | | | | |
| SWEND | 131-2 | 131-4 | 131-6 | 131-8 | 131-10# | | | | | | | | | |
| TSSAUT | 125-10# | 125-12 | | | | | | | | | | | | |
| TSSCLE | 126-8# | 126-13 | | | | | | | | | | | | |
| TSSDAT | 133-3 | 133-3# | 133-11 | | | | | | | | | | | |
| TSSHAR | 128-14 | 128-14# | 129-6 | | | | | | | | | | | |
| TSSHW | 29-10 | 29-10# | 29-17 | | | | | | | | | | | |
| TSSINI | 116-8# | 122-14 | 124-104 | | | | | | | | | | | |
| TSSMSG | 54-14# | 54-16 | 54-18# | 54-20 | 54-22# | 54-24 | 54-26# | 54-28 | 54-30# | 54-32 | 54-34# | 54-36 | 54-38# | 54-40 |
| | 54-42# | 54-44 | 54-46# | 54-53 | 54-55# | 54-59 | 54-61# | 54-73 | 54-75# | 54-77 | 54-79# | 54-81 | 54-83# | 54-85 |

| | 54-87# | 54-89 | 54-91# | 54-94 | 54-96# | 54-98 | 54-100# | 54-102 | 54-104# | 54-106 | 54-108# | 54-110 | 54-112# | 54-114 |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| TSSPC | 54-87# | 54-89 | | | | | | | | | | | | |
| TSSPRO | 54-116# | 54-118 | | | | | | | | | | | | |
| TSSPTA | 133-1# | 133-13 | | | | | | | | | | | | |
| TSSSOF | 115-10# | | | | | | | | | | | | | |
| TSSSRV | 133-1# | 133-3 | 133-3# | | | | | | | | | | | |
| TSSSW | 130-12 | 130-12# | 131-10 | | | | | | | | | | | |
| TSSTES | 102-10# | 102-14 | 103-18# | 103-21 | 113-5# | 113-9 | | | | | | | | |
| TSARGC | 30-10 | 30-10# | 30-14 | | | | | | | | | | | |
| | 127-5# | 127-11 | 127-17 | 127-68 | 127-69 | | | | | | | | | |
| | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34# | 27-34# | 27-34# |
| | 27-34# | 27-34# | 27-34# | 54-58 | 54-58 | 54-58# | 91-12 | 91-12 | 91-12 | 91-12# | 91-12# | 91-14 | 91-14 | 91-14 |
| | 91-14# | 91-14# | 91-16 | 91-16 | 91-16 | 91-16# | 91-16# | 91-18 | 91-18 | 91-18 | 91-18# | 91-18# | 127-50 | 127-50 |
| | 127-50 | 127-50# | 127-50# | | | | | | | | | | | |
| TSCODE | 124-1 | 124-1 | 124-1 | 124-1# | 124-1# | 124-1# | 127-20 | 127-20 | 127-20 | 127-20# | 127-20# | 127-20# | 127-32 | 127-32 |
| | 127-32 | 127-32# | 127-32# | 127-32# | 129-1 | 129-1 | 129-1 | 129-1# | 129-1# | 129-1# | 129-1# | 129-2 | 129-2 | 129-2# |
| | 129-2# | 129-2# | 129-3 | 129-3 | 129-3 | 129-3# | 129-3# | 129-3# | 129-4 | 129-4 | 129-4 | 129-4# | 129-4# | 129-4# |
| | 129-5 | 129-5 | 129-5 | 129-5# | 129-5# | 129-5# | 131-1 | 131-1 | 131-1 | 131-1# | 131-1# | 131-1# | 131-2 | 131-2 |
| | 131-2 | 131-2 | 131-2 | 131-2 | 131-2# | 131-2# | 131-2# | 131-2# | 131-3 | 131-3 | 131-3 | 131-3# | 131-3# | 131-3# |
| | 131-4 | 131-4 | 131-4 | 131-4 | 131-4 | 131-4 | 131-4# | 131-4# | 131-4# | 131-4# | 131-4# | 131-5 | 131-5 | 131-5# |
| | 131-5# | 131-5# | 131-6 | 131-6 | 131-6 | 131-6 | 131-6 | 131-6 | 131-6# | 131-6# | 131-6# | 131-6# | 131-7 | 131-7 |
| | 131-7 | 131-7# | 131-7# | 131-7# | 131-8 | 131-8 | 131-8 | 131-8 | 131-8 | 131-8 | 131-8# | 131-8# | 131-8# | 131-8# |
| | 131-9 | 131-9# | | | | | | | | | | | | |
| TSERRN | 27-8# | 56-7 | 56-7# | 60-24 | 60-24# | 61-36 | 61-36# | 62-21 | 62-21# | 63-8 | 63-8# | 67-22 | 67-22# | 69-30 |
| | 69-30# | 96-3 | 96-3# | 100-29 | 100-29# | 101-5 | 101-5# | 105-35 | 105-35# | 106-11 | 106-11# | 107-33 | 107-33# | 109-44 |
| | 109-44# | 110-36 | 110-36# | 111-5 | 111-5# | 123-3 | 123-3# | 123-8 | 123-8# | 123-14 | 123-14# | 123-20 | 123-20# | 127-10 |
| | 127-10# | 127-16 | 127-16# | | | | | | | | | | | |
| TSEXCP | 124-1 | 124-1# | 127-20 | 127-20# | 127-32 | 127-32# | 129-1 | 129-1# | 129-2 | 129-2# | 129-3 | 129-3# | 129-4 | 129-4# |
| | 129-5 | 129-5# | | | | | | | | | | | | |
| TSFLAG | 122-14 | 122-14 | 122-14# | 122-14# | 127-11 | 127-11 | 127-11# | 127-11# | 127-17 | 127-17 | 127-17# | 127-17# | 127-68 | 127-68 |
| | 127-68# | 127-68# | | | | | | | | | | | | |
| TSFREE | 132-8 | 133-13# | | | | | | | | | | | | |
| TSGMAN | 27-8# | 124-1 | 124-1# | 124-1# | 127-20 | 127-20# | 127-20# | 127-32 | 127-32# | 127-32# | | | | |
| TSHILI | 124-1 | 124-1# | 127-20 | 127-20# | 127-32 | 127-32# | 129-1 | 129-1# | 129-2 | 129-2# | 129-3 | 129-3# | 129-4 | 129-4# |
| | 129-5 | 129-5# | | | | | | | | | | | | |
| TSLAST | 27-8# | 132-8# | 133-1 | | | | | | | | | | | |
| TSLOLI | 124-1 | 124-1# | 127-20 | 127-20# | 127-32 | 127-32# | 129-1 | 129-1# | 129-2 | 129-2# | 129-3 | 129-3# | 129-4 | 129-4# |
| | 129-5 | 129-5# | | | | | | | | | | | | |
| TSLSYM | 27-8 | 27-8# | 29-17 | 30-14 | 54-16 | 54-20 | 54-24 | 54-28 | 54-32 | 54-36 | 54-40 | 54-44 | 54-53 | 54-59 |
| | 54-73 | 54-77 | 54-81 | 54-85 | 54-89 | 54-94 | 54-98 | 54-102 | 54-106 | 54-110 | 54-114 | 54-118 | 102-14 | 103-21 |
| | 113-9 | 124-104 | 125-12 | 126-13 | 127-69 | 129-6 | 131-10 | | | | | | | |
| TSLTNO | 132-8# | | | | | | | | | | | | | |
| TSNEST | 27-8# | 27-26 | 27-26 | 27-26# | 29-10 | 29-10 | 29-10# | 29-17 | 29-17 | 29-17 | 29-17# | 30-10 | 30-10 | 30-10# |
| | 30-14 | 30-14 | 30-14 | 30-14# | 30-16 | 30-16 | 30-16 | 30-16# | 31-3 | 31-3 | 31-3# | 54-14 | 54-14 | 54-14# |
| | 54-16 | 54-16 | 54-16 | 54-16# | 54-18 | 54-18 | 54-18# | 54-20 | 54-20 | 54-20 | 54-20# | 54-22 | 54-22 | 54-22# |
| | 54-24 | 54-24 | 54-24 | 54-24# | 54-26 | 54-26 | 54-26# | 54-28 | 54-28 | 54-28 | 54-28# | 54-30 | 54-30 | 54-30# |
| | 54-32 | 54-32 | 54-32 | 54-32# | 54-34 | 54-34 | 54-34# | 54-36 | 54-36 | 54-36 | 54-36# | 54-38 | 54-38 | 54-38# |
| | 54-40 | 54-40 | 54-40 | 54-40# | 54-42 | 54-42 | 54-42# | 54-44 | 54-44 | 54-44 | 54-44# | 54-46 | 54-46 | 54-46# |
| | 54-53 | 54-53 | 54-53 | 54-53# | 54-55 | 54-55 | 54-55# | 54-59 | 54-59 | 54-59 | 54-59# | 54-61 | 54-61 | 54-61# |
| | 54-73 | 54-73 | 54-73 | 54-73# | 54-75 | 54-75 | 54-75# | 54-77 | 54-77 | 54-77 | 54-77# | 54-79 | 54-79 | 54-79# |
| | 54-81 | 54-81 | 54-81 | 54-81# | 54-83 | 54-83 | 54-83# | 54-85 | 54-85 | 54-85 | 54-85# | 54-87 | 54-87 | 54-87# |
| | 54-89 | 54-89 | 54-89 | 54-89# | 54-91 | 54-91 | 54-91# | 54-94 | 54-94 | 54-94 | 54-94# | 54-96 | 54-96 | 54-96# |
| | 54-98 | 54-98 | 54-98 | 54-98# | 54-100 | 54-100 | 54-100# | 54-102 | 54-102 | 54-102 | 54-102# | 54-104 | 54-104 | 54-104# |
| | 54-106 | 54-106 | 54-106 | 54-106# | 54-108 | 54-108 | 54-108# | 54-110 | 54-110 | 54-110 | 54-110# | 54-112 | 54-112 | 54-112# |
| | 54-114 | 54-114 | 54-114 | 54-114# | 54-116 | 54-116 | 54-116# | 54-118 | 54-118 | 54-118 | 54-118# | 102-10 | 102-10 | 102-10# |
| | 102-14 | 102-14 | 102-14 | 102-14# | 103-18 | 103-18 | 103-18# | 103-21 | 103-21 | 103-21 | 103-21# | 113-5 | 113-5 | 113-5# |
| | 113-9 | 113-9 | 113-9 | 113-9# | 114-38 | 114-38 | 114-38 | 114-38# | 115-3 | 115-3 | 115-3# | 115-10 | 115-10 | 115-10# |

| | | | | | | |
|---------|---------|---------|----------|---------|--------|--------|
| UF.CMR | 39-3# | | | | | |
| UF.CMW | 39-4# | | | | | |
| UF.INA | 39-6# | | | | | |
| UF.RPL | 39-5# | | | | | |
| UF.SCH | 39-7# | | | | | |
| UF.SCL | 39-8# | | | | | |
| UF.WBN | 39-9# | | | | | |
| UF.WPH | 39-10# | | | | | |
| UF.WPS | 39-11# | | | | | |
| UFREEZ | 47-22# | 59-35* | 61-3 | 61-13* | 69-19 | 69-21* |
| URNING | 47-20# | 59-16* | 59-31* | 59-40 | 61-32* | |
| URUN | 47-19# | 59-15* | 59-20 | 60-7 | | |
| WAITMS | 95-9 | 100-11# | | | | |
| XSALWA | 27-8# | | | | | |
| XSFALS | 27-8# | | | | | |
| XSOFFS | 27-8# | 131-2 | 131-4 | 131-6 | 131-8 | |
| X\$TRUE | 27-8# | 131-2 | 131-4 | 131-6 | 131-8 | |
| X1 | 52-5# | 54-15 | | | | |
| X10 | 52-13# | 54-35 | | | | |
| X100 | 52-37# | 54-113 | | | | |
| X101 | 52-38# | 54-117 | | | | |
| X1A | 52-1# | 54-15 | | | | |
| X2 | 52-6# | 54-19 | | | | |
| X20 | 52-14# | 54-43 | | | | |
| X21 | 52-18# | 54-52 | | | | |
| X22 | 52-20# | 54-57 | | | | |
| X23A | 52-22# | 54-62 | | | | |
| X23B | 52-26# | 54-66 | | | | |
| X24 | 52-27# | 54-76 | | | | |
| X25 | 52-29# | 54-80 | | | | |
| X2A | 52-2# | 54-19 | | | | |
| X3 | 52-7# | 54-23 | | | | |
| X30 | 52-31# | 54-84 | | | | |
| X31 | 52-32# | 54-88 | | | | |
| X32 | 52-33# | 54-92 | | | | |
| X36 | 52-34# | 54-105 | | | | |
| X37 | 52-36# | 54-109 | | | | |
| X3A | 52-3# | 54-23 | | | | |
| X4 | 52-8# | 54-27 | | | | |
| X8 | 52-10# | 54-39 | | | | |
| X8A | 52-4# | 54-39 | | | | |
| X9 | 52-11# | 54-31 | | | | |
| XFRU | 53-8# | 54-58 | 54-72 | 85-5 | | |
| XMSG1 | 53-1# | 54-133 | | | | |
| XMSG2 | 53-2# | 54-137 | | | | |
| XPKT1 | 53-3# | 54-120 | | | | |
| XPKT2 | 53-6# | 54-126 | | | | |
| XSA | 53-7# | 86-5 | | | | |
| YEAR19 | 48-31# | 124-92 | | | | |
| YEAR20 | 48-32# | 124-95 | | | | |
| YER1 | 124-69# | 124-82 | | | | |
| YER2 | 124-70 | 124-72 | 124-83# | | | |
| YER3 | 124-86 | 124-92# | | | | |
| YER4 | 124-94 | 124-96# | 124-99 | | | |
| YER5 | 124-91 | 124-97 | 124-100# | 124-101 | | |

| | | | | | | | | | | | | | | |
|--------|---------|---------|---------|---------|---------|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| MSGETS | 29-17 | 29-17# | 30-14 | 30-14# | 30-16 | 30-16# | 54-16 | 54-16# | 54-20 | 54-20# | 54-24 | 54-24# | 54-28 | 54-28# |
| | 54-32 | 54-32# | 54-36 | 54-36# | 54-40 | 54-40# | 54-44 | 54-44# | 54-53 | 54-53# | 54-59 | 54-59# | 54-73 | 54-73# |
| | 54-77 | 54-77# | 54-81 | 54-81# | 54-85 | 54-85# | 54-89 | 54-89# | 54-94 | 54-94# | 54-98 | 54-98# | 54-102 | 54-102# |
| | 54-106 | 54-106# | 54-110 | 54-110# | 54-114 | 54-114# | 54-118 | 54-118# | 102-14 | 102-14# | 103-21 | 103-21# | 113-9 | 113-9# |
| | 114-38 | 114-38# | 115-16 | 115-16# | 124-104 | 124-104# | 125-12 | 125-12# | 126-13 | 126-13# | 126-15 | 126-15# | 127-69 | 127-69# |
| | 127-70 | 127-70# | 129-6 | 129-6# | 131-2 | 131-2# | 131-4 | 131-4# | 131-6 | 131-6# | 131-8 | 131-8# | 131-10 | 131-10# |
| | 132-10 | 132-10# | | | | | | | | | | | | |
| MSGETT | 122-14# | 127-11# | 127-17# | 127-68# | 131-2 | 131-2# | 131-4 | 131-4# | 131-6 | 131-6# | 131-8 | 131-8# | | |
| MSGNGB | 27-26# | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 |
| | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 |
| | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 |
| | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# |
| | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# |
| | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# |
| | 30-10# | 31-3# | 49-12 | 49-12# | 49-16 | 49-16# | 54-14 | 54-14# | 54-18 | 54-18# | 54-22 | 54-22# | 54-26 | 54-26# |
| | 54-30 | 54-30# | 54-34 | 54-34# | 54-38 | 54-38# | 54-42 | 54-42# | 54-46 | 54-46# | 54-55 | 54-55# | 54-61 | 54-61# |
| | 54-75 | 54-75# | 54-79 | 54-79# | 54-83 | 54-83# | 54-87 | 54-87# | 54-91 | 54-91# | 54-96 | 54-96# | 54-100 | 54-100# |
| | 54-104 | 54-104# | 54-108 | 54-108# | 54-112 | 54-112# | 54-116 | 54-116# | 102-10 | 102-10# | 103-18 | 103-18# | 113-5 | 113-5# |
| | 115-3# | 115-10 | 115-10# | 116-8 | 116-8# | 125-10 | 125-10# | 126-8 | 126-8# | 127-3# | 128-3# | 128-14 | 128-14# | 130-12 |
| | 130-12# | 132-8 | 132-8# | | | | | | | | | | | |
| MSGNIN | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 |
| | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 |
| | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 |
| | 27-34 | 27-34 | 27-34 | 27-34 | 27-34 | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# |
| | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# |
| | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# |
| | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# | 27-34# |
| | 49-12 | 49-12 | 49-12# | 49-12# | 49-16 | 49-16# | 49-16# | 49-16# | 54-16 | 54-16# | 54-20 | 54-20# | 54-24 | 54-24# |
| | 54-28 | 54-28# | 54-32 | 54-32# | 54-36 | 54-36# | 54-40 | 54-40# | 54-44 | 54-44# | 54-53 | 54-53# | 54-58 | 54-58# |
| | 54-58 | 54-58 | 54-58 | 54-58# | 54-58# | 54-58# | 54-58# | 54-59 | 54-59# | 54-73 | 54-73# | 54-77 | 54-77# | 54-81 |
| | 54-81# | 54-85 | 54-85# | 54-89 | 54-89# | 54-94 | 54-94# | 54-98 | 54-98# | 54-102 | 54-102# | 54-106 | 54-106# | 54-110 |
| | 54-110# | 54-114 | 54-114# | 54-118 | 54-118# | 56-7 | 56-7 | 56-7 | 56-7 | 56-7# | 56-7# | 56-7# | 56-7# | 56-7# |
| | 56-8 | 56-8# | 60-8 | 60-8# | 60-24 | 60-24 | 60-24 | 60-24 | 60-24 | 60-24# | 60-24# | 60-24# | 60-24# | 60-24# |
| | 61-36 | 61-36 | 61-36 | 61-36# | 61-36# | 61-36# | 61-36# | 61-36# | 61-36# | 62-21 | 62-21 | 62-21 | 62-21# | 62-21# |
| | 62-21# | 62-21# | 62-21# | 63-8 | 63-8 | 63-8 | 63-8 | 63-8 | 63-8# | 63-8# | 63-8# | 63-8# | 63-8# | 63-8# |
| | 67-22 | 67-22 | 67-22# | 67-22# | 67-22# | 67-22# | 67-22# | 67-22# | 69-30 | 69-30 | 69-30 | 69-30 | 69-30# | 69-30# |
| | 69-30# | 69-30# | 72-18 | 72-18# | 72-19 | 72-19 | 72-19# | 72-19# | 72-23 | 72-23 | 72-23# | 72-23# | 72-24 | 72-24# |
| | 91-12 | 91-12 | 91-12 | 91-12# | 91-12 | 91-12 | 91-12# | 91-12# | 91-12# | 91-12# | 91-12# | 91-12# | 91-14 | 91-14 |
| | 91-14 | 91-14 | 91-14 | 91-14# | 91-14# | 91-14# | 91-14# | 91-14# | 91-16 | 91-16 | 91-16 | 91-16 | 91-16 | 91-16 |
| | 91-16# | 91-16# | 91-16# | 91-16# | 91-16# | 91-18 | 91-18 | 91-18 | 91-18 | 91-18 | 91-18 | 91-18# | 91-18# | 91-18# |
| | 91-18# | 91-18# | 94-20 | 94-20 | 94-20 | 94-20 | 94-20 | 94-20 | 94-20 | 94-20# | 94-20# | 94-20# | 94-20# | 94-20# |
| | 96-3 | 96-3 | 96-3 | 96-3 | 96-3# | 96-3# | 96-3# | 96-3# | 96-3# | 96-3# | 100-21 | 100-21# | 100-29 | 100-29 |
| | 100-29 | 100-29# | 100-29# | 100-29# | 100-29# | 100-29# | 101-5 | 101-5 | 101-5 | 101-5 | 101-5# | 101-5# | 101-5# | 101-5# |
| | 101-5# | 102-14 | 102-14# | 103-21 | 103-21# | 105-35 | 105-35 | 105-35 | 105-35 | 105-35# | 105-35# | 105-35# | 105-35# | 105-35# |
| | 106-11 | 106-11 | 106-11 | 106-11 | 106-11# | 106-11# | 106-11# | 106-11# | 106-11# | 107-12 | 107-12# | 107-27 | 107-27 | 107-27 |
| | 107-27 | 107-27 | 107-27 | 107-27# | 107-27# | 107-27# | 107-27# | 107-27# | 107-27# | 107-30 | 107-30 | 107-30# | 107-30# | 107-33 |
| | 107-33 | 107-33 | 107-33 | 107-33# | 107-33# | 107-33# | 107-33# | 107-33# | 107-33# | 109-44 | 109-44 | 109-44 | 109-44# | 109-44# |
| | 109-44# | 109-44# | 109-44# | 110-27 | 110-27# | 110-36 | 110-36 | 110-36 | 110-36 | 110-36# | 110-36# | 110-36# | 110-36# | 110-36# |
| | 111-5 | 111-5 | 111-5 | 111-5 | 111-5# | 111-5# | 111-5# | 111-5# | 111-5# | 113-9 | 113-9# | 116-10 | 116-10 | 116-10# |
| | 116-10# | 116-11 | 116-11# | 116-12 | 116-12 | 116-12# | 116-12# | 116-13 | 116-13# | 116-14 | 116-14 | 116-14# | 116-14# | 116-15 |
| | 116-15# | 116-16 | 116-16 | 116-16# | 116-16# | 116-17 | 116-17# | 116-18 | 116-18# | 116-33 | 116-33# | 116-34 | 116-34 | 116-34# |
| | 116-34# | 116-43 | 116-43 | 116-43# | 116-43# | 116-43# | 116-44 | 116-44# | 116-45 | 116-45# | 116-45# | 116-45# | 116-45# | 116-46 |
| | 116-46# | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54 | 116-54# | 116-54# | 116-54# | 116-54# | 116-54# | 116-54# | 117-10 |
| | 117-10 | 117-10# | 117-10# | 117-10# | 117-11 | 117-11# | 120-4 | 120-4 | 120-4 | 120-4# | 120-4# | 120-5 | 120-5# | 122-12 |
| | 122-12 | 122-12# | 122-12# | 122-13 | 122-13# | 122-14 | 122-14 | 122-14 | 122-14# | 123-3 | 123-3 | 123-3 | 123-3 | 123-3# |
| | 123-3# | 123-3# | 123-3# | 123-3# | 123-4 | 123-4# | 123-8 | 123-8 | 123-8 | 123-8 | 123-8# | 123-8# | 123-8# | 123-8# |

| | | | | | | | | | | | | | | |
|--------|---------|---------|---------|---------|---------|----------|----------|---------|---------|---------|----------|---------|---------|----------|
| | 123-8# | 123-9 | 123-9# | 123-14 | 123-14 | 123-14 | 123-14 | 123-14# | 123-14# | 123-14# | 123-14# | 123-14# | 123-15 | 123-15# |
| | 123-20 | 123-20 | 123-20 | 123-20 | 123-20# | 123-20# | 123-20# | 123-20# | 123-20# | 123-21 | 123-21# | 124-1 | 124-1 | 124-1 |
| | 124-1 | 124-1 | 124-1 | 124-1 | 124-1# | 124-1# | 124-1# | 124-1# | 124-1# | 124-104 | 124-104# | 125-12 | 125-12# | 126-10 |
| | 126-10# | 126-11 | 126-11# | 126-13 | 126-13# | 127-8 | 127-8# | 127-9 | 127-9# | 127-10 | 127-10 | 127-10 | 127-10 | 127-10# |
| | 127-10# | 127-10# | 127-10# | 127-10# | 127-11 | 127-11 | 127-11# | 127-11# | 127-11# | 127-16 | 127-16 | 127-16 | 127-16 | 127-16# |
| | 127-16# | 127-16# | 127-16# | 127-17 | 127-17 | 127-17# | 127-17# | 127-17# | 127-17# | 127-20 | 127-20 | 127-20 | 127-20 | 127-20 |
| | 127-20 | 127-20# | 127-20# | 127-20# | 127-20# | 127-21 | 127-21 | 127-21# | 127-21# | 127-21# | 127-21# | 127-32 | 127-32 | 127-32 |
| | 127-32 | 127-32 | 127-32 | 127-32# | 127-32# | 127-32# | 127-32# | 127-32# | 127-32# | 127-50 | 127-50 | 127-50 | 127-50 | 127-50# |
| | 127-50# | 127-50# | 127-50# | 127-50# | 127-68 | 127-68 | 127-68# | 127-68# | 127-68# | 127-69 | 127-69# | 128-14 | 128-14# | 129-1 |
| | 129-1 | 129-1 | 129-1# | 129-2 | 129-2 | 129-2 | 129-2# | 129-2# | 129-2# | 129-3 | 129-3 | 129-3 | 129-3 | 129-3# |
| | 129-4 | 129-4 | 129-4 | 129-4 | 129-4 | 129-4# | 129-4# | 129-5 | 129-5 | 129-5 | 129-5 | 129-5 | 129-5# | 129-6 |
| | 130-12 | 130-12# | 131-1 | 131-1 | 131-1 | 131-1# | 131-1# | 131-2 | 131-2# | 131-3 | 131-3 | 131-3 | 131-3# | 131-4 |
| | 131-5 | 131-5 | 131-5 | 131-5# | 131-6 | 131-6# | 131-6# | 131-7 | 131-7 | 131-7 | 131-7# | 131-8 | 131-8# | 131-9 |
| | 131-9# | 131-9# | 131-10 | 131-10# | 132-8 | 132-8 | 132-8 | 132-8# | 133-3 | 133-3 | 133-3# | 133-3# | 133-3# | 133-3# |
| MSGNLS | 124-1 | 124-1# | 127-20 | 127-20# | 127-32 | 127-32# | 127-32# | 127-32# | 127-32# | 127-32# | 127-32# | 127-32# | 127-32# | 127-32# |
| MSGNTA | 29-17 | 29-17# | 30-14 | 30-14# | 54-16 | 54-16# | 54-20 | 54-20# | 54-24 | 54-24# | 54-28 | 54-28# | 54-32 | 54-32# |
| | 54-36 | 54-36# | 54-40 | 54-40# | 54-44 | 54-44# | 54-53 | 54-53# | 54-59 | 54-59# | 54-73 | 54-73# | 54-77 | 54-77# |
| | 54-81 | 54-81# | 54-85 | 54-85# | 54-89 | 54-89# | 54-94 | 54-94# | 54-98 | 54-98# | 54-102 | 54-102# | 54-106 | 54-106# |
| | 54-110 | 54-110# | 54-114 | 54-114# | 54-118 | 54-118# | 102-14 | 102-14# | 103-21 | 103-21# | 113-9 | 113-9# | 124-104 | 124-104# |
| MSGNTE | 125-12 | 125-12# | 126-13 | 126-13# | 127-69 | 127-69# | 129-6 | 129-6# | 131-10 | 131-10# | 133-3 | 133-3# | 133-11 | 133-11# |
| MSHAPT | 127-5 | 127-5# | | | | | | | | | | | | |
| MSHNAP | 27-34 | 27-34# | | | | | | | | | | | | |
| MSINCR | 27-34 | 27-34# | | | | | | | | | | | | |
| | 27-26 | 27-26# | 29-10 | 29-10 | 29-10# | 29-10# | 30-10 | 30-10 | 30-10# | 30-10# | 30-10# | 31-3 | 31-3# | 54-14 |
| | 54-14# | 54-14# | 54-16# | 54-18 | 54-18 | 54-18# | 54-18# | 54-20# | 54-22 | 54-22 | 54-22# | 54-22# | 54-24# | 54-26 |
| | 54-26 | 54-26# | 54-26# | 54-28# | 54-30 | 54-30 | 54-30# | 54-30# | 54-32# | 54-34 | 54-34 | 54-34# | 54-34# | 54-36# |
| | 54-38 | 54-38 | 54-38# | 54-38# | 54-40# | 54-42 | 54-42 | 54-42# | 54-42# | 54-44# | 54-46 | 54-46 | 54-46# | 54-46# |
| | 54-53# | 54-55 | 54-55 | 54-55# | 54-55# | 54-58# | 54-59# | 54-61 | 54-61 | 54-61# | 54-61# | 54-73# | 54-75 | 54-75 |
| | 54-75# | 54-75# | 54-77# | 54-79 | 54-79 | 54-79# | 54-79# | 54-81# | 54-83 | 54-83 | 54-83# | 54-83# | 54-85# | 54-87 |
| | 54-87 | 54-87# | 54-87# | 54-89# | 54-91 | 54-91 | 54-91# | 54-91# | 54-94# | 54-96 | 54-96 | 54-96# | 54-96# | 54-98# |
| | 54-100 | 54-100 | 54-100# | 54-100# | 54-102# | 54-104 | 54-104 | 54-104# | 54-104# | 54-106# | 54-108 | 54-108 | 54-108# | 54-108# |
| | 54-110# | 54-112 | 54-112 | 54-112# | 54-112# | 54-114# | 54-116 | 54-116 | 54-116# | 54-116# | 54-118# | 56-7# | 56-8# | 60-8# |
| | 60-24# | 61-36# | 62-21# | 63-8# | 67-22# | 69-30# | 72-18# | 72-19# | 72-23# | 91-12# | 91-14# | 91-16# | 91-18# | 94-20# |
| | 96-3# | 100-21# | 100-29# | 101-5# | 102-10 | 102-10 | 102-10# | 102-10# | 103-18 | 103-18 | 103-18# | 103-18# | 105-35# | 106-11# |
| | 107-12# | 107-27# | 107-30# | 107-33# | 109-44# | 110-27# | 110-36# | 111-5# | 113-5 | 113-5 | 113-5# | 113-5# | 115-3 | 115-3# |
| | 115-10 | 115-10 | 115-10# | 115-10# | 116-8 | 116-8 | 116-8# | 116-8# | 116-10# | 116-12# | 116-14# | 116-16# | 116-18# | 116-33# |
| | 116-34# | 116-43# | 116-45# | 116-54# | 117-10# | 120-4# | 122-12# | 122-13# | 122-14# | 123-3# | 123-4# | 123-8# | 123-9# | 123-14# |
| | 123-15# | 123-20# | 123-21# | 124-1 | 124-1# | 124-1# | 124-104# | 125-10 | 125-10 | 125-10# | 125-10# | 125-12# | 126-8 | 126-8 |
| | 126-8# | 126-8# | 126-10# | 126-11# | 126-13# | 127-3 | 127-3# | 127-5 | 127-5 | 127-5 | 127-5# | 127-5# | 127-5# | 127-8# |
| | 127-10# | 127-11# | 127-16# | 127-17# | 127-20 | 127-20# | 127-20# | 127-21# | 127-32 | 127-32# | 127-32# | 127-50# | 127-68# | 127-69# |
| | 128-3 | 128-3# | 128-14 | 128-14 | 128-14# | 128-14# | 130-12 | 130-12 | 130-12# | 130-12# | 133-1 | 133-1# | 133-3 | 133-3 |
| | 133-3 | 133-3# | | | | | | | | | | | | |
| MSLDRO | 72-19 | 72-19# | 107-30 | 107-30# | 116-10 | 116-10# | 116-12 | 116-12# | 116-14 | 116-14# | 116-16 | 116-16# | 116-43 | 116-43# |
| | 116-45 | 116-45# | 117-10 | 117-10# | 120-4 | 120-4# | 122-12 | 122-12# | 127-21 | 127-21# | | | | |
| MSMCHI | 27-8 | 27-8# | | | | | | | | | | | | |
| MSMCLO | 27-8 | 27-8# | | | | | | | | | | | | |
| MSPOP | 29-17 | 29-17# | 30-14 | 30-14# | 30-16 | 30-16# | 54-16 | 54-16# | 54-20 | 54-20# | 54-24 | 54-24# | 54-28 | 54-28# |
| | 54-32 | 54-32# | 54-36 | 54-36# | 54-40 | 54-40# | 54-44 | 54-44# | 54-53 | 54-53# | 54-59 | 54-59# | 54-73 | 54-73# |
| | 54-77 | 54-77# | 54-81 | 54-81# | 54-85 | 54-85# | 54-89 | 54-89# | 54-94 | 54-94# | 54-98 | 54-98# | 54-102 | 54-102# |
| | 54-106 | 54-106# | 54-110 | 54-110# | 54-114 | 54-114# | 54-118 | 54-118# | 102-14 | 102-14# | 103-21 | 103-21# | 113-9 | 113-9# |
| | 114-38 | 114-38# | 115-16 | 115-16# | 124-104 | 124-104# | 125-12 | 125-12# | 126-13 | 126-13# | 126-15 | 126-15# | 127-69 | 127-69# |
| | 127-70 | 127-70# | 129-6 | 129-6# | 131-10 | 131-10# | 132-10 | 132-10# | | | | | | |
| MSPRIN | 54-58 | 54-58# | 91-12 | 91-12# | 91-14 | 91-14# | 91-16 | 91-16# | 91-18 | 91-18# | 127-50 | 127-50# | | |
| MSPUSH | 27-26 | 27-26# | 29-10 | 29-10# | 30-10 | 30-10# | 31-3 | 31-3# | 54-14 | 54-14# | 54-18 | 54-18# | 54-22 | 54-22# |
| | 54-26 | 54-26# | 54-30 | 54-30# | 54-34 | 54-34# | 54-38 | 54-38# | 54-42 | 54-42# | 54-46 | 54-46# | 54-55 | 54-55# |
| | 54-61 | 54-61# | 54-75 | 54-75# | 54-79 | 54-79# | 54-83 | 54-83# | 54-87 | 54-87# | 54-91 | 54-91# | 54-96 | 54-96# |

| | | | | | | | | | | | | | | |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|---------|---------|---------|----------|
| | 54-100 | 54-100# | 54-104 | 54-104# | 54-108 | 54-108# | 54-112 | 54-112# | 54-116 | 54-116# | 102-10 | 102-10# | 103-18 | 103-18# |
| | 113-5 | 113-5# | 115-3 | 115-3# | 115-10 | 115-10# | 116-8 | 116-8# | 125-10 | 125-10# | 126-8 | 126-8# | 127-3 | 127-3# |
| MSPUT | 127-5 | 127-5# | 128-3 | 128-3# | 128-14 | 128-14# | 130-12 | 130-12# | | | | | | |
| | 54-58 | 54-58 | 54-58# | 91-12 | 91-12 | 91-12 | 91-12# | 91-14 | 91-14 | 91-14 | 91-14# | 91-16 | 91-16 | 91-16 |
| | 91-16# | 91-18 | 91-18 | 91-18 | 91-18# | 94-20 | 94-20 | 94-20 | 94-20 | 94-20# | 107-27 | 107-27 | 107-27 | 107-27 |
| | 107-27# | 116-54 | 116-54 | 116-54 | 116-54 | 116-54# | 127-50 | 127-50 | 127-50 | 127-50# | | | | |
| MSPUT1 | 54-58 | 54-58 | 54-58# | 54-58# | 91-12 | 91-12 | 91-12 | 91-12# | 91-12# | 91-12# | 91-14 | 91-14 | 91-14 | 91-14# |
| | 91-14# | 91-14# | 91-16 | 91-16 | 91-16 | 91-16# | 91-16# | 91-16# | 91-16# | 91-18 | 91-18 | 91-18# | 91-18# | 91-18# |
| | 94-20 | 94-20 | 94-20 | 94-20 | 94-20# | 94-20# | 94-20# | 94-20# | 94-20# | 107-27 | 107-27 | 107-27 | 107-27# | 107-27# |
| | 107-27# | 107-27# | 116-54 | 116-54 | 116-54 | 116-54 | 116-54# | 116-54# | 116-54# | 116-54# | 127-50 | 127-50 | 127-50 | 127-50# |
| | 127-50# | 127-50# | | | | | | | | | | | | |
| MSRADI | 124-1 | 124-1# | 127-20 | 127-20# | 127-32 | 127-32# | 129-1 | 129-1# | 129-2 | 129-2# | 129-3 | 129-3# | 129-4 | 129-4# |
| | 129-5 | 129-5# | 131-1 | 131-1# | 131-3 | 131-3# | 131-5 | 131-5# | 131-7 | 131-7# | | | | |
| MSRBRO | 72-23 | 72-23# | | | | | | | | | | | | |
| MSRNRO | 116-34 | 116-34# | 116-43 | 116-43# | 116-45 | 116-45# | 117-10 | 117-10# | 120-4 | 120-4# | | | | |
| MSSETS | 27-26 | 27-26# | 29-10 | 29-10# | 30-10 | 30-10# | 31-3 | 31-3# | 54-14 | 54-14# | 54-18 | 54-18# | 54-22 | 54-22# |
| | 54-26 | 54-26# | 54-30 | 54-30# | 54-34 | 54-34# | 54-38 | 54-38# | 54-42 | 54-42# | 54-46 | 54-46# | 54-55 | 54-55# |
| | 54-61 | 54-61# | 54-75 | 54-75# | 54-79 | 54-79# | 54-83 | 54-83# | 54-87 | 54-87# | 54-91 | 54-91# | 54-96 | 54-96# |
| | 54-100 | 54-100# | 54-104 | 54-104# | 54-108 | 54-108# | 54-112 | 54-112# | 54-116 | 54-116# | 102-10 | 102-10# | 103-18 | 103-18# |
| | 113-5 | 113-5# | 115-3 | 115-3# | 115-10 | 115-10# | 116-8 | 116-8# | 125-10 | 125-10# | 126-8 | 126-8# | 127-3 | 127-3# |
| | 127-5 | 127-5# | 128-3 | 128-3# | 128-14 | 128-14# | 130-12 | 130-12# | | | | | | |
| MSSVC | 54-16 | 54-16# | 54-20 | 54-20# | 54-24 | 54-24# | 54-28 | 54-28# | 54-32 | 54-32# | 54-36 | 54-36# | 54-40 | 54-40# |
| | 54-44 | 54-44# | 54-53 | 54-53# | 54-58 | 54-58# | 54-59 | 54-59# | 54-73 | 54-73# | 54-77 | 54-77# | 54-81 | 54-81# |
| | 54-85 | 54-85# | 54-89 | 54-89# | 54-94 | 54-94# | 54-98 | 54-98# | 54-102 | 54-102# | 54-106 | 54-106# | 54-110 | 54-110# |
| | 54-114 | 54-114# | 54-118 | 54-118# | 56-7 | 56-7# | 56-8 | 56-8# | 60-8 | 60-8# | 60-24 | 60-24# | 61-36 | 61-36# |
| | 69-30 | 72-18 | 72-18# | 72-19 | 72-19# | 72-23 | 72-23# | 72-23# | 91-12 | 91-12# | 91-14 | 91-14# | 91-16 | 91-16# |
| | 91-18# | 94-20 | 94-20# | 96-3 | 100-21 | 100-21# | 100-29 | 100-29# | 101-5 | 105-35 | 106-11 | 107-12 | 107-12# | 107-27 |
| | 107-30 | 107-30# | 107-33 | 109-44 | 110-27 | 110-27# | 110-36 | 110-36# | 111-5 | 116-10 | 116-10# | 116-12 | 116-12# | 116-14 |
| | 116-16 | 116-16# | 116-18 | 116-18# | 116-33 | 116-33# | 116-34 | 116-34# | 116-43 | 116-43# | 116-45 | 116-45# | 116-54 | 116-54# |
| | 117-10 | 117-10# | 120-4 | 120-4# | 122-12 | 122-12# | 122-13 | 122-13# | 122-14 | 122-14# | 123-3 | 123-3# | 123-4 | 123-4# |
| | 123-9 | 123-9# | 123-14 | 123-14# | 123-15 | 123-15# | 123-20 | 123-20# | 123-21 | 123-21# | 124-1 | 124-1# | 124-104 | 124-104# |
| | 126-10 | 126-10# | 126-11 | 126-11# | 126-13 | 126-13# | 127-8 | 127-8# | 127-10 | 127-10# | 127-11 | 127-11# | 127-16 | 127-16# |
| | 127-20 | 127-20# | 127-21 | 127-21# | 127-32 | 127-32# | 127-50 | 127-50# | 127-68 | 127-68# | 127-69 | 127-69# | 127-17 | 127-17# |
| MSTLAB | 54-16# | 54-20# | 54-24# | 54-28# | 54-32# | 54-36# | 54-40# | 54-44# | 54-53# | 54-58# | 54-59# | 54-73# | 54-77# | 54-81# |
| | 54-85# | 54-89# | 54-94# | 54-98# | 54-102# | 54-106# | 54-110# | 54-114# | 54-118# | 56-7# | 56-8# | 60-8# | 60-24# | 61-36# |
| | 62-21# | 63-8# | 67-22# | 69-30# | 72-18# | 72-19# | 72-23# | 91-12# | 91-14# | 91-16# | 91-18# | 94-20# | 96-3# | 100-21# |
| | 100-29# | 101-5# | 105-35# | 106-11# | 107-12# | 107-27# | 107-30# | 107-33# | 109-44# | 110-27# | 110-36# | 111-5# | 116-10# | 116-12# |
| | 116-14# | 116-16# | 116-18# | 116-33# | 116-34# | 116-43# | 116-45# | 116-54# | 117-10# | 120-4# | 122-12# | 122-13# | 122-14# | 123-3# |
| | 123-4# | 123-8# | 123-9# | 123-14# | 123-15# | 123-20# | 123-21# | 124-1# | 124-104# | 125-12# | 126-10# | 126-11# | 126-13# | 127-8# |
| | 127-10# | 127-11# | 127-16# | 127-17# | 127-20# | 127-21# | 127-32# | 127-50# | 127-68# | 127-69# | | | | |
| MSTSTL | 54-16 | 54-16# | 54-20 | 54-20# | 54-24 | 54-24# | 54-28 | 54-28# | 54-32 | 54-32# | 54-36 | 54-36# | 54-40 | 54-40# |
| | 54-44 | 54-44# | 54-53 | 54-53# | 54-58 | 54-58# | 54-59 | 54-59# | 54-73 | 54-73# | 54-77 | 54-77# | 54-81 | 54-81# |
| | 54-85 | 54-85# | 54-89 | 54-89# | 54-94 | 54-94# | 54-98 | 54-98# | 54-102 | 54-102# | 54-106 | 54-106# | 54-110 | 54-110# |
| | 54-114 | 54-114# | 54-118 | 54-118# | 56-7 | 56-7# | 56-7# | 56-8 | 56-8# | 60-8 | 60-8# | 60-24 | 60-24# | 60-24# |
| | 61-36 | 61-36# | 61-36# | 62-21 | 62-21# | 62-21# | 63-8 | 63-8# | 63-8# | 67-22 | 67-22# | 67-22# | 69-30 | 69-30# |
| | 69-30# | 72-18 | 72-18# | 72-19 | 72-19# | 72-23 | 72-23# | 91-12 | 91-12# | 91-14 | 91-14# | 91-14# | 91-16# | 91-18 |
| | 91-18# | 94-20 | 94-20# | 96-3 | 96-3# | 96-3# | 100-21 | 100-21# | 100-29 | 100-29# | 100-29# | 100-29# | 101-5 | 101-5# |
| | 105-35 | 105-35# | 105-35# | 106-11 | 106-11# | 106-11# | 107-12 | 107-12# | 107-27 | 107-27# | 107-30 | 107-30# | 107-33 | 107-33# |
| | 107-33# | 109-44 | 109-44# | 109-44# | 110-27 | 110-27# | 110-36 | 110-36# | 110-36# | 111-5 | 111-5# | 111-5# | 116-10 | 116-10# |
| | 116-12 | 116-12# | 116-14 | 116-14# | 116-16 | 116-16# | 116-18 | 116-18# | 116-33 | 116-33# | 116-34 | 116-34# | 116-43 | 116-43# |
| | 116-45 | 116-45# | 116-54 | 116-54# | 117-10 | 117-10# | 120-4 | 120-4# | 122-12 | 122-12# | 122-13 | 122-13# | 122-14 | 122-14# |
| | 123-3 | 123-3# | 123-3# | 123-4 | 123-4# | 123-8 | 123-8# | 123-8# | 123-9 | 123-9# | 123-14 | 123-14# | 123-14# | 123-15 |
| | 123-15# | 123-20 | 123-20# | 123-20# | 123-21 | 123-21# | 124-1 | 124-1# | 124-104 | 124-104# | 125-12 | 125-12# | 126-10 | 126-10# |
| | 126-11 | 126-11# | 126-13 | 126-13# | 127-8 | 127-8# | 127-10 | 127-10# | 127-10# | 127-10# | 127-11 | 127-11# | 127-16 | 127-16# |
| | 127-17 | 127-17# | 127-20 | 127-20# | 127-21 | 127-21# | 127-32 | 127-32# | 127-50 | 127-50# | 127-68 | 127-68# | 127-69 | 127-69# |
| MSWORD | 27-34 | 27-34# | 28-9 | 28-9 | 28-9# | 56-7 | 56-7 | 56-7 | 56-7# | 60-24 | 60-24 | 60-24 | 60-24# | 61-36 |

