B1

IDENTIFICATION

PRODUCT CODE: AC T938B-MC

PRODUCT NAME: CZUDKB0 RA SERIES DISK DRIVE FORMATTER

PRODUCT DATE: 23-DEC-1985

MAINTAINER: RON BOWSER

AUTHOR: RON BOWSER

```
1                       .TITLE  CZUDKO UDA50A/KDA50-Q FORMATTER
        .REM
```

TABLE OF CONTENTS

E1

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

This program will format any disk drive connected to a UDA50A or KDA50-Q disk controller. At the time of this writing, there are three such drives in existance -- the RA60, RA80 and RA81. No changes to this program will be needed to format new disk drives as they become available.

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

1. Reformat - Format the disk with the bad sector information that was written onto the disk at the factory. This is the normal way to format a disk.

2. Reconstruct - Format the disk without using any bad sector information. This should be used only when the bad sector information has been destroyed or for some reason can no longer be read from the disk. This method may also be specified in the disk drive's maintenance manual for special cases (eg. changing an RM/RA80 spare HDA from RM80 format to RA80 format).

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

The format operation is performed by a Diagnostic Utilities and Protocol (DUP) program loaded into the disk controller. The host program simply downline loads the DUP program into the controller 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 controller can only format one disk at a time, so each disk on a controller are connected to different controllers, all controllers 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 controller and unit 3 is connected to a different controller. This program will automatically start 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 or Q-bus processor
- 28K words of memory (minimum)
- Console terminal
- XXDP+ load media containing this program
- One or more UDA50A or KDA50-Q 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 "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	Execute only those tests specified in the list. List is a string of test numbers, for example - /TESTS:1:5:7-10. This list will cause tests 1,5,7,8,9,10 to be run. All other tests will not be run.
/PASS:DDDDD	Execute DDDDD passes (DDDDD = 1 to 64000)
/FLAGS:FLGS	Set specified flags. Flags are described in section 2.3.
/EOP:DDDDD	Report end of pass message after every DDDDD passes only. (DDDDD = 1 to 64000)
/UNITS:LIST	TEST/ADD/DROP only those units specified in the list. List example - /UNITS:0:5:10-12 use units 0,5,10,11,12 (unit numbers = 0-63).

Example of switch usage:

```
START/TESTS:1-5/PASS:1000/EOP:100
```

The effect of this command will be: 1) tests 1 through 5 will be executed, 2) all units will tested 1000 times and 3) the end of pass messages will be printed after each 100 passes only. A switch can be recognized by the first three characters. You may, for example, type "/TES:1-5" instead of "/TESTS:1-5".

Below is a table that specifies which switches can be used by each command.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

Flags are used to set up certain operational parameters such as looping on error. All flags are cleared at startup and remain cleared until explicitly set using the flags switch. Flags are also cleared after a START or RESTART command unless set using the flag switch. The ZFLAGS command may also be used to clear all flags. With the exception of the START, RESTART and ZFLAGS commands, no commands affect the state of the flags; they remain set or cleared as specified by the last flag switch.

FLAG	EFFECT
HOE	Halt on error - control is returned to runtime services command mode
LOE	Loop on error
IER*	Inhibit all error reports
IBE*	Inhibit all error reports except first level (first level contains error type, number, PC, test and unit)
IXE*	Inhibit extended error reports (those called by PRINTX macro's)
PRI	Direct messages to line printer
PNT	Print test number as test executes
BOE	"BELL" on error
UAM	Unattended mode (no manual intervention)
IDU	Inhibit program dropping of units
LOT	Loop on test

*Error messages are described in section 3.1

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

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

2.4 HARDWARE QUESTIONS

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

CSR ADDRESS (O) 172150 ?

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

VECTOR (O) 154 ?

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

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 controller at a time.

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 formatter and this mode is not recommended. The format will begin only on a "YES" answer. Otherwise the following message will be printed and the program will abort.

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

2.6 MANUAL INTERVENTION QUESTIONS

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

WARNING:
THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC TOOL. RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK DRIVE'S SERVICE MANUAL.
WARNING:
THIS PROGRAM WILL TAKE APPROXIMATELY 45 MINUTES ON A RA60, 30 MINUTES ON A RA80, 60 MINUTES ON A RA81, AND 120 MINUTES ON A RA82.

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

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

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

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

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

1. If the number typed is 70 or greater, a 19 is prefixed. Eg., 1-JAN-70 translates to year 1970 and 25-DEC-99 translates to year 1999.

2. If the number typed is less than 70, a 20 is prefixed. Eg., 1 APR-21 is translated to year 2021.

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

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

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

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

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

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

2.7 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

As you can see in the above dialogue, the runtime services will build as many entries as it can with the information given in any one pass through the questions. In the first pass, two entries are built since two sub-devices and q-factors were specified. The Services assume that the CSR address is 160000 for both since it was specified only once. In the second pass, four entries were built. This is because four sub-devices were specified. The "-" construct tells the Runtime Services to increment the data from the first number to the second. In this case, sub-devices 2, 3, 4 and 5 were specified. (If the sub-devices 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 ZUDKA0"
4. Type "START"
5. Answer the "CHANGE HW" question with "Y"
6. Answer all the hardware questions
7. Answer the "CHANGE SW" question with "N"
8. Answer "Y" to the "ARE YOU SURE ..." question following the warning. Please read the disk drive's service manual before answering this question.
9. Type today's date.

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

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

```
DR>STA

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
CSR ADDRESS (O)  172150 ?
VECTOR (O)  154 ?
DRIVE NUMBER (D)  0 ? 0,1

CHANGE SW (L)  ? N

WARNING:
      THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC
      TOOL.  RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK
      DRIVE'S SERVICE MANUAL.
WARNING:
      THIS PROGRAM WILL TAKE APPROXIMATELY 45 MINUTES ON
      A RA60, 30 MINUTES ON A RA80, 60 MINUTES ON A RA81, AND
      120 MINUTES ON A RA82.

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

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

UNIT 0 CONTROLLER AT 172150 DRIVE 0   RUNTIME 0:00:20
Format begun Version 11
STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK
UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN
BROUGHT ONLINE.

UNIT 1 CONTROLLER AT 172150 DRIVE 1   RUNTIME 0:00:23
Format begun Version 11
STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK
UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN
BROUGHT ONLINE.

UNIT 0 CONTROLLER AT 172150 DRIVE 0   RUNTIME 0:42:20
Format completed
     2 Revectored LBNS
     2 Primary revectored LBNS
     0 Secondary/tertiary revectored LBNS
     0 Bad RBNS
     0 Bad blocks in the RCT area due to data errors
     0 Bad blocks in the DBN area due to data errors
     0 Bad blocks in the XBN area due to data errors
     2 Blocks retried on the check pass
FCT used successfully
```

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

CZUDK 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

WARNING:
      THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC
      TOOL.  RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK
      DRIVE'S SERVICE MANUAL.
WARNING:
      THIS PROGRAM WILL TAKE APPROXIMATELY 45 MINUTES ON
      A RA60, 30 MINUTES ON A RA80, 60 MINUTES ON A RA81, AND
      120 MINUTES ON A RA82.

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

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

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

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

```
CZUDK EOP     1
    0 CUMULATIVE ERRORS
DR>
```

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

There are three levels of error messages that may be issued by the formatter: 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 = formatter 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 formatter program. The formatter 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 the controller 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 formatter are always one line each. The basic message defines what program detected the error, the controller being used and the time of the error:

```
HOST  PROGRAM CONTROLLER AT xxxxxx  RUNTIME hhh:mm:ss
```

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

Sample error message:

```
CZUDK DVC FTL ERR  00021 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx      -  general message
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME 0:00:12                 -  basic message
CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE                   \
  SA CONTAINS 104041                                                }-  extended message
REPLACE CONTROLLER PROCESSOR MODULE                                /
```

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 formatter 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, controller address and data in registers are filled with sample numbers. Additional information about the error may follow the error message.

00001 CZUDK SYS FTL ERR 00001 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS
CONTROLLER HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE

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

00002 CZUDK SYS FTL ERR 00002 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS
MULTIPLE 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 the hardware questions may be changed.

00003 CZUDK SYS FTL ERR 00003 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS
MORE THAN EIGHT DRIVES SELECTED ON THIS CONTROLLER

Up to four physical disk drives can be attached to a UDA50A or KDA50-Q 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 formatter program. Even though more than eight logical disk drives can be attached to one UDA50A or KDA50-Q, the controller only supports eight. The program is aborted and returns to the Runtime Services prompt so that the hardware questions may be changed.

00004 CZUDK 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 FORMAT THE UNITS SELECTED
PLEASE START PROGRAM OVER AND FORMAT 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. The number of units that are testable at one time has been exceeded. Start program over and select fewer units.

00008 CZUDK SYS FTL ERR 00008 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS
TWO CONTROLLERS USE THE SAME VECTOR

The hardware questions for two units specified different CSR addresses but identical vector addresses. The program is aborted and returns to the Runtime Services prompt so that the hardware questions may be changed.

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

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

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

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

00014 CZUDK DVC FTL ERR 00014 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
CONTROLLER IS NOT SUPPORTED BY THIS FORMATTER PROGRAM. THIS
PROGRAM REQUIRES A UDA50-A (MODEL 6) OR A KDA50-Q (MODEL 13)
CONTROLLER. CONTROLLER REPORTED MODEL CODE xx.

All UDA50-0's (modules M7161-2) are not supported by this formatter. The module sets M7485-6 and M????-? are the only ones that can be used by this formatter. If the controller is a UDA50-0 (M7161-2) it will not be tested. If the controller consists of the M7161-2 modules, install one with M7485-6 modules. Replace both modules, mixing the module sets will not work.

00020 CZUDK DVC FTL ERR 00020 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS
CHECK CSR SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE OR BUS
OR REPLACE CONTROLLER PROCESSOR MODULE

A non-existant memory error occurred when the host program tried to access the IP and SA registers. The controller is at another address (check the CSR selection switches) or the BUS or the controller processor module is broken.

00021 CZUDK DVC FTL ERR 00021 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE
SA CONTAINS 105154
REPLACE CONTROLLER SDI MODULE

The controller Resident diagnostic detected a failure. The error is displayed in the SA. 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 - Unexpectant 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 in the last line of the error message.

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

                The controller did not respond as expected during the
                initialization sequence which communicates using data
                in the SA register. A normal response from the controller
                contains either a STEP bit or an ERROR bit defined as
                follows:
                        Bit 15 (100000)   Error bit
                        Bit 14 (040000)   Step 4 bit
                        Bit 13 (020000)   Step 3 bit
                        Bit 12 (010000)   Step 2 bit
                        bit 11 (004000)   Step 1 bit
                Neither the expected step bit nor the error bit set within
                the expected time.

00023   CZUDK DVC FTL ERR  00023 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
        HOST PROGRAM  CONTROLLER AT 172150  RUNTIME x:xx:xx
        CONTROLLER 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 CONTROLLER PROCESSOR MODULE

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

00024   CZUDK DVC FTL ERR  00024 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
        HOST PROGRAM  CONTROLLER AT 172150  RUNTIME x:xx:xx
        SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION
        PURGE/POLE DIAGNOSTICS WERE REQUESTED
          SA CONTAINS 004400
        REPLACE CONTROLLER PROCESSOR MODULE

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

C 5 CZUDK DVC FTL ERR 00025 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
CONTROLLER DID NOT RETURN CORRECT DATA IN SA REGISTER DURING
INITIALIZATION
SA EXPECTED 004400
SA CONTAINS 004000
REPLACE CONTROLLER PROCESSOR MODULE

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

00030 CZUDK DVC FTL ERR 00030 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE RUNNING FORMATTER
SA CONTAINS 100004

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

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

00031 CZUDK DVC FTL ERR 00031 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
FORMATTER 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 locp. If the DM program has not done so, the host program assumes the DM is hung and this message appears.

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

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

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

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

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

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

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

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

00037 CZUDK DVC FTL ERR 00037 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE LOADING FORMATTER
SA CONTAINS 100004
REPLACE CONTROLLER PROCESSOR MODULE

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

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

The formatter 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 CZUDK DVC FTL ERR 00101 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
FORMATTER REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION

After the operator inputs the date/serial number, the formatter will ask the host program for them. If for some reason the date/serial number was unacceptable to the formatter, 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 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 controller 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 controller. Again, there may be a cable/receiver/transmitter problem.

BUS read error

This is caused by one of two problems. While trying to read an overlay into the controller 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 BUS or the controller processor module.

Formatter initialization error

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

Non-existant unit number

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

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 be 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 BUS or controller processor module problem.

Drive init timeout

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

Illegal response to start-up question

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

FCT corrupted - Format Invalid

A problem was detected while using the data in the FCT. Either the data was not written properly or it has been corrupted since the last format. The format on the disk is no good and the disk will not be usable by any DEC operating system. Running the formatter again may have a slight chance of succeeding. Otherwise, replace the disk or HDA. If you do not have a spare disk or HDA you may try to format the disk in RECONSTRUCT mode. If the disk is not an RA80, order a replacement disk or HDA immediately.

```
DRIVE ERROR ENCOUNTERED - STATUS RESPONSE:
STATUS (R TO L):  1AF1  0304  E100  8800  0080  0013  1000
LAST BLOCK ACCESSED (16-BIT OCTAL): 000000  000000
```

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

MORE THAN 12.5% OF TRACK IS BAD

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

An example of how the errors are presented is below:

```
  RUNTIME  0:00:18
Non-existant unit number
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

The DUP program issues the following messages upon normal completion:

Format completed

n Revectored LBNS

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

n Primary revectored LBNS

Where n is the number of LBNs which were primary revectors.

n Secondary/tertiary revectored LBNS

Where n is the number of the LBNs which were secondary or tertiary revectors.

n Bad RBNS

Where n is the number of RBNs which were bad.

n Bad blocks in the RCT area due to data errors

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

n Bad blocks in the DBN area due to data errors

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

n Bad blocks in the XBN area due to data errors

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

n Blocks retried on the check pass

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

FCT used successfully or
FCT was not used

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

An example of how the messages are presented is below.

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

5.0 TEST SUMMARIES

There is only one test in this program - Test #1. Its only purpose is to load and run the format program in a UDA50A or KDA50-Q.

```
                                .SBTTL  PROGRAM

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

                                        POINTER BGNSW, BGNSFT, BGNSETUP

                                        HEADER  CZUDK,B,0,7200.,1,PRI07

002000                                                          L$NAME::
002000  103                                                             .ASCII /C/
002001  132                                                             .ASCII /Z/
002002  125                                                             .ASCII /U/
002003  104                                                             .ASCII /D/
002004  113                                                             .ASCII /K/
002005  000                                                             .BYTE   0
002006  000                                                             .BYTE   0
002007  000                                                             .BYTE   0
002010                                                          L$REV::
002010  102                                                             .ASCII  /B/
002011                                                          L$DEPO::
002011  060                                                             .ASCII  /0/
002012                                                          L$UNIT::
002012  000001                                                          .WORD   T$PTHV
002014                                                          L$TIML::
002014  016040                                                          .WORD   7200.
002016                                                          L$HPCP::
002016  023234                                                          .WORD   L$HARD
002020                                                          L$SPCP::
002020  023312                                                          .WORD   L$SOFT
002022                                                          L$HPTP::
002022  002130                                                          .WORD   L$HW
002024                                                          L$SPTP::
002024  002136                                                          .WORD   L$SW
002026                                                          L$LADP::
002026  000124'                                                         .WORD   L$LAST
002030                                                          L$STA::
002030  000000                                                          .WORD   0
002032                                                          L$CO::
002032  000000                                                          .WORD   0
002034                                                          L$DTYP::
002034  000001                                                          .WORD   1
002036                                                          L$APT::
002036  000000                                                          .WORD   0
002040                                                          L$DTP::
002040  002124                                                          .WORD   L$DISPATCH
002042                                                          L$PRIO::
002042  000340                                                          .WORD   PRI07
002044                                                          L$ENVI::
002044  000000                                                          .WORD   0
002046                                                          L$EXP1::
002046  000000                                                          .WORD   0
002050                                                          L$MREV::
002050     003                                                          .BYTE   C$REVISION
002051     003                                                          .BYTE   C$EDIT
```

```
002052
002052  000000                  L$EF::
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::
002060  003454                          .WORD   L$DVTYP
002062                          L$REPP::
002062  000000                          .WORD   0
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::
002070  000000                          .WORD   0
002072                          L$DUT::
002072  000000                          .WORD   0
002074                          L$LUN::
002074  000000                          .WORD   0
002076                          L$DESP::
002076  003502                          .WORD   L$DESC
002100                          L$LOAD::
002100  104035                          EMT     E$LOAD
002102                          L$ETP::
002102  000000                          .WORD   0
002104                          L$ICP::
002104  021512                          .WORD   L$INIT
002106                          L$CCP::
002106  022450                          .WORD   L$CLEAN
002110                          L$ACP::
002110  022446                          .WORD   L$AUTO
002112                          L$PRT::
002112  021504                          .WORD   L$PROT
002114                          L$TEST::
002114  000000                          .WORD   0
002116                          L$DLY::
002116  000000                          .WORD   0
002120                          L$HIME::
002120  000000                          .WORD   0
```

```
                                .SBTTL  DISPATCH TABLE

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


002122                          DISPATCH 1
002122  000001                                                         .WORD   1
002124                                                          L$DISPATCH::
002124  022534                                                          .WORD   T1
```

J3


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

002126                                  BGNHW   DFPTBL
002126  000002                                                                 .WORD   L10000-L$HW/2
002130                                                                  L$HW::
002130                                                                  DFPTBL::

002130  172150                          .WORD   172150          ; UNIBUS ADDRESS
002132  000000                          .WORD   0.              ; LOGICAL DRIVE NUMBER
002134                                  ENDHW
002134                                                                  L10000:
```

```
                                .SBTTL  SOFTWARE P-TABLE

                                ;++
                                ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
                                ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
                                ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
                                ; AT RUN TIME.
                                ;--

002134                                  BGNSW   SFPTBL
002134  000001                                                                 .WORD   L10001-L$SW/2
002136                                                                  L$SW::
002136                                                                  SFPTBL::

                                                                ;OFFSET         USE
002136  000007                          .WORD   7                       ; 0.    YES/NO ANSWERS
002140                                  ENDSW
002140                                                                  L10001:

002140                                  ENDMOD
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

002140                                  BGNMOD

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

002140                                  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
```

M3


```
                000140                  PRIO3== 140
                000100                  PRIO2== 100
                000040                  PRIO1== 40
                000000                  PRIO0== 0
                                        ;
                                        ;OPERATOR FLAG BITS
                                        ;
                000004                  EVL==       4
                000010                  LOT==      10
                000020                  ADR==      20
                000040                  IDU==      40
                000100                  ISR==     100
                000200                  UAM==     200
                000400                  BOE==     400
                001000                  PNT==    1000
                002000                  PRI==    2000
                004000                  IXE==    4000
                010000                  IBE==   10000
                020000                  IER==   20000
                040000                  LOE==   40000
                100000                  HOE==  100000
    11
    12          000015                  CR=     15                      ;VALUE TO PASS TO PRINT MACRO TO END LINE
```

```
                        ;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
```

```
                                ;UDA BIT DEFINITIONS
                                ;
                                ;UDASA REGISTER UNIVERSAL READ BITS
                                ;
004000                          SA.S1=  004000          ;STEP 1 STATUS BIT
010000                          SA.S2=  010000          ;STEP 2 STATUS BIT
020000                          SA.S3=  020000          ;STEP 3 STATUS BIT
040000                          SA.S4=  040000          ;STEP 4 STATUS BIT
100000                          SA.ERR= 100000          ;ERROR INDICATOR
001000                          SA.QB=  1000            ;QB BIT MASK
000100                          SA.MP=  100             ;MP BIT MASK
000040                          SA.SM=  40              ;SA BIT MASK
                                ;
                                ;UDASA REGISTER ERROR STATUS BITS
                                ;
003777                          SA.ERC= 003777          ;ERROR CODE
                                ;
                                ;UDASA REGISTER STEP ONE READ BITS
                                ;
002000                          SA.NV=  002000          ;NON SETTABLE INTERRUPT VECTOR
001000                          SA.A2=  001000          ;22 BIT ADDRESS BUS
000400                          SA.DI=  000400          ;ENHANCED DIAGNOSTICS
                                ;       000377          ;ALL BITS RESERVED
                                ;
                                ;UDASA REGISTER STEP ONE WRITE BITS
                                ;
000177                          SA.VEC= 000177          ;INTERRUPT VECTOR (DIVIDED BY 4)
000200                          SA.INT= 000200          ;INTERRUPT ENABLE DURING INITIALIZATION
003400                          SA.MSG= 003400          ;MESSAGE RING LENGTH
034000                          SA.CMD= 034000          ;COMMAND RING LENGTH
040000                          SA.WRP= 040000          ;WRAP BIT
100000                          SA.STP= 100000          ;STEP - MUST ALWAYS BE WRITTEN A ONE
                                ;
000400                          SA.MS1= 000400          ;LSB OF MESSAGE RING LENGTH
004000                          SA.CM1= 004000          ;LSB OF COMMAND RING LENGTH
                                ;
                                ;UDASA REGISTER STEP TWO READ BITS
                                ;
000007                          SA.MSE= 000007          ;MESSAGE RING LENGTH ECHO
000070                          SA.CME= 000070          ;COMMAND RING LENGTH ECHO
                                ;       000100          ;RESERVED
000200                          SA.STE= 000200          ;STEP ECHO
003400                          SA.CTP= 003400          ;CONTROLLER TYPE
                                ;
                                ;UDASA REGISTER STEP TWO WRITE BITS
                                ;
000001                          SA.PRG= 000001          ;ENABLE VAX UNIBUS ADAPTER PURGE INTERRUPT
                                ;       177776          ;LOW ORDER MESSAGE RING BYTE ADDRESS
```

```
                                ;UDASA REGISTER STEP THREE READ BITS
                                
000177                          SA.VCE= 000177          ;INTERRUPT VECTOR ECHO
000200                          SA.INE= 000200          ;INTERRUPT ENABLE ECHO
000400                          SA.NVE= 000400          ;VECTOR NOT PROGRAMMABLE
                                ;       003000          ;RESERVED

                                ;UDASA REGISTER STEP THREE WRITE BITS

                                ;       077777          ;HIGH ORDER MESSAGE RING BYTE ADDRESS
100000                          SA.TST= 100000          ;PURGE POLE TEST ENABLE

                                ;UDASA REGISTER STEP FOUR READ BITS

000017                          SA.MCV= 000017          ;UDA MICROCODE VERSION
003760                          SA.CNT= 003760          ;CONTROLLER MODEL

                                ;UDASA REGISTER STEP FOUR WRITE BITS

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

                                ;INIT ROUTINE FLAGS

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

```
                                ;COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS
                                ;
100000                          RG.OWN= 100000                  ;SET WHEN UDA OWNS RING
040000                          RG.FLG= 040000                  ;FLAG BIT
                                ;
                                ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
                                ;AND TWO PACKET AND BUFFER AREAS.
                                ;
000004                          HC.ISZ=  4.                     ;SIZE OF INTERRUPT INDICATOR WORDS
000004                          HC.RSZ=  4.                     ;SIZE OF RING IN BYTES
000004                          HC.ESZ=  4.                     ;SIZE OF ENVELOPE WORDS BEFORE PACKET
000060                          HC.PSZ= 48.                     ;SIZE OF COMMAND AND MESSAGE PACKETS
000244                          HC.BSZ= 164.                    ;SIZE OF BUFFER
                                ;
000000                          HC.INT=  0.                     ;INTERRUPT INDICATOR WORDS START
000004                          HC.MSG= HC.INT+HC.ISZ           ;MESSAGE RING START
000006                          HC.MCT= HC.MSG+2.               ;MESSAGE RING CONTROL WORD
000010                          HC.CMD= HC.MSG+HC.RSZ           ;COMMAND RING START
000012                          HC.CCT= HC.CMD+2.               ;COMMAND RING CONTROL WORDS
000014                          HC.MEV= HC.CMD+HC.RSZ           ;MESSAGE ENVELOPE START
000020                          HC.MPK= HC.MEV+HC.ESZ           ;MESSAGE PACKET START
000100                          HC.CEV= HC.MPK+HC.PSZ           ;COMMAND ENVELOPE START
000104                          HC.CPK= HC.CEV+HC.ESZ           ;COMMAND PACKET START
000164                          HC.BF1= HC.CPK+HC.PSZ           ;FIRST BUFFER
000430                          HC.BF2= HC.BF1+HC.BSZ           ;SECOND BUFFER
                                ;
000674                          HC.SIZ= HC.BF2+HC.BSZ           ;TOTAL SIZE OF HOST COMM AREA
                                ;
                                ;VIRTUAL CIRCUIT IDENTIFIERS
                                ;
000000                          MSCP=    0                      ;MSCP CIRCUIT
000001                          LOG=     1                      ;LOG CIRCUIT
177777                          DIAG=   -1                      ;DIAGNOSTIC CIRCUIT
001000                          DUP=    1000                    ;DIAGNOSTIC AND UTILITIES PROTOCOL
```

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

```
                                ;COMMAND PACKET OPCODES
                                
000001                          OP.ABO=   1                     ;ABORT COMMAND
000020                          OP.ACC=  20                     ;ACCESS COMMAND
000010                          OP.AVL=  10                     ;AVAILABLE COMMAND
000021                          OP.CCD=  21                     ;COMPARE CONTROLLER DATA COMMAND
000040                          OP.CMP=  40                     ;COMPARE HOST DATA COMMAND
000022                          OP.ERS=  22                     ;ERASE COMMAND
000023                          OP.FLU=  23                     ;FLUSH COMMAND
000002                          OP.GCS=   2                     ;GET COMMAND STATUS COMMAND
000003                          OP.GUS=   3                     ;GET UNIT STATUS COMMAND
000011                          OP.ONL=  11                     ;ONLINE COMMAND
000041                          OP.RD=   41                     ;READ COMMAND
000024                          OP.RPL=  24                     ;REPLACE COMMAND
000004                          OP.SCC=   4                     ;SET CONTROLLER CHARACTERISTICS COMMAND
000012                          OP.SUC=  12                     ;SET UNIT CHARACTERISTICS COMMAND
000042                          OP.WR=   42                     ;WRITE COMMAND
000030                          OP.MRD=  30                     ;MAINTENANCE READ COMMAND
000031                          OP.MWR=  31                     ;MAINTENANCE WRITE COMMAND
000200                          OP.END= 200                     ;END PACKET FLAG
000007                          OP.SEX=   7                     ;SERIOUS EXCEPTION END PACKET
000100                          OP.AVA= 100                     ;AVAILABLE ATTENTION MESSAGE
000101                          OP.DUP= 101                     ;DUPLICATE UNIT NUMBER ATTENTION MESSAGE
000102                          OP.SHC= 102                     ;SHADOW COPY COMPLETE ATTENION MESSAGE
000103                          OP.RLC= 103                     ;RESET COMMAND LIMIT ATTENTION MESSAGE

000001                          OP.GDS=   1                     ;DUP GET DUST STATUS
000001                          OP.GSS=   1                     ;DUP GET DUST STATUS
000002                          OP.ESP=   2                     ;DUP EXECUTE SUPPLIED PROGRAM
000003                          OP.ELP=   3                     ;DUP EXECUTE LOCAL PROGRAM
000004                          OP.SSD=   4                     ;DUP SEND STUD DATA
000005                          OP.RSD=   5                     ;DUP RECEIVE STUD DATA

                                ;NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
                                ;PACKET FLAG TO THE COMMAND OPCODE. FOE EXAMPLE, A READ COMMAND'S END PACKET
                                ;CONTAINS THE VALUE OP.RD+OP.END IN ITS OPCODE FIELD. THE INVALID COMMAND END
                                ;PACKET CONTAINS JUST THE END PACKET FLAG (I.E., OP.END) IN ITS OPCODE FIELD.
                                ;THE SERIOUS EXCEPTION END PACKET CONTAINS THE SUM OF THE END PACKET FLAG
                                ;PLUS THE SERIOUS EXCEPTION OPCODE SHOWN ABOVE (I.E., OP.SEX+OP.END) IN ITS
                                ;OPCODE FIELD.
                                ;
                                ;COMMAND OPCODE BITS 3 THROUGH 5 INDICATE THE COMMAND CLASS, WHICH IS ENCODED
                                ;AS FOLLOWS:
                                ;    000  IMMEDIATE COMMANDS
                                ;    001  SEQUENTIAL COMMANDS
                                ;    010  NON-SEQUENTIAL COMMANDS THAT DO NOT INCLUDE A BUFFER DESCRIPTOR
                                ;    100  NON-SEQUENTIAL COMMANDS THAT DO INCLUDE A BUFFER DESCRIPTOR
```

```
                                ;COMMAND MODIFIERS
                                ;       = 020000                ;CLEAR SERIOUS EXCEPTION
040000                          MD.CMP= 040000                  ;COMPARE
100000                          MD.EXP= 100000                  ;EXPRESS REQUEST
010000                          MD.ERR= 010000                  ;FORCE ERROR
004000                          MD.SCH= 004000                  ;SUPPRESS CACHING (HIGH SPEED)
002000                          MD.SCL= 002000                  ;SUPPRESS CACHING (LOW SPEED)
000100                          MD.SEC= 000100                  ;SUPPRESS ERROR CORRECTION
000400                          MD.SER= 000400                  ;SUPPRESS ERROR RECOVERY
000200                          MD.SSH= 000200                  ;SUPPRESS SHADOWING
000100                          MD.WBN= 000100                  ;WRITE-BACK (NON-VOLATILE)
000400                          MD.WBV= 000400                  ;WRITE BACK (VOLATILE)
000020                          MD.SEQ= 000020                  ;WRITE SHADOW SET ONE UNIT AT A TIME
000001                          MD.SPD= 000001                  ;SPIN-DOWN
000001                          MD.FEU= 000001                  ;FLUSH ENTIRE UNIT
000002                          MD.VOL= 000002                  ;VOLATILE ONLY
000001                          MD.NXU= 000001                  ;NEXT UNIT
000001                          MD.RIP= 000001                  ;ALLOW SELF DESTRUCTION
000002                          MD.IMF= 000002                  ;IGNORE MEDIA FORMAT ERROR
000004                          MD.SWP= 000004                  ;SET WRITE PROTECT
000010                          MD.CWB= 000010                  ;CLEAR WRITE-BACK DATA LOST
000001                          MD.PRI= 000001                  ;PRIMARY REPLACEMENT BLOCK

                                ;END PACKET FLAGS

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

                                ;CONTROLLER FLAGS

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

```
                        ;UNIT FLAGS
                        ;
000001                  UF.CMR= 000001          ;COMPARE READS
000002                  UF.CMW= 000002          ;COMPARE WRITES
100000                  UF.RPL= 100000          ;HOST INITIATED BAD BLOCK REPLACEMENT
040000                  UF.INA= 040000          ;INACTIVE SHADOW SET UNIT
004000                  UF.SCH= 004000          ;SUPPRESS CACHING (HIGH SPEED)
002000                  UF.SCL= 002000          ;SUPPRESS CACHING (LOW SPEED)
000100                  UF.WBN= 000100          ;WRITE-BACK (NON-VOLATILE)
020000                  UF.WPH= 020000          ;WRITE PROTECT (HARDWARE)
001000                  UF.WPS= 001000          ;WRITE PROTECT (SOFTWARE OR VOLUME)
000004                  UF.576= 000004          ;576 BYTE SECTORS
                        ;
                        ;COMMAND PACKET OFFSETS
                        ;
                        ;                       GENERIC COMMAND PACKET OFFSETS:
000000                  P.CRF=   0.             ;COMMAND REFERENCE NUMBER
000004                  P.UNIT=  4.             ;UNIT NUMBER
000010                  P.OPCD=  8.             ;OPCODE
000012                  P.MOD=  10.             ;MODIFIERS
000014                  P.BCNT= 12.             ;BYTE COUNT
000020                  P.BUFF= 16.             ;BUFFER DESCRIPTOR
000020                  P.UADR= 16.             ;UNIBUS ADDRESS OF BUFFER DESCRIPTOR
000034                  P.LBN=  28.             ;LOGICAL BLOCK NUMBER
                        ;
                        ;                       ABORT AND GET COMMAND STATUS COMMAND PACKET OFFSETS:
000014                  P.OTRF= 12.             ;OUTSTANDING REFERENCE NUMBER
                        ;
                        ;                       ONLINE AND SET UNIT CHARACTERISTICS COMMAND PACKET OFFSETS:
000016                  P.UNFL= 14.             ;UNIT FLAGS
000020                  P.HSTI= 16.             ;HOST IDENTIFIER / RESERVED
000034                  P.ELGF= 28.             ;ERROR LOG FLAGS
000040                  P.SHUN= 32.             ;SHADOW UNIT
000042                  P.CPSP= 34.             ;COPY SPEED
                        ;                       REPLACE COMMAND PACKET OFFSETS:
000014                  P.RBN=  12.             ;REPLACEMENT BLOCK NUMBER
                        ;
                        ;                       SET CONTROLLER CHARACTERISTICS COMMAND PACKET OFFSETS:
000014                  P.VRSN= 12.             ;MSCP VERSION
000016                  P.CNTF= 14.             ;CONTROLLER FLAGS
000020                  P.HTMO= 16.             ;HOST TIMEOUT
000022                  P.USEF= 18.             ;USE FRACTION
000024                  P.TIME= 20.             ;QUAD-WORD TIME AND DATE
                        ;
                        ;                       MAINTENANCE READ AND MAINTENA CE WRITE COMMAND PACKET OFFSETS:
000034                  P.RGID= 28.             ;REGION ID
000040                  P.RGOF= 32.             ;REGION OFFSET
                        ;
                        ;                       EXECUTE SUPPLIED PROGRAM COMMAND PACKET OFFSETS:
000024                  P.DMDT= 20.             ;DMDT TERMINAL ADDRESS (MAINT WRITE ONLY)
000034                  P.OVRL= 28.             ;BUFFER DESCRIPTOR FOR OPERLAYS
```

```
                        ;END PACKET OFFSETS
                        ;
                                GENERIC END PACKET OFFSETS:
000000                  P.CRF=   0.             ;COMMAND REFERENCE NUMBER
000004                  P.UNIT=  4.             ;UNIT NUMBER
000010                  P.OPCD=  8.             ;OPCODE (ALSO CALLED ENDCODE)
000011                  P.FLGS=  9.             ;END PACKET FLAGS
000012                  P.STS=   10.            ;STATUS
000014                  P.BCNT=  12.            ;BYTE COUNT
000034                  P.FBBK=  28.            ;FIRST BAD BLOCK
                        ;
                                GET COMMAND STATUS END PACKET OFFSETS:
000014                  P.OTRF=  12.            ;OUTSTANDING REFERENCE NUMBER
000020                  P.CMST=  16.            ;COMMAND STATUS
                        ;
                                GET UNIT STATUS END PACKET OFFSETS:
000014                  P.MLUN=  12.            ;MULTI-UNIT CODE
000016                  P.UNFL=  14.            ;UNIT FLAGS
000020                  P.HSTI=  16.            ;HOST IDENTIFIER
000024                  P.UNTI=  20.            ;UNIT IDENTIFIER
000034                  P.MEDI=  28.            ;MEDIA TYPE IDENTIFIER
000040                  P.SHUN=  32.            ;SHADOW UNIT
000042                  P.SHST=  34.            ;SHADOW STATUS
000044                  P.TRCK=  36.            ;TRACK SIZE
000046                  P.GRP=   38.            ;GROUP SIZE
000050                  P.CYL=   40.            ;CYLINDER SIZE
000054                  P.RCTS=  44.            ;RCT TABLE SIZE
000056                  P.RBNS=  46.            ;RBNS / TRACK
000057                  P.RCTC=  47.            ;RCT COPIES
                        ;
                                ONLINE AND SET UNIT CHARACTERISTICS END PACKET AND AVAILABLE
                                ATTENTION MESSAGE OFFSETS:
000014                  P.MLUN=  12.            ;MULTI-UNIT CODE
000016                  P.UNFL=  14.            ;UNIT FLAGS
000020                  P.HSTI=  16.            ;HOST IDENTIFIER
000024                  P.UNTI=  20.            ;UNIT IDENTIFIER
000034                  P.MEDI=  28.            ;MEDIA TYPE IDENTIFIER
000040                  P.SHUN=  32.            ;SHADOW UNIT
000042                  P.SHST=  34.            ;SHADOW STATUS
000044                  P.UNCL=  36.            ;UNIT COMMAND LIMIT
000050                  P.UNSZ=  40.            ;UNIT SIZE
000054                  P.VSER=  44.            ;VOLUME SERIAL NUMBER
                        ;
                                SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS:
000014                  P.VRSN=  12.            ;MSCP VERSION
000016                  P.CNTF=  14.            ;CONTROLLER FLAGS
000020                  P.CTMO=  16.            ;CONTROLLER TIMEOUT
000022                  P.CNCL=  18.            ;CONTROLLER COMMAND LIMIT
000024                  P.CNTI=  20.            ;CONTROLLER ID
                        ;
                                GET DUST STATUS END PACKET OFFSETS:
000014                  P.DEXT=  12.            ;DUST PROGRAM EXTENSION
000017                  P.DFLG=  15.            ;STATUS FLAGS
000020                  P.DPI=   16.            ;PROGRESS INDICATOR
000024                  P.DTO=   20.            ;TIMEOUT VALUE
```

```
                        ;STATUS AND EVENT CODE DEFINITIONS

000037                  ST.MSK= 37              ;STATUS / EVENT CODE MASK
000040                  ST.SUB= 40              ;SUB-CODE MULTIPLIER
000000                  ST.SUC=  0              ;SUCCESS
000001                  ST.CMD=  1              ;INVALID COMMAND
000002                  ST.ABO=  2              ;COMMAND ABORTED
000003                  ST.OFL=  3              ;UNIT-OFFLINE
000004                  ST.AVL=  4              ;UNIT-AVAILABLE
000005                  ST.MFE=  5              ;MEDIA FORMAT ERROR
000006                  ST.WPR=  6              ;WRITE PROTECTED
000007                  ST.CMP=  7              ;COMPARE ERROR
000010                  ST.DAT= 10              ;DATA ERROR
000011                  ST.HST= 11              ;HOST BUFFER ACCESS ERROR
000012                  ST.CNT= 12              ;CONTROLLER ERROR
000013                  ST.DRV= 13              ;DRIVE ERROR
000037                  ST.DIA= 37              ;MESSAGE FROM AN INTERNAL DIAGNOSTIC

                        ;GET DUST STATUS FLAGS

000010                  DF.ACT= 010             ;SET IF THIS DUST CURRENTLY ACTIVE
000004                  DF.NES= 004             ;SET IF THIS DUST WILL NOT ACCEPT THE EXECUTE
                                                ;SUPPLIED PROGRAM COMMAND
000002                  DF.LCL= 002             ;SET IF THIS DUST HAS A LOCAL LOAD MEDIA FOR LOADING
                                                ;DIAGNOSTICS AND OTHER UTILITIES
000001                  DF.SA=  001             ;SET IF ANY PROGRAM EXECUTION UNDER THIS DUST
                                                ;DISABLES THE OPERATION OF ALL OTHER SERVERS IN THE
                                                ;SAME SYSTEM AS THE DUST

                        ;DUP MESSAGE TYPES

010000                  DU.QUE = 10000          ;QUESTION
020000                  DU.DFL = 20000          ;DEFAULT QUESTION
030000                  DU.INF = 30000          ;INFORMATION
040000                  DU.TER = 40000          ;TERMINATOR
050000                  DU.FTL = 50000          ;FATAL ERROR
060000                  DU.SPC = 60000          ;SPECIAL

170000                  DU.TYP= 170000          ;MESSAGE TYPE FIELD

                        ;DM PROGRAM HEADER DEFINITIONS

000000                  DMTRLN= 0               ;OFFSET TO SIZE OF PROGRAM NEEDING DOWNLINE LOAD
000004                  DMOVRL= 4               ;OFFSET TO SIZE OF OVERLAY
000021                  DMTMO=  21              ;TIMEOUT VALUE IN SECONDS (ONE BYTE)
000040                  DMMAIN= 40              ;OFFSET TO FIRST WORD OF MAIN PROGRAM
001000                  DMFRST= 1000            ;ADDRESS IN DM FILE CONTAINING FIRST BYTE OF HEADER
```

```
                                ;CONTROLLER TABLE DEFINITIONS
                                ;
                                ;ONE TABLE WILL BE SET UP BY INITIALIZE SECTION FOR EACH UDA SELECTED
                                ;FOR TESTING.  TABLES ARE CONTIGUOUS.  THE END OF THE TABLES IS
                                ;MARKED BY A WORD OF ZEROS.
                                ;
                                ;THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF CTABS.
                                ;THE NUMBER OF TABLES IS CONTAINED IN CTRLRS.

002140                          TABLE                                   ;START A TABLE DEFINITION

002140                          ITEM C.UADR     2                       ;UNIBUS ADDRESS OF UDAIP REGISTER
002140                          ITEM C.UNIT     2
        000077                          CT.UNT= 000077                  ; LOGICAL UNIT NUMBER (FIRST)
        100000                          CT.AVL= BIT15                   ; SET WHEN NOT AVAILABLE FOR TESTING
002140                          ITEM C.VEC      2
        000777                          CT.VEC= 000777                  ; VECTOR ADDRESS
        007000                          CT.BRL= 007000                  ; BR LEVEL
002140                          ITEM C.JSR      2                       ;INTERRUPT SERVICE ROUTINE FOR CONTROLLER
002140                          ITEM C.JAD      2                       ;  THESE TWO WORDS LOADED WITH [JSR R0,UDASRV]
002140                          ITEM C.FLG      2                       ;FLAGS
        000002                          CT.RN=  BIT1                    ;DM PROGRAM RUNNING
        000004                          CT.CMD= BIT2                    ;COMMAND ISSUED, WAITING FOR RESPONSE
        000010                          CT.MSG= BIT3                    ;MESSAGE RESPONSE RECEIVED
                                                                        ;WHENEVER THIS BIT IS SET, CT.CMD IS CLEARED
        000020                          CT.REQ= BIT4                    ;BUFFER HAS BEEN GIVEN TO UDA FOR REQUEST
                                                                        ;SET WHENEVER READ STUD DATA COMMAND
                                                                        ;GIVEN TO UDA
        000040                          CT.STA= BIT5                    ;GET DUST STATUS COMMAND HAS BEEN SENT
        000100                          CT.TM1= BIT6                    ;ONE TIMEOUT PERIOD HAS EXPIRED BETWEEN SEND OR
                                                                        ;RECEIVE DATA RESPONSE
        000200                          CT.TM2= BIT7                    ;SECOND TIMEOUT HAS EXPIRED
002140                          ITEM C.RING     2                       ;RING BUFFER ADDRESS
002140                          ITEM C.DR0      2                       ;POINTER TO DRIVE TABLES
002140                          ITEM C.DR1      2                       ;  IF ZERO, NO DRIVE TABLE EXISTS
002140                          ITEM C.DR2      2
002140                          ITEM C.DR3      2
002140                          ITEM C.DR4      2
002140                          ITEM C.DR5      2
002140                          ITEM C.DR6      2
002140                          ITEM C.DR7      2
002140                          ITEM C.TO       2                       ;TIMEOUT COUNTER
002140                          ITEM C.TOH      2                       ;  (TWO WORDS)
002140                          ITEM C.TOT      2                       ;DUP PROGRAM TIMEOUT VALUE IN SECONDS
002140                          ITEM C.PRI      4                       ;DUP PROGRAM PROGRESS INDICATOR
002140                          ITEM C.REF      2                       ;COMMAND REFERENCE NUMBER

002140                          END C.SIZE                              ;SIZE OF CONTROLLER TABLE IN BYTES
```

```
                                ;DRIVE TABLE DEFINITIONS
                                ;
                                ;ONE DRIVE TABLE WILL BE SET UP BY THE INITIALIZE SECTION FOR EACH
                                ;DRIVE SELECTED FOR TESTING.  EACH TABLE IS POINTED TO BY A
                                ;WORD IN THE CONTROLLER TABLE ON WHICH THE DRIVE EXISTS.

 002140                         TABLE                           ;START A TABLE DEFINITION

 002140                         ITEM D.DRV      2                       ;DRIVE NUMBER
 002140                         ITEM D.UNIT     2
        000077                          DT.UNT= 000077                  ; LOGICAL UNIT NUMBER OF DRIVE
        100000                          DT.AVL= BIT15                   ; SET WHEN NOT AVAILABLE FOR TESTING
 002140                         ITEM D.SERN     22.                     ;DISK SERIAL NUMBER

 002140                         END D.SIZE                              ;SIZE OF DRIVE TABLE IN BYTES
```

M4


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

```
                        ;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
                                                                                MOVB ARG1,R0
                                        .NLIST
                                .ENDC
                                .LIST
                                                                                CALL CPNT
                                .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
                                                                                MOV AA,-(SP)
                                                .NLIST
                                                PNT.CT=PNT.CT+2
                                        .ENDC
                                .ENDM
                                .LIST
                                                                                JSR R1,RTN
                                                                                .WORD ADR
                                                                                .WORD PNT.CT
                                .NLIST
                        .ENDM
```

```
                        ;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
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

002140                          FFREE:: .BLKW 1                         ;FIRST FREE WORD IN MEMORY
002142                          FSIZE:: .BLKW 1                         ;SIZE OF FREE MEMORY IN WORDS
002144                          FMEM:   .BLKW 1                 ;COPY OF FFREE AT END OF INIT SECTION
002146                          FMEMS:  .BLKW 1                 ;COPY OF FSIZE AT END OF INIT SECTION
002150                          CTABS:: .BLKW 1                         ;START OF CONTROLLER TABLE STORAGE
002152                          CTRLRS: .BLKW 1                         ;COUNT OF UDA CONTROLLERS IN PTABLES
002154                          TSTTAB: .BLKW 1                         ;POINTER TO FIRST CONTROLLER TABLE UNDER TEST
                                        .GLOBL RAFMT
002156  000000G                 DMPROG: .WORD RAFMT                     ;START ADDRESS OF DM PROGRAM
002160                          URUN:   .BLKW 1                         ;NUMBER OF UNITS TO RUN AT ONE TIME
002162                          URNING: .BLKW 1                         ;NUMBER OF UNITS STILL RUNNING
002164                          UCNT:   .BLKW 1                         ;COUNTER OF UNITS UNDER TEST
002166  000000                  FILOPN: .WORD 0                         ; FILE OPEN
002170                          UFREEZ: .BLKW 1                         ;FREEZE ON UNIT WHEN NOT ZERO
002172                          NXMAD:  .BLKW 1                         ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
002174  000000                  FDATA:  .WORD 0
002176                          FCTBUF: .BLKB 512.              ;STORAGE FOR FCT BLOCK
003176                          FCTNUM: .BLKW 1         ;FCT BLOCK NUMBER
003200                          MODE:   .BLKW 1 ;MODE WORD, SAME BIT DEFS AS S0.BIT

                                ;INIT ROUTINE DATA

003202                          DTABS:: .BLKW 1                         ;START OF DRIVE TABLE STORAGE
003204                          IFLAGS::.BLKW 1                         ;FLAGS FROM INIT CODE

                                ;CLOCK CONTROL

003206  000000                  KW.CSR: .WORD 0                         ;CSR OF CLOCK
003210                          KW.BRL: .BLKW 1                         ;BR LEVEL
003212                          KW.VEC: .BLKW 1                         ;VECTOR
003214                          KW.HZ:  .BLKW 1                         ;HERTZ (50. OR 60.)
003216                          KW.EL:  .BLKW 2                         ;ELAPSED TIME

003222  016540                  PTYPE:  .WORD PF                        ;PRINT TYPE
003224    000       000         ERRCHR: .BYTE 0,0                       ;FIRST BYTE LOADED WITH OUTPUT CHARACTER
003226  000000                  NULL:   .WORD 0                         ;USED TO PRINT A NULL CHARACTER
003230                          FNAME:  .BLKB 10.
```

```
                                        TEMP:   .BLKB 22.                       ;USED TO GET ANSWER FROM GMANID CALL
                061     055     112     DATEI:  .ASCIZ\1-JAN-70\                ;DEFAULT DATE
                                                .BLKB 3
                000000                  DATEO:  .WORD 0 ;DATE STRING IN FORMATTER FORMAT
                                                .BLKB 10.       ;(FIRST WORD ZERO SAYS NO DATE HERE YET)
                061     070     064     HIGHEST:.ASCIZ\18446744073709551615\    ;HIGHEST DISK SERIAL NUMBER
                104     105     103     MONTHS: .ASCII\DEC\     ;NAME OF MONTHS
                116     117     126             .ASCII\NOV\
                117     103     124             .ASCII\OCT\
                123     105     120             .ASCII\SEP\
                101     125     107             .ASCII\AUG\
                112     125     114             .ASCII\JUL\
                112     125     116             .ASCII\JUN\
                115     101     131             .ASCII\MAY\
                101     120     122             .ASCII\APR\
                115     101     122             .ASCII\MAR\
                106     105     102             .ASCII\FEB\
                112     101     116             .ASCII\JAN\
                037                     DAYS:   .BYTE 31.       ;NUMBER OF DAYS IN EACH MONTH
                035                             .BYTE 29.
                037                             .BYTE 31.
                036                             .BYTE 30.
                037                             .BYTE 31.
                036                             .BYTE 30.
                037                             .BYTE 31.
                037                             .BYTE 31.
                036                             .BYTE 30.
                037                             .BYTE 31.
                036                             .BYTE 30.
                037                             .BYTE 31.
                061     071     000     YEAR19: .ASCIZ\19\
                062     060     000     YEAR20: .ASCIZ\20\

                                                .EVEN
000000                          IPADRS: .WORD 0
000000                                  .WORD 0
000000                                  .WORD 0
000000                                  .WORD 0
000000                                  .WORD 0
000000                                  .WORD 0
000000                                  .WORD 0
```

```
      1                                 .SBTTL  GLOBAL TEXT SECTION
      2
      3                                 ;++
      4                                 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
      5                                 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
      6                                 ; MORE THAN ONE TEST.
      7                                 ;--
      8
      9                                 ;
     10                                 ; NAMES OF DEVICES SUPPORTED BY PROGRAM
     11                                 ;
     12 003454                                 DEVTYP <RA SERIES DISK DRIVE>
        003454                                                                  L$DVTYP::
        003454  122     101     040                                                     .ASCIZ  /RA SERIES DISK DRIVE
                                                                                        .EVEN
     13
     14                                 ; TEST DESCRIPTION
     15                                 ;
     16 003502                                 DESCRIPT <CZUDKO UDA50A,KDA50A-Q FORMATTER>
        003502                                                                  L$DESC::
        003502  103     132     125                                                     .ASCIZ  /CZUDKO UDA50A,KDA50A
                                                                                        .EVEN
```

-Q FORMATTER

F5

GLOBAL TEXT SECTION

```
                                        ;UNFORMATTED MESSAGES
                                        
003544     105     116     124  DATEQ:  .ASCIZ\ENTER DATE AS DD-MMM-YY\
003574     040     106     117  FILNAQ: .ASCIZ\ FOR DISK TO BE FORMATTED\
003626     040     000          SERNQ:  .ASCIZ\ \
003630     101     122     105  WNQUES: .ASCIZ\ARE YOU SURE YOU WANT TO RUN THIS FORMATTER\
```

```
                                        ; FORMAT STATEMENTS USED IN PRINT CALLS

003704      045     124     000 ERRONE: .ASCIZ\%T\
003707      045     116     000 ERRNL:  .ASCIZ\%N\
003712      042     040     040 RNTIM:  .ASCIZ\"  RUNTIME "D16":"\
003735      104     071     042 RNTIM1: .ASCIZ\D9":"\
003743      104     071     000 RNTIM2: .ASCIZ\D9\
003746      042     040     040 ERRME1: .ASCIZ\"  *  *  *  ERROR PROCESSING MESSAGE STRING  *  *  *"N\
004035      116     042     125 MESSG:  .ASCIZ\N"UNIT "D6" CONTROLLER AT "O16" DRIVE "D9S\
004110      042     116     117 NOCLOCK:.ASCIZ\"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\
004165      042     110     117 BASNO:  .ASCIZ\"HOST PROGRAM"\
004204      042     040     040 BASL2:  .ASCIZ\"  CONTROLLER AT "O16\
004232      042     040     040 BASL3:  .ASCIZ\"  DRIVE "D9\
004247      000                 BAS:    .BYTE 0                         ;NULL TO PRINT NOTHING

004250      122     066     122 BASLN:  .ASCIZ\R6R6R6R6\                ;USED TO PRINT BASIC LINE OF ERROR MESSAGE
004261      116     042     123 SERNUM: .ASCIZ\N"SERIAL NUMBER FOR UNIT "D6" CONTROLLER AT "O16" DRIVE "D9\
004355      042     123     124 WNSTOP: .ASCII\"STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK"N\
004450      042     125     116         .ASCII\"UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN"N\
004541      042     102     122         .ASCIZ\"BROUGHT ONLINE."NN\
004565      116     042     127 WNSTRT: .ASCII\N"WARNING:"N\
004601      042     040     040         .ASCII\"      THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC"N\
004703      042     040     040         .ASCII\"      TOOL.  RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK'N\
005002      042     040     040         .ASCIZ\"      DRIVE'S SERVICE MANUAL."N\
005043      116     042     127 WNTIME: .ASCII\N"WARNING:"N\
005057      042     040     040         .ASCII\"      THIS PROGRAM WILL TAKE APPROXIMATELY 45 MINUTES ON"N\
005152      042     040     040         .ASCII\"      A RA60, 30 MINUTES ON A RA80, 60 MINUTES ON A RA81, AND "N\
005253      042     040     040         .ASCIZ\"      120 MINUTES ON A RA82."N\
```

```
      1 005313                          X1A:
      2 005313                          X2A:
      3 005313                          X3A:
      4 005313    042     111     116   X8A:    .ASCIZ\"INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS"N\
      5 005372    122     065     122   X1:     .ASCIZ\R5R6"CONTROLLER HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE"N\
      6 005475    122     065     122   X2:     .ASCIZ\R5R6"MULTIPLE UNITS SELECT THE SAME DRIVE"N\
      7 005551    122     065     122   X3:     .ASCIZ\R5R6"MORE THAN EIGHT DRIVES SELECTED ON THIS CONTROLLER"N\
      8 005643    122     064     042   X4:     .ASCII\R4"NOT ENOUGH ROOM IN MEMORY TO FORMAT THE UNITS SELECTED"N\
      9 005736    042     120     114           .ASCIZ\"PLEASE START PROGRAM OVER AND FORMAT FEWER UNITS AT A TIME"N\
     10 006034    122     065     122   X8:     .ASCIZ\R5R6"TWO CONTROLLERS USE THE SAME VECTOR"N\
     11 006107    122     064     042   X9:     .ASCII\R4"ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE."N\
     12 006212    042     120     114           .ASCIZ\"PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK."N\
     13 006301    122     064     042   X10:    .ASCIZ\R4"THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE."N\
     14 006400    122     065     042   X14:    .ASCII\R5"CONTROLLER IS NOT SUPPORTED BY THIS FORMATTER PROGRAM.  THIS"N\
     15 006501    042     120     122           .ASCII\"PROGRAM REQUIRES A UDA50-A (MODEL 6) OR A KDA50-Q (MODEL 13)"N\
     16 006600    042     103     117           .ASCIZ\"CONTROLLER.  CONTROLLER REPORTED MODEL CODE "D4" "N\
     17 006665    122     065     042   X20:    .ASCII\R5"MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS"N\
     18 006752    042     103     110           .ASCII\"CHECK CSR SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE OR BUS"N\
     19 007057    042     117     122           .ASCIZ\"OR REPLACE CONTROLLER PROCESSOR MODULE"N\
     20 007131    122     065     042   X21:    .ASCII\R5"CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE"NR8\
     21 007220    042     122     105           .ASCIZ\"REPLACE CONTROLLER SDI MODULE"N\
     22 007261    122     065     042   X21A:   .ASCIZ\R5"CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE"NR8R7\
     23 007353    122     065     042   X22:    .ASCII\R5"STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION"N\
     24 007451    042     123     124           .ASCIZ\"STEP BIT EXPECTED "O16NR8R7\
N"   25 007506    122     065     042   X23A:   .ASCII\R5"CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATIO
     26 007627    104     071     042           .ASCII\D9" WORDS WERE TO BE CLEARED STARTING AT ADDRESS "O16N\
     27 007715    042     106     111           .ASCII\"FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):"N\
     28 007772    123     066     042           .ASCIZ\S6"ADDRESS"S4"CONTENTS"N\
     29 010023    123     067     117   X23B:   .ASCIZ\S7016S5016N\
     30 010037    122     065     042   X24:    .ASCII\R5"SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION"N\
     31 010147    042     120     125           .ASCIZ\"PURGE/POLE DIAGNOSTICS WERE REQUESTED"NR8R7\
     32 010224    122     065     042   X25:    .ASCII\R5"CONTROLLER DID NOT RETURN CORRECT DATA IN SA REGISTER DURING"N\
     33 010325    042     111     116           .ASCII\"INITIALIZATION"N\
     34 010346    042     040     040           .ASCIZ\"  SA EXPECTED  "O16NR8R7\
     35 010400    122     065     042   X30:    .ASCIZ\R5"CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE RUNNING FORMATTER"NR8\
     36 010516    122     065     042   X31:    .ASCIZ\R5"FORMATTER PROGRAM IS HUNG"N\
     37 010555    122     065     042   X32:    .ASCIZ\R5"MESSAGE BUFFER RECEIVED FROM FORMATTER WITH UNKNOWN REQUEST NUMBER"N\
     38 010665    122     065     042   X36:    .ASCII\R5"NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS"N\
     39 010756    042     127     110           .ASCIZ\"WHILE LOADING FORMATTER"N\
7\   40 011011    122     065     042   X37:    .ASCIZ\R5"CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE LOADING FORMATTER'NR8R
     41 011131    122     065     042   X100:   .ASCIZ\R5"FORMATTER ASKED UNEXPECTED QUESTION ("D12")"N\
     42 011212    122     065     042   X101:   .ASCIZ\R5"FORMATTER REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION"N\
```

GLOBAL TEXT SECTION

```
 1 011313  042  115  105  XMSG1:  .ASCIZ\"MESSAGE BUFFER CONTAINS:"N\
 2 011347  123  063  117  XMSG2:  .ASCIZ\S3016S1016S1016S1016S1016S1016S1016N\
 3 011414  122  065  042  XPKT1:  .ASCII\R5"RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA"N\
 4 011517  042  105  111          .ASCII\"EITHER CONTROLLER RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED"N\
 5 011624  042  103  117          .ASCII\"CORRECTLY"N\
 6 011640  123  063  042          .ASCIZ\S3"COMMAND PACKET SENT"S6"RESPONSE PACKET RECEIVED"N\
 7 011725  123  066  117  XPKT2:  .ASCIZ\S6016S1016S14016S1016N\
 8 011754  042  040  040  XSA:    .ASCIZ\"  SA CONTAINS  "016N\
 9 012002  042  122  105  XFRU:   .ASCIZ\"REPLACE CONTROLLER PROCESSOR MODULE"N\
10
11
12 012051  045  101  111  SERNX:  .ASCIZ\%AINPUT ERROR. ANSWER WITH DECIMAL NUMBER LO= 0  HI= %T\
13 012141  042  111  116  DATEX:  .ASCIZ\"INPUT ERROR."\
14 012160  042  116  101  FILNAM: .ASCIZ\"NAME OF FILE CONTAINING BAD SECTOR INFORMATION"\
15                               .EVEN
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;++
                                ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
                                ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
                                ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
                                ;--
            177777              $VCINS= -1                  ; LIST INSTRUCTIONS, SHIFTED RIGHT
            177777              SVCTST= -1                  ; LIST TEST TAGS, SHIFTED RIGHT
            177777              SVCSUB= -1                  ; LIST SUBTEST TAGS, SHIFTED RIGHT
            177777              SVCGBL= -1                  ; LIST GLOBAL TAGS, SHIFTED RIGHT
            177777              SVCTAG= -1                  ; LIST OTHER TAGS, SHIFTED RIGHT

012242                          BGNMSG ERR001
012242                                  PNTB X1,#X1A
012242  012746  005313                                                  MOV #X1A,-(SP)
012246  004137  016672                                                  JSR R1,LPNTB
012252  005372                                                          .WORD X1
012254  000002                                                          .WORD PNT.CT
012256                          ENDMSG

012260                          BGNMSG ERR002
012260                                  PNTB X2,#X2A
012260  012746  005313                                                  MOV #X2A,-(SP)
012264  004137  016672                                                  JSR R1,LPNTB
012270  005475                                                          .WORD X2
012272  000002                                                          .WORD PNT.CT
012274                          ENDMSG

012276                          BGNMSG ERR003
012276                                  PNTB X3,#X3A
012276  012746  005313                                                  MOV #X3A,-(SP)
012302  004137  016672                                                  JSR R1,LPNTB
012306  005551                                                          .WORD X3
012310  000002                                                          .WORD PNT.CT
012312                          ENDMSG

012314                          BGNMSG ERR004
012314                                  PNTB X4
012314  004137  016672                                                  JSR R1,LPNTB
012320  005643                                                          .WORD X4
012322  000000                                                          .WORD PNT.CT
012324                          ENDMSG

012326                          BGNMSG ERR008
012326                                  PNTB X8,#X8A
012326  012746  005313                                                  MOV #X8A,-(SP)
012332  004137  016672                                                  JSR R1,LPNTB
012336  006034                                                          .WORD X8
012340  000002                                                          .WORD PNT.CT
012342                          ENDMSG

012344                          BGNMSG ERR009
012344                                  PNTB X9
012344  004137  016672                                                  JSR R1,LPNTB
012350  006107                                                          .WORD X9
012352  000000                                                          .WORD PNT.CT
```

```
 36 012354                              ENDMSG
 37
 38 012356                              BGNMSG ERR010
 39 012356                                      PNTB X10
    012356  004137  016672                                          JSR R1,LPNTB
    012362  006301                                                  .WORD X10
    012364  000000                                                  .WORD PNT.CT
 40 012366                              ENDMSG
 41
 42 012370                              BGNMSG ERR014
 43 012370                                      PNTB X14,R2
    012370  010246                                                  MOV R2,-(SP)
    012372  004137  016672                                          JSR R1,LPNTB
    012376  006400                                                  .WORD X14
    012400  000002                                                  .WORD PNT.CT
 44 012402                              ENDMSG
 45
 46 012404                              BGNMSG ERR020
 47 012404                                      PNTB X20
    012404  004137  016672                                          JSR R1,LPNTB
    012410  006665                                                  .WORD X20
    012412  000000                                                  .WOPD PNT.CT
 48 012414                              ENDMSG
 49
 50 012416                              BGNMSG ERR021
 51 012416  010201                              MOV R2,R1
 52 012420  000301                              SWAB R1
 53 012422                                      AND 2,R1
    012422  042701  177775                                          BIC #^C<2>,R1
 54 012426  001406                              BEQ ERR21A
 55 012430                                      PNTB X21,R2
    012430  010246                                                  MOV R2,-(SP)
    012432  004137  016672                                          JSR R1,LPNTB
    012436  007131                                                  .WORD X21
    012440  000002                                                  .WORD PNT.CT
 56 012442  000405                              BR EOFMSG
 57 012444                              ERR21A:
 58 012444                                      PNTB X21A,R2
    012444  010246                                                  MOV R2,-(SP)
    012446  004137  016672                                          JSR R1,LPNTB
    012452  007261                                                  .WORD X21A
    012454  000002                                                  .WORD PNT.CT
 59 012456                              EOFMSG:
 60 012456                              ENDMSG
 61
 62 012460                              BGNMSG ERR022
 63 012460  042737  100000  020626              BIC #SA.ERR,UDARSD
 64 012466                                      PNTB X22,UDARSD,R2
    012466  010246                                                  MOV R2,-(SP)
    012470  013746  020626                                          MOV UDARSD,-(SP)
    012474  004137  016672                                          JSR R1,LPNTB
    012500  007353                                                  .WORD X22
    012502  000004                                                  .WORD PNT.CT
 65 012504                              ENDMSG
 66
 67 012506                              BGNMSG ERR023
 68 012506                                      PNTB X23A,R3,R1
```

```
        012506  010146                                                  MOV R1,-(SP)
        012510  010346                                                  MOV R3,-(SP)
        012512  004137  016672                                          JSR R1,LPNTB
        012516  007506                                                  .WORD X23A
        012520  000004                                                  .WORD PNT.CT
    69  012522  005742                  TST -(R2)
    70  012524  005712          ERR23A: TST (R2)
    71  012526  001406                  BEQ ERR23B
    72  012530                          PNTB X23B,R2,(R2)
        012530  011246                                                  MOV (R2),-(SP)
        012532  010246                                                  MOV R2,-(SP)
        012534  004137  016672                                          JSR R1,LPNTB
        012540  010023                                                  .WORD X23B
        012542  000004                                                  .WORD PNT.CT
    73  012544  005722          ERR23B: TST (R2)+
    74  012546  005303                  DEC R3
    75  012550  001365                  BNE ERR23A
    76  012552                  ERR23C: PNTB XFRU
        012552  004137  016672                                          JSR R1,LPNTB
        012556  012002                                                  .WORD XFRU
        012560  000000                                                  .WORD PNT.CT
    77  012562                          ENDMSG
    78
    79  012564                          BGNMSG ERR024
    80  012564                                 PNTB X24,R2
        012564  010246                                                  MOV R2,-(SP)
        012566  004137  016672                                          JSR R1,LPNTB
        012572  010037                                                  .WORD X24
        012574  000002                                                  .WORD PNT.CT
    81  012576                          ENDMSG
    82
    83  012600                          BGNMSG ERR025
    84  012600                                 PNTB X25,R1,R2
        012600  010246                                                  MOV R2,-(SP)
        012602  010146                                                  MOV R1,-(SP)
        012604  004137  016672                                          JSR R1,LPNTB
        012610  010224                                                  .WORD X25
        012612  000004                                                  .WORD PNT.CT
    85  012614                          ENDMSG
    86
    87  012616                          BGNMSG ERR030
    88  012616                                 PNTB X30,R1
        012616  010146                                                  MOV R1,-(SP)
        012620  004137  016672                                          JSR R1,LPNTB
        012624  010400                                                  .WORD X30
        012626  000002                                                  .WORD PNT.CT
    89  012630                          ENDMSG
    90
    91  012632                          BGNMSG ERR031
    92  012632                                 PNTB X31
        012632  004137  016672                                          JSR R1,LPNTB
        012636  010516                                                  .WORD X31
        012640  000000                                                  .WORD PNT.CT
    93  012642                          ENDMSG
    94
    95  012644                          BGNMSG ERR032
    96  012644                                 PNTB X32
```

```
        012644  004137  016672                                                  JSR R1,LPNTB
        012650  010555                                                          .WORD X32
        012652  000000                                                          .WORD PNT.CT
     97 012654  004737  013044                  CALL MSGPKT
     98 012660                          ENDMSG
     99
    100 012662                          BGNMSG ERR033
    101 012662  004737  012752                  CALL PNTPKT
    102 012666                          ENDMSG
    103
    104 012670                          BGNMSG ERR034
    105 012670  004737  012752                  CALL PNTPKT
    106 012674                          ENDMSG
    107
    108 012676                          BGNMSG ERR036
    109 012676                                  PNTB X36
        012676  004137  016672                                                  JSR R1,LPNTB
        012702  010665                                                          .WORD X36
        012704  000000                                                          .WORD PNT.CT
    110 012706                          ENDMSG
    111
    112 012710                          BGNMSG ERR037
    113 012710                                  PNTB X37,R1
        012710  010146                                                          MOV R1,-(SP)
        012712  004137  016672                                                  JSR R1,LPNTB
        012716  011011                                                          .WORD X37
        012720  000002                                                          .WORD PNT.CT
    114 012722                          ENDMSG
    115
    116 012724                          BGNMSG ERR100
    117 012724                                  PNTB X100,(R4)
        012724  011446                                                          MOV (R4),-(SP)
        012726  004137  016672                                                  JSR R1,LPNTB
        012732  011131                                                          .WORD X100
        012734  000002                                                          .WORD PNT.CT
    118 012736                          ENDMSG
    119
    120 012740                          BGNMSG ERR101
    121 012740                                  PNTB X101
        012740  004137  016672                                                  JSR R1,LPNTB
        012744  011212                                                          .WORD X101
        012746  000000                                                          .WORD PNT.CT
    122 012750                          ENDMSG
    123
    124 012752                          PNTPKT: PNTB XPKT1
        012752  004137  016672                                                  JSR R1,LPNTB
        012756  011414                                                          .WORD XPKT1
        012760  000000                                                          .WORD PNT.CT
    125 012762  010401                          MOV R4,R1
    126 012764  062701  000104                  ADD #HC.CPK,R1
    127 012770  010402                          MOV R4,R2
    128 012772  062702  000020                  ADD #HC.MPK,R2
    129 012776  012703  000014                  MOV #12.,R3
    130 013002                          PNTPKL: PNTB XPKT2,2(R1),(R1),2(R2),(R2)
        013002  011246                                                          MOV (R2),-(SP)
        013004  016246  000002                                                  MOV 2(R2),-(SP)
        013010  011146                                                          MOV (R1),-(SP)
```

```
        013012  016146  000002                                                  MOV 2(R1),-(SP)
        013016  004137  016672                                                  JSR R1,LPNTB
        013022  011725                                                          .WORD XPKT2
        013024  000010                                                          .WORD PNT.CT
    131 013026  062701  000004          ADD #4,R1
    132 013032  062702  000004          ADD #4,R2
    133 013036  005303                  DEC R3
    134 013040  001360                  BNE PNTPKL
    135 013042  000207                  RETURN
    136
    137 013044                  MSGPKT: PNTB XMSG1
        013044  004137  016672                                                  JSR R1,LPNTB
        013050  011313                                                          .WORD XMSG1
        013052  000000                                                          .WORD PNT.CT
    138 013054  016504  000014          MOV C.RING(R5),R4
    139 013060  062704  000430          ADD #HC.BF2,R4
    140 013064  012703  000005          MOV #5,R3
    141 013070                  MSGPKL: PNTB XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)
        013070  016446  000014                                                  MOV 12.(R4),-(SP)
        013074  016446  000012                                                  MOV 10.(R4),-(SP)
        013100  016446  000010                                                  MOV 8.(R4),-(SP)
        013104  016446  000006                                                  MOV 6(R4),-(SP)
        013110  016446  000004                                                  MOV 4(R4),-(SP)
        013114  016446  000002                                                  MOV 2(R4),-(SP)
        013120  011446                                                          MOV (R4),-(SP)
        013122  004137  016672                                                  JSR R1,LPNTB
        013126  011347                                                          .WORD XMSG2
        013130  000016                                                          .WORD PNT.CT
    142 013132  062704  000016          ADD #14.,R4
    143 013136  005303                  DEC R3
    144 013140  001353                  BNE MSGPKL
    145 013142  000207                  RETURN
```

B6

GLOBAL ERROR REPORT SECTION

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

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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

013144                          FMERR:  ERRSF 4,,ERR004
013144  104454                                                         TRAP    C$ERSF
013146  000004                                                         .WORD   4
013150  000000                                                         .WORD   0
013152  012314                                                         .WORD   ERR004
013154                                  DOCLN                   ;ABORT
013154  104444                                                         TRAP    C$DCLN
```

```
                                ;ALOCM
                                ;
                                ;ALLOCATE A BLOCK OF FREE MEMORY.  REPORT ERROR IF MEMORY EXHAUSTED.
                                ;
                                ;INPUTS:
                                ;       R1 - NUMBER OF WORDS TO ALLOCATE
                                ;       FFREE - FIRST FREE WORD IN MEMORY
                                ;       FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS
                                ;OUTPUTS:
                                ;       R1 - ADDRESS OF FIRST WORD OF ALLOCATED MEMORY
                                ;       FFREE - NEW FIRST FREE WORD IN MEMORY
                                ;       FSIZE - SIZE OF FREE MEMORY LEFT AFTER ALLOCATION
                                ;SYSTEM FATAL ERROR WILL BE REPORTED IF NOT ENOUGH MEMORY AVAILABLE
                                ;AND ENTIRE PROGRAM WILL BE STOPPED.
                                
013156                          ALOCM:  PUSH FFREE                      ;SAVE FFREE AT ENTRY
013156  013746  002140                                                                MOV FFREE,-(SP)
013162  160137  002142                  SUB R1,FSIZE                    ;REDUCE SIZE OF FREE MEMORY
013166  002766                          BLT FMERR                       ;REPORT ERROR IF NOT ENOUGH MEMORY
013170  060101                          ADD R1,R1                       ;CHANGE WORDS TO BYTES
013172  060137  002140                  ADD R1,FFREE                    ;CALCULATE NEW START OF FREE MEMORY
013176                                  POP R1                          ;GET START OF ALLOCATED MEMORY
013176  01260'                                                                        MOV (SP)+,R1
013200  000207                          RETURN
```

```
                                        .HCOMM
                                        ;
                                        ;ALLOCATES MEMORY FOR HOST COMM AREA AND PACKET BUFFERS WITH ONE
                                        ;DESCRIPTOR IN EACH RING. TO BE CALLED WHEN INITIALIZING
                                        ;A CONTROLLER WITH SA.MSG=0 AND SA.CMD=0.
                                        ;
                                        ;INPUTS:
                                        ;       R5 - ADDRESS OF CONTROLLER TABLE
                                        ;OUTPUTS:
                                        ;       CONTROLLER TABLE POINTING TC HOST COMM AREA
                                        ;       R4 - ADDRESS OF HOST COMM AREA

013202  012701  000336          HCOMM:  MOV #HC.SIZ/2,R1                ;GET SIZE OF AREA TO ALLOCATE
013206  004737  013156                  CALL ALOCM                      ;ALLOCATE THE MEMORY
013212  010165  000014                  MOV R1,C.RING(R5)                       ;GET ADDRESS OF HOST COMM AREA
                                                                        ;PLACE IN CONTROLLER TABLE
013216  000207                          RETURN
```

```
                                ;RESET
                                ; RESET ALL UDA-50S IN THE CONTROLLER TABLES
                                ;
                                ; INPUTS:
                                ;       IPADRS - CONTAINS ALL IP ADDRESSES
                                ; OUTPUTS:
                                ;       NONE
                                ;
013220                          RESET:  PUSH <R3,R4>
013220  010346                                                                  MOV R3,-(SP)
013222  010446                                                                  MOV R4,-(SP)
013224  005037  002172                  CLR     NXMAD
013230                                  SETVEC #4,#NXMI,#PRI07
013230  012746  000340                                                          MOV     #PRI07,-(SP)
013234  012746  017602                                                          MOV     #NXMI,-(SP)
013240  012746  000004                                                          MOV     #4,-(SP)
013244  012746  000003                                                          MOV     #3,-(SP)
013250  104437                                                                  TRAP    C$SVEC
013252  062706  000010                                                          ADD     #10,SP
013256                                  BREAK
013256  104422                                                                  TRAP    C$BRK
013260  012703  000010                  MOV     #8.,R3          ; R3 = COUNTER OF ENTRIES
013264  012704  003434                  MOV     #IPADRS,R4      ; R4 -> IP ADDRESS
013270  005714                  1$:     TST     (R4)            ; IS THERE AN ENTRY?
013272  001406                          BEQ     2$              ; IF NOT, DONE
013274  005034                          CLR     @(R4)+          ; INIT UDA
013276  005737  002172                  TST     NXMAD           ; WAS THERE AN ERROR?
013302  001010                          BNE     3$              ; IF SO, EXIT
013304  005303                          DEC     R3              ; MAKE SURE WE DO NOT EXTEND OVER AREA
013306  001370                          BNE     1$              ; IF NOT DONE, BRANCH
013310                          2$:     CLRVEC  #4
013310  012700  000004                                                          MOV     #4,R0
013314  104436                                                                  TRAP    C$CVEC
013316                                  POP     <R4,R3>
013316  012604                                                                  MOV (SP)+,R4
013320  012603                                                                  MOV (SP)+,R3
013322  000207                          RETURN
                                ;
013324  005744                  3$:     TST     -(R4)           ; R4 -> UDAIP THAT FAILED
013326  010405                          MOV     R4,R5           ; SAVE IN R5 FOR REPORT
013330                                  ERRDF   20.,ERR020
013330  104455                                                                  TRAP    C$ERDF
013332  000024                                                                  .WORD   20
013334  000000                                                                  .WORD   0
013336  012404                                                                  .WORD   ERR020
013340  005014                          CLR     (R4)            ; DESTROY ENTRY SO NOT TO FALL INTO RESET ERROP LOOP
013342                                  DOCLN
013342  104444                                                                  TRAP    C$DCLN
```

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

013344  010137  002160          RUNDM:  MOV R1,URUN                     ;SAVE NUMBER OF UNITS TO RUN
013350  005037  002162                  CLR URNING                      ;CLEAR NUMBER OF UNITS RUNNING

                                        ;LOAD DM PROGRAM INTO EACH CONTROLLER

013354  013737  002160  002164          MOV URUN,UCNT                   ;SET COUNTER OF UNITS
013362  013705  002154                  MOV TSTTAB,R5                   ;GET FIRST CONTROLLER TABLE
013366                          LDDM:
013366  005065  000012                  CLR C.FLG(R5)                   ;CLEAR ALL FLAGS
013372  116537  000002  002074          MOVB C.UNIT(R5),L$LUN           ;SEE IF UNIT TO BE TESTED
013400  005765  000002                  TST C.UNIT(R5)
013404  100407                          BMI LDNEXT                      ;IF NOT, DON'T LOAD THIS UNIT
013406                                          ASSUME CT.AVL EQ BIT15
013406  004737  013202                  CALL HCOMM                      ;ALLOCATE SPACE FOR HOST COMM AREA
013412  004737  017024                  CALL LOADDM                     ;LOAD THE DM PROGRAM
013416  001402                          BEQ LDNEXT                      ;IF ERROR, GO TO NEXT CONTROLLER
013420  005237  002162                  INC URNING                      ;IF NO ERROR, COUNT UNIT RUNNING
013424  062705  000052          LDNEXT: ADD #C.SIZE,R5                  ;MOVE TO NEXT CONTROLLER TABLE
013430  005337  002164                  DEC UCNT                        ;CHECK IF MORE CONTROLLERS
013434  001354                          BNE LDDM                        ;LOAD NEXT
013436  005037  002170                  CLR UFREEZ                      ;CLEAR UNIT FREEZE FLAG
013442  012737  177777  003176          MOV #-1,FCTNUM  ;INVALIDATE FCT BLOCK NUMBER (BLOCK IN MEMORY)

                                        ;CHECK IF ANY CONTROLLERS LOADED

013450  005737  002162                  TST URNING                      ;ANY UNITS LOADED?

                                        ;THE DM PROGRAMS ARE NOW IN CONTROL
                                        ;RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS

013454  000207                          RETURN
```

```
                                ;CLOSEF
                                ;
                                ;CLOSE DATA FILE FOR DM PROGRAMS
                                ;
                                ;INPUTS:
                                ;       FILOPN - ZERO IF FILE NOT OPEN
                                ;OUTPUTS:
                                ;       NONE
                                ;
013456  005737  002166          CLOSEF: TST FILOPN              ;SEE IF FILE CURRENTLY OPEN
013462  001403                          BEQ 1$
013464                                  CLOSE                   ; IF SO, CLOSE IT
013464  104435                                                          TRAP    C$CLOS
013466  005037  002166                  CLR FILOPN              ;AND MARK AS SO
013472  000207                  1$:     RETURN
```

```
                                        ;RESPDM
                                        ;
                                        ;RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
                                        ;HAVE TERMINATED.

013474  013705  002154          RESPDM: MOV TSTTAB,R5                   ;GET CONTROLLER TABLE ADDRESS
013500  013737  002160  002164          MOV URUN,UCNT                   ;SET COUNTER OF UNITS
013506                        RESPCT: BREAK   ;ALLOW DRS TO SEE TERMINAL INPUT
013506  104422                                                                TRAP    C$BRK
013510  016504  000014                MOV C.RING(R5),R4               ;GET HOST COMM AREA ADDRESS
013514  032765  000002  000012        BIT #CT.RN,C.FLG(R5)            ;CHECK IF PROGRAM RUNNING
013522  001502                        BEQ RSPNXT                      ;IF NOT, LOOK AT NEXT
013524  116537  000002  002074        MOVB C.UNIT(R5),L$LUN           ;STORE UNIT NUMBER UNDER TEST
013532  032765  000010  000012        BIT #CT.MSG,C.FLG(R5)           ;SEE IF INTERRUPT RECEIVED
013540  001150                        BNE RSPIN                       ;IF SO, LOOK AT PACKET
013542  032765  000004  000012        BIT #CT.CMD,C.FLG(R5)           ;SEE IF COMMAND HAS BEEN SENT
013550  001002                        BNE 1$                          ;IF NOT, SEND ONE
013552  000137  014320                JMP RSPOUT

                                        ;CHECK IF UDA STILL RUNNING

013556  011503                1$:     MOV (R5),R3                     ;GET ADDRESS OF UDAIP
013560  016301  000002                MOV 2(R3),R1                    ;LOOK AT UDASA REGISTER
013564  001405                        BEQ RSPTM                       ;IF ZERO, UDA STILL RUNNING
013566                                ERRDF 30,,ERR030                ;REPORT UDA HAS FATAL ERROR
013566  104455                                                                TRAP    C$ERDF
013570  000036                                                                .WORD   30
013572  000000                                                                .WORD   0
013574  012616                                                                .WORD   ERR030
013576  000465                        BR RSPDRP                       ;DROP CONTROLLER FROM TESTING

                                        ;CHECK FOR TIMEOUT OF RESPONSE

013600  005765  000042          RSPTM:  TST C.TOT(R5)                   ;SEE IF DUP PROGRAM TO BE TIMED
013604  001451                        BEQ RSPNTO
013606  005737  003206                TST KW.CSR                      ;SEE IF A CLOCK ON SYSTEM
013612  001446                        BEQ RSPNTO                      ;DON'T TIME IF NO CLOCK
013614  023765  003220  000040        CMP KW.EL+2,C.TOH(R5)           ;COMPARE TO TIMEOUT COUNTER
013622  101005                        BHI RSPTMO
013624  001041                        BNE RSPNTO
013626  023765  003216  000036        CMP KW.EL,C.TO(R5)
013634  103435                        BLO RSPNTO                      ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
013636  032765  000040  000012  RSPTMO: BIT #CT.STA,C.FLG(R5)         ;SEE IF A GET DUST STATUS COMMAND OUTSTANDING
013644  001101                        BNE RSPTOE                      ;REPORT ERROR IF SO
013646  005764  000012                TST HC.CCT(R4)                  ;SEE IF UDA TOOK LAST COMMAND PACKET
013652  100476                        BMI RSPTOE                      ;REPORT ERROR IF NOT
013654  012700  000100                MOV #CT.TM1,R0                  ;SEE IF FIRST TIMEOUT ALREADY HAPPENED
013660  032765  000100  000012        BIT #CT.TM1,C.FLG(R5)
013666  001401                        BEQ 1$                          ;IF SO,
013670  006300                        ASL R0                          ;SET SECOND TIME OUT FLAG
013672  052700  000040          1$:     BIS #CT.STA,R0                ;SET THE PROPER TIMEOUT BIT
013676  050065  000012                BIS R0,C.FLG(R5)                ;  AND STATUS REQUESTED BIT
013702  012700  000001                MOV #OP.GDS,R0                  ;BUILD GET DUST STATUS COMMAND
013706  004737  017224                CALL BLDCMD
013712  012764  100000  000012        MOV #RG.OWN,HC.CCT(R4)          ;MARK COMMAND TO UDA
013720  005775  000000                TST @(R5)                       ;TELL UDA COMMAND IS THERE
013724  000137  014400                JMP RSPOU4
```

```
     53 013730                          RSPNTO:
```

```
                                        ;SWITCH TO NEXT CONTROLLER

013730  005737  002170        RSPNXT: TST UFREEZ      ;FROZEN TO ONE UNIT?
013734  001264                        BNE RESPCT      ;STAY THERE IF SO
013736  062705  000052                ADD #C.SIZE,R5                  ;MOVE TO NEXT TABLE
013742  005337  002164                DEC UCNT                        ;CHECK IF MORE CONTROLLERS
013746  001257                        BNE RESPCT                      ;LOOK AT NEXT CONTROLLER
013750  000651                        BR RESPDM                       ;LOOK AT FIRST CONTROLLER AGAIN

                                        ;REMOVE A CONTROLLER FROM TESTING

013752  00506?  000012        RSPDRP: CLR C.FLG(R5)   ;CLEAR PROGRAM RUNNING
013756  00503?  002170                CLR UFREEZ
013762  010504                        MOV R5,R4
013764  062704  000016                ADD #C.DRO,R4
013770  012702  000010                MOV #8.,R2
013774  012403                  1$:   MOV (R4)+,R3
013776  001420                        BEQ 3$
014000  005763  000002                TST D.UNIT(R3)
014004                                      ASSUME DT.AVL EQ BIT15
014004  100003                        BPL 2$
014006  005302                        DEC R2
014010  001371                        BNE 1$
014012  000412                        BR 3$
014014  052763  100000  000002  2$:   BIS #DT.AVL,D.UNIT(R3)
014022  005302                        DEC R2
014024  001405                        BEQ 3$
014026  005714                        TST (R4)
014030  001403                        BEQ 3$
014032  004737  017024                CALL LOADDM     ;START DM PROGRAM AGAIN
014036  001223                        BNE RESPCT
014040  005337  002162          3$:   DEC URNING                      ;REDUCE RUNNING CONTROLLERS COUNT
014044  001331                        BNE RSPNXT                      ;IF ANY STILL RUNNING, LOOK AT THEM
014046  000207                        RETURN                          ;ELSE RETURN TO TEST SECTION

014050                          RSPTOE: ERRDF 31,,ERR031                ;REPORT TIMEOUT ERROR
014050  104455                                                                  TRAP    C$ERDF
014052  000037                                                                  .WORD   31
014054  000000                                                                  .WORD   0
014056  012632                                                                  .WORD   ERR031
014060  000734                        BR RSPDRP                       ;DROP CONTROLLER FROM TESTING
```

```
                                        ;CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
                                        ;
                                        ;CHECK FOR PROPER OPCODE IN END PACKET
                                        ;
014062  012700  000204          RSPIN:  MOV #CP.END+OP.SSD,R0           ;GET SEND DATA END PACKET OPCODE
014066  032765  000020  000012          BIT #CT.REQ,C.FLG(R5)           ;LOOK IF SEND DATA OR RECEIVE DATA
014074  001402                          BEQ RSPMWR
014076  012700  000205                  MOV #OP.END+OP.RSD,R0           ;CHANGE TO RECEIVE DATA END PACKET OPCODE
014102  120064  000030          RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)       ;COMPARE TO OPCODE IN END PACKET
014106  001145                          BNE RSPERR
                                        ;
                                        ;LOOK AT STATUS CODE
                                        ;
014110  032764  000037  000032          BIT #ST.MSK,HC.MPK+P.STS(R4)    ;CHECK FOR STATUS CODE ST.SUC (ZERO)
014116  001004                          BNE RSPERW
                                        ;
                                        ;CHECK FOR EXPECTED REFERENCE NUMBER
                                        ;
014120  026564  000050  000020          CMP C.REF(R5),HC.MPK+P.CRF(R4)  ;CHECK IF CORRECT REF NUMBER
014126  001405                          BEQ RSPPTW
014130                          RSPERW: ERRDF 33,,ERR033
014130  104455                                                          TRAP    C$ERDF
014132  000041                                                          .WORD   33
014134  000000                                                          .WORD   0
014136  012662                                                          .WORD   ERR033
014140  000704                          BR RSPDRP                       ;DROP UNIT FROM TESTING
                                        ;
                                        ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
                                        ;
014142  032765  000020  000012  RSPPTW: BIT #CT.REQ,C.FLG(R5)           ;CHECK IF RESPONSE FROM DM PROGRAM
014150  001463                  RSPOU:  BEQ RSPOUT                      ;LOOK AT REQUEST NUMBER IF SO
```

```
                                        ;MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM

014152  016401  000430        RSPPT2: MOV HC.BF2(R4),R1               ;GET REQUEST NUMBER
014156  042701  007777                BIC     #^C<DU.TYP>,R1          ;CHECK TYPE
014162  001403                        BEQ 1$                          ;IF ZERO, ERROR
014164  020127  060000                CMP R1,#DU.SPC                  ;CHECK IF IN EXPECTED RANGE
014170  101405                        BLOS RSPPT3
014172                          1$:     ERRDF 32,,ERR032                ;BAD REQUEST NUMBER
014172  104455                                                                  TRAP    C$ERDF
014174  000040                                                                  .WORD   32
014176  000000                                                                  .WORD   0
014200  012644                                                                  .WORD   ERR032
014202  000663                        BR RSPDRP                       ;DROP UNIT FROM TESTING

014204  016403  000034        RSPPT3: MOV HC.MPK+P.BCNT(R4),R3        ;GET BYTE COUNT OF CHARACTERS RECEIVED IN R3
014210  162703  000002                SUB #2,R3                       ;(FIRST TWO CHARACTERS ARE TYPE WORD)
014214  012700  000004                MOV #OP.SSD,R0                  ;BUILD A SEND DATA COMMAND PACKET
014220  004737  017224                CALL BLDCMD                     ;  FOR ANSWER TO DM PROGRAM
014224  012700  000164                MOV #HC.BF1,R0                  ;POINT TO BUFFER IN PACKET
014230  004737  017366                CALL CLRBUF                     ; AND CLEAR BUFFER
014234  010402                        MOV R4,R2                       ;R2 POINTS TO SEND BUFFER
014236  062704  000244                ADD #HC.BSZ,R4          ;R4 POINTS TO CHARACTERS IN RECEIVE BUFFER
014242  042724  170000                BIC #DU.TYP,(R4)+       ;CLEAR TYPE FIELD IN BUFFER
014246  000301                        SWAB R1                         ;GET TYPE RIGHT JUSTIFIED
014250  006201                        ASR R1                          ;TIMES TWO
014252  006201                        ASR R1
014254  006201                        ASR R1
014256  010100                        MOV R1,R0                       ;COPY MESSAGE TYPE TO R0
014260  005001                        CLR R1                          ;R1 CONTAINS ZERO SEND BYTE COUNT
014262  004770  014546                CALL @RSPDSP-2(R0)              ;CALL REQUESTED ROUTINE
014266  001231                        BNE RSPDRP                      ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
                                                                        ; Z SET IF UNIT TO CONTINUE RUNNING
014270  016504  000014                MOV C.RING(R5),R4               ;GET RING ADDRESS
014274  032701  000001                BIT #1,R1                       ;LOOK AT CHARACTER COUNT TO SEND TO DUP PROGRAM
014300  001401                        BEQ 1$                          ;IF AN ODD COUNT
014302  005201                        INC R1                          ;  INCREASE BY ONE
014304  010164  000120          1$:   MOV R1,HC.CPK+P.BCNT(R4)        ;PUT CHARACTER COUNT IN COMMAND PACKET
014310  100003                        BPL RSPOUT                      ;IF NEGATIVE BYTE COUNT RETURNED
014312  042765  000020  000012        BIC #CT.REQ,C.FLG(R5)           ;  DON'T SEND ANY DATA TO UDA

                                        ;SEND COMMAND BACK TO UDA

014320  042765  000350  000012  RSPOUT: BIC #CT.MSG+CT.STA+CT.TM1+CT.TM2,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
014326  032765  000020  000012        BIT #CT.REQ,C.FLG(R5)           ;CHECK WHICH COMMAND TO SEND
014334  001014                        BNE RSPOU2                      ;BRANCH IF RESPONSE TO REQUEST

014336  012700  000005                MOV #OP.RSD,R0                  ;BUILD RECEIVE DATA COMMAND
014342  004737  017224                CALL BLDCMD
014346  012700  000430                MOV #HC.BF2,R0                  ;POINT TO MESSAGE BUFFER
014352  004737  017366                CALL CLRBUF                     ;  AND CLEAR IT
014356  052765  000020  000012        BIS #CT.REQ,C.FLG(R5)           ;SET REQUEST BIT
014364  000403                        BR RSPOU3

014366  042765  000020  000012  RSPOU2: BIC #CT.REQ,C.FLG(R5)         ;CLEAR REQUEST BIT
014374                          RSPOU3:
014374  004737  017310                CALL SNDCMD                     ;SEND COMMAND TO UDA
014400  016500  000042          RSPOU4: MOV C.TOT(R5),R0                ;SET TIMEOUT
```

```
 54 014404  010501                          MOV R5,R1
 55 014406  062701  000036                  ADD #C.TO,R1                    ;PUT TIME IN CONTROLLER TABLE
 56 014412  004737  017622                  CALL SETTO
 57 014416  000137  013730                  JMP RSPNXT                      ;NOW WAIT FOR END PACKET
 58 014422  122764  000201  000030  RSPERR: CMPB #OP.END+OP.GDS,HC.MPK+P.OPCD(R4)   ;SEE IF GET DUST STATUS OPCODE
 59 014430  001237                          BNE RSPERW
 60 014432  132764  000010  000037          BITB #DF.ACT,HC.MPK+P.DFLG(R4)  ;IF DUST NO LONGER RUNNING
 61 014440  001603                          BEQ RSPTOE                      ;  REPORT ERROR
 62 014442  042765  000050  000012          BIC #CT.STA+CT.MSG,C.FLG(R5)    ;CLEAR CONTROL BITS
 63 014450  032765  000200  000012          BIT #CT.TM2,C.FLG(R5)           ;IF AT SECOND TIMEOUT
 64 014456  001413                          BEQ 1$
 65 014460  026465  000040  000044          CMP HC.MPK+P.DPI(R4),C.PRI(R5)  ;COMPARE PROGRESS INDICATOR
 66 014466  001004                          BNE 2$
 67 014470  026465  000042  000046          CMP HC.MPK+P.DPI+2(R4),C.PRI+2(R5)      ;COMPARE PROGRESS INDICATOR
 68 014476  001422                          BEQ 4$                          ;REPORT ERROR IF NOT CHANGED
 69 014500  042765  000200  000012  2$:     BIC #CT.TM2,C.FLG(R5)           ;CLEAR TIMEOUT 2 FLAG
 70 014506  032765  000100  000012  1$:     BIT #CT.TM1,C.FLG(R5)           ;IF AT FIRST TIMEOUT
 71 014514  001406                          BEQ 3$
 72 014516  016465  000040  000044          MOV HC.MPK+P.DPI(R4),C.PRI(R5)  ;GET COPY OF PROGRESS INDICATOR
 73 014524  016465  000042  000046          MOV HC.MPK+P.DPI+2(R4),C.PRI+2(R5)      ;GET COPY OF PROGRESS INDICATOR
 74 014532  012764  140000  000006  3$:     MOV #RG.OWN+RG.FLG,HC.MCT(R4)   ;GIVE MESSAGE BUFFER BACK TO UDA
 75 014540  000137  013730                  JMP RSPNXT
 76 014544  000137  014050          4$:     JMP RSPTOE
```

```
                                        ;PESPONSE REQUEST DISPATCH TABLE

014550  014564                RSPDSP: .WORD QUEST                     ;QUESTION
014552  014636                        .WORD DQUEST                    ;QUESTION WITH DEFAULT ANSWER
014554  015010                        .WORD INFO                      ;INFORMATION MESSAGE FOR OPERATOR
014556  015136                        .WORD TERM                      ;NORMAL TERMINATION
014560  015146                        .WORD ERRTRM                    ;FATAL ERROR TERMINATION
014562  015156                        .WORD SPECL                     ;SPECIAL
        000006                       DSPSIZ=<.-RSPDSP>/2              ;LEGAL NUMBERS ARE LOWER THAN THIS
```

```
;NORMAL DUP RECEIVE DATA BUFFER DESCRIPTION

;BYTE OFFSET FROM
;START OF BUFFER                +-----------------------------------+
;        0                      ! TYPE !     MESSAGE NUMBER          !  USED TO SELECT ROUTINE
;                               +-----------------------------------+
;        2                      !            DATA BYTES              !  R4 CONTAINS THIS ADDRESS
;                               +-----------------------------------+
;        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              !
;                               +-----------------------------------+
```

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

;BYTE OFFSET FROM
;START OF BUFFER   +-------------------------------+
;       0          !          DATA BYTES           !  R2 CONTAINS THIS ADDRESS
;                  +-------------------------------+
;       2          !          DATA BYTES           !
;                  +-------------------------------+
;       4          !          DATA BYTES           !
;                  +-------------------------------+
;       6          !          DATA BYTES           !
;                  +-------------------------------+
;       8          !          DATA BYTES           !
;                  +-------------------------------+
;      10          !          DATA BYTES           !
;                  +-------------------------------+
;      12          !          DATA BYTES           !
;                  +-------------------------------+
;      14          !          DATA BYTES           !
;                  +-------------------------------+
;      16          !          DATA BYTES           !
;                  +-------------------------------+
;      18          !          DATA BYTES           !
;                  +-------------------------------+
;      20          !          DATA BYTES           !
;                  +-------------------------------+
;      22          !          DATA BYTES           !
;                  +-------------------------------+
;
;       .                         .
;
;       .                         .
;
;       .                         .
;
;                  +-------------------------------+
;      80          !          DATA BYTES           !
;                  +-------------------------------+
```

```
                                ;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
                                ;
014564  004737  015310          QUEST:  CALL GTDRVT     ;GET POINTER TO DRIVE TABLE
014570  062700  000004                  ADD #D.SERN,R0  ;BUMP POINTER TO SERIAL NUMBER
014574  014403                          MOV -(R4),R3    ;GET QUESTION NUMBER
014576  001411                          BEQ QUE0        ;BRANCH IF QUESTION NUMBER 0
014600  020327  000007                  CMP R3,#7       ;IF NOT, SEE IF QUESTION NUMBER 7
014604  001410                          BEQ QUE7
014606                                  ERRDF 100,,ERR100       ;ANY OTHER NUMBER IS AN ERROR
014606  104455                                                          TRAP    C$ERDF
014610  000144                                                          .WORD   100
014612  000000                                                          .WORD   0
014614  012724                                                          .WORD   ERR100
014616  000244                          CLZ     ;CLEAR Z TO STOP DUP PROGRAM
014620  000207                          RETURN
                                ;
014622  012700  003304          QUE0:   MOV #DATE0,R0   ;POINT TO DATE STRING
014626                          QUE7:
014626  005201                  QUEL:   INC R1  ;COUNT THE CHARACTERS
014630  112022                          MOVB (R0)+,(R2)+        ; AND PUT THEM IN OUTPUT BUFFER
014632  001375                          BNE QUEL        ;  UNTIL A NUL CHARACTER FOUND
014634  000207                          RETURN  ;RETURN WITH Z SET
```

```
                                        ;MESSAGE TYPE 2
                                        ;
                                        ;ANSWER QUESTIuN FOR DUP PROGRAM WITH DEFAULT ANSWER
                                        ;
                                        ;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
                                        
014636  004737  015310          DQUEST: CALL GTDRVT     ;GET DRIVE TABLE ADDRESS INTO R0
014642  014403                          MOV -(R4),R3    ;GET QUESTION NUMBER
014644  020327  000006                  CMP R3,#DQUESZ
014650  101035                          BHI DQUEX
014652  006303                          ASL R3
014654  000173  014660                  JMP @DQUEJP(R3)
014660  014744                  DQUEJP: .WORD DQUEX     ; 0  (NOT USED)
014662  014676                          .WORD DQUNIT    ; 1  ENTER UNIT NUMBER TO FORMAT
014664  014744                          .WORD DQUEX     ; 2  (NOT USED)
014666  014744                          .WORD DQUEX     ; 3  (NOT USED)
014670  014750                          .WORD DQRFMT    ; 4  USE EXISTING BAD SECTOR INFORMATION
014672  014770                          .WORD DQRSTR    ; 5  DOWN-LINE LOAD BAD SECTOR BLOCK INFORMATION
014674  015000                          .WORD DQCONT    ; 6  CONTINUE IF BAD BLOCK INFO INACCESSIBLE
000006                                          DQUESZ=<<.-DQUEJP>/2>-1
                                        
                                        ;ENTER UNIT NUMBER TO FORMAT
                                        
014676                          DQUNIT: PUSH R5                                         MOV R5,-(SP)
014676  010546
014700  005004                          CLR R4
014702  011003                          MOV (R0),R3     ;GET DRIVE NUMBER
014704                                          ASSUME D.DRV EQ 0
014704  012700  000012                  MOV #10.,R0     ;RADIX 10.
014710  004737  016766          DQUNL1: CALL DIVIDE
014714                                  PUSH R5                                         MOV R5,-(SP)
014714  010546
014716  005201                          INC R1
014720  005703                          TST R3
014722  001372                          BNE DQUNL1
014724  010100                          MOV R1,R0
014726                          DQUNL2: POP R5                                          MOV (SP)+,R5
014726  012605
014730  062705  000060                  ADD #'0,R5
014734  110522                          MOVB R5,(R2)+
014736  005300                          DEC R0
014740  001372                          BNE DQUNL2
014742                                  POP R5                                          MOV (SP)+,R5
014742  012605
014744  000264                  DQUEX:  SEZ
014746  000207                          RETURN

014750  032737  000003  003200  DQRFMT: BIT #SO.FMT,MODE
```

GLOBAL SUBROUTINES SECTION

```
     54 014756  001410                          BEQ DQNO
     55 014760  112712  000131          DQYES:  MOVB #'Y,(R2)
     56 014764  005201                          INC R1
     57 014766  000766                          BR DQUEX
     58
     59 014770  032737  000010  003200  DQRSTR: BIT #SO.STR,MODE
     60 014776  001370                          BNE DQYES
     61 015000                          DQCONT:
     62 015000  112712  000116          DQNO:   MOVB #'N,(R2)
     63 015004  005201                          INC R1
     64 015006  000756                          BR DQUEX
```

```
                                        ;MESSAGE TYPE 3
                                        ;
                                        ;PRINT INFORMATION FROM DUP PROGRAM
                                        ;
                                        ;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:
                                        ;       R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM
                                        ;       Z SET TO CONTINUE RUNNING DUP PROGRAM
                                        ;
015010  016400  177776          INFO:   MOV -2(R4),R0   ;GET MESSAGE NUMBER
015014  001434                          BEQ INFOB       ;IF ZERO, PRINT BEGUN MESSAGE
015016  020027  000100                  CMP R0,#100     ;IF OCTAL 100
015022  001423                          BEQ INFOE       ;  PRINT ERROR MESSAGE
015024  020027  000200                  CMP R0,#200     ;SEE IF 200 OR GREATER
015030  002005                          BGE INFOH       ;  IF SO, PRINT WITHOUT FREEZING
015032  005737  002170                  TST UFREEZ
015036  001007                          BNE INFOP
015040  005237  002170                  INC UFREEZ
015044  004737  015310          INFOH:  CALL GTDRVT
015050  010002                          MOV R0,R2
015052  004737  015334                  CALL HEADER
015056  004737  015254          INFOP:  CALL MESG        ;PRINT THE MESSAGE
015062  012701  100000          INFOX:  MOV #BIT15,R1                   ;RETURN A NEGATIVE BYTE COUNT
015066  000264                          SEZ
015070  000207                          RETURN                          ;RETURN WITH Z SET
                                        
015072                          INFOE:  ERRDF 101 ,ERR101       ;ANSWER WAS REJECTED BY DUP PROGRAM
015072  104455                                                                  TRAP    C$ERDF
015074  000145                                                                  .WORD   101
015076  000000                                                                  .WORD   0
015100  012740                                                                  .WORD   ERR101
015102  000244                          CLZ     ;RETURN WITH Z CLEAR TO STOP DUP PROGRAM
015104  000207                          RETURN
                                        
015106  004737  015310          INFOB:  CALL GTDRVT      ;PRINT FORMAT BEGUN MESSAGE
015112  010002                          MOV R0,R2
015114  004737  015334                  CALL HEADER
015120  004737  015254                  CALL MESG
015124                                  PNT WNSTOP       ;PRINT WARNING NOT TO STOP NOW
015124  004137  016720                                                      JSR R1,LPNT
015130  004355                                                              .WORD WNSTOP
015132  000000                                                              .WORD PNT.CT
015134  000752                          BR INFOX
```

```
                                ;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
                                ;
015136  004737  015010          TERM:   CALL INFO       ;PRINT THE MESSAGE
015142  000244                          CLZ
015144  000207                          RETURN                          ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM
```

```
                                ;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
                                
015146  004737  015010          ERRTRM: CALL INFO
015152  C00244                         CLZ
015154  000207                         RETURN                          ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM
```

```
                                        ;MESSAGE TYPE 6
                                        ;
                                        ;SPECIAL TYPE - READ FCT BLOCK FROM FILE
                                        ;
                                        ;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 SET TO SEND DATA TO PROGRAM

015156  023714  003176          SPECL:  CMP FCTNUM,(R4) ;SEE IF DESIRED BLOCK IS IN MEMORY
015162  001425                          BEQ SPECLX      ; IF SO, SEND TO DUP PROGRAM
015164  002407                          BLT SPECLR      ; IF LOWER NUMBERED BLOCK IN MEMORY,
                                                        ;   GO READ NEXT BLOCK
015166                          SPECLC:
015166                                  CLOSE   ;OTHERWISE, START READING FROM BEGINNING AGAIN
015166  104435                                                                  TRAP    C$CLOS
015170                                  OPEN #FNAME
015170  012700  003230                                                         MOV     #FNAME,R0
015174  104434                                                                  TRAP    C$OPEN
015176  012737  177777  003176          MOV #-1,FCTNUM
015204  012703  001000          SPECLR: MOV #512.,R3    ;GET BYTE COUNT IN A BLOCK
015210  012701  002176                  MOV #FCTBUF,R1  ;POINT TO STORAGE AREA
015214                          SPECLL: GETBYTE (R1)+   ;READ THE FILE
015214  104426                                                                  TRAP    C$GETB
015216  110021                                                                  MOVB    R0,(R1)+
015220                                  BNCOMPLETE SPECLE       ;PRINT ERROR IF NO MORE BYTES IN FILE
015220  103005                                                                  BCC     SPECLE
015222  005303                          DEC R3  ;COUNT THE BYTES
015224  001373                          BNE SPECLL
015226  005237  003176                  INC FCTNUM      ;KEEP COUNT OF BLOCK IN MEMORY
015232  000751                          BR SPECL

015234  005212                  SPECLE: INC (R2)        ;TELL DUP PROGRAM DATA NOT AVAILABLE
015236  012762  002176  000002  SPECLX: MOV #FCTBUF,2(R2)       ;PUT ADDRESS OF DATA IN OUTPUT BUFFER
015244  012701  000006                  MOV #6,R1       ;SEND 3 WORDS TO DUP PROGRAM
015250  000264                          SEZ
015252  000207                          RETURN                  ;RETURN WITH Z SET TO SEND DATA TO DUP PROGRAM
```

```
                                        ;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
                                        
015254                                  MESG:
015254  112400                          1$:     MOVB (R4)+,R0           ;PRINT CHARACTERS FROM DUP PROGRAM
015256  001405                                  BEQ 2$                  ;  DISCARDING LF AND NULL CHARACTERS
015260  020027  000012                          CMP R0,#12
015264  001402                                  BEQ 2$
015266                                          PRINT R0
015266  004737  016510                                                  CALL CPNT
015272  005303                          2$:     DEC R3                  ;COUNT THE CHARACTERS
015274  003367                                  BGT 1$
015276                                          PRINT #CR
015276  112700  000015                                                  MOVB #CR,R0
015302  004737  016510                                                  CALL CPNT
015306  000207                                  RETURN
```

M7


```
                                        ;GTDRVT
                                        ;
                                        ;GET DRIVE TABLE ADDRESS FROM CONTROLLER TABLE
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;OUTPUTS:
                                        ;       R0 - ADDRESS OF FIRST DRIVE TABLE AVAILABLE FOR TESTING
                                        ;               (WITH DT.AVL BIT CLEAR)
                                        ;
015310                                  GTDRVT: PUSH R5
015310  010546                                                                         MOV R5,-(SP)
015312  062705  000016                          ADD #C.DR0,R5
015316  012500                          GTDRVL: MOV (R5)+,R0
015320  016037  000002  002074                  MOV D.UNIT(R0),L$LUN
015326                                                  ASSUME DT.AVL EQ BIT15
015326  100773                                  BMI GTDRVL
015330                                          POP R5
015330  012605                                                                         MOV (SP)+,R5
015332  000207                                  RETURN
```

```
                                        ;HEADER
                                        ;
                                        ;PRINT A HEADER IN FRONT OF EACH MESSAGE FROM DUP PROGRAM.
                                        ;A UDA ADDRESS IS PRINTED IF MORE THAN ONE UDA IS IN HARDWARE P-TABLE.
                                        ;A RUNTIME IS PRINTED IF A CLOCK IS BEING USED TO TIME PROGRAM EXECUTION.
                                        ;
                                        ;INPUT:
                                        ;       R5 - POINTER TO CONTROLLER TABLE
                                        ;OUTPUT:
                                        ;       R0 - POINTER TO DRIVE TABLE
                                        ;       PRINTED MESSAGE
                                        ;
015334  022737  000001  00201c HEADER: CMP #1,L$UNIT                    ;IF MORE THAN ONE UNIT BEING TESTED
015342  001411                        BEQ 1$
015344                                PNTF MESSG,D.UNIT(R2),(R5),(R2)         ;PRINT UDA ADDRESS
015344  011246                                                                  MOV (R2),-(SP)
015346  011546                                                                  MOV (R5),-(SP)
015350  016246  000002                                                          MOV D.UNIT(R2),-(SP)
015354  004137  016662                                                          JSR R1,LPNTF
015360  004035                                                                  .WORD MESSG
015362  000006                                                                  .WORD PNT.CT
                                              ASSUME C.UADR EQ 0
                                              ASSUME D.DRV EQ 0
015364  000407                        BR 2$
015366  005737  003206          1$:   TST KW.CSR                        ;IF NO CLOCK BEING USED
015372  001406                        BEQ 3$                            ;BYPASS RUNTIME MESSAGE
015374                                PRINT #CR
015374  112700  000015                                                          MOVB #CR,R0
015400  004737  016510                                                          CALL CPNT
015404  004737  020652          2$:   CALL RNTIME                       ;PRINT RUNTIME IF A CLOCK IN USE
015410                          3$:   PRINT #CR
015410  112700  000015                                                          MOVB #CR,R0
015414  004737  016510                                                          CALL CPNT
015420  000207                        RETURN
```

```
                                ;OSTRNG
                                ;
                                ;FORMAT OF THE ASCIZ STRING IS AS FOLLOWS:
                                ;
                                ;CHARACTERS ENCLOSED IN QUOTES ARE TO BE PRINTED AS THEY ARE.
                                ;
                                ;OTHERWISE CODE IS A SINGLE LETTER FOLLOWED BY AN OPTIONAL DECIMAL
                                ;NUMBER:
                                ;  ON - PRINT OCTAL NUMBER. N REPRESENTS SIZE OF BINARY NUMBER PASSED
                                ;       IN PARAMETER IN BITS. MAY BE IN RANGE 1 TO 32. IF N>16, TWO PARAMETER
                                ;       WORDS ARE USED, OTHERWISE ONLY ONE WORD. LEADING ZEROS ARE PRINTED.
                                ;       N IS ALWAYS SPECIFIED.
                                ;  DN - PRINT UNSIGNED DECIMAL NUMBER FROM N BIT PARAMETER. LEADING ZEROS
                                ;       ARE NOT PRINTED. A 16 BIT NUMBER EQUAL TO ZERO WILL PRINT "0".
                                ;  HN - PRINT HEX NUMBER FROM PARAMETER OF N BITS. IF N>16 TWO PARAMETERS
                                ;       ARE USED, OTHERWISE ONLY ONE PARAMETER. LEADING ZEROS ARE PRINTED.
                                ;  SN - PRINT N SPACES. N ASSUMED TO BE 1.
                                ;  NN - START NEW LINE (CR-LF SEQUENCE). N ASSUMED TO BE 1.
                                ;  AN - PRINT N ASCII CHARACTERS FROM PARAMETERS, N ASSUMED TO BE 1.
                                ;       N/2 PARAMETER WORDS USED.
                                ;  RN - EXECUTE ROUTINE #N. N MUST BE GIVEN AND DEFINED IN HOST PROGRAM.
                                ;
                                ;A NULL CHARACTER MEANS END OF MESSAGE. A NULL AS FIRST CHARACTER IN STRING
                                ;MUST BE IGNORED.
                                ;
                                ;OUTPUT A MESSAGE ACCORDING TO A FORMAT STRING
                                ;
                                ;INPUTS:
                                ;       R2 - ADDRESS OF START OF FORMAT STRING
                                ;       R4 - ADDRESS OF PARAMETERS
                                ;OUTPUTS:
                                ;       R2 AND R4 UPDATED TO END OF STRING AND PARAMETERS
                                ;
015422  112201                  OSTRNG: MOVB (R2)+,R1           ;GET CONTROL CHARACTER
015424  001421                          BEQ OSTRE               ;EXIT IF NULL CHARACTER
015426  012700  015722                  MOV #ERRC,R0            ;GET POINTER TO CHARACTER TABLE
015432  120110                  NCONS:  CMPB R1,(R0)            ;COMPARE CHARACTER WITH TABLE ENTRY
015434  001407                          BEQ NCONF               ;BRANCH IF MATCH FOUND
015436  105720                          TSTB (R0)+              ;INCREMENT POINTER
015440  001374                          BNE NCONS               ;CONTINUE SEARCH IF NOT END OF TABLE
015442                                  PNTF ERRME1             ;REPORT BAD CONTROL CHARACTER
015442  004137  016662                                                  JSR R1,LPNTF
015446  003746                                                          .WORD ERRME1
015450  000000                                                          .WORD PNT.CT
015452  000406                          BR OSTRE
015454  162700  015722          NCONF:  SUB #ERRC,R0            ;GET INCREMENT INTO TABLE
015460  006300                          ASL R0                  ;DOUBLE TO WORD COUNT
015462  004770  015734                  CALL @ERRD(R0)          ;DISPATCH TO PRINT ROUTINE
015466  000755                          BR OSTRNG               ;GET NEXT
015470  000207                  OSTRE:  RETURN
```

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

015472  112200                CON.QU: MOVB (R2)+,R0           ;GET CHARACTER
015474  120027  000042                CMPB R0,#'"             ;CHECK IF ENDING QUOTE
015500  001403                        BEQ CON.QX              ;IF SO, GO GET NEXT CONTROL CHARACTER
015502                                PRINT R0                ;PRINT THE CHARACTER
015502  004737  016510                                                        CALL CPNT
015506  000771                        BR CON.QU               ;CONTINUE PRINTING
015510  000207                CON.QX: RETURN

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

015512  004737  016170        CON.A:  CALL GETCNT              ;GET COUNT OF CHARACTERS
015516                        CON.A1: PRINT (R4)+              ;PRINT THE CHARACTER
015516  112400                                                        MOVB (R4)+,R0
015520  004737  016510                                                CALL CPNT
015524  005301                        DEC R1                  ;COUNT THE CHARACTERS
015526  001373                        BNE CON.A1              ;PRINT UNTIL COUNT REACHES ZERO
015530  032704  000001                BIT #1,R4               ;CHECK IF R4 NOW ODD
015534  001401                        BEQ CON.A2
015536  005204                        INC R4                  ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
015540  000207                CON.A2: RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.

015542  012701  000012        CON.D:  MOV #10.,R1              ;LOAD RADIX
015546  004737  016246                CALL PNTNUM             ;PRINT NUMBER
015552  000207                        RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.

015554  012701  000020        CON.H:  MOV #16.,R1              ;LOAD RADIX
015560  004737  016246                CALL PNTNUM             ;PRINT NUMBER
015564  000207                        RETURN                  ;NOW GET NEXT CONTROL CHARACTER
```

```
                                ;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.

015566  012701  000010        CON.O:  MOV #8.,R1              ;LOAD RADIX
015572  004737  016246                CALL PNTNUM             ;PRINT NUMBER
015576  000207                        RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                ;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.

015600  004737  016170        CON.N:  CALL GETCNT             ;GET COUNT
015604                        CON.N1: PRINT #CR               ;PRINT NEW LINE SEQUENCE
015604  112700  000015                                                MOVB #CR,R0
015610  004737  016510                                                CALL CPNT
015614  005301                        DEC R1                  ;COUNT THE SEQUENCES
015616  001372                        BNE CON.N1
015620  000207                        RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                ;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.

015622  004737  016170        CON.R:  CALL GETCNT              ;GET ROUTINE NUMBER
015626  020127  000010                CMP R1,#ERRRSZ          ;CHECK IF DEFINED ROUTINE NUMBER
015632  101004                        BHI CON.R1
015634  060101                        ADD R1,R1               ;DOUBLE COUNT TO GET WORD INDEX
015636  004771  015700                CALL @ERRRTB-2(R1)      ;CALL ROUTINE
015642  000207                        RETURN                  ;NOW GET NEXT CONTROL CHARACTER
015644                        CON.R1: PNTF ERRME1             ;REPORT BAD MESSAGE STRING
015644  004137  016662                                                JSR R1,LPNTF
015650  003746                                                        .WORD ERRME1
015652  000000                                                        .WORD PNT.CT
015654                                POP R1                  ;FIX THE STACK
015654  012601                                                        MOV (SP)+,R1
015656  000207                        RETURN

                                ;CONTROL CHARACTER WAS AN S. PRINT SPACES.

015660  004737  016170        CON.S:  CALL GETCNT              ;GET COUNT
015664                        CON.S1: PRINT <#' >             ;PRINT A SPACE
015664  112700  000040                                                MOVB #' ,R0
015670  004737  016510                                                CALL CPNT
015674  005301                        DEC R1                  ;COUNT THE SPACES
015676  001372                        BNE CON.S1
015700  000207                        RETURN                  ;NOW GET NEXT CONTROL CHARACTER
```

E8


```
      1                                 ;ERROR ROUTINE DISPATCH TABLE
      2
      3 015702  015754                  ERRRTB: .WORD CALRE                     ;NOT USED
      4 015704  015754                          .WORD CALRE                     ;NOT USED
      5 015706  015754                          .WORD CALRE                     ;NOT USED
      6 015710  015766                          .WORD CALR4                     ;PRINT BASIC LINE WITHOUT UDA ADDRESS
      7 015712  016042                          .WORD CALR5                     ;PRINT BASIC LINE WITH UDA ADDRESS
      8 015714  016120                          .WORD CALR6                     ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
      9 015716  016134                          .WORD CALR7                     ;PRINT  "REPLACE PROCESSOR MODULE"
     10 015720  016152                          .WORD CALR8                     ;PRINT  "  UDASA CONTAINS  XXXXXX"
     11         000010                  ERRRSZ=<.-ERRRTB>/2
     12
     13                                 ;BUILD TWO TABLES
     14                                 ;       FIRST CONTAINING CONTROL CHARACTERS
     15                                 ;       SECOND CONTAINING ROUTINE ADDRESSES
     16
     17                                 .MACRO BUILD
     18                                         ENTRY ",CON.QU
     19                                         ENTRY A,CON.A
     20                                         ENTRY D,CON.D
     21                                         ENTRY H,CON.H
     22                                         ENTRY O,CON.O
     23                                         ENTRY N,CON.N
     24                                         ENTRY R,CON.R
     25                                         ENTRY S,CON.S
     26                                 .ENDM
```

F8


```
                                        ;HERE IS FIRST TABLE
                                        
                                        .MACRO ENTRY ARG1,ARG2
                                                .LIST
                                                .BYTE ''ARG1
                                                .NLIST
                                        .ENDM
                                        
015722                                  ERRC:   BUILD
015722          042                             .BYTE '"
015723          101                             .BYTE 'A
015724          104                             .BYTE 'D
015725          110                             .BYTE 'H
015726          117                             .BYTE 'O
015727          116                             .BYTE 'N
015730          122                             .BYTE 'R
015731          123                             .BYTE 'S
015732          000                             .BYTE 0                 ;FOLLOW WITH A NULL BYTE
                                                .EVEN
                                        
                                        ;HERE IS SECOND TABLE
                                        
                                        .MACRO ENTRY ARG1,ARG2
                                                .LIST
                                                .WORD ARG2
                                                .NLIST
                                        .ENDM
                                        
015734                                  ERRD:   BUILD
015734  015472                                  .WORD CON.QU
015736  015512                                  .WORD CON.A
015740  015542                                  .WORD CON.D
015742  015554                                  .WORD CON.H
015744  015566                                  .WORD CON.O
015746  015600                                  .WORD CON.N
015750  015622                                  .WORD CON.R
015752  015660                                  .WORD CON.S
```

```
                                ;PRE-PROGRAMMED ROUTINES 1, 2 AND 3
                                ;NOT USED - PRINTS ERROR MESSAGE
                                
4  015754                       CALRE:  PNTF ERRME1                    ;PRINT ERROR MESSAGE
   015754  004137  016662                                                       JSR R1,LPNTF
   015760  003746                                                              .WORD ERRME1
   015762  000000                                                              .WORD PNT.CT
5  015764  000207                       RETURN
```

H8


```
      1                                 ;PRE-PROGRAMMED ROUTINE 4
      2                                 ;PRINT BASIC LINE FOR HOST PROGRAM ERROR WITHOUT UDA ADDRESS
      3                                 ;THEN SWITCH TO EXTENDED FORMAT
      4
      5 015766                          CALR4:  PNTB BASLN,#BASNO,#BAS,#BAS,#BAS
        015766  012746  004247                                                    MOV #BAS,-(SP)
        015772  012746  004247                                                    MOV #BAS,-(SP)
        015776  012746  004247                                                    MOV #BAS,-(SP)
        016002  012746  004165                                                    MOV #BASNO,-(SP)
        016006  004137  016672                                                    JSR R1,LPNTB
        016012  004250                                                            .WORD BASLN
        016014  000010                                                            .WORD PNT.CT
      6 016016  004737  020652                  CALL RNTIME
      7 016022                                  PRINT #CR
        016022  112700  000015                                                    MOVB #CR,R0
        016026  004737  016510                                                    CALL CPNT
      8 016032  012737  016610  003222          MOV #PX,PTYPE
      9 016040  000207                          RETURN
```

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

016042                          CALR5:  PNTB BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
016042  012746  004247                                                  MOV #BAS,-(SP)
016046  012746  004247                                                  MOV #BAS,-(SP)
016052  011546                                                          MOV (R5),-(SP)
016054  012746  004204                                                  MOV #BASL2,-(SP)
016060  012746  004165                                                  MOV #BASNO,-(SP)
016064  004137  016672                                                  JSR R1,LPNTB
016070  004250                                                          .WORD BASLN
016072  000012                                                          .WORD PNT.CT
016074  004737  020652                  CALL RNTIME
016100                                  PRINT #CR
016100  112700  000015                                                  MOVB #CR,R0
016104  004737  016510                                                  CALL CPNT
016110  012737  016610  003222          MOV #PX,PTYPE
016116  000207                          RETURN
```

```
                                ;PRE-PROGRAMMED ROUTINE 6
                                ;CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
                                
016120                          CALR6:  PUSH R2                         ;SAVE CURRENT STRING POINTER
016120  010246                                                                  MOV R2,-(SP)
016122  012402                          MOV (R4)+,R2                    ;GET NEW STRING POINTER
016124  004737  015422                  CALL OSTRNG                     ;OUTPUT USING THIS STRING
016130                                  POP R2                          ;GET OLD POINTER BACK
016130  012602                                                                  MOV (SP)+,R2
016132  000207                          RETURN                          ;NOW CONTINUE THE OLD STRING
```

```
                                ;PRE-PROGRAMMED ROUTINE 7
                                ;PRINT "REPLACE PROCESSOR MODULE"

016134                          CALR7:  PUSH R2
016134  010246                                                    MOV R2 -(SP)
016136  012702  012002                  MOV #XFRU,R2
016142  004737  015422                  CALL OSTRNG
016146                                  POP R2
016146  012602                                                    MOV (SP)+,R2
016150  000207                          RETURN
```

L8


```
                                ;PRE-PROGRAMMED ROUTINE 8
                                ;PRINT   "  UDASA CONTAINS  XXXXXX"

016152                          CALR8:  PUSH R2
016152  010246                                                    MOV R2,-(SP)
016154  012702  011754                  MOV #XSA,R2
016160  004737  015422                  CALL OSTRNG
016164                                  POP R2
016164  012602                                                    MOV (SP)+,R2
016166  000207                          RETURN
```

M8


```
                                        ;GETCNT
                                        ;
                                        ;GET COUNT IN NEXT CHARACTERS OF STRING POINTED TO BY R2.
                                        ;NUMBER WILL BE IN DECIMAL. IF NO NUMBER, RETURN A
                                        ;DEFAULT OF 1.
                                        ;
                                        ;INPUTS:
                                        ;       R2 - POINTER TO ASCII STRING
                                        ;OUTPUTS:
                                        ;       R1 - NUMBER READ OR A ONE
                                        ;       R2 - POINTING TO CHARACTER AFTER NUMBER
                                        ;
016170                                  GETCNT: PUSH R0
016170  010046                                                                  MOV R0,-(SP)
016172  005001                                  CLR R1                          ;START WITH ZERO COUNT
016174  121227  000060                  GETCNX: CMPB (R2),#'0                   ;CHECK IF CHARACTER A DIGIT
016200  103415                                  BLO GETCDN                      ;BRANCH IF LOWER THAN ZERO
016202  121227  000071                          CMPB (R2),#'9
016206  101012                                  BHI GETCDN                      ;BRANCH IF HIGHER THAN NINE
016210  006301                                  ASL R1                          ;MULTIPLY NUMBER BY 10
016212  010100                                  MOV R1,R0                       ; SAVE 2N
016214  006301                                  ASL R1                          ; COMPUTE 4N
016216  006301                                  ASL R1                          ; COMPUTE 8N
016220  060001                                  ADD R0,R1                       ; 8N + 2N = 10N
016222  112200                                  MOVB (R2)+,R0                   ;GET DIGIT FROM STING
016224  162700  000060                          SUB #'0,R0                      ;GET RID OF ASCII
016230  060001                                  ADD R0,R1                       ;ADD TO NUMBER
016232  000760                                  BR GETCNX                       ;GO TO NEXT CHARACTER
016234  005701                          GETCDN: TST R1                          ;CHECK IF NUMBER IS ZERO
016236  001001                                  BNE GETCXX                      ;IF ZERO, CHANGE
016240  005201                                  INC R1                          ; TO DEFAULT OF ONE
016242                                  GETCXX: POP R0
016242  012600                                                                  MOV (SP)+,R0
016244  000207                                  RETURN
```

```
                                ;PNTNUM
                                ;
                                ;PRINT A NUMBER
                                ;
                                ;INPUTS:
                                ;       R1 - RADIX OF NUMBER
                                ;       R2 - ASCII STRING TO COUNT OF BITS IN NUMBER
                                ;       R4 - POINTER TO NUMBER (LOW WORD)
                                ;OUTPUTS:
                                ;       NUMBER IS PRINTED. LEADING ZEROS ARE PRINTED EXCEPT FOR
                                ;           DECIMAL NUMBERS.
                                ;       R0 - CONTENTS DESTROYED
                                ;
016246  010100          PNTNUM: MOV R1,R0                       ;SAVE RADIX
016250  004737  016170          CALL GETCNT                     ;GET COUNT OF BITS
016254                  PNTNUS: PUSH <R2,R3,R5>
016254  010246                                                          MOV R2,-(SP)
016256  010346                                                          MOV R3,-(SP)
016260  010546                                                          MOV R5,-(SP)
016262  012403                  MOV (R4)+,R3                    ;GET ONE PARAMETER WORD
016264  005005                  CLR R5                          ;CLEAR STORAGE FOR OTHER
016266  020127  000020          CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
016272  003401                  BLE 1$
016274  012405                  MOV (R4)+,R5                    ;YES, GET SECOND PARAMETER WORD
016276                  1$:     PUSH R4
016276  010446                                                          MOV R4,-(SP)
016300  010504                  MOV R5,R4                       ;PUT HIGH WORD IN R4
016302  012702  000020          MOV #16.,R2                     ;COMPUTE BITS NOT WANTED
016306  160102                  SUB R1,R2                       ;BY SUBTRACTING BITS TO USE
016310  002002                  BGE 2$                          ;FROM 16.
016312  062702  000020          ADD #16.,R2                     ;IF NEGATIVE, ADD 16 FOR FIRST WORD
016316  001414          2$:     BEQ 6$                          ;IF ZERO, NO BITS NEED BE CLEARED
016320  012705  100000          MOV #BIT15,R5                   ;START MASK WITH SIGN BIT SET
016324  005302          3$:     DEC R2                          ;COUNT BITS IN MASK
016326  001402                  BEQ 4$
016330  006205                  ASR R5                          ;SHIFT MORE BITS TO RIGHT
016332  000774                  BR 3$
016334  020127  000020          CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
016340  003402                  BLE 5$
016342  040504                  BIC R5,R4                       ;YES, CLEAR IN HIGH WORD
016344  000401                  BR 6$
016346  040503          5$:     BIC R5,R3                       ;NO, CLEAR IN LOW WORD
016350  004737  016766  6$:     CALL DIVIDE                     ;DIVIDE BY RADIX IN R0
016354                          PUSH R5                         ;PUSH REMAINDER ON STACK
016354  010546                                                          MOV R5,-(SP)
016356  005202                  INC R2                          ;COUNT DIGITS ON STACK
016360  005703                  TST R3                          ;CHECK IF QUOTIENT IS ZERO
016362  001372                  BNE 6$
016364  005704                  TST R4
016366  001370                  BNE 6$
```

```
      1 016370  020027  000012                CMP R0,#10.                ;IF RADIX IS DECIMAL
      2 016374  001423                        BEQ 10$                    ; JUST GO PRINT DIGITS ON STACK
      3 016376  010103                        MOV R1,R3                  ;OTHERWISE COMPUTE NUMBER OF LEADING ZEROS
      4 016400  162700  000014                SUB #12.,R0                ;DIVIDEND IS BITS IN NUMBER
      5 016404  003002                        BGT 7$                     ;DIVISOR IS BITS PER DIGIT PRINTED
      6 016406  012700  000003                MOV #3,R0                  ;   (3 OR 4)
      7 016412  004737  016766          7$:   CALL DIVIDE
      8 016416  005705                        TST R5                     ;IF REMAINDER NOT ZERO
      9 016420  001401                        BEQ 8$                     ;INCREMENT QUOTIENT
     10 016422  005203                        INC R3
     11 016424  160203                  8$:   SUB R2,R3                  ;SUBTRACT DIGITS ON STACK
     12 016426  001406                        BEQ 10$                    ;NO LEADING ZEROS IF ZERO
     13 016430                          9$:   PRINT #'0                  ;PRINT A ZERO
        016430  112700  000060                                                    MOVB #'0,R0
        016434  004737  0 6510                                                     CALL CPNT
     14 016440  005303                        DEC R3
     15 016442  001372                        BNE 9$                     ;REPEAT UNTIL COUNT REACHES ZERO
     16
     17 016444                          10$:  POP R5                     ;GET CHACACTER FROM STACK
        016444  012605                                                             MOV (SP)+,R5
     18 016446  062705  000060                ADD #'0,R5                 ;CNVERT TO ASCII DIGIT
     19 016452  020527  000071                CMP R5,#'9                 ;IF GREATER THAN A 9
     20 016456  003402                        BLE 11$                    ; CONVERT TO A OR HIGHER
     21 016460  062705  000007                ADD #<'A-'9-1>,R5          ; FOR HEX DIGIT
     22 016464                          11$:  PRINT R5                   ;PRINT THE CHARACTER
        016464  110500                                                             MOVB R5,R0
        016466  004737  016510                                                     CALL CPNT
     23 016472  005302                        DEC R2                     ;REPEAT FOR ALL DIGITS
     24 016474  001363                        BNE 10$                    ; ON STACK
     25 016476                                POP <R4,R5,R3,R2>
        016476  012604                                                             MOV (SP)+,R4
        016500  012605                                                             MOV (SP)+,R5
        016502  012603                                                             MOV (SP)+,R3
        016504  012602                                                             MOV (SP)+,R2
     26 016506  000207                        RETURN
```

```
                                        ;PRINT ONE CHARACTER
                                        ;
                                        ;CALL WITH MACRO PRINT
016510  110037  003224          CPNT:   MOVB R0,ERRCHR
016514                                  PUSH R1
016514  010146                                                  MOV R1,-(SP)
016516  012701  003704                  MOV #ERRONE,R1
016522  120027  000015                  CMPB R0,#CR
016526  001002                          BNE 1$
016530  012701  003707                  MOV #ERRNL,R1
016534  000177  164462          1$:     JMP @PTYPE
016540                          PF:     PRINTF R1,#ERRCHR
016540  012746  003224                                          MOV     #ERRCHR,-(SP)
016544  010146                                                  MOV     R1,-(SP)
016546  012746  000002                                          MOV     #2,-(SP)
016552  010600                                                  MOV     SP,R0
016554  104417                                                  TRAP    C$PNTF
016556  062706  000006                                          ADD     #6,SP
016562  000435                          BR CPNTX
016564                          PB:     PRINTB R1,#ERRCHR
016564  012746  003224                                          MOV     #ERRCHR,-(SP)
016570  010146                                                  MOV     R1,-(SP)
016572  012746  000002                                          MOV     #2,-(SP)
016576  010600                                                  MOV     SP,R0
016600  104414                                                  TRAP    C$PNTB
016602  062706  000006                                          ADD     #6,SP
016606  000423                          BR CPNTX
016610                          PX:     PRINTX R1,#ERRCHR
016610  012746  003224                                          MOV     #ERRCHR,-(SP)
016614  010146                                                  MOV     R1,-(SP)
016616  012746  000002                                          MOV     #2,-(SP)
016622  010600                                                  MOV     SP,R0
016624  104415                                                  TRAP    C$PNTX
016626  062706  000006                                          ADD     #6,SP
016632  000411                          BR CPNTX
016634                          PS:     PRINTS R1,#ERRCHR
016634  012746  003224                                          MOV     #ERRCHR,-(SP)
016640  010146                                                  MOV     R1,-(SP)
016642  012746  000002                                          MOV     #2,-(SP)
016646  010600                                                  MOV     SP,R0
016650  104416                                                  TRAP    C$PNTS
016652  062706  000006                                          ADD     #6,SP
016656                          CPNTX:  POP R1
016656  012601                                                  MOV (SP)+,R1
016660  000207                          RETURN
```

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

016662  012737  016540  003222  LPNTF:  MOV #PF,PTYPE
016670  000413                        BR LPNT
016672  012737  016564  003222  LPNTB:  MOV #PB,PTYPE
016700  000407                        BR LPNT
016702  012737  016610  003222  LPNTX:  MOV #PX,PTYPE
016710  000403                        BR LPNT
016712  012737  016634  003222  LPNTS:  MOV #PS,PTYPE
016720                          LPNT:   PUSH <R2,R3,R4,R5>
016720  010246                                                          MOV R2,-(SP)
016722  010346                                                          MOV R3,-(SP)
016724  010446                                                          MOV R4,-(SP)
016726  010546                                                          MOV R5,-(SP)
016730  012102                        MOV (R1)+,R2              ;GET ADDRESS OF STRING
016732  010604                        MOV SP,R4                 ;COMPUTE ADDRESS OF ARGUMENTS
016734  062704  000012                ADD #10.,R4               ;  WHICH ARE NOW ON STACK (IF ANY)
016740                                PUSH R1                   ;SAVE RETURN ADDRESS
016740  010146                                                          MOV R1,-(SP)
016742  004737  015422                CALL OSTRNG               ;PRINT THE FORMATTED MESSAGE
016746                                POP <R0,R5,R4,R3,R2,R1>   ;RESTORE ALL REGISTERS
016746  012600                                                          MOV (SP)+,R0
016750  012605                                                          MOV (SP)+,R5
016752  012604                                                          MOV (SP)+,R4
016754  012603                                                          MOV (SP)+,R3
016756  012602                                                          MOV (SP)+,R2
016760  012601                                                          MOV (SP)+,R1
016762  062006                        ADD (R0)+,SP              ;ADJUST STACK POINTER OVER ARGUMENTS
016764  000110                        JMP @R0                   ;RETURN
```

```
                                        ;DIVIDE
                                        ;
                                        ;DIVIDE A 32 BIT UNSIGNED NUMBER BY A 16 BIT UNSIGNED NUMBER.
                                        ;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
                                        ;WILL NOT CHECK FOR DIVIDE BY ZERO.
                                        ;
                                        ;INPUTS:
                                        ;       R3 - LOW 16 BITS OF DIVIDEND
                                        ;       R4 - HIGH 16 BITS OF DIVIDEND
                                        ;       R0 - DIVISOR
                                        ;OUTPUTS:
                                        ;       R3 - LOW 16 BITS OF QUOTIENT
                                        ;       R4 - HIGH 16 BITS OF QUOTIENT
                                        ;       R5 - REMAINDER
                                        ;
016766                                  DIVIDE: PUSH R2
016766  010246                                                                 MOV R2,-(SP)
016770  012702  000040                          MOV #32.,R2           ;SET UP SHIFT COUNT
016774  005005                                  CLR R5                ;START WITH ZERO REMAINDER
016776  006303                          1$:     ASL R3                ;SHIFT LEFT INTO R5
017000  006104                                  ROL R4
017002  006105                                  ROL R5
017004  020005                                  CMP R0,R5             ;WILL DIVISOR GO INTO REMAINDER
017006  101002                                  BHI 2$                ;ONLY SUBTRACT IF IT WILL
017010  160005                                  SUB R0,R5             ;SUBTRACT DIVISOR
017012  005203                                  INC R3                ;PUT A ONE INTO QUOTIENT
017014  005302                          2$:     DEC R2                ;COUNT THE SHIFTS
017016  001367                                  BNE 1$
017020                                          POP R2
017020  012602                                                                 MOV (SP)+,R2
017022  000207                                  RETURN
```

```
                                        ;LOADDM
                                        ;
                                        ;LOAD AND START A DM PROGRAM INTO A CONTROLLER
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
                                        ;OUTPUTS:
                                        ;       IF LOAD SUCCEEDS - Z CLEAR
                                        ;                          CONTROLLER TABLE MARKED LOADED
                                        ;       IF ERROR - Z SET
                                        ;
017024  013701  002156          LOADDM: MOV DMPROG,R1                   ;GET STORAGE ADDRESS OF DM PROGRAM
017030  116165  000021  000042          MOVB DMTMO(R1),C.TOT(R5)        ;GET TIMEOUT VALUE
017036  105065  000043                  CLRB C.TOT+1(R5)
017042  016504  000004                  MOV C.VEC(R5),R4                ;GET VECTOR OF UDA
017046                                  AND CT.VEC,R4
017046  042704  177000                                                          BIC #^C<CT.VEC>,R4
017052  010501                          MOV R5,R1                       ;GET INTERRUPT SERVICE LINK
017054  062701  000006                  ADD #C.JSR,R1
017060                                  SETVEC R4,R1,#PRI07             ;SET UP INTERRUPT VECTOR
017060  012746  000340                                                          MOV     #PRI07,-(SP)
017064  010146                                                                  MOV     R1,-(SP)
017066  010446                                                                  MOV     R4,-(SP)
017070  012746  000003                                                          MOV     #3,-(SP)
017074  104437                                                                  TRAP    C$SVEC
017076  062706  000010                                                          ADD     #10,SP
                                                                        ;INITIALIZE UDA WITH SMALLEST
017102  004737  017704                  CALL UDAINT                     ; RING BUFFER AND INTERRUPTS ENABLED
017106  001444                          BEQ LOADER                      ;BRANCH IF AN ERROR
```

```
    1 017110  012700  000002          MOV #OP.ESP,R0                  ;BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET
    2 017114  004737  017224          CALL BLDCMD
    3 017120  013764  002156  000124  MOV DMPROG,HC.CPK+P.UADR(R4)    ;LOAD MAIN PROGRAM ADDRESS
    4 017126  017764  163024  000120  MOV @DMPROG,HC.CPK+P.BCNT(R4)   ;   AND SIZE
    5 017134  013764  002156  000140  MOV DMPROG,HC.CPK+P.OVRL(R4)    ;LOAD OVERLAY ADDRESS
    6 017142  067764  163010  000140  ADD @DMPROG,HC.CPK+P.OVRL(R4)
    7 017150  004737  017310          CALL SNDCMD                     ;SEND COMMAND TO UDA
    8 017154  004737  017430          CALL WAITMS                     ;WAIT FOR MESSAGE RESPONSE
    9 017160  001417                  BEQ LOADER      ;ABORT IF NO RESPONSE
   10 017162  032764  000037  000032  BIT #ST.MSK,HC.MPK+P.STS(R4)    ;CHECK FOR ERRORS
   11 017170  001007                  BNE LOADE1
   12 017172  042765  000024  000012  BIC #CT.CMD+CT.REQ,C.FLG(R5)    ;CLEAR COMMAND OUTSTANDING FLAG
   13 017200  052765  000002  000012  BIS #CT.RN,C.FLG(R5)             ;SET DM PROGRAM RUNNING FLAG
   14 017206  000207                  RETURN
```

```
                                ;UDA FAILED TO DOWNLINE LOAD DM PROGRAM

                                LOADE1: ERRDF 34,,ERR034
017210  104455                                                          TRAP    C$ERDF
017212  000042                                                          .WORD   34
017214  000000                                                          .WORD   0
017216  012670                                                          .WORD   ERR034
017220  000264                LOADER: SEZ                     ;SET Z TO INDICATE ERROR OCCURRED
017222  000207                        RETURN
```

```
                                        ;BLDCMD
                                        ;
                                        ;BUILD A COMMAND IN COMMAND PACKET
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       R0 - COMMAND CODE
                                        ;OUTPUTS:
                                        ;       R4 - ADDRESS OF HOST COMM AREA
                                        ;       COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL OTHER FIELDS CLEARED.
                                        ;       CMD REFERRENCE NUMBER IN CONTROLLER TABLE INCREMENTED AND RESULT
                                        ;       IN COMMAND PACKET.
                                        ;       R0 - CONTENTS DESTROYED
                                        ;
 017224                                 BLDCMD: PUSH <R1,R0>
 017224  010146                                                                        MOV R1,-(SP)
 017226  010046                                                                        MOV R0,-(SP)
 017230  016504  000014                         MOV C.RING(R5),R4          ;GET ADDRESS OF HOST COMM AREA
 017234  010400                                 MOV R4,R0                  ;COPY TO R0
 017236  062700  000100                         ADD #HC.CEV,R0             ;COMPUTE ADDRESS OF COMMAND ENVELOPE
 017242  012720  000060                         MOV #HC.PSZ,(R0)+          ;LOAD PACKET LENGTH
 017246  012701  001000                         MOV #DUP,R1                ;LOAD DIAG CIRCUIT IDENTIFIER
 017252  022716  000031                         CMP #OP.MWR,(SP)           ;IF CODE IS MAINTENANCE WRITE
 017256  001002                                 BNE BLDC0                  ;  GET OTHER CIRCUIT IDENTIFIER
 017260  012701  177777                         MOV #DIAG,R1
 017264  010120                         BLDC0:  MOV R1,(R0)+               ;PUT IDENTIFIER INTO PACKET
 017266  012701  000030                         MOV #<HC.PSZ>/2,R1         ;GET WORDS TO CLEAR
 017272  005020                         BLDC1:  CLR (R0)+                  ;CLEAR PACKET
 017274  005301                                 DEC R1
 017276  001375                                 BNE BLDC1
 017300                                         POP HC.CPK+P.OPCD(R4)      ;PUT OPCODE IN PACKET
 017300  012664  000114                                                             MOV (SP)+,HC.CPK+P.OPCD(R4)
 017304                                         POP R1                     ;RESTORE R1
 017304  012601                                                                     MOV (SP)+,R1
 017306  000207                                 RETURN
```

J9


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


 017310                         SNDCMD: PUSH <R0,R1>
 017310  010046                                                                 MOV R0,-(SP)
 017312  010146                                                                 MOV R1,-(SP)
 017314  016504  000014                 MOV C.RING(R5),R4       ;LOAD R4 WITH HOST COMM AREA ADDRESS
 017320  005265  000050                 INC C.REF(R5)           ;INCREMENT CMD REFERENCE NUMBER
 017324  016564  000050  000104         MOV C.REF(R5),HC.CPK+P.CRF(R4) ;PUT IN PACKET
 017332  012764  140000  000006         MOV #RG.OWN+RG.FLG,HC.MCT(R4) ;MARK MESSAGE PACKET AVAILABLE
 017340  012764  100000  000012         MOV #RG.OWN,HC.CCT(R4)  ;MARK COMMAND TO UDA
 017346  005775  000000                 TST @(R5)               ;TELL UDA COMMAND IS THERE
 017352  052765  000004  000012         BIS #CT.CMD,C.FLG(R5)   ;MARK COMMAND ISSUED
 017360                                 POP <R1,R0>
 017360  012601                                                                 MOV (SP)+,R1
 017362  012600                                                                 MOV (SP)+,R0
 017364  000207                         RETURN
```

```
                                        ;CLRBUF
                                        ;
                                        ;CLEAR THE SPECIFIED DATA BUFFER IN THE HOST COMM AREA
                                        ;AND LOAD BUFFER DESCRIPTOR IN COMMAND PACKET TO THE BUFFER
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       R4 - ADDRESS OF HOST COMM AREA
                                        ;       R0 - OFFSET INTO HOST COMM AREA TO DATA BUFFER
                                        ;OUTPUTS:
                                        ;       DATA BUFFER CLEARED
                                        ;       COMMAND PACKET POINTING TO BUFFER
                                        ;       BYTE COUNT SET TO SIZE OF BUFFER
                                        ;       R4 - ADDRESS OF DATA BUFFER
                                        ;
017366                                  CLRBUF: PUSH <R0,R1>
017366  010046                                                                         MOV R0,-(SP)
017370  010146                                                                         MOV R1,-(SP)
017372  060400                                  ADD R4,R0                       ;ADD START OF HOST COMM AREA TO OFFSET
017374  010064  000124                         MOV R0,HC.CPK+P.UADR(R4)        ;PUT BUFFER ADDRESS IN COMMAND PACKET
017400  012764  000244  000120                 MOV #HC.BSZ,HC.CPK+P.BCNT(R4)   ;PUT SIZE OF BUFFER IN COMMAND PACKET
017406  010004                                  MOV R0,R4                       ;PUT BUFFER ADDRESS IN R4
017410  012701  000122                         MOV #HC.BSZ/2,R1                ;GET SIZE OF BUFFER IN WORDS
017414  005020                          CLRBFL: CLR (R0)+                       ;CLEAR ALL THE WORDS
017416  005301                                  DEC R1
017420  001375                                  BNE CLRBFL
017422                                          POP <R1,R0>
017422  012601                                                                         MOV (SP)+,R1
017424  012600                                                                         MOV (SP)+,R0
017426  000207                                  RETURN
```

L 9


```
                                        ;WAITMS
                                        ;
                                        ;WAIT FOR UDA TO RESPOND WITH A MESSAGE PACKET
                                        ;
                                        ;INPUTS:
                                        ;       R5 - ADDRESS OF CONTROLLER TABLE
                                        ;OUTPUTS:
                                        ;       Z CLEAR IF NO ERROR
                                        ;       Z SET IF ERROR, MESSAGE PRINTED
                                        ;
017430                                  WAITMS: PUSH <R0,R1>
017430  010046                                                                  MOV R0,-(SP)
017432  010146                                                                  MOV R1,-(SP)
017434  012700  000036                       MOV #30.,R0             ;SET TIME OUT VALUE OF 30 SECONDS
017440  010501                               MOV R5,R1               ;POINT TO TIME OUT COUNTER
017442  062701  000036                       ADD #C.TO,R1
017446  004737  017622                       CALL SETTO
017452  011500                         1$:   MOV (R5),R0             ;GET ADDRESS OF UDAIP REGISTER
017454  032765  000010  000012             BIT #CT.MSG,C.FLG(R5)   ;LOOK IF INTERRUPT OCCURRED
017462  001030                             BNE 3$                  ;BRANCH IF SO
017464  016001  000002                     MOV 2(R0),R1            ;LOOK AT UDASA REGISTER
017470  001034                             BNE 4$                  ;BRANCH IF ERROR CODE PRESENT
017472                                     BREAK
017472  104422                                                                  TRAP    C$BRK
017474  005737  003206                     TST KW.CSR              ;SEE IF A CLOCK ON SYSTEM
017500  001764                             BEQ 1$
017502  023765  003220  000040             CMP KW.EL+2,C.TOH(R5)   ;CHECK IF TIMEOUT HAS HAPPENED
017510  101005                             BHI 2$
017512  001357                             BNE 1$
017514  023765  003216  000036             CMP KW.EL,C.TO(R5)
017522  103753                             BLO 1$
017524                                 2$:   ERRDF 36.,,ERR036
017524  104455                                                                  TRAP    C$ERDF
017526  000044                                                                  .WORD   36
017530  000000                                                                  .WORD   0
017532  012676                                                                  .WORD   ERR036
017534                                     POP <R1,R0>
017534  012601                                                                  MOV (SP)+,R1
017536  012600                                                                  MOV (SP)+,R0
017540  000264                             SEZ
017542  000207                             RETURN
```

M9


```
      1 017544  042765  000010  000012  3$:     BIC #CT.MSG,C.FLG(R5)          ;CLEAR MESSAGE RECEIVED FLAG
      2 017552                                  POP <R1,R0>
        017552  012601                                                          MOV (SP)+,R1
        017554  012600                                                          MOV (SP)+,R0
      3 017556  000244                          CLZ                            ;GIVE NO ERROR RETURN
      4 017560  000207                          RETURN
      5 017562                            4$:   ERRDF 37,,ERR037
        017562  104455                                                          TRAP    C$ERDF
        017564  000045                                                          .WORD   37
        017566  000000                                                          .WORD   0
        017570  012710                                                          .WORD   ERR037
      6 017572                                  POP <R1,R0>
        017572  012601                                                          MOV (SP)+,R1
        017574  012600                                                          MOV (SP)+,R0
      7 017576  000264                          SEZ
      8 017600  000207                          RETURN
```

N9


```
                                ;NXMI
                                ;
                                ;NON-EXISTANT MEMORY SERVICE ROUTINE
                                ;
                                ;INPUTS:
                                ;       NXMAD SET TO ZERO
                                ;OUTPUTS:
                                ;       NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
                                
017602                          BGNSRV NXMI
017602                                                                  NXMI::
                                
017602  012737  177777  002172          MOV #-1,NXMAD
                                
017610                          ENDSRV
017610                                                                  L10031:
017610  000002                                                                  RTI
```

```
                                ;UDASRV
                                ;
                                ;UDA INTERRUPT SERVICE ROUTINE. MARKS UDA CONTROLLER TABLE THAT AN
                                ;INTERRUPT HAS BEEN RECEIVED.
                                ;
                                ;THIS ROUTINE IS CALLED BY A [JSR R0,UDASRV] INSTRUCTION FROM WITHIN
                                ;THE CONTROLLER TABLE. THE PC STORED IN R0 IS THE ADDRESS OF THE C.FLG
                                ;WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
                                ;OF R0 FOLLOWED BY THE INTERRUPTED PC AND PS.
                                ;
                                ;INPUTS:
                                ;       R0 - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
                                ;       STACK - SAVED CONTENTS OF R0
                                ;OUTPUTS:
                                ;       CT.CMD CLEARED AND CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
                                ;       R0 - RESTORED FROM STACK
                                
017612                          BGNSRV UDASRV
017612                                                                  UDASRV::
017612  052710  000010                  BIS #CT.MSG,(R0)               ;SET CT.MSG
017616                                   POP R0                         ;RESTORE R0
017616  012600                                                                  MOV (SP)+,R0
017620                          ENDSRV
017620                                                                  L10032:
017620  000002                                                                  RTI
```

```
                                ;SETTO
                                ;
                                ;SET TIMEOUT COUNTER TO SOME NUMBER OF SECONDS FROM CURRENT TIME.
                                ;
                                ;INPUTS:
                                ;       R0 - NUMBER OF SECONDS FOR TIMEOUT
                                ;       R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT
                                ;OUTPUTS:
                                ;       R0 - CONTENTS DESTROYED
                                ;       R1 - INCREMENTED BY 2
                                ;
                                ;COMPUTE CLOCK TICKS TIL TIMEOUT
                                ;
017622                          SETTO:  PUSH <R2,R3>
017622  010246                                                                  MOV R2,-(SP)
017624  010346                                                                  MOV R3,-(SP)
017626  005002                          CLR R2                  ;CLEAR PRODUCT
017630  013703  003214                  MOV KW.HZ,R3            ;GET MULTIPLICAND
017634  006200                  SETO0:  ASR R0                  ;SHIFT MULTIPLIER TO RIGHT
017636  103001                          BCC SETO1               ;IF A ONE BIT SHIFTED OUT
017640  060302                          ADD R3,R2               ;  ADD MULTIPLICAND TO PRODUCT
017642  006303                  SETO1:  ASL R3                  ;DOUBLE THE MULTIPLICAND
017644  005700                          TST R0
017646  001372                          BNE SETO0               ;CONTINUE UNTIL MULTIPLIER IS ZERO
                                ;
                                ;GET CURRENT TIME
                                ;
017650  013700  003216          SETO2:  MOV KW.EL,R0            ;GET TIME
017654  013703  003220                  MOV KW.EL+2,R3
017660  020037  003216                  CMP R0,KW.EL            ;IF CHANGED DURING RETRIEVAL
017664  001371                          BNE SETO2               ;  GET IT AGAIN
                                ;
                                ;ADD TIME TIL TIMEOUT
                                ;
017666  060200                          ADD R2,R0               ;ADD
017670  005503                          ADC R3
                                ;
                                ;PUT RESULT IN STORAGE
                                ;
017672  010021                          MOV R0,(R1)+
017674  010311                          MOV R3,(R1)
                                ;
017676                                  POP <R3,R2>
017676  012603                                                                  MOV (SP)+,R3
017700  012602                                                                  MOV (SP)+,R2
017702  000207                          RETURN
```

```
                                ;UDAINT
                                ;
                                ;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

017704  016502  000014  UDAINT: MOV C.RING(R5),R2                ;GET FIRST ADDRESS OF RING BUFFER
017710  012703  000006          MOV #<HC.RSZ*2+HC.ISZ>/2,R3       ;GET SIZE OF RING BUFFER
017714  012722  177777  UDAI1L: MOV #-1,(R2)+                    ;WRITE ONES TO BUFFER
017720  005303                  DEC R3                          ;COUNT THE WORDS IN BUFFER
017722  003374                  BGT UDAI1L                      ;LOOP UNTIL ENTIRE BUFFER WRITTEN

                                ;DO THE INITIALIZATION

017724  004737  020152          CALL UDAIST                     ;DO FIRST THREE STEPS
017730  103506                  BCS UDAIEX                      ;GET OUT IF UDA MICROCODE REPORTED FAILURE
017732  012364  000002          MOV (R3)+,2(R4)                 ;WRITE NEXT WORD TO UDASA REGISTER
017736  012703  000310          MOV #200.,R3                    ;GET TRY COUNTER
017742  016402  000002  UDAI1A: MOV 2(R4),R2                   ;LOOK AT UDASA
017746  001407                  BEQ UDAI1C
017750  005303                  DEC R3
017752  001373                  BNE UDAI1A
017754                          ERRDF 24.,ERR024
017754  104455                                                  TRAP    C$ERDF
017756  000030                                                  .WORD   24
017760  000000                                                  .WORD   0
017762  012564                                                  .WORD   ERR024
017764  000470                  BR UDAIEX
017766  010264  000002  UDAI1C: MOV R2,2(R4)                   ;WRITE 0 TO UDASA (PURGE)
017772  011402                  MOV (R4),R2                     ;READ FROM UDAIP (POLL)
017774  004737  020470          CALL UDARSP                     ;WAIT FOR STEP OR ERROR BIT
020000  103462                  BCS UDAIEX                      ;GET OUT IF UDA MICROCODE REPORTED FAILURE
020002  042702  174017          BIC #^C<SA.CNT>,R2              ;CLEAR OTHER BITS
020006  006202                  ASR R2                          ;MOVE TO RIGHT OF REGISTER
020010  006202                  ASR R2
020012  006202                  ASR R2
020014  006202                  ASR R2
020016  020227  000006          CMP R2,#6                       ;CONTROLLER MODEL MUST BE 6
020022  001410                  BEQ UDAI2
020024  020227  000015          CMP R2,#13.                     ;  OR 13
020030  001405                  BEQ UDAI2
020032                          ERRDF 14.,ERR014                ;REPORT CONTROLLER NEEDS NEW REVISION
020032  104455                                                  TRAP    C$ERDF
020034  000016                                                  .WORD   14
```

E10


```
        020036  000000                                          .WORD   0
        020040  012370                                          .WORD   ERR014
     52 020042  000441          BR UDAIEX
```

```
                                        ;CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
                                        ;
020044  016502  000014          UDAI2:  MOV C.RING(R5),R2               ;GET FIRST ADDRESS OF RING BUFFER
020050  010201                          MOV R2,R1                       ;SAVE FOR ERROR MESSAGE
020052  012703  000006                  MOV #<HC.RSZ*2+HC.ISZ>/2,R3     ;GET SIZE OF RING BUFFER
020056  005722                  UDAI2L: TST (R2)+                       ;CHECK WORD IN BUFFER
020060  001003                          BNE UDAI2E                      ;GO TO ERROR REPORTER IF NOT ZERO
020062  005303                          DEC R3                          ;COUNT THE WORDS IN BUFFER
020064  003374                          BGT UDAI2L                      ;LOOP UNTIL ALL WORDS CHECKED
020066  000405                          BR UDAI3
                                        ;
020070                          UDAI2E: ERRDF 23,,ERR023                ;REPORT BUFFER NOT CLEARED
020070  104455                                                                  TRAP    C$ERDF
020072  000027                                                                  .WORD   23
020074  000000                                                                  .WORD   0
020076  012506                                                                  .WORD   ERR023
020100  000422                          BR UDAIEX
                                        ;
                                        ;SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION
                                        ;
020102                          UDAI3:
020102  012700  000001                  MOV #SA.GO,R0
020106  010064  000002                  MOV R0,2(R4)                    ;SEND TO UDA
020112  016501  000014                  MOV C.RING(R5),R1
020116  010161  000004                  MOV R1,HC.MSG(R1)
020122  062761  000020  000004          ADD #HC.MPK,HC.MSG(R1)
020130  010161  000010                  MOV R1,HC.CMD(R1)
020134  062761  000104  000010          ADD #HC.CPK,HC.CMD(R1)
020142  000244                          CLZ                             ;CLEAR Z AS NO ERROR INDICATION
020144  000207                          RETURN
                                        ;
                                        ;ERROR RETURN
                                        ;
020146  000264                  UDAIEX: SEZ                             ;SET Z TO INDICATE ERROR OCCURRED
020150  000207                          RETURN
```

```
                                ;UDAIST
                                ;
                                ;START THE INITIALIZATION PROCESS ON THE SELECTED UDA.
                                ;STOP BEFORE WRITING THE THIRD WORD SO UDA DOES NOT
                                ;ATTEMPT ANY UNIBUS TRANSFERS.
                                ;
                                ;INPUTS:
                                ;       R5 - ADDRESS OF CONTROLLER TABLE

                                ;LOAD TABLE OF DATA TO SEND TO UDASA REGISTER

020152                          UDAIST: BREAK
020152  104422                                                          TRAP    C$BRK
020154                                  PUSH R1
020154  010146                                                          MOV R1,-(SP)
020156  016504  000004                 MOV C.VEC(R5),R4
020162                                  AND CT.VEC,R4
020162  042704  177000                                                  BIC #^C<CT.VEC>,R4
020166  006204                          ASR R4
020170  006204                          ASR R4
020172  052704  100000                 BIS #SA.STP,R4          ;SET STEP BIT IN DATA WORD
020176  010437  020370                 MOV R4,UDAID1           ;LOAD INTERRUPT VECTOR
020202  016537  000014  020374         MOV C.RING(R5),UDAID2   ;  LOAD MEMORY ADDRESS
020210  062737  000004  020374         ADD #HC.MSG,UDAID2      ;  OF FIRST RESPONSE RING

                                ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER

020216  016504  000000                 MOV C.UADR(R5),R4       ;GET ADDRESS OF UDAIP REGISTER
020222  005037  002172                 CLR NXMAD               ;CLEAR MEMORY ERROR FLAG
020226                                  SETVEC #4,#NXMI,#PRI07  ;SET UP VECTOR 4
020226  012746  000340                                                  MOV     #PRI07,-(SP)
020232  012746  017602                                                  MOV     #NXMI,-(SP)
020236  012746  000004                                                  MOV     #4,-(SP)
020242  012746  000003                                                  MOV     #3,-(SP)
020246  104437                                                          TRAP    C$SVEC
020250  062706  000010                                                  ADD     #10,SP
020254  005764  000002                 TST 2(R4)               ;ACCESS UDASA REGISTER
020260  005014                          CLR (R4)                ;WRITE TO UDAIP
020262                                  CLRVEC #4               ;GIVE UP THE VECTOR
020262  012700  000004                                                  MOV     #4,R0
020266  104436                                                          TRAP    C$CVEC
020270  005737  002172                 TST NXMAD               ;SEE IF A MEMORY ERROR OCCURRED
020274  001406                          BEQ UDAISG
020276                                  ERRDF 20,,ERR020
020276  104455                                                          TRAP    C$ERDF
020300  000024                                                          .WORD   20
020302  000000                                                          .WORD   0
020304  012404                                                          .WORD   ERR020
020306  000261                          SEC
020310  000424                          BR UDAISE
```

H10


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

020312  012737  004000  020626  UDAISG: MOV #SA.S1,UDARSD               ;STORE RESPONSE MASK
020320  012703  020366                  MOV #UDAIDT,R3                  ;AND INDEX TO TABLE

                                        ;WAIT FOR AND CHECK RESPONSE DATA

020324  004737  020470          UDAISL: CALL UDARSP                     ;WAIT FOR STEP OR ERROR BITS
020330  103414                          BCS UDAISE                      ;EXIT IF ERROR
020332  004733                          CALL @(R3)+                     ;CALL RESPONSE CHECKER FOR STEP
020334  103412                          BCS UDAISE                      ;GET OUT IF ERROR
020336  006337  020626                  ASL UDARSD                      ;SHIFT TO NEXT STEP BIT
020342  032737  040000  020626          BIT #SA.S4,UDARSD               ;CHECK IF NOW AT STEP 4
020350  001003                          BNE UDAISX                      ;GET OUT IF SO
020352  012364  000002                  MOV (R3)+,2(R4)                 ;WRITE DATA TO UDASA REGISTER
020356  000762                          BR UDAISL                       ;STAY IN LOOP

020360  000241          UDAISX: CLC                                     ;CLEAR CARRY FOR NO ERROR INDICATION
020362                  UDAISE: POP R1
020362  012601                                                          MOV (SP)+,R1
020364  000207                  RETURN
```

```
                                ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION
                                
020366  020402                  UDAIDT: .WORD UDAIR1                    ;FIRST WORD RESPONSE CHECK ROUTINE
020370  000000                  UDAID1: .WORD 0                         ;FIRST WORD TO SEND TO UDASA
020372  020414                          .WORD UDAIR2                    ;SECOND WORD RESPONSE CHECK ROUTINE
020374  000000                  UDAID2: .WORD 0                         ;SECOND WORD TO SEND TO UDASA
020376  020434                          .WORD UDAIR3                    ;THIRD WORD RESPONSE CHECK ROUTINE
020400  100000                  UDAID3: .WORD SA.TST                    ;THIRD WORD TO SEND TO UDASA
                                
                                ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
                                ;CHECK FOR PROPER CONTROLLER TYPE
                                
020402  012701  004400          UDAIR1: MOV #SA.S1+SA.DI,R1             ;SET STEP ONE BIT
020406  042702  001140                  BIC #<SA.QB+SA.MP+SA.SM>,R2     ;MASK OFF UNWANTED BITS
020412  000416                          BR UDAIRC                       ;NOW COMPARE
                                
                                ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
                                ;CHECK FOR ECHO OF INTI AND VECTOR
                                
020414  013701  020370          UDAIR2: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
020420  000301                          SWAB R1                         ;GET HIGH 8 BITS
020422  042701  177400                  BIC #177400,R1
020426  052701  010000                  BIS #SA.S2,R1                   ;SET STEP 2 BIT
020432  000406                          BR UDAIRC                       ;NOW COMPARE
                                
                                ;RESPONSE CHECK FOR THIRD WORD FROM UDASA
                                ;CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
                                
020434  013701  020370          UDAIR3: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
020440  042701  177400                  BIC #177400,R1                  ;JUST LOW 8 BITS
020444  052701  020000                  BIS #SA.S3,R1                   ;SET STEP 3 BIT
                                
                                ;COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
                                
020450  020102                  UDAIRC: CMP R1,R2                       ;COMPARE THE DATA
020452  001405                          BEQ UDAIRX                      ;EXIT IF COMPARED CORRECTLY
020454                                  ERRDF 25,,ERR025                ;REPORT ERROR
020454  104455                                                                  TRAP    C$ERDF
020456  000031                                                                  .WORD   25
020460  000000                                                                  .WORD   0
020462  012600                                                                  .WORD   ERR025
020464  000261                          SEC
020466  000207                  UDAIRX: RETURN
```

J10


```
                                ;UDARSP
                                ;
                                ;WAIT FOR UDA TO RESPOND WITH DATA IN UDASA REGISTER.
                                ;EITHER STEP BIT FROM MASK IN LOCATION UDARSD OR ERROR BIT
                                ;WILL CAUSE A TERMINATION.
                                ;AN ERROR MESSAGE WILL BE PRINTED IF THE UDA DOES NOT RESPOND
                                ;IN 10 SECONDS OR IF ERROR SETS.
                                ;
                                ;INPUTS:
                                ;       UDASRD - MASK OF STEP BIT TO LOOK FOR
                                ;       R5 - ADDRESS OF CONTROLLER TABLE
                                ;       R4 - ADDRESS OF UDAIP REGISTER
                                ;OUTPUTS:
                                ;       ERROR MESSAGE IF TIME OUT ON RESPONSE OR ERROR BIT SETS
                                ;       R2 - DATA FROM UDASA REGISTER
                                ;       CARRY SET IF ERROR BIT SETS OR TIME OUT
                                
018 020470                      UDARSP: PUSH R1
    020470  010146                                                              MOV R1,-(SP)
019 020472  052737  100000  020626      BIS #SA.ERR,UDARSD          ;SET ERROR BIT IN MASK WORD
020 020500  012700  000012              MOV #10.,R0                 ;SET UP FOR 10 SECOND TIMEOUT
021 020504  010501                      MOV R5,R1                   ;POINT TO COUNTER IN CONTROLLER TABLE
022 020506  062701  000036              ADD #C.TO,R1
023 020512  004737  017622              CALL SETTO
024 020516                              POP R1
    020516  012601                                                              MOV (SP)+,R1
025 020520  033764  020626  000002  UDARS1: BIT UDARSD,2(R4)           ;LOOK AT ERROR AND STEP BIT
026 020526  001024                      BNE UDARS2                  ;BRANCH IF EITHER SET
027 020530                              BREAK
    020530  104422                                                              TRAP    C$BRK
028 020532  005737  003206              TST KW.CSR                  ;SEE IF CLOCK ON SYSTEM
029 020536  001770                      BEQ UDARS1
030 020540  023765  003220  000040      CMP KW.EL+2,C.TOH(R5)       ;CHECK IF TIME OUT OCCURRED
031 020546  101005                      BHI 1$
032 020550  001363                      BNE UDARS1
033 020552  023765  003216  000036      CMP KW.EL,C.TO(R5)
034 020560  103757                      BLO UDARS1
035 020562  016402  000002      1$:     MOV 2(R4),R2                ;GET REGISTER CONTENTS
036 020566                              ERRDF 22,,ERR022            ;REPORT TIME OUT ERROR
    020566  104455                                                              TRAP    C$ERDF
    020570  000026                                                              .WORD   22
    020572  000000                                                              .WORD   0
    020574  012460                                                              .WORD   ERR022
037 020576  000407                      BR UDARSE
```

K10


```
                                        ;CHECK IF ERROR BIT SET
                                        
020600  016402  000002          UDARS2: MOV 2(R4),R2                    ;GET REGISTER CONTENTS
020604  100006                          BPL UDARSX                      ;EXIT IF ERROR NOT SET
020606                                  ERRDF 21,,ERR021                ;REPORT ERROR INFO
020606  104455                                                                  TRAP    C$ERDF
020610  000025                                                                  .WORD   21
020612  000000                                                                  .WORD   0
020614  012416                                                                  .WORD   ERR021
020616  000261          UDARSE: SEC
020620  000207                  RETURN
                                        
                                        ;NORMAL EXIT
                                        
020622  000241          UDARSX: CLC                                     ;CLEAR CARRY AS NO ERROR INDICATION
020624  000207                  RETURN
                                        
                                        ;LOCATION FOR STEP BIT MASK
                                        
020626  000000          UDARSD: .WORD 0                                 ;LOAD BY CALLING ROUTINE
```

```
                                ;KW11I
                                ;
                                ;CLOCK INTERRUPT SERVICE ROUTINE
                                ;
020630                          BGNSRV KW11I
020630                                                                  KW11I::
020630  062737  000001  003216          ADD #1,KW.EL                  ;COUNT THE INTERRUPT
020636  005537  003220                  ADC KW.EL+2
020642  012777  000105  162336          MOV #KWOUT.,@KW.CSR           ;RESTART THE CLOCK
020650                          ENDSRV
020650                                                                  L10033:
020650  000002                                                                  RTI
```

```
                                        ;RNTIME
                                        ;
                                        ;PRINT RUNTIME
                                        ;
                                        ;INPUTS:
                                        ;       KW.EL - CONTAINS ELAPSED TIME
                                        ;       KW.HZ - HERTZ OF CLOCK
                                        ;OUTPUTS:
                                        ;       IF CLOCK ON SYSTEM:
                                        ;          "  RUNTIME HH:MM:SS " PRINTED
                                        ;       IF NO CLOCK: ONE SPACE IS PRINTED
                                        ;
020652  005737  003206          RNTIME: TST KW.CSR                      ;CHECK IF A CLOCK PRESENT
020656  001465                          BEQ RNTIMX                      ;BRANCH IF NOT
020660                                  PUSH <R0,R3,R4,R5>
020660  010046                                                                  MOV R0,-(SP)
020662  010346                                                                  MOV R3,-(SP)
020664  010446                                                                  MOV R4,-(SP)
020666  010546                                                                  MOV R5,-(SP)
020670  013703  003216                  MOV KW.EL,R3                    ;GET ELAPSED TIME
020674  013704  003220                  MOV KW.EL+2,R4
020700  013700  003214                  MOV KW.HZ,R0                    ;GET SPEED OF CLOCK
020704  004737  016766                  CALL DIVIDE                     ;COMPUTE SECONDS OF ELAPSED TIME
020710  012700  000074                  MOV #60.,R0                     ;NOW DIVIDE BY 60
020714  004737  016766                  CALL DIVIDE                     ; TO COMPUTE MINUTES
020720                                  PUSH R5                         ;SAVE REMAINDER AS SECONDS
020720  010546                                                                  MOV R5,-(SP)
020722  004737  016766                  CALL DIVIDE                     ;DIVIDE BY 60 AGAIN
020726                                  PNT RNTIM,R3                    ;PRINT HOURS
020726  010346                                                                  MOV R3,-(SP)
020730  004137  016720                                                          JSR R1,LPNT
020734  003712                                                                  .WORD RNTIM
020736  000002                                                                  .WORD PNT.CT
020740  020527  000011                  CMP R5,#9.                      ;IF MINUTES 9 OR LESS
020744  003004                          BGT 1$
020746                                  PRINT #'0                       ;PRINT A LEADING ZERO
020746  112700  000060                                                          MOVB #'0,R0
020752  004737  016510                                                          CALL CPNT
020756                          1$:     PNT RNTIM1,R5                   ;NOW PRINT MINUTES
020756  010546                                                                  MOV R5,-(SP)
020760  004137  016720                                                          JSR R1,LPNT
020764  003735                                                                  .WORD RNTIM1
020766  000002                                                                  .WORD PNT.CT
020770                                  POP R5                          ;GET SECONDS
020770  012605                                                                  MOV (SP)+,R5
020772  020527  000011                  CMP R5,#9.                      ;IF 9 OR LESS
020776  003004                          BGT 2$
021000                                  PRINT #'0                       ;PRINT A LEADING ZERO
021000  112700  000060                                                          MOVB #'0,R0
021004  004737  016510                                                          CALL CPNT
021010                          2$:     PNT RNTIM2,R5                   ;NOW PRINT SECONDS
021010  010546                                                                  MOV R5,-(SP)
021012  004137  016720                                                          JSR R1,LPNT
021016  003743                                                                  .WORD RNTIM2
021020  000002                                                                  .WORD PNT.CT
021022                                  POP <R5,R4,R3,R0>               ;HOUPS IN R3
021022  012605                                                                  MOV (SP)+,R5
```

N10

GLOBAL SUBROUTINES SECTION

```
                021024  012604                                          MOV (SP)+,R4
                021026  012603                                          MOV (SP)+,R3
                021030  012600                                          MOV (SP)+,R0
     35         021032                  RNTIMX: PRINT <#' >                     ;PRINT A SPACE
                021032  112700  000040                                  MOVB #' ,R0
                021036  004737  016510                                  CALL CPNT
     36         021042  000207                  RETURN
```

```
      1 021044                          DATE:   GMANID DATEQ,DATEI,A,-1,1,11.,YES       ;GET DATE
        021044  104443                                                                    TRAP    C$GMAN
        021046  000406                                                                    BR      10000$
        021050  003270                                                                    .WORD   DATEI
        021052  000152                                                                    .WORD   T$CODE
        021054  003544                                                                    .WORD   DATEQ
        021056  177777                                                                    .WORD   -1
        021060  000001                                                                    .WORD   T$LOLIM
        021062  000013                                                                    .WORD   T$HILIM
        021064                                                                  10000$:
      2 021064  012705  003270                      MOV #DATEI,R5   ;GET POINTER TO ANSWER
      3 021070  121527  000060                      CMPB (R5),#'0
      4 021074  103443                              BLO DERR
      5 021076  122527  000071          DAY:        CMPB (R5)+,#'9
      6 021102  101040                              BHI DERR
      7 021104  121527  000055                      CMPB (R5),#'-
      8 021110  001406                              BEQ DAS1
      9 021112  121527  000060                      CMPB (R5),#'0
     10 021116  103432                              BLO DERR
     11 021120  122527  000071                      CMPB (R5)+,#'9
     12 021124  101027                              BHI DERR
     13 021126  122527  000055          DAS1:       CMPB (R5)+,#'-
     14 021132  001024                              BNE DERR
     15 021134  012704  000014                      MOV #12.,R4     ;GET NUMBER OF MONTH
     16 021140  012703  003345                      MOV #MONTHS,R3  ;GET POINTER TO MONTH NAMES
     17 021144  005000                  MON1:       CLR R0
     18 021146  121523                              CMPB (R5),(R3)+
     19 021150  001401                              BEQ MON2
     20 021152  005200                              INC R0
     21 021154  126523  000001          MON2:       CMPB 1(R5),(R3)+
     22 021160  001401                              BEQ MON3
     23 021162  005200                              INC R0
     24 021164  126523  000002          MON3:       CMPB 2(R5),(R3)+
     25 021170  001401                              BEQ MON4
     26 021172  005200                              INC R0
     27 021174  005700                  MON4:       TST R0
     28 021176  001407                              BEQ MON5
     29 021200  005304                              DEC R4
     30 021202  001360                              BNE MON1
     31 021204                          DERR:       PNTF DATEX
        021204  004137  016662                                                          JSR R1,LPNTF
        021210  012141                                                                  .WORD DATEX
        021212  000000                                                                  .WORD PNT.CT
     32 021214  000713                              BR DATE
     33 021216  012701  003304          MON5:       MOV #DATEQ,R1   ;GET POINTER TO DATE FOR FORMATTER
     34 021222  010403                              MOV R4,R3       ;GET COPY OF MONTH NUMBER
     35 021224  020327  000012                      CMP R3,#10.     ;  IF 10 OR GREATER
     36 021230  103404                              BLO MON6
     37 021232  112721  000061                      MOVB #'1,(R1)+  ;PUT A "1" IN OUTPUT
     38 021236  162703  000012                      SUB #10.,R3
     39 021242  062703  000060          MON6:       ADD #'0,R3      ;CONVERT MONTH NUMBER TO ASCII
     40 021246  110321                              MOVB R3,(R1)+   ;PUT A NUMBER IN OUTPUT
     41 021250  112721  000055                      MOVB #'-,(R1)+  ;PUT A "-" IN OUTPUT
     42 021254  062704  003410                      ADD #DAYS-1,R4  ;GET POINTER TO DAYS IN MONTH
     43                                                             ;INDEXED BY NUMBER OF MONTH
     44 021260  012703  003270                      MOV #DATEI,R3   ;GET POINTER TO DATE INPUT
     45 021264  005000                              CLR R0
```

GLOBAL SUBROUTINES SECTION

```
 46 021266  121327  000055          DAY1:   CMPB (R3),#'-
 47 021272  001413                          BEQ DAY2
 48 021274  111321                          MOVB (R3),(R1)+ ;PUT DAY CHARACTER IN OUTPUT
 49 021276  006300                          ASL R0
 50 021300  010002                          MOV R0,R2
 51 021302  006300                          ASL R0
 52 021304  006300                          ASL R0
 53 021306  060200                          ADD R2,R0
 54 021310  112302                          MOVB (R3)+,R2
 55 021312  162702  000060                  SUB #'0,R2
 56 021316  060200                          ADD R2,R0
 57 021320  000762                          BR DAY1
 58 021322  120014                  DAY2:   CMPB R0,(R4)
 59 021324  101327                          BHI DERR
 60 021326  005700                          TST R0  ;SEE IF DATE IS ZERO
 61 021330  001725                          BEQ DERR        ;ERROR IF SO
 62 021332  062705  000003                  ADD #3,R5
 63 021336  121527  000055                  CMPB (R5),#'-   ;CHECK FOR "-" BETWEEN DAY
 64 021342  001320                          BNE DERR        ;  AND YEAR IN OUTPUT
 65 021344  112521                          MOVB (R5)+,(R1)+        ;PUT "-" IN OUTPUT
 66 021346  010504                          MOV R5,R4       ;GET COPY OF INPUT STRING POINTER
 67 021350  005000                          CLR R0
 68 021352  005002                          CLR R2
 69 021354  121427  000060          YER1:   CMPB (R4),#'0
 70 021360  103416                          BLO YER2
 71 021362  121427  000071                  CMPB (R4),#'9
 72 021366  101013                          BHI YER2
 73 021370  006300                          ASL R0
 74 021372  010003                          MOV R0,R3
 75 021374  006300                          ASL R0
 76 021376  006300                          ASL R0
 77 021400  060300                          ADD R3,R0
 78 021402  112403                          MOVB (R4)+,R3
 79 021404  162703  000060                  SUB #'0,R3
 80 021410  060300                          ADD R3,R0
 81 021412  005202                          INC R2
 82 021414  000757                          BR YER1
 83 021416  105714                  YER2:   TSTB (R4)
 84 021420  001271                          BNE DERR
 85 021422  020227  000002                  CMP R2,#2
 86 021426  001407                          BEQ YER3
 87 021430  020227  000004                  CMP R2,#4
 88 021434  001263                          BNE DERR
 89 021436  020027  003554                  CMP R0,#1900.
 90 021442  103660                          BLO DERR
 91 021444  000413                          BR YER5
 92 021446  012702  003425          YER3:   MOV #YEAR19,R2
 93 021452  020027  000106                  CMP R0,#70.
 94 021456  103002                          BHIS YER4
 95 021460  012702  003430                  MOV #YEAR20,R2
 96 021464  105712                  YER4:   TSTB (R2)
 97 021466  001402                          BEQ YER5
 98 021470  112221                          MOVB (R2)+,(R1)+
 99 021472  000774                          BR YER4
100 021474  112521                  YER5:   MOVB (R5)+,(R1)+
101 021476  001376                          BNE YER5
102 021500  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
103
104 021502  000000                  BRSAV:  .WORD   0                       ;DEFAULT BR LEVEL AND VECTOR
105
106 021504                                  ENDMOD
```

```
                                .SBTTL  PROTECTION TABLE

021504                                  BGNMOD

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

021504                                  BGNPROT
021504                                                                  L$PROT::

021504  177777                          -1                      ;OFFSET INTO P-TABLE FOR CSR ADDRESS
021506  177777                          -1                      ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
021510  177777                          -1                      ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

021512                                  ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

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

021512                                  BGNINIT
021512                                                                  L$INIT::
                                                                ;HERE FROM START COMMAND?
021512                                  READEF  #EF.STA
021512  012700  000040                                                  MOV     #EF.STA,R0
021516  104447                                                          TRAP    C$REFG
                                                                ;BRANCH TO 1$ IF NOT, ELSE
021520                                  BNCOMPLETE      1$
021520  103004                                                          BCC     1$
021522  012737  000010  003204          MOV     #ISTRT,IFLAGS   ;SET START BIT IN FLAG.
021530  000432                          BR      INIT1
021532                          1$:
                                                                ;HERE FROM RESTART COMMAND?
021532                                  READEF  #EF.RES
```

INITIALIZE SECTION

```
                021532  012700  000037                                                  MOV     #EF.RES,R0
                021536  104447                                                          TRAP    C$REFG
     54
     55         021540                          BNCOMPLETE      2$                      ;BRANCH TO 2$ IF NOT, ELSE
                021540  103004                                                          BCC     2$
     56         021542  052737  000004  003204          BIS     #IREST,IFLAGS           ;SET RESTART BIT IN FLAG.
     57         021550  000422                  BR      INIT1
     58         021552                  2$:
     59         021552                          READEF  #EF.CON                         ;HERE FROM CONTINUE COMMAND?
                021552  012700  000036                                                  MOV     #EF.CON,R0
                021556  104447                                                          TRAP    C$REFG
     60                                                                                 ;BRANCH TO 3$ IF NOT, ELSE
     61         021560                          BNCOMPLETE      3$
                021560  103007                                                          BCC     3$
     62         021562  042737  000020  003204          BIC     #ISTRTH,IFLAGS          ;CLEAR 1ST TIME THRU FLAG AND
     63         021570  052737  000002  003204          BIS     #ICONT,IFLAGS           ;SET CONTINUE BIT IN FLAG.
     64         021576  000405                  BR      INIT0
     65         021600                  3$:
     66         021600                          READEF  #EF.PWR                         ;HERE FROM POWER FAIL?
                021600  012700  000034                                                  MOV     #EF.PWR,R0
                021604  104447                                                          TRAP    C$REFG
     67                                                                                 ;BRANCH TO INIT0 IF POWER FAIL, ELSE
     68         021606                          BCOMPLETE       INIT0
                021606  103401                                                          BCS     INIT0
     69         021610                  INITQT: DOCLN                                   ; ABORT PROGRAM ON NEW PASS
                021610  104444                                                          TRAP    C$DCLN
     70
     71         021612  000137  022376  INIT0:  JMP     INITXX                          ; EXIT THE INITIALIZE SECTION.
     72
     73                                 ;
     74                                 ;       INITIALIZE KW11 CLOCK, FREE MEMORY AND IP ADDRESS TABLE
     75                                 ;       DURING START OR RESTART COMMAND ONLY
     76                                 ;
     77
     78         021616  012700  000003  INIT1:  MOV     #S0.FMT,R0                      ; GET BITS FOR REFORMAT MODE FLAG
     79         021622  030037  002136          BIT     R0,SFPTBL                       ; CHECK IF REFORMAT
     80         021626  001011                  BNE     1$                              ; IF SO, CONTINUE
     81         021630  012700  000004          MOV     #S0.CNS,R0                      ; GET BIT FOR RECONSTRUCT FLAG
     82         021634  030037  002136          BIT     R0,SFPTBL                       ; CHECK IF RECONSTRUCT MODE
     83         021640  001004                  BNE     1$                              ; IF SO, CONTINUE
     84         021642  006300                  ASL     R0                              ; GET BIT FOR RESTORE MODE
     85         021644  030037  002136          BIT     R0,SFPTBL                       ; CHECK IF RESTORE MODE
     86         021650  001757                  BEQ     INITQT                          ; IF NONE OF ABOVE, ABORT TEST
     87         021652  010037  003200  1$:     MOV     R0,MODE                         ; SAVE MODE FLAGS
     88
     89                 000105                  KWOUT.=105                              ; DATA TO START CLOCK
     90
     91         021656  005037  003216          CLR     KW.EL                           ;CLEAR ELAPSED TIME
     92         021662  005037  003220          CLR     KW.EL+2
     93         021666                          CLOCK   L,R0                            ;SEE IF L-CLOCK PRESENT
                021666  012700  000114                                                  MOV     #'L,R0
                021672  104462                                                          TRAP    C$CLCK
     94         021674                          BCOMPLETE       2$
                021674  103413                                                          BCS     2$
     95         021676                          CLOCK   P,R0                            ;SEE IF P-CLOCK PRESENT
                021676  012700  000120                                                  MOV     #'P,R0
                021702  104462                                                          TRAP    C$CLCK
```

```
     96 021704                                  BCOMPLETE       2$
        021704  103407                                                          BCS     2$
     97 021706  005037  003206                  CLR     KW.CSR                  ;IF NEITHER, CLEAR CSR STORAGE WORD
     98 021712                                  PNTF    NOCLOCK
        021712  004137  016662                                                  JSR R1,LPNTF
        021716  004110                                                          .WORD NOCLOCK
        021720  000000                                                          .WORD PNT.CT
     99 021722  000426                          BR      3$
    100
    101 021724  012037  003206          2$:     MOV     (R0)+,KW.CSR            ;STORE DATA RETURNED
    102 021730  012037  003210                  MOV     (R0)+,KW.BRL
    103 021734  012037  003212                  MOV     (R0)+,KW.VEC
    104 021740  012037  003214                  MOV     (R0)+,KW.HZ
    105                                                                         ;SETUP KW11 VECTOR ADDRESS
    106 021744                                  SETVEC  KW.VEC,#KW11I,#PRI07
        021744  012746  000340                                                  MOV     #PRI07,-(SP)
        021750  012746  020630                                                  MOV     #KW11I,-(SP)
        021754  013746  003212                                                  MOV     KW.VEC,-(SP)
        021760  012746  000003                                                  MOV     #3,-(SP)
        021764  104437                                                          TRAP    C$SVEC
        021766  062706  000010                                                  ADD     #10,SP
    107 021772  012777  000105  161206          MOV     #KWOUT.,@KW.CSR         ;START THE CLOCK
    108 022000  004737  013220          3$:     CALL    RESET                   ;RESET ALL CONTROLLERS
    109 022004                                  MEMORY  FFREE                   ;RESET START OF FREE MEMORY
        022004  104431                                                          TRAP    C$MEM
        022006  010037  002140                                                  MOV     R0,FFREE
    110 022012  017737  160122  002142          MOV     @FFREE,FSIZE            ;RESET SIZE OF FREE MEMORY
    111
    112                                 ;
    113                                 ;       ALLOCATE DRIVE TABLES TO MEMORY
    114                                 ;
    115
    116 022020  013737  002140  003202  INIT2:  MOV     FFREE,DTABS             ;STORE START OF DRIVE TABLES AND
    117 022026  005077  161150                  CLR     @DTABS                  ;MARK ZERO END.
    118 022032  013700  002012                  MOV     L$UNIT,R0               ;GET NUMBER OF LOGICAL UNITS TO RUN,
    119 022036  012701  000001                  MOV     #1,R1                   ;GET INITIAL SIZE OF DRIVE TABLE AND
    120 022042  062701  000015          1$:     ADD     #<D.SIZE>/2,R1          ;ACCUMULATE DRIVE TABLE SIZE.
    121 022046  005300                          DEC     R0                      ;SEE IF ANY MORE LOGICAL UNITS,
    122 022050  001374                          BNE     1$                      ;BRANCH IF NOT, ELSE
    123 022052  004737  013156                  CALL    ALOCM                   ;ALLOCATE ALL DRIVE TABLES TO MEMORY.
    124                                                                         ; R1 POINTS TO 1ST WORD IN DRIVE TABLE
    125
    126                                 ;
    127                                 ;       INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
    128                                 ;
    129
    130 022056  013737  002140  002150  INIT3:  MOV     FFREE,CTABS             ; STORE START OF CONTROLLER TABLES AND
    131 022064  005077  160060                  CLR     @CTABS                  ; MARK ZEROS END.
    132 022070  005037  002152                  CLR     CTRLRS                  ; CLEAR CONTROLLER COUNT
    133 022074  012701  003434                  MOV     #IPADRS,R1              ; R1 -> IP ADDRESS
    134 022100  012702  000010                  MOV     #8.,R2                  ; GET MAXIMUM # OF CONTROLLERS
    135 022104  005021                  1$:     CLR     (R1)+                   ; CLEAR ENTRY
    136 022106  005302                          DEC     R2                      ; DONE?
    137 022110  001375                          BNE     1$                      ; IF NOT, BRANCH
    138
    139                                 ;
    140                                 ;       BUILD CONTROLLER TABLES
```

```
                                        ;
                                        ;
022112  005005                  INIT4:  CLR     R5                      ;CLEAR CUSTOMER DATA FLAG
022114  005002                          CLR     R2                      ;START WITH LOGICAL UNIT 0
022116  012737  005160  021502          MOV     #5160,BRSAV             ; SAVE DEFAULT FOR BR LEVEL & VECTOR
022124                          1$:     GPHARD  R2,R0                   ; GET POINTER TO IT'S P-TABLE
022124  010200                                                                  MOV     R2,R0
022126  104442                                                                  TRAP    C$GPHRD
022130                                  BNCOMPLETE      16$             ; BRANCH TO 16$ IF NOT AVAILABLE
022130  103104                                                                  BCC     16$
022132  013703  002150                  MOV     CTABS,R3                ; GET ADDRESS OF 1ST CONTROLLER TABLE
022136  005713                  2$:     TST     (R3)                    ; CHECK IF ANY MORE TABLES
022140  001405                          BEQ     6$                      ; BUILD NEW TABLE IF FOUND ZERO WORD
022142  021013                          CMP     (R0),(R3)               ; CHECK IF SAME CSR ADDRESS,
022144                                          ASSUME  C.UADR EQ 0
022144                                          ASSUME  H0.UBA EQ 0
022144  001444                          BEQ     11$                     ; BRANCH IF SO
                                        ;
                                        ;
022146  062703  000052          5$:     ADD     #C.SIZE,R3              ;POINT TO BEGINNING OF NEXT CONTROLLER
022152  000771                          BR      2$                      ;TABLE IN MEMORY.
                                        ;
                                        ;
                                        ;       BUILD NEW CONTROLLER TABLE
                                        ;
                                        ;
022154  012704  003434          6$:     MOV     #IPADRS,R4              ;GET BEGINNING OF IP ADDRESS TABLE
022160  020427  003444          7$:     CMP     R4,#IPADRS+8.           ;SEE IF END OF IP ADDRESS TABLE,
022164  101004                          BHI     9$                      ;BRANCH IF SO, ELSE
022166  005724                          TST     (R4)+                   ;DID WE FIND AN OPEN ENTRY ?
022170  001401                          BEQ     8$                      ;BRANCH IF SO, ELSE
022172  000772                          BR      7$                      ;LOOK AGAIN.
                                        ;
022174  011044                  8$:     MOV     (R0),-(R4)              ;TAKE CSR ADDRESS FROM P-TABLE
                                        ;                               ;AND STORE IT IN THE IP ADDRESS TABLE.
022176  012701  000025          9$:     MOV     #<C.SIZE>/2,R1          ;GET # OF ENTRIES IN CONTROLLER TABLE
022202  004737  013156                  CALL    ALOCM                   ;AND ALLOCATE A TABLE TO MEMORY.
                                        ;                               ; R0 => 1ST WORD P-TABLE
                                        ;                               ; R1 => 1ST WORD IN CONTROLLER TABLE
022206  011021                          MOV     (R0),(R1)+              ; STORE CSR ADDRESS AND
022210  010221                          MOV     R2,(R1)+                ; UNIT NUMBER IN THE CONTROLLER TABLE.
022212  013704  021502                  MOV     BRSAV,R4                ; GET DEFAULT VECTOR & BR LEVEL
022216  162704  000004                  SUB     #4,R4                   ; GET NEXT VECTOR
022222  010437  021502                  MOV     R4,BRSAV                ; SAVE NEXT VECTOR
022226  010421                          MOV     R4,(R1)+                ;STORE IT IN THE CONTROLLER TABLE.
022230  012721  004037                  MOV     #4037,(R1)+             ;THE 'JSR R0' INSTRUCION AND
022234  012721  017612                  MOV     #UDASRV,(R1)+           ;THE ADDRESS OF THE INTERRUPT SERVICE
                                        ;                               ;ROUTINE IN THE CONTROLLER TABLE.
022240  012704  000020                  MOV     #<C.SIZE-C.FLG>/2,R4    ;GET # OF ENTRIES TO END OF TABLE,
022244  005021                  10$:    CLR     (R1)+                   ;CLEAR REST OF TABLE AND
022246  005304                          DEC     R4                      ;ADD ZERO WORD AT END.
022250  002375                          BGE     10$                     ;LOOP TIL ALL CLEARED
022252  005237  002152                  INC     CTRLRS                  ;KEEP TRACK OF CONTROLLER COUNT
                                        ;
                                        ;
                                        ;       BUILD DRIVE TABLES
                                        ;
```

```
    022256  013701  003202          11$:    MOV     DTABS,R1                ;GET ADDRESS OF CURRENT DRIVE TABLE
    022262  062703  000016                  ADD     #C.DR0,R3               ; INDEX TO 1ST DRIVE IN TABLE
    022266  012704  000010                  MOV     #8.,R4                  ; GET # OF DRIVES PER CONTROLLER
    022272  005713                  12$:    TST     (R3)                    ;ANY ENTRY TO DRIVE TABLE,
    022274  001411                          BEQ     14$                     ;BRANCH IF NOT, ELSE
    022276  026033  000002                  CMP     H0.LDR(R0),@(R3)+       ;COMPARE DRIVE NUMBER IN DRIVE TABLE,
    022302  001002                          BNE     13$                     ;BRANCH IF DIFFERNENT, ELSE
    022304  000137  022410                  JMP     MLDRER                  ;FOUND TWO P-TABLES WITH SAME DRIVE.

    022310  005304                  13$:    DEC     R4                      ; COUNT DRIVES
    022312  001367                          BNE     12$                     ; IF FOUR DRIVE TABLES ALREADY EXIST,
    022314  000137  022426                  JMP     TOOMER                  ;  THEN REPORT ERROR

    022320  010113                  14$:    MOV     R1,(R3)                 ;STORE ADDRESS OF DRIVE TABLE IN
                                                                            ;CONTROLLER TABLE.
    022322  016021  000002                  MOV     H0.LDR(R0),(R1)+        ;STORE DRIVE NUMBER AND
    022326  010221                          MOV     R2,(R1)+                ;LOGICAL UNIT NUMBER IN DRIVE TABLE.

    022330  062737  000032  003202          ADD     #D.SIZE,DTABS           ;NEXT DRIVE TABLE ADDRESS AND
    022336  005077  160640                  CLR     @DTABS                  ;MARK ZERO END.
    022342  005202                  16$:    INC     R2                      ;INCREMENT LOGICAL UNIT NUMBER
    022344  020237  002012                  CMP     R2,L$UNIT               ;CHECK IF GOT ALL TABLES
    022350  002665                          BLT     1$                      ;IF NOT, GO BACK FOR NEXT, ELSE
    022352  012701  000001                  MOV     #1,R1                   ;GET 1 WORD TO TERMINATE ALL CONTROLLER
    022356  004737  013156                  CALL    ALOCM                   ;TABLES AND ALLOCATE IT TO MEMORY.


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

    022362  013737  002140  002144  INIT6:  MOV     FFREE,FMEM              ;SAVE START ADDRESS
    022370  013737  002142  002146          MOV     FSIZE,FMEMS             ;SAVE SIZE


                                    ;
                                    ;       EXIT INITIALIZE SECTION
                                    ;

    022376                          INITXX: SETPRI  #PRI00                  ;SET RUNNING PRIORITY TO ZERO
    022376  012700  000000                                                                          MOV     #PRI00,R0
    022402  104441                                                                                  TRAP    C$SPRI

    022404                                  EXIT    INIT
    022404  104432                                                                                  TRAP    C$EXIT
    022406  000036                                                                                  .WORD   L10035-.


                                    ;TWO P-TABLES FOR SAME DRIVE
    022410  013705  003242          MLDRER: MOV TEMP,R5             ;GET CONTROLLER ADDRESS
    022414                                  ERRSF 2,,ERR002
    022414  104454                                                                                  TRAP    C$ERSF
    022416  000002                                                                                  .WORD   2
    022420  000000                                                                                  .WORD   0
    022422  012260                                                                                  .WORD   ERR002
    022424                                  DOCLN
    022424  104444                                                                                  TRAP    C$DCLN

                                    ;MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER
```

```
   243
   244 022426  013705  003242         TOOMER: MOV TEMP,R5             ;GET CONTROLLER ADDRESS
   245 022432                                 ERRSF 3,,ERR003
       022432  104454                                                                 TRAP    C$ERSF
       022434  000003                                                                 .WORD   3
       022436  000000                                                                 .WORD   0
       022440  012276                                                                 .WORD   ERR003
   246 022442                                 DOCLN
       022442  104444                                                                 TRAP    C$DCLN
   247
   248
   249 022444                                 ENDINIT
       022444                                                                 L10035:
       022444  104411                                                                 TRAP    C$INIT
```

```
                                .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.
                                ;-

022446                                  BGNAUTO
022446                                                                  L$AUTO::

022446                                  ENDAUTO
022446                                                                  L10036:
022446  104461                                                                  TRAP    C$AUTO
```

```
      1                                 .SBTTL  CLEANUP CODING SECTION
      2
      3                                 ;++
      4                                 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
      5                                 ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
      6                                 ;--
      7
      8 022450                                  BGNCLN
        022450                                                                          L$CLEAN::
      9
     10 022450  004737  013456                  CALL CLOSEF                     ;CLOSE DATA FILE
     11 022454                                  SETVEC #4,#NXMI,#PRI07
        022454  012746  000340                                                          MOV     #PRI07,-(SP)
        022460  012746  017602                                                          MOV     #NXMI,-(SP)
        022464  012746  000004                                                          MOV     #4,-(SP)
        022470  012746  000003                                                          MOV     #3,-(SP)
        022474  104437                                                                  TRAP    C$SVEC
        022476  062706  000010                                                          ADD     #10,SP
     12 022502  012703  000010                  MOV     #8.,R3          ; R3 = COUNTER OF ENTRIES
     13 022506  012704  003434                  MOV     #IPADRS,R4      ; R4 -> IP ADDRESS
     14 022512  005714                  1$:     TST     (R4)            ; IS THERE AN ENTRY?
     15 022514  001403                          BEQ     2$              ; IF NOT, DONE
     16 022516  005034                          CLR     @(R4)+          ; INIT UDA
     17 022520  005303                          DEC     R3              ; MAKE SURE WE DO NOT EXTEND OVER AREA
     18 022522  001373                          BNE     1$              ; IF NOT DONE, BRANCH
     19 022524                          2$:     CLRVEC  #4
        022524  012700  000004                                                          MOV     #4,R0
        022530  104436                                                                  TRAP    C$CVEC
     20
     21 022532                                  ENDCLN
        022532                                                                  L10037:
        022532  104412                                                                  TRAP    C$CLEAN
     22
     23 022534                                  ENDMOD
```

```
                                .SBTTL  TEST 1: DUP PROGRAM DRIVER

022534                                  BGNMOD

022534                                  BGNTST
022534                                                                  T1::
022534                                  PNTX WNSTRT             ;PRINT WARNING MESSAGE
022534  004137  016702                                                  JSR R1,LPNTX
022540  004565                                                          .WORD WNSTRT
022542  000000                                                          .WORD PNT.CT
022544                                  PNTX WNTIME             ;PRINT WARNING TIMES
022544  004137  016702                                                  JSR R1,LPNTX
022550  005043                                                          .WORD WNTIME
022552  000000                                                          .WORD PNT.CT
022554                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
022554  104450                                                          TRAP    C$MANI
022556                                  BNCOMPLETE T1MODE       ;IF NOT, JUST RUN THE PROGRAM
022556  103020                                                          BCC     T1MODE
022560  005037  003242                  CLR TEMP                ;CLEAR WORD FOR ANSWER
022564                                  GMANIL WNQUES,TEMP,1,YES        ;ASK IF STILL WANT TO RUN
022564  104443                                                          TRAP    C$GMAN
022566  000404                                                          BR      10000$
022570  003242                                                          .WORD   TEMP
022572  000130                                                          .WORD   T$CODE
022574  003630                                                          .WORD   WNQUES
022576  000001                                                          .WORD   1
022600                                                          10000$:
022600  005737  003242                  TST TEMP                        ;LOOK AT ANSWER
022604  001417                          BEQ T1QUIT              ;IF NO, QUIT NOW
022606  005737  003304                  TST DATEO       ;SEE IF ALREADY ASKED FOR DATE
022612  001002                          BNE T1MODE
022614  004737  021044                  CALL DATE       ;IF NOT, GET IT NOW

022620  032737  000003  003200  T1MODE: BIT #SO.FMT,MODE
022626  001164                          BNE T1FMT
022630                                  MANUAL
022630  104450                                                          TRAP    C$MANI
022632                                  BCOMPLETE T1GO
022632  103406                                                          BCS     T1GO
022634                                  ERRSF 10,,ERR010
022634  104454                                                          TRAP    C$ERSF
022636  000012                                                          .WORD   10
022640  000000                                                          .WORD   0
022642  012356                                                          .WORD   ERR010
022644                          T1QUIT: EXIT TST
022644  104432                                                          TRAP    C$EXIT
022646  000362                                                          .WORD   L10040-.
022650  032737  000010  003200  T1GO:   BIT #SO.STR,MODE
022656  001435                          BEQ T1CNS
022660  023727  002012  000001          CMP L$UNIT,#1
022666  001406                          BEQ T1RST
022670                                  ERRSF 9,,ERR009
022670  104454                                                          TRAP    C$ERSF
022672  000011                                                          .WORD   9
022674  000000                                                          .WORD   0
022676  012344                                                          .WORD   ERR009
022700                                  EXIT TST
```

```
                022700  104432                                                          TRAP    C$EXIT
                022702  000326                                                          .WORD   L10040-.
     30
     31 022704                          T1RST:  PNTF FILNAM
                022704  004137  016662                                                  JSR R1,LPNTF
                022710  012160                                                          .WORD FILNAM
                022712  000000                                                          .WORD PNT.CT
     32 022714                                  GMANID FILNAQ,FNAME,A,-1,1,10.,NO       ;GET FILE NAME
                022714  104443                                                          TRAP    C$GMAN
                022716  000406                                                          BR      10001$
                022720  003230                                                          .WORD   FNAME
                022722  000142                                                          .WORD   T$CODE
                022724  003574                                                          .WORD   FILNAQ
                022726  177777                                                          .WORD   -1
                022730  000001                                                          .WORD   T$LOLIM
                022732  000012                                                          .WORD   T$HILIM
                022734                                                          10001$:
     33 022734                                  OPEN #FNAME
                022734  012700  003230                                                  MOV     #FNAME,R0
                022740  104434                                                          TRAP    C$OPEN
     34 022742  012737  177777  002166          MOV #-1,FILOPN  ;MARK FLAG AS FILE OPEN
     35 022750  000513                          BR T1FMT
     36 022752  013705  002150          T1CNS:  MOV CTABS,R5
     37 022756  010504                  T1SER1: MOV R5,R4
     38 022760  062704  000016                  ADD #C.DRQ,R4
     39 022764  012703  000010                  MOV #8.,R3
     40 022770  011402                  T1SER2: MOV (R4),R2     ;GET DRIVE TABLE POINTER
     41 022772  001476                          BEQ T1SERN
     42 022774                                  PNTF SERNUM,D.UNIT(R2),(R5),(R2)
                022774  011246                                                          MOV (R2),-(SP)
                022776  011546                                                          MOV (R5),-(SP)
                023000  016246  000002                                                  MOV D.UNIT(R2),-(SP)
                023004  004137  016662                                                  JSR R1,LPNTF
                023010  004261                                                          .WORD SERNUM
                023012  000006                                                          .WORD PNT.CT
     43 023014                                      ASSUME C.UADR EQ 0
     44 023014                                      ASSUME D.DRV EQ 0
     45 023014                          T1SER3: GMANID SERNQ,TEMP,A,-1,1,20.,NO ;GET SERIAL NUMBER
                023014  104443                                                          TRAP    C$GMAN
                023016  000406                                                          BR      10002$
                023020  003242                                                          .WORD   TEMP
                023022  000142                                                          .WORD   T$CODE
                023024  003626                                                          .WORD   SERNQ
                023026  177777                                                          .WORD   -1
                023030  000001                                                          .WORD   T$LOLIM
                023032  000024                                                          .WORD   T$HILIM
                023034                                                          10002$:
     46 023034  012701  003242                  MOV #TEMP,R1
     47 023040  005000                          CLR R0
     48 023042  105711                  T1SER4: TSTB (R1)
     49 023044  001410                          BEQ T1SER5
     50 023046  005200                          INC R0
     51 023050  121127  000060                  CMPB (R1),#'0
     52 023054  103420                          BLO T1SER7
     53 023056  122127  000071                  CMPB (R1)+,#'9
     54 023062  101767                          BLOS T1SER4
     55 023064  000414                          BR T1SER7
```

```
     56 023066  020027  000024          T1SER5: CMP R0,#20.
     57 023072  103424                          BLO T1SER8
     58 023074  012701  003242                  MOV #TEMP,R1
     59 023100  012700  003320                  MOV #HIGHEST,R0
     60 023104  105710                  T1SER6: TSTB (R0)
     61 023106  001416                          BEQ T1SER8
     62 023110  122120                          CMPB (R1)+,(R0)+
     63 023112  001774                          BEQ T1SER6
     64 023114  103413                          BLO T1SER8
     65 023116                          T1SER7: PRINTF #SERNX,#HIGHEST
        023116  012746  003320                                                  MOV     #HIGHEST,-(SP)
        023122  012746  012051                                                  MOV     #SERNX,-(SP)
        023126  012746  000002                                                  MOV     #2,-(SP)
        023132  010600                                                          MOV     SP,R0
        023134  104417                                                          TRAP    C$PNTF
        023136  062706  000006                                                  ADD     #6,SP
     66 023142  000724                          BR T1SER3
     67 023144  062702  000004          T1SER8: ADD #D.SERN,R2  ;PUT ANSWER INTO DRIVE TABLE
     68 023150  012701  003242                  MOV #TEMP,R1
     69 023154  112122                  T1SER9: MOVB (R1)+,(R2)+
     70 023156  001376                          BNE T1SER9
     71 023160  005303                          DEC R3
     72 023162  001402                          BEQ T1SERN
     73 023164  005724                          TST (R4)+
     74 023166  000700                          BR T1SER2
     75 023170  062705  000052          T1SERN: ADD #C.SIZE,R5
     76 023174  005715                          TST (R5)
     77 023176  001267                          BNE T1SER1
     78 023200  013737  002150  002154  T1FMT:  MOV CTABS,TSTTAB               ;GET FIRST TABLE ADDRESS
     79 023206  013701  002152                  MOV CTRLRS,R1                 ;RUN DM PROGRAM ON ALL CONTROLLERS
     80 023212  004737  013344                  CALL RUNDM                    ;  RUN ALL CONTROLLERS OF ONE TYPE AT ONCE
     81 023216  001402                          BEQ 6$
     82 023220  004737  013474                  CALL RESPDM
     83 023224                          6$:     EXIT TST
        023224  104432                                                          TRAP    C$EXIT
        023226  000002                                                          .WORD   L10040-.
     84 023230                                  ENDTST
        023230                                                          L10040:
        023230  104401                                                          TRAP    C$ETST
     85 023232                                  ENDMOD
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION

023232                                  BGNMOD

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

023232                                  BGNHRD
023232  000011                                                          .WORD L10041-L$HARD/2
023234                                                          L$HARD::

                                ;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:

023234                          TABLE                                   ;START A TEBLE DEFINITION

023234                          ITEM HO.UBA     2                       ; UNIBUS ADDRESS
023234                          ITEM HO.LDR     2                       ; DRIVE NUMBER
023234                          END
```

```
      1 023234                                  GPRMA   H.UBA,HO.UBA,0,160000,177774,YES        ,BUS ADDRESS
        023234  000031                                                                            .WORD   T$CODE
        023236  023256                                                                            .WORD   H.UBA
        023240  160000                                                                            .WORD   T$LOLIM
        023242  177774                                                                            .WORD   T$HILIM
      2 023244                                  GPRMD   H.LDR,HO.LDR,D,-1,0.,255.,YES   ; DRIVE SELECT NUMBER
        023244  001052                                                                            .WORD   T$CODE
        023246  023272                                                                            .WORD   H.LDR
        023250  177777                                                                            .WORD   -1
        023252  000000                                                                            .WORD   T$LOLIM
        023254  000377                                                                            .WORD   T$HILIM
      3 023256                                  ENDHRC
                                                                                                  .EVEN
        023256                                                                          L10041:
      4
      5 023256      103     123     122  H.UBA:  .ASCIZ  \CSR ADDRESS\
      6 023272      104     122     111  H.LDR:  .ASCIZ  \DRIVE NUMBER\
      7                                         .EVEN
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION

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

023310                                  BGNSFT
023310  000022                                                          .WORD L10042-L$SOFT/2
023312                                                          L$SOFT::

                                ;FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:

023312                          TABLE                           ;START A TABLE DEFINITION

023312                          ITEM SO.BIT     2               ;YES/NO ANSWERS
        000001                          SO.FM1 = BIT0           ;  REFORMAT MODE
        000002                          SO.FM2 = BIT1           ;  (AGAIN)
        000003                            SO.FMT = SO.FM1+SO.FM2
        000004                          SO.CNS = BIT2           ;  RECONSTRUCT MODE
        000010                          SO.STR = BIT3           ;  RESTORE MODE

023312                          END
```

```
      1 023312                                  GPRML S.FMT,SO.BIT,SO.FM1,YES   ;REFORMAT?
        023312  000130                                                          .WORD   T$CODE
        023314  023527                                                          .WORD   S.FMT
        023316  000001                                                          .WORD   SO.FM1
      2 023320                                  XFERT SWEND
        023320  017024                                                          .WORD   T$CODE
      3 023322                                  GPRML S.NRF,SO.BIT,SO.FM2,YES   ;AGAIN - REFORMAT?
        023322  000130                                                          .WORD   T$CODE
        023324  023356                                                          .WORD   S.NRF
        023326  000002                                                          .WORD   SO.FM2
      4 023330                                  XFERT SWEND
        023330  013024                                                          .WORD   T$CODE
      5 023332                                  GPRML S.CNS,SO.BIT,SO.CNS,YES   ;RECONSTRUCT
        023332  000130                                                          .WORD   T$CODE
        023334  023606                                                          .WORD   S.CNS
        023336  000004                                                          .WORD   SO.CNS
      6 023 40                                  XFERT SWEND
        023340  007024                                                          .WORD   T$CODE
      7 023342                                  GPRML S.RST,SO.BIT,SO.STR,YES   ;RESTORE?
        023342  000130                                                          .WORD   T$CODE
        023344  023651                                                          .WORD   S.RST
        023346  000010                                                          .WORD   SO.STR
      8 023350                                  XFERT SWEND
        023350  003024                                                          .WORD   T$CODE
      9 023352                                  DISPLAY S.NOF   ;WARNING
        023352  000003                                                          .WORD   T$CODE
        023354  023772                                                          .WORD   S.NOF
     10 023356                          SWEND:  ENDSFT
                                                                                .EVEN
        023356                                                          L10042:
     11
     12 023356      015     012                 S.NRF:  .BYTE 15,12
     13 023360      116     117     124                 .ASCII\NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR\
     14 023462      015     012                         .BYTE 15,12
     15 023464      111     116     106                 .ASCII\INFORMATION ON THE DISKS.\
     16 023515      015     012                         .BYTE 15,12
     17 023517      101     107     101                 .ASCII\AGAIN - \
     18 023527      122     105     106         S.FMT:  .ASCIZ\REFORMAT USING EXISTING BAD SECTOR INFORMATION\
     19 023606      122     105     103         S.CNS:  .ASCIZ\RECONSTRUCT BAD SECTOR INFORMATION\
     20 023651      104     117     040         S.RST:  .ASCII\DO YOU HAVE A FILE ON THE SYSTEM LOAD DEVICE\
     21 023725      015     012                         .BYTE 15,12
     22 023727      040     103     117                 .ASCIZ\ CONTAINING BAD SECTOR INFORMATION\
     23 023772      131     117     125         S.NOF:  .ASCIZ\YOU CANNOT PROCEED WITHOUT SUCH A FILE.\
     24 024042      122     105     123                 .ASCIZ\RESTART PROGRAM AND SELECT TO REFORMAT OR RECONSTRUCT DISK.\
     25 024136      000                                 .BYTE 0
     26                                                 .EVEN
     27
     28                                         .DSABL AMA
     29 000000                                  .PSECT END
```

```
                                        .SBTTL  PATCH AREA
                                ;
000000                          $PATCH::
        000050                          .REPT 40.
                                        .WORD 0
                                        .ENDR
                                ;
000120                                  LASTAD

                                                                        .EVEN
000120  000134'                                                        .WORD T$FREE
000122  000004                                                        .WORD T$SIZE
000124                          L$LAST::
                                ;
000124                                  ENDMOD
```

PATCH AREA

```
      1 000124                                  BGNSETUP        1
      2
      3 000124                                  BGNPTAB
        000124  000000                                                                  .WORD   0
        000126  000002                                                                  .WORD   L10045 ./2-1
        000130                                                                 L10043:
      4
      5 000130  172150                          .WORD   172150              ; UNIBUS ADDRESS
      6 000132  000000                          .WORD   0.                  ; LOGICAL DRIVE NUMBER
      7
      8 000134                                  ENDPTAB
        000134                                                                 L10045:
      9
     10 000134                                  ENDSETUP
     11
     12
     13
     14
     15
     16
     17
     18         000001                  .END
Errors detected:  0

*** Assembler statistics


Work  file  reads: 597
Work  file writes: 517
Size of work file: 29648 Words  ( 116 Pages)
Size of core pool: 14080 Words  ( 55 Pages)
Operating  system: RT-11 (Under RTEM-11)

Elapsed time: 00:02:22.00
ZUDKBO,ZUDKBO/C=SVC34R.MLB/P:1,ZUDKBO.DOC,ZUDKBO
```

J12

Cross reference table (CREF V05.03)

Cross reference table (CREF V05.03)

M12

Cross reference table (CREF V05.03)

```
DATE    114-1#   114-32   119-16
DATEI    47-2#   114-1    114-2    114-44
DATEO    47-4#    68-26   114-33   119-14
DATEQ    49-3#   114-1
DATEX    52-13#  114-31
DAY     114-5#
DAY1    114-46#  114-57
DAY2    114-47   114-58#
DAYS     47-19#  114-42
DERR    114-4    114-6    114-10   114-12   114-14   114-31#  114-59   114-61   114-64   114-84   114-88   114-90
DF.ACT   40-21#   64-60
DF.LCL   40-24#
DF.NES   40-22#
DF.SA    40-26#
DFPTBL   28-10#
DIAG     34-33#   97-23
DIAGMC   26-8     26-8
DIVIDE   69-38    89-39    90-7     93-16#   113-19   113-21   113-23
DMFRST   40-47#
DMMAIN   40-46#
DMOVRL   40-44#
DMPROG   46-16#   94-13    95-3     95-4     95-5     95-6
DMTMO    40-45#   94-14
DMTRLN   40-43#
DQCONT   69-28    69-61#
DQNO     69-54    69-62#
DQRFMT   69-26    69-53#
DQRSTR   69-27    69-59#
DQUEJP   69-21    69-22#   69-29
DQUEST   65-4     69-16#
DQUESZ   69-18    69-29#
DQUEX    69-19    69-22    69-24    69-25    69-50#   69-57    69-64
DQUNIT   69-23    69-33#
DQUNL1   69-38#   69-42
DQUNL2   69-44#   69-48
DQYES    69-55#   69-60
DSPSIZ   65-9#
DT.AVL   42-12#   62-20    62-25    75-15
DT.UNT   42-11#
DTABS    46-30#   116-116* 116-117* 116-196  116-214* 116-215*
DU.DFL   40-33#
DU.FTL   40-36#
DU.INF   40-34#
DU.QUE   40-32#
DU.SPC   40-37#   64-6
DU.TER   40-35#
DU.TYP   40-39#   64-4     64-19
DUP      34-34#   97-20
E$END    26-8#
E$LOAD   26-8#    26-34
EF.BBR   37-27#
EF.BBU   37-28#
EF.CON   30-10#   116-59
EF.LOG   37-29#
EF.NEW   30-10#
EF.PWR   30-10#   116-66
```

Cross reference table (CREF V05.03)

```
EF.RES   30-10#  116-53
EF.SEX   37-30#
EF.STA   30-10#  116-47
EOFMSG   53-56    53-59#
ERR001   53 14#
ERR002   53-18#  116-239
ERR003   53-22#  116-245
ERR004   53-26#   55-7
ERR008   53-30#
ERR009   53-34#  119-28
ERR010   53-38#  119-22
ERR014   53-42#  105-51
ERR020   53-46#   58-28   107-33
ERR021   53-50#  111-5
ERR022   53-62#  110-36
ERR023   53-67#  106-12
ERR024   53-79#  105-36
ERR025   53-83#  109-37
ERR030   53-87#   61-24
ERR031   53 91#   62-36
ERR032   53-95#   64-8
ERR033   53-100#  63-21
ERR034   53 104#  96-3
ERR036   53 108# 100-29
ERR037   53 112# 101-5
ERR100   53-116#  68-22
ERR10    53-120#  70-32
ERR21A   53 54     53 57#
ERR23A   53-70#    53-75
ERR23B   53 71     53-73#
ERR23C   53 76#
ERRC     77 36     77-43    81-9#
ERRCHR   46-42#    91-5*    91-12    91-14    91-16    91-18
ERRD     77-45     81-21#
ERRME1   50-8#     77-41    79-23    82-4
ERRNL    50-4#     91-10
ERRONE   50-3#     91-7
ERRRSZ   79-18     80-11#
ERRRTB   79-21     80-3#    80 11
ERRTRM   65-7      72-14#
EVL      30-10#
F$AU     26-8#
F$AUTO   26-8#    117-10   117-12
F$BGN    26-8#     26-26    29-16    30-3     53-14    53-18    53-22    53-26    53-30    53-34    53-38    53-42    53-46    53-50
         53-62     53-67    53-79    53-83    53-87    53-91    53-95    53-100   53-104   53-108   53-112   53-116   53-120   102-10
        103-18    112-5    114-106  115-3    115-10   116-45   116-235  117-10   118-8    118-23   119-3    119-5    119-23   119-29
        119-83    119-84   119-85   120-3    120-14   122-12   123-9    123-9    124-10   125-1    125-3    125-3    125-8    125 10
F$CLEA   26-8#    118-8    118 21
F$DU     26-8#
F$END    26-8      26-8     26-8     26-8     26-8     26-8     26-8     26-8     26-8     26-8     26-8     26-8     26-8     26-8
         26-8      26-8     26-8#    26-26    29-16    30-3     53-16    53-20    53-24    53-28    53-32    53-36    53-40    53-44
         53-48     53-60    53-65    53-77    53-81    53-85    53-89    53-93    53-98    53-102   53-106   53-110   53-114   53-118
         53-122   102-14   103-21   112-9    114-106  115-3    116-235  116-249  117-12   118-21   118-23   119-3    119-5    119-5
        119-5     119-23   119-29   119-83   119-84   119-84   119-85   120-3    121-3    123-10   124-10   125-1    125-3    125-8
        125-10
F$HARD   26-8#    120-14   121-3    123 2    123-4    123-6    123-8
```

Cross reference table (CREF V05.03)

```
F$MW    26-8#    28-10    28 14
F$INIT  26-8#   116-45   116-249
F$JMP   26 8#   116-235  119-23   119-29   119-83
F$MOD   26 8#    26-26    29-10    30-3    114-106  115-3   118-23  119-3   119-85  120-3   124-10
F$MSG   26-8#    53-14    53-16    53-18    53-20    53-22   53-24   53-26   53-28   53-30   53-32   53-34   53-36   53-38
        53-40    53-42    53-44    53-46    53-48    53-50   53-60   53-62   53-65   53-67   53-77   53-79   53-81   53-83
        53 85    53-87    53-89    53-91    53-93    53-95   53-98   53-100  53-102  53-104  53-106  53-108  53-110  53-112
        53 114   53-116   53-118   53-120   53-122
F$PROT  26 8#   115-10   115-16
F$PWR   26-8#
F$RPT   26 8#
F$SEG   26-8#
F$SOFT  26-8#   122-12   123-2    123-4    123-6    123-8   123-10
F$SRV   26 8#   102-10   102-14   103-18   103-21   112-5   112-9
F$SUB   26-8#
F$SW    26-8#    29-10    29-14
F$TEST  26-8#   119-5    119-84
FCTBUF  46-24#   73-23    73 32
FCTNUM  46-25#   59-36*   73-14    73-21*   73-28*
FDATA   46-23#
FFREE   46-8#    56-16    56-20*  116-109* 116-110  116-116  116-130  116-226
FILNAM  52-14#  119-31
FILNAQ  49-4#   119-32
FILOPN  46-20#   60-10    60-13*   119-34*
FMEM    46-10#  116-226*
FMEMS   46-11#  116-227*
FMERR   55-7#    56-18
FNAME   46-44#   73-20    119-32   119-33
FSIZE   46-9#    56-17*  116-110* 116-227
G$CNTO  26-8#
G$DELM  26-8#
G$DISP  26-8#   123-9
G$EXCP  26-8#
G$HILI  26-8#
G$LOLI  26-8#
G$NO    26-8#   119-32   119-45
G$OFFS  26-8#   114-1    119-11   119-32   119-45   121-1   121-2   123-1   123-3   123-5   123-7
G$OFSI  26-8#   114-1    119-11   119-32   119-45   121-1   121-2   123-1   123-3   123-5   123-7
G$PPMA  26-8#   121-1
G$PRMD  26-8#   114-1    119-32   119-45   121-2
G$PRML  26-8#   119-11   123-1    123-3    123-5    123-7
G$RADA  26-8#   114-1    119-32   119-45
G$RADB  26-8#
G$RADD  26-8#   121-2
G$RADL  26-8#   119-11   123-1    123-3    123-5    123-7
G$PADO  26-8#   121-1
G$XFER  26-8#   123-2    123-4    123-6    123-8
G$YES   26-8#   114-1    119-11   121-1    121-2    123-1   123-3   123-5   123-7
GETCDN  88-16    88-18    88-28#
GETCNT  78-12    79-9     79-17    79-29    88-13#   89-15
GETCNx  88-15#   88-27
GETCXX  88-29    88-31#
GTDRVL  75-13#   75-16
GTDRVT  68-16    69-16    70-24    70-36    75-11#
H LDR  121-2    121-6#
H UBA  121-1    121-5#
```

Cross reference table (CREF V05.03)

Cross reference table (CREF V05.03)

```
L$LOAD  26-34#
L$LUN   26-34#   59-24*   61-12*   75-14*
L$MREV  26-34#
L$NAME  26-34#
L$PRIO  26-34#
L$PROT  26-34    115-10#
L$PRT   26-34#
L$REPP  26-34#
L$REV   26-34#
L$SOFT  26-34    122-12   122-12#
L$SPC   26-34#
L$SPCP  26-34#
L$SPTP  26-34#
L$STA   26-34#
L$SW    26-34     29-10    29-10#
L$TEST  26-34#
L$TIML  26-34#
L$UNIT  26-34#    76-13   116-118  116-217  119-26
L10000  28-10     28-14#
L10001  29-10     29-14#
L10002  53-16#
L10003  53-20#
L10004  53-24#
L10005  53-28#
L10006  53-32#
L10007  53-36#
L10010  53-40#
L10011  53-44#
L10012  53-48#
L10013  53-60#
L10014  53-65#
L10015  53-77#
L10016  53-81#
L10017  53-85#
L10020  53-89#
L10021  53-93#
L10022  53-98#
L10023  53-102#
L10024  53-106#
L10025  53-110#
L10026  53-114#
L10027  53-118#
L10030  53-122#
L10031 102-14#
L10032 103-21#
L10033 112-9#
L10035 116-235  116-249#
L10036 117-12#
L10037 118-21#
L10040 119-23   119-29   119-83   119-84#
L10041 120-14   121-3#
L10042 122-12   123-10#
L10043 125-3#
L10045 125-3    125-8#
LDDM    59-22#   59-34
LDNEXT  59-26    59-30    59-32#
```

Cross reference table (CREF V05.03)

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.03b Monday 23-Dec-85 11:22 Page S-11
Cross reference table (CREF V05.03)

Cross reference table (CREF V05.03)

K13


```
ST.DRV  40-16#
ST.HST  40-14#
ST.MFE  40-10#
ST.MSK  40 3#    63-14    95-10
ST.OFL  40-8#
ST.SUB  40-4#
ST.SUC  40-5#
ST.WPR  40-11#
SVCGBL  26-8#    26-13#   26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    27-9     28-10
        28-10    29-10    29-10    48-12    48-16    53-11#   53-14    53-14    53-14    53-18    53-18    53-18    53-22    53-22
        53-22    53-26    53-26    53-26    53-30    53-30    53-30    53-34    53-34    53-34    53-38    53-38    53-38    53-42
        53-42    53-42    53-46    53-46    53-46    53-50    53-50    53-50    53-62    53-62    53-62    53-67    53-67    53-67
        53 79    53-79    53-79    53-83    53-83    53-83    53-87    53-87    53-87    53-91    53-91    53-91    53-95    53 95
        53-95    53-100   53-100   53-100   53-104   53-104   53-104   53-108   53-108   53-108   53-112   53-112   53-112   53-116
        53-116   53-116   53-120   53-120   53-120   54-4#    102-10   103-18   112-5    115-10   116-45   117-10   118-8    120-14
        122-12   124-8    124-8    124-8    124-8#
SVCINS  26-8#    26-10#   26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26 34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    27-9     27-9     27-9     27-9     27-9     27-9     28-10    28-10    28-10    29-10    29-10
        29-10    48-12    48-12    48-12    48-12    48-12    48-12    48-16    48-16    48-16    48-16    48-16    48-16    53-8#
        53-16    53-20    53-24    53-28    53-32    53-36    53-40    53-44    53-48    53-60    53-65    53-77    53-81    53-85
        53-89    53-93    53-98    53-102   53-106   53-110   53-114   53-118   53-122   54-1#    55-7     55-7     55-7     55-7
        55-7     55-7     55-7     55-7     55-7     55-7     55-7     55-7     55-8     55-8     55-8     58-11    58 11    58-11
        58-11    58-11    58-11    58-11    58-11    58-11    58-11    58-11    58-11    58-11    58-11    58-11    58-11    58-11
        58-11    58-12    58-12    58-12    58-22    58-22    58-22    58-22    58-22    58-22    58-28    58-28    58-28    58-28
        58-28    58-28    58-28    58-28    58-28    58-28    58-28    58-28    58-30    58-30    58-30    60-12    60-12    60-12
        61-8     61-8     61-8     61-24    61-24    61-24    61-24    61-24    61-24    61-24    61-24    61-24    61-24    61-24
        61-24    62-36    62-36    62-36    62-36    62-36    62-36    62-36    62-36    62-36    62-36    62-36    62-36    63-21
        63-21    63-21    63-21    63-21    63-21    63-21    63-21    63-21    63-21    63-21    63-21    64-8     64-8     64-8
        64-8     64-8     64-8     64-8     64-8     64-8     64-8     64-8     64-8     68-22    68-22    68-22    68-22    68-22
        68-22    68-22    68-22    68-22    68-22    68-22    68-22    70-32    70-32    70-32    70-32    70-32    70-32    70-32
        70-32    70-32    70-32    70-32    70-32    73-19    73-19    73-19    73-20    73-20    73-20    73-20    73-20    73-20
        73-24    73-24    73-24    73-24    73-24    73-24    73-25    73-25    73-25    91-12    91-12    91-12    91-12    91-12
        91-12    91-12    91-12    91-12    91-12    91-12    91-12    91-12    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    91-14    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
        91-16    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    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    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     96-3     96-3     100-21
        100-21   100-21   100-29   100-29   100-29   100-29   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    101-5    101-5    101-5    102-14   102-14
        102 14   103-21   103-21   103-21   105-36   105-36   105-36   105-36   105-36   105-36   105-36   105-36   105-36   105-36
        105-36   105-36   105-51   105-51   105-51   105-51   105-51   105-51   105-51   105-51   105-51   105-51   105-51   105 51
        106-12   106-12   106-12   106-12   106-12   106-12   106-12   106-12   106-12   106-12   106-12   106-12   107-12   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   107-27
```

	107-27	107-27	107-27	107-27	107-27	107-30	107-30	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	107-33	107-33	109-37	109-37	109-37	109-37	109-37
	109-37	109-37	109-37	109-37	109-37	109-37	109-37	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	111-5	111-5	111-5	111-5	111-5	111-5
	111-5	111-5	111-5	111-5	111-5	111-5	112-9	112-9	112-9	114-1	114-1	114-1	114-1	114-1
	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1
	114-1	114-1	114-1	114-1	114-1	116-47	116-47	116-47	116-47	116-47	116-47	116-49	116-49	116-49
	116-53	116-53	116-53	116-53	116-53	116-53	116-55	116-55	116-55	116-59	116-59	116-59	116-59	116-59
	116-59	116-61	116-61	116-61	116-66	116-66	116-66	116-66	116-66	116-66	116-68	116-68	116-68	116-69
	116-69	116-69	116-93	116-93	116-93	116-93	116-93	116-93	116-94	116-94	116-94	116-95	116-95	116-95
	116-95	116-95	116-95	116-96	116-96	116-96	116-106	116-106	116-106	116-106	116-106	116-106	116-106	116-106
	116-106	116-106	116-106	116-106	116-106	116-106	116-106	116-106	116-106	116-106	116-109	116-109	116-109	116-109
	116-109	116-109	116-146	116-146	116-146	116-146	116-146	116-146	116-147	116-147	116-233	116-233	116-233	116-233
	116-233	116-233	116-233	116-235	116-235	116-235	116-235	116-235	116-235	116-239	116-239	116-239	116-239	116-239
	116-239	116-239	116-239	116-239	116-239	116-239	116-239	116-240	116-240	116-240	116-245	116-245	116-245	116-245
	116-245	116-245	116-245	116-245	116-245	116-245	116-245	116-245	116-246	116-246	116-246	116-249	116-249	116-249
	117-12	117-12	117-12	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11
	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-19	118-19	118-19	118-19	118-19	118-19	118-21
	118-21	118-21	119-8	119-8	119-8	119-9	119-9	119-9	119-11	119-11	119-11	119-11	119-11	119-11
	119-11	119-11	119-11	119-11	119-11	119-11	119-11	119-11	119-11	119-11	119-11	119-11	119-20	119-20
	119-20	119-21	119-21	119-21	119-22	119-22	119-22	119-22	119-22	119-22	119-22	119-22	119-22	119-22
	119-22	119-22	119-23	119-23	119-23	119-23	119-23	119-23	119-28	119-28	119-28	119-28	119-28	119-28
	119-28	119-28	119-28	119-28	119-28	119-28	119-29	119-29	119-29	119-29	119-29	119-29	119-32	119-32
	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32
	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-33	119-33	119-33	119-33	119-33	119-33
	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45
	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-45	119-65	119-65	119-65	119-65
	119-65	119-65	119-65	119-65	119-65	119-65	119-65	119-65	119-65	119-65	119-65	119-65	119-65	119-65
	119-83	119-83	119-83	119-83	119-83	119-83	119-84	119-84	119-84	120-14	120-14	120-14	121-1	121-1
	121-1	121-1	121-1	121-1	121-1	121-1	121-1	121-1	121-1	121-1	121-2	121-2	121-2	121-2
	121-2	121-2	121-2	121-2	121-2	121-2	121-2	121-2	121-2	121-2	121-2	121-3	121-3	121-3
	122-12	122-12	122-12	123-1	123-1	123-1	123-1	123-1	123-1	123-1	123-1	123-1	123-2	123-2
	123-2	123-3	123-3	123-3	123-3	123-3	123-3	123-3	123-3	123-3	123-4	123-4	123-4	123-5
	123-5	123-5	123-5	123-5	123-5	123-5	123-5	123-6	123-6	123-6	123-7	123-7	123-7	123-7
	123-7	123-7	123-7	123-7	123-7	123-7	123-8	123-8	123-8	123-9	123-9	123-9	123-9	123-9
	123-9	123-10	123-10	123-10	124-8	124-8	124-8	124-8	124-8	124-8	124-8	124-8	124-8	125-3
	125-3	125-3	125-3	125-3	125-3									
SVCSUB	26-8#	26-12#	53-10#	54-3#										
SVCTAG	26-8#	26-14#	28-14	29-14	53-12#	53-16	53-16	53-16	53-20	53-20	53-20	53-24	53-24	53-24
	53-28	53-28	53-28	53-32	53-32	53-32	53-36	53-36	53-36	53-40	53-40	53-40	53-44	53-44
	53-44	53-48	53-48	53-48	53-60	53-60	53-60	53-65	53-65	53-65	53-77	53-77	53-77	53-81
	53-81	53-81	53-85	53-85	53-85	53-89	53-89	53-89	53-93	53-93	53-93	53-98	53-98	53-98
	53-102	53-102	53-102	53-106	53-106	53-106	53-110	53-110	53-110	53-114	53-114	53-114	53-118	53-118
	53-118	53-122	53-122	53-122	54-5#	102-14	103-21	112-9	114-1	116-249	117-12	118-21	119-11	119-32
	119-45	119-84	121-3	123-10	125-3	125-8								
SVCTST	26-8#	26-11#	53-9#	54-2#	119-5									
SWEND	123-2	123-4	123-6	123-8	123-10#									
T$$AUT	117-10#	117-12												
T$$CLE	118-8#	118-21												
T$$DAT	125-3	125-3#	125-8											
T$$HAR	120-14	120-14#	121-3											
T$$HW	28-10	28-10#	28-14											
T$$INI	116-45#	116-235	116-249											
T$$MSG	53-14#	53-16	53-18#	53-20	53-22#	53-24	53-26#	53-28	53-30#	53-32	53-34#	53-36	53-38#	53-40
	53-42#	53-44	53-46#	53-48	53-50#	53-60	53-62#	53-65	53-67#	53-77	53-79#	53-81	53-83#	53-85
	53 87#	53-89	53-91#	53-93	53-95#	53-98	53-100#	53-102	53-104#	53-106	53-108#	53-110	53-112#	53-114

M13

Cross reference table (CREF V05.03)

	53-116#	53-118	53-120#	53-122										
T$$PC	125-1#	125-10												
T$$PRO	115-10#													
T$$PTA	125-1#	125-3	125-3#											
T$$SOF	122-12	122-12#	123-10											
T$$SRV	102-10#	102-14	103-18#	103-21	112-5#	112-9								
T$$SW	29-10	29-10#	29-14											
T$$TES	119-5#	119-23	119-29	119-83	119-84									
T$ARGC	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34#	26-34#	26-34#
	26-34#	26-34#	26-34#	91-12	91-12	91-12	91-12#	91-12#	91-14	91-14	91-14	91-14#	91-14#	91-16
	91-16	91-16	91-16#	91-16#	91-18	91-18	91-18	91-18#	91-18#	119-65	119-65	119-65	119-65#	119-65#
T$CODE	114-1	114-1	114-1	114-1#	114-1#	114-1#	119-11	119-11	119-11	119-11#	119-11#	119-11#	119-32	119-32
	119-32	119-32#	119-32#	119-32#	119-45	119-45	119-45	119-45#	119-45#	119-45#	121-1	121-1	121-1	121-1
	121-1#	121-1#	121-2	121-2	121-2	121-2#	121-2#	121-2#	123-1	123-1	123-1	123-1#	123-1#	123-1#
	123-2	123-2	123-2	123-2	123-2	123-2	123-2#	123-2#	123-2#	123-2#	123-3	123-3	123-3	123-3#
	123-3#	123-3#	123-4	123-4	123-4	123-4	123-4	123-4	123-4#	123-4#	123-4#	123-4#	123-5	123-5
	123-5	123-5#	123-5#	123-5#	123-6	123-6	123-6	123-6	123-6	123-6	123-6#	123-6#	123-6#	123-6#
	123-7	123-7	123-7	123-7#	123-7#	123-7#	123-8	123-8	123-8	123-8	123-8	123-8	123-8#	123-8#
	123-8#	123-8#	123-9	123-9#										
T$ERRN	26-8#	55-7	55-7#	58-28	58-28#	61-24	61-24#	62-36	62-36#	63-21	63-21#	64-8	64-8#	68-22
	68-22#	70-32	70-32#	96-3	96-3#	100-29	100-29#	101-5	101-5#	105-36	105-36#	105-51	105-51#	106-12
	106-12#	107-33	107-33#	109-37	109-37#	110-36	110-36#	111-5	111-5#	116-239	116-239#	116-245	116-245#	119-22
	119-22#	119-28	119-28#											
T$EXCP	114-1	114-1#	119-32	119-32#	119-45	119-45#	121-1	121-1#	121-2	121-2#				
T$FLAG	116-235	116-235	116-235#	116-235#	119-23	119-23	119-23#	119-23#	119-29	119-29	119-29#	119-29#	119-83	119-83
	119-83#	119-83#												
T$FREE	124-8	125-10#												
T$GMAN	26-8#	114-1	114-1#	114-1#	119-32	119-32#	119-32#	119-45	119-45#	119-45#				
T$HILI	114-1	114-1#	119-32	119-32#	119-45	119-45#	121-1	121-1#	121-2	121-2#				
T$LAST	26-8#	124-8#	125-1											
T$LOLI	114-1	114-1#	119-32	119-32#	119-45	119-45#	121-1	121-1#	121-2	121-2#				
T$LSYM	26-8	26-8#	28-14	29-14	53-16	53-20	53-24	53-28	53-32	53-36	53-40	53-44	53-48	53-60
	53-65	53-77	53-81	53-85	53-89	53-93	53-98	53-102	53-106	53-110	53-114	53-118	53-122	102-14
	103-21	112-9	116-249	117-12	118-21	119-84	121-3	123-10						
T$LTNO	124-8#													
T$NEST	26-8#	26-26	26-26	26-26#	28-10	28-10	28-10#	28-14	28-14	28-14	28-14#	29-10	29-10	29-10#
	29-14	29-14	29-14	29-14#	29-16	29-16	29-16	29-16#	30-3	30-3	30-3#	53-14	53-14	53-14#
	53-16	53-16	53-16	53-16#	53-18	53-18	53-18#	53-20	53-20	53-20	53-20#	53-22	53-22	53-22#
	53-24	53-24	53-24	53-24#	53-26	53-26	53-26#	53-28	53-28	53-28	53-28#	53-30	53-30	53-30#
	53-32	53-32	53-32	53-32#	53-34	53-34	53-34#	53-36	53-36	53-36	53-36#	53-38	53-38	53-38#
	53-40	53-40	53-40	53-40#	53-42	53-42	53-42#	53-44	53-44	53-44	53-44#	53-46	53-46	53-46#
	53-48	53-48	53-48	53-48#	53-50	53-50	53-50#	53-60	53-60	53-60	53-60#	53-62	53-62	53-62#
	53-65	53-65	53-65	53-65#	53-67	53-67	53-67#	53-77	53-77	53-77	53-77#	53-79	53-79	53-79#
	53-81	53-81	53-81	53-81#	53-83	53-83	53-83#	53-85	53-85	53-85	53-85#	53-87	53-87	53-87#
	53-89	53-89	53-89	53-89#	53-91	53-91	53-91#	53-93	53-93	53-93	53-93#	53-95	53-95	53-95#
	53-98	53-98	53-98	53-98#	53-100	53-100	53-100#	53-102	53-102	53-102	53-102#	53-104	53-104	53-104#
	53-106	53-106	53-106	53-106#	53-108	53-108	53-108#	53-110	53-110	53-110	53-110#	53-112	53-112	53-112#
	53-114	53-114	53-114	53-114#	53-116	53-116	53-116#	53-118	53-118	53-118	53-118#	53-120	53-120	53-120#
	53-122	53-122	53-122	53-122#	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#	112-5	112-5	112-5#	112-9	112-9	112-9	112-9#	114-106	114-106	114-106
	114-106#	115-3	115-3	115-3#	115-10	115-10	115-10#	115-16	115-16	115-16	115-16#	116-45	116-45	116-45#
	116-249	116-249	116-249	116-249#	117-10	117-10	117-10#	117-12	117-12	117-12	117-12#	118-8	118-8	118-8#
	118-21	118-21	118-21	118-21#	118-23	118-23	118-23	118-23#	119-3	119-3	119-3#	119-5	119-5	119-5#
	119-84	119-84	119-84	119-84#	119-85	119-85	119-85	119-85#	120-3	120-3	120-3#	120-14	120-14	120-14#
	121-3	121-3	121-3	121-3#	122-12	122-12	122-12#	123-2	123-4	123-6	123-8	123-10	123-10	123-10
	123-10#	124-10	124-10	124-10	124-10#									

Cross reference table (CREF V05.03)

Cross reference table (CREF V05.03)

Cross reference table (CREF V05.03)

Cross reference table (CREF V05.03)

Cross reference table (CREF V05.03)

```
GPHARD 116-146
GPRMA  121-1
GPRMD  114-1    114-1#   119-32   119-32#  119-45   119-45#  121-2
GPRML  119-11   119-11#  123-1    123-3    123-5    123-7
HEADER  26-34
ITEM    31-24#   41-12    41-13    41-16    41-19    41-20    41-21    41-33    41-34    41-35    41-36    41-37    41-38    41-39
        41-40    41-41    41-42    41-43    41-44    41-45    41-46    42-9     42-10    42-13    120-20   120-21   122-18
LASTAD 124-8
M$BYTE  26-34    26-34    26-34    26-34#
M$CHEC 116-235   116-235# 119-23   119-23#  119-29   119-29#  119-83   119-83#
M$CNTO 114-1     114-1#   119-11   119-11#  119-32   119-32#  119-45   119-45#  121-1    121-1#   121-2    121-2#   123-1    123-1#
       123-3     123-3#   123-5    123-5#   123-7    123-7#
M$COUN  91-12    91-12#   91-14    91-14#   91-16    91-16#   91-18    91-18#   119-65   119-65#
M$DATA  26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34#   26-34#   48-12    48-12#   48-16
        48-16#
M$DECR  28-14    28-14#   29-14    29-14#   29-16    29-16#   53-16    53-16#   53-20    53-20#   53-24    53-24#   53-28    53-28#
        53-32    53-32#   53-36    53-36#   53-40    53-40#   53-44    53-44#   53-48    53-48#   53-60    53-60#   53-65    53-65#
        53-77    53-77#   53-81    53-81#   53-85    53-85#   53-89    53-89#   53-93    53-93#   53-98    53-98#   53-102   53-102#
        53-106   53-106#  53-110   53-110#  53-114   53-114#  53-118   53-118#  53-122   53-122#  102-14   102-14#  103-21   103-21#
       112-9     112-9#   114-106  114-106# 115-16   115-16#  116 249  116-249# 117-12   117-12#  118-21   118-21#  118-23   118-23#
       119-84    119-84#  119-85   119-85#  121-3    121-3#   123-10   123-10#  124-10   124-10#  125-3    125-3#
M$DEFA  114-1    114-1#   119-11   119-11#  119-32   119-32#  119-45   119-45#  121-1    121-1#   121-2    121-2#   123-1    123-1#
        123-3    123-3#   123-5    123-5#   123-7    123-7#
M$ENDE   28-14#   29-14#   29-16#   53-16#   53-20#   53-24#   53-28#   53-32#   53-36#   53-40#   53-44#   53-48#   53-60#   53-65#
         53-77#   53-81#   53-85#   53-89#   53-93#   53-98#   53-102#  53-106#  53-110#  53-114#  53-118#  53-122#  102-14#  103-21#
        112-9#   114-106#  116-249#  117-12#  118-21#  118-23#  119-84#  119-85#  121-3#   123-10#  124-10#
M$ERRI   55-7     55-7#    58-28    58-28#   61-24    61-24#   62-36    62-36#   63-21    63-21#   64-8     64-8#    68-22    68-22#
         70-32    70-32#   96-3     96-3#    100-29   100-29#  101-5    101-5#   105-36   105-36#  105-51   105-51#  106-12   106-12#
        107-33   107-33#  109-37   109-37#  110-36   110-36#  111-5    111-5#   116-239  116-239# 116-245  116-245# 119-22   119-22#
        119-28   119-28#
M$EXCP  114-1    114-1    114-1#   119-32   119-32   119-32#  119-45   119-45   119-45#  121 1    121-1    121-1#   121-2    121-2
        121-2#
M$EXIT 116-235   116-235# 119-23   119-23#  119-29   119-29#  119-83   119-83#
M$EXSE 116-235#  119-23#  119-29#  119-83#
M$EXTJ 116-235#  119-23#  119-29#  119-83#
M$GEN   26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26 34    26-34    26-34
        26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34    26-34#   26-34#   26-34#   26-34#
        26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#
        26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   26-34#
        26-34#   26-34#   26-34#   26-34#   26-34#   26-34#   27-9     27-9#    28-10    28-10    28-10#   28-10#   28-14    28-14#
        29-10    29-10    29-10#   29-10#   29-14    29-14#   48-12    48-12#   48-16    48-16#   53-14    53-14#   53-16    53-16#
        53-18    53-18#   53-20    53-20#   53-22    53-22#   53-24    53-24#   53-26    53-26#   53-28    53-28#   53-30    53-30#
        53-32    53-32#   53-34    53-34#   53-36    53-36#   53-38    53-38#   53-40    53-40#   53-42    53-42#   53-44    53-44#
        53-46    53-46#   53-48    53-48#   53-50    53-50#   53-60    53-60#   53-62    53-62#   53-65    53-65#   53-67    53-67#
        53-77    53-77#   53-79    53-79#   53-81    53-81#   53-83    53-83#   53-85    53-85#   53-87    53-87#   53-89    53-89#
        53-91    53-91#   53-93    53-93#   53-95    53-95#   53-98    53-98#   53-100   53-100#  53-102   53-102#  53-104   53-104#
        53-106   53-106#  53-108   53-108#  53-110   53-110#  53-112   53-112#  53-114   53-114#  53-116   53-116#  53-118   53-118#
        53-120   53-120#  53-122   53-122#  102-10   102-10#  102-14   102-14#  103 18   103-18#  103-21   103-21#  112-5    112-5#
       112-9     112-9#   114-1    114-1#   115-10   115-10#  116-45   116-45#  116-249  116-249# 117-10   117-10#  117-12   117-12#
       118-8     118-8#   118-21   118-21#  119-5    119-5#   119-11   119-11#  119-32   119-32#  119-45   119-45#  119-84   119-84#
       120-14    120-14#  121-3    121-3#   122-12   122-12#  123-10   123-10#  124-8    124 8#   125-3    125-3#   125-8    125 8#
M$GENB 114-1     114-1#   119-11   119-11#  119-32   119-32#  119-45   119-45#
M$GETS  28-14    28-14#   29-14    29-14#   29-16    29-16#   53-16    53-16#   53-20    53-20#   53-24    53-24#   53-28    53-28#
```

	53-32	53-32#	53 36	53-36#	53-40	53-40#	53 44	53-44#	53-48	53-48#	53-60	53-60#	53 65	53 65#
	53-77	53-77#	53-81	53-81#	53-85	53-85#	53-89	53-89#	53-93	53-93#	53-98	53-98#	53-102	53-102#
	53-106	53-106#	53-110	53-110#	53-114	53-114#	53-118	53-118#	53-122	53-122#	102-14	102-14#	103-21	103 21#
	112-9	112-9#	114-106	114-106#	115-16	115-16#	116-249	116-249#	117-12	117-12#	118-21	118-21#	118-23	118-23#
	119-84	119-84#	119-85	119-85#	121-3	121-3#	123-2	123-2#	123-4	123-4#	123-6	123-6#	123-8	123-8#
	123-10	123-10#	124-10	124-10#										
M$GETT	116-235#	119-23#	119-29#	119-83#	123-2	123-2#	123-4	123-4#	123-6	123-6#	123-8	123-8#		
M$GNGB	26-26#	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26 34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26 34
	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26 34#	26-34#	26 34#	26-34#
	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#
	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#
	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	27-9	27-9#	28-10	28-10	28-10#	29-10	29-10
	29-10#	30-3#	48-12	48-12#	48-16	48-16#	53-14	53-14#	53-18	53-18#	53-22	53-22#	53-26	53 26#
	53-30	53-30#	53-34	53-34#	53-38	53-38#	53-42	53-42#	53-46	53-46#	53-50	53-50#	53-62	53-62#
	53-67	53-67#	53-79	53-79#	53-83	53-83#	53-87	53-87#	53-91	53-91#	53-95	53-95#	53 100	53 100#
	53-104	53-104#	53-108	53-108#	53-112	53-112#	53-116	53-116#	53-120	53-120#	102-10	102-10#	103-18	103-18#
	112-5	112-5#	115-3#	115-10	115-10#	116-45	116-45#	117-10	117-10#	118-8	118-8#	119-3#	120-3#	120-14
	120-14#	122-12	122-12#	124-8	124-8#									
M$GNIN	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26-34	26-34	26-34	26-34	26-34	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#
	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#
	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#
	26-34#	26-34#	26-34#	26-34#	26-34#	26-34#	27-9	27-9	27-9#	27-9#	28-10	28-10#	29-10	29 10#
	48-12	48-12	48-12#	48-12#	48-16	48-16	48-16#	48-16#	53-16	53-16#	53-20	53-20#	53-24	53-24#
	53-28	53-28#	53-32	53-32#	53-36	53-36#	53-40	53-40#	53-44	53-44#	53-48	53-48#	53-60	53-60#
	53-65	53-65#	53-77	53-77#	53-81	53-81#	53-85	53-85#	53-89	53-89#	53-93	53-93#	53-98	53 98#
	53-102	53-102#	53-106	53-106#	53-110	53-110#	53-114	53-114#	53-118	53-118#	53-122	53-122#	55-7	55-7
	55-7	55-7	55-7#	55-7#	55-7#	55-7#	55-7#	55-8	55-8#	58-11	58-11	58-11	58-11	58-11
	58-11	58-11#	58-11#	58-11#	58-11#	58-11#	58-11#	58-12	58-12#	58-22	58-22	58-22#	58-22#	58-28
	58-28	58-28	58-28	58-28#	58-28#	58-28#	58-28#	58-28#	58-30	58-30#	60-12	60-12#	61-8	61-8#
	61-24	61-24	61-24	61-24	61-24#	61-24#	61-24#	61-24#	61-24#	62-36	62-36	62-36	62-36	62-36#
	62-36#	62-36#	62-36#	62-36#	63-21	63-21	63-21	63-21	63-21#	63-21#	63-21#	63-21#	63-21#	64-8
	64-8	64-8	64-8	64-8#	64-8#	64-8#	64-8#	64-8#	68-22	68-22	68-22	68-22	68-22#	68-22#
	68-22#	68-22#	68-22#	70-32	70-32	70-32	70-32	70-32#	70-32#	70-32#	70-32#	70-32#	73-19	73-19#
	73-20	73-20	73-20#	73-20#	73-24	73-24	73-24#	73-24#	73-25	73 25#	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#	91-14#	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#	100-21	100-21#	100-29	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#	105-36	105-36	105-36	105-36	105-36#	105-36#	105-36#	105-36#	105-36#	105-51	105-51	105-51	105-51
	105-51#	105-51#	105-51#	105-51#	105-51#	106-12	106-12	106-12	106-12	106-12#	106-12#	106-12#	106-12#	106-12#
	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#	109-37
	109-37	109-37	109-37	109-37#	109-37#	109-37#	109-37#	109-37#	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#
	112-9	112-9#	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1	114-1#	114-1#	114-1#	114-1#
	116-47	116-47	116-47#	116-47#	116-49	116-49#	116-53	116-53	116-53#	116-53#	116-55	116-55#	116-59	116-59
	116-59#	116-59#	116-61	116-61#	116-66	116-66	116-66#	116-66#	116-68	116-68#	116-69	116-69#	116-93	116-93
	116-93#	116-93#	116-93#	116-94	116-94#	116-95	116-95	116-95#	116-95#	116-95#	116-96	116-96#	116-106	116-106
	116-106	116-106	116-106	116-106	116-106#	116-106#	116-106#	116-106#	116-106#	116-106#	116-109	116-109	116-109#	116-109#
	116-146	116-146	116-146#	116-146#	116-146#	116-147	116-147#	116-233	116-233	116-233#	116-233#	116-235	116-235	116-235#

	116-235#	116 239	116-239	116-239	116-239	116-239#	116 239#	116-239#	116-239#	116-239#	116-240	116-240#	116-245	116-245
	116-245	116-245	116-245#	116-245#	116-245#	116-245#	116-245#	116-246	116-246#	116-249	116-249#	117-12	117-12#	118-11
	118-11	118-11	118-11	118-11	118-11	118-11#	118-11#	118-11#	118-11#	118-11#	118-11#	118-19	118-19	118-19#
	118-19#	118-21	118-21#	119-8	119-8#	119-9	119-9#	119-11	119-11	119-11	119-11	119-11	119-11	119-11#
	119-11#	119-11#	119-11#	119-20	119-20#	119-21	119-21#	119-22	119-22	119-22	119-22	119-22#	119-22#	119-22#
	119-22#	119-22#	119-23	119-23	119-23#	119-23#	119-28	119-28	119-28	119-28	119-28#	119-28#	119-28#	119-28#
	119-28#	119-29	119-29	119-29#	119-29#	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32	119-32#
	119-32#	119-32#	119-32#	119-33	119-33	119-33#	119-33#	119-45	119-45	119 45	119-45	119-45	119-45	119-45
	119-45	119-45#	119-45#	119-45#	119-45#	119-65	119-65	119-65	119-65	119-65	119-65	119-65#	119-65#	119-65#
	119-65#	119-65#	119-83	119-83	119-83#	119-83#	119-84	119-84#	120-14	120-14#	121-1	121-1	121-1	121-1
	121-1#	121-2	121-2	121-2	121-2	121-2	121-2#	121-3	121-3#	122-12	122-12#	123-1	123-1	123-1
	123-1#	123-2	123-2#	123-3	123-3	123-3	123-3#	123-4	123-4#	123-5	123-5	123-5	123-5#	123-6
	123-6#	123-7	123-7	123-7	123-7#	123-8	123-8#	123-9	123-9	123-9#	123-9#	123-10	123-10#	124-8
	124-8	124-8	124-8#	125-3	125-3	125-3#	125-3#							
M$GNLS	114-1	114-1#	119-11	119-11#	119-32	119-32#	119-45	119-45#						
M$GNTA	28-14	28-14#	29-14	29-14#	53-16	53-16#	53-20	53-20#	53-24	53-24#	53-28	53-28#	53-32	53-32#
	53-36	53-36#	53-40	53-40#	53-44	53-44#	53-48	53-48#	53-60	53-60#	53-65	53-65#	53-77	53-77#
	53-81	53-81#	53-85	53-85#	53-89	53-89#	53-93	53-93#	53-98	53-98#	53-102	53-102#	53-106	53-106#
	53-110	53-110#	53-114	53-114#	53-118	53-118#	53-122	53-122#	102-14	102-14#	103-21	103-21#	112-9	112-9#
	116-249	116-249#	117-12	117-12#	118-21	118-21#	119-84	119-84#	121-3	121-3#	123-10	123-10#	125-3	125-3#
	125-8	125-8#												
M$GNTE	119-5	119-5#												
M$HAPT	26-34	26-34#												
M$HNAP	26-34	26-34#												
M$INCR	26-26	26-26#	28-10	28-10	28-10#	28-10#	29-10	29-10	29-10#	29-10#	30-3	30-3#	53-14	53-14
	53-14#	53-14#	53-16#	53-18	53-18	53-18#	53-18#	53-20#	53-22	53-22	53-22#	53-22#	53-24#	53-26
	53-26	53-26#	53-26#	53-28#	53-30	53-30	53-30#	53-30#	53-32#	53-34	53-34	53-34#	53-34#	53-36#
	53-38	53-38	53-38#	53-38#	53-40#	53-42	53-42	53-42#	53-42#	53-44#	53-46	53-46	53-46#	53-46#
	53-48#	53-50	53-50	53-50#	53-50#	53-60#	53-62	53-62	53-62#	53-62#	53-65#	53-67	53-67	53-67#
	53-67#	53-77#	53-79	53-79	53-79#	53-79#	53-81#	53-83	53-83	53-83#	53-83#	53-85#	53-87	53-87
	53-87#	53-87#	53-89#	53-91	53-91	53-91#	53-91#	53-93#	53-95	53-95	53-95#	53-95#	53-98#	53-100
	53-100	53-100#	53-100#	53-102#	53-104	53-104	53-104#	53-104#	53-106#	53-108	53-108	53 108#	53-108#	53-110#
	53-112	53-112	53-112#	53-112#	53-114#	53-116	53-116	53-116#	53-116#	53-118#	53-120	53-120	53-120#	53-120#
	53-122#	55-7#	55-8#	58-11#	58-12#	58-22#	58-28#	58-30#	60-12#	61-8#	61-24#	62-36#	63-21#	64-8#
	68-22#	70-32#	73-19#	73-20#	73-24#	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#	105-36#	105-51#	106-12#	107-12#	107-27#	107-30#
	107-33#	109-37#	110-27#	110-36#	111-5#	112-5	112-5	112-5#	112-5#	114-1	114-1#	114-1#	115-3	115-3#
	115-10	115-10	115-10#	115-10#	116-45	116-45	116-45#	116-45#	116-47#	116-53#	116-59#	116-66#	116-69#	116-93#
	116-95#	116-106#	116-109#	116-146#	116-233#	116-235#	116-239#	116-240#	116-245#	116-246#	116-249#	117-10	117-10	117-10#
	117-10#	117-12#	118-8	118-8	118-8#	118-8#	118-11#	118-19#	118-21#	119-3	119-3#	119-5	119-5	119-5
	119-5#	119-5#	119-5#	119-8#	119-11	119-11#	119-11#	119-20#	119-22#	119-23#	119-28#	119-29#	119-32	119-32#
	119-32#	119-33#	119-45	119-45#	119-45#	119-65#	119-83#	119-84#	120-3	120-3#	120-14	120-14	120-14#	120-14#
	122-12	122-12	122-12#	122-12#	125-1	125-1#	125-3	125-3	125-3	125-3#				
M$LDRO	58-22	58-22#	73-20	73-20#	107-30	107-30#	116-47	116-47#	116-53	116-53#	116-59	116-59#	116-66	116-66#
	116-93	116-93#	116-95	116-95#	116-146	116-146#	116-233	116-233#	118-19	118-19#	119-33	119-33#		
M$MCHI	26-8	26-8#												
M$MCLO	26-8	26-8#												
M$POP	28-14	28-14#	29-14	29-14#	29-16	29-16#	53-16	53-16#	53-20	53-20#	53-24	53-24#	53-28	53-28#
	53-32	53-32#	53-36	53-36#	53-40	53-40#	53-44	53-44#	53-48	53-48#	53-60	53-60#	53-65	53-65#
	53-77	53-77#	53-81	53-81#	53-85	53-85#	53-89	53-89#	53-93	53-93#	53-98	53-98#	53-102	53-102#
	53-106	53-106#	53-110	53-110#	53-114	53-114#	53-118	53-118#	53-122	53-122#	102-14	102-14#	103-21	103-21#
	112-9	112-9#	114-106	114-106#	115-16	115-16#	116-249	116-249#	117-12	117-12#	118-21	118-21#	118-23	118-23#
	119-84	119-84#	119-85	119-85#	121-3	121-3#	123-10	123-10#	124-10	124-10#				
M$PRIN	91-12	91-12#	91-14	91-14#	91-16	91-16#	91-18	91-18#	119-65	119-65#				
M$PUSH	26-26	26-26#	28-10	28-10#	29-10	29-10#	30-3	30-3#	53-14	53-14#	53 18	53-18#	53-22	53-22#
	53-26	53-26#	53-30	53-30#	53-34	53-34#	53-38	53-38#	53-42	53-42#	53-46	53-46#	53-50	53-50#

Cross reference table (CREF V05.03)