RA60/80/81 UDA50/KDA50-Q FRMTR
UDA50A KDA50Q CZUDKA0

AH-T939A-MC FICHE
FICHE 01 OF 01

FEB 1985
digital
Made In USA

IDENTIFICATION

PRODUCT CODE: AC-T938A-MC

PRODUCT NAME: CZUDKAO UDA50A/KDA50-Q FORMATTER

PRODUCT DATE: 2-OCT-1984

MAINTAINER: ROGER OAKEY

AUTHOR: BRIAN SCHOW

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

TABLE OF CONTENTS

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.

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 (O) ?  160000<CR>
SUB-DEVICE # (O) ?  5<CR>
Q-FACTOR (O) O ?  <CR>

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

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

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

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

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

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

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

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

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

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 ECP    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.

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 SEL_CTED 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-existent 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.

00025 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 loop. 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  8B00  0080  0013  1000
LAST BLOCK ACCESSED (16-BIT OCTAL): 000000  000000
```

The disk drive reported an error. You may see the drive's fault light come on. The formatter will attempt to clear the error in the drive and continue. This error does not mean that anything is necessarily wrong unless this error is printed many times. If you see many of these errors, you may wish to stop the format and run diagnostics on the disk drive. But remember, if you stop the formatter the disk will not be usable and the diagnostics will report 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-existent unit number
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

The DUP program issues the following messages upon normal completion:

Format completed

n Revectored LBNS

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

n Primary revectored LBNS

Where n is the number of LBNs which were primary 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,A,0,7200.,1,PRIO7

002000                                                                  L$NAME::
002000
002000  103                                                                     .ASCII /C/
002001  132                                                                     .ASCII /Z/
002002  125                                                                     .ASCII /U/
002003  104                                                                     .ASCII /D/
002004  113                                                                     .ASCII /K/
002005  000                                                                     .BYTE   0
002006  000                                                                     .BYTE   0
002007  000                                                                     .BYTE   0
002010                                                                  L$REV::
002010  101                                                                     .ASCII  /A/
002011                                                                  L$DEPO::
002011  060                                                                     .ASCII  /0/
002012                                                                  L$UNIT::
002012  000001                                                                  .WORD   T$PTHV
002014                                                                  L$TIML::
002014  016040                                                                  .WORD   7200.
002016                                                                  L$HPCP::
002016  022754                                                                  .WORD   L$HARD
002020                                                                  L$SPCP::
002020  023032                                                                  .WORD   L$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$PRIO::
002042  000340                                                                  .WORD   PRIO7
002044                                                                  L$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  021242                          .WORD   L$INIT
002106                          L$CCP::
002106  022200                          .WORD   L$CLEAN
002110                          L$ACP::
002110  022176                          .WORD   L$AUTO
002112                          L$PRT::
002112  021234                          .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
                                                        L$DISPATCH::
002124                                                          .WORD   T1
002124  022264
```

```
                                .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:
```

```
      1                                 .SBTTL  SOFTWARE P-TABLE
      2
      3                                 ;++
      4                                 ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
      5                                 ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
      6                                 ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
      7                                 ; AT RUN TIME.
      8                                 ;--
      9
     10 002134                                  BGNSW   SFPTBL
        002134  000001                                                          .WORD   L10001-L$SW/2
        002136                                                          L$SW::
        002136                                                          SFPTBL::
     11
     12                                                         ;OFFSET         USE
     13 002136  000007                          .WORD   7       ; 0.    YES/NO ANSWERS
     14 002140                                  ENDSW
        002140                                                          L10001:
     15
     16 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
```

GLOBAL EQUATES SECTION

```
                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
```

GLOBAL EQUATES SECTION

```
                        ;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
```

GLOBAL EQUATES SECTION

```
                                ;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
```

GLOBAL EQUATES SECTION

```
                                ;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 MAINTENANCE 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.MSTI=  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.MSTI=  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
```

GLOBAL EQUATES SECTION

```
                                ;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
```

```
                        ;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 PYABLES
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 SO.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  016270                  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.
```

GLOBAL DATA SECTION

```
                                                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
        000000                                          .WORD 0
```

GLOBAL TEXT SECTION

```
                                .SBTTL  GLOBAL TEXT SECTION

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

                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
003454                                  DEVTYP <RA SERIES DISK DRIVE>
003454                                                                  L$DVTYP::
003454  122     101     040                                                     .ASCIZ  /RA SERIES DISK DRIV
E/
                                                                                .EVEN

                                ; TEST DESCRIPTION
                                ;
003502                                  DESCRIPT <CZUDKO UDA50A,KDA50A-Q FORMATTER>
003502                                                                  L$DESC::
003502  103     132     125                                                     .ASCIZ  /CZUDKO UDA50A,KDA50
A-Q FORMATTER/
                                                                                .EVEN
```

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  UNQUES: .ASCIZ\ARE YOU SURE YOU WANT TO RUN THIS FORMATTER\
```

GLOBAL TEXT SECTION

```
                                        ; FORMAT STATEMENTS USED IN PRINT CALLS
 1
 2
 3 003704  045  124  000  ERRONE: .ASCIZ\%T\
 4 003707  045  116  000  ERRNL:  .ASCIZ\%N\
 5 003712  042  040  040  RNTIM:  .ASCIZ\"  RUNTIME "D16":"\
 6 003735  104  071  042  RNTIM1: .ASCIZ\D9":"\
 7 003743  104  071  000  RNTIM2: .ASCIZ\D9\
 8 003746  042  040  040  ERRME1: .ASCIZ\"  *  *  *  ERROR PROCESSING MESSAGE STRING  *  *  *"N\
 9 004035  116  042  125  MESSG:  .ASCIZ\N"UNIT "D6" CONTROLLER AT "O16" DRIVE "D9S\
10 004110  042  116  117  NOCLOCK:.ASCIZ\"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\
11 004165  042  110  117  BASNO:  .ASCIZ\"HOST PROGRAM"\
12 004204  042  040  040  BASL2:  .ASCIZ\"  CONTROLLER AT "O16\
13 004232  042  040  040  BASL3:  .ASCIZ\"  DRIVE "D9\
14 004247  000              BAS:    .BYTE 0                                 ;NULL TO PRINT NOTHING
15
16 004250  122  066  122  BASLN:  .ASCIZ\R6R6R6R6\                         ;USED TO PRINT BASIC LINE OF ERROR MESSAGE
17 004261  116  042  123  SERNUM: .ASCIZ\N"SERIAL NUMBER FOR UNIT "D6" CONTROLLER AT "O16" DRIVE "D9\
18 004355  042  123  124  WNSTOP: .ASCII\"STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK"N\
19 004450  042  125  116          .ASCII\"UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN"N\
20 004541  042  102  122          .ASCIZ\"BROUGHT ONLINE."NN\
21 004565  116  042  127  WNSTRT: .ASCII\N"WARNING:"N\
22 004601  042  040  040          .ASCII\"      THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC"N\
23 004703  042  040  040          .ASCII\"      TOOL.  RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK"N\
24 005002  042  040  040          .ASCIZ\"      DRIVE'S SERVICE MANUAL."N\
```

GLOBAL TEXT SECTION

```
      1 005043                          X1A:
      2 005043                          X2A:
      3 005043                          X3A:
      4 005043    042     111     116   X8A:    .ASCIZ\"INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS"N\
      5 005122    122     065     122   X1:     .ASCIZ\R5R6"CONTROLLER HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE"N\
      6 005225    122     065     122   X2:     .ASCIZ\R5R6"MULTIPLE UNITS SELECT THE SAME DRIVE"N\
      7 005301    122     065     122   X3:     .ASCIZ\R5R6"MORE THAN EIGHT DRIVES SELECTED ON THIS CONTROLLER"N\
      8 005373    122     064     042   X4:     .ASCII\R4"NOT ENOUGH ROOM IN MEMORY TO FORMAT THE UNITS SELECTED"N\
      9 005466    042     120     114           .ASCIZ\"PLEASE START PROGRAM OVER AND FORMAT FEWER UNITS AT A TIME"N\
     10 005564    122     065     122   X8:     .ASCIZ\R5R6"TWO CONTROLLERS USE THE SAME VECTOR"N\
     11 005637    122     064     042   X9:     .ASCII\R4"ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE."N\
     12 005742    042     120     114           .ASCIZ\"PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK."N\
     13 006031    122     064     042   X10:    .ASCIZ\R4"THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE."N\
     14 006130    122     065     042   X14:    .ASCII\R5"CONTROLLER IS NOT SUPPORTED BY THIS FORMATTER PROGRAM.  THIS"N\
     15 006231    042     120     122           .ASCII\"PROGRAM REQUIRES A UDA50-A (MODEL 6) OR A KDA50-Q (MODEL 13)"N\
     16 006330    042     103     117           .ASCIZ\"CONTROLLER.  CONTROLLER REPORTED MODEL CODE "D4"."N\
     17 006415    122     065     042   X20:    .ASCII\R5"MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS"N\
     18 006502    042     103     110           .ASCII\"CHECK CSR SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE OR BUS"N\
     19 006607    042     117     122           .ASCIZ\"OR REPLACE CONTROLLER PROCESSOR MODULE"N\
     20 006661    122     065     042   X21:    .ASCII\R5"CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE"NR8\
     21 006750    042     122     105           .ASCIZ\"REPLACE CONTROLLER SDI MODULE"N\
     22 007011    122     065     042   X21A:   .ASCIZ\R5"CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE"NR8R7\
     23 007103    122     065     042   X22:    .ASCII\R5"STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION"N\
     24 007201    042     123     124           .ASCIZ\"STEP BIT EXPECTED "O16NR8R7\
     25 007236    122     065     042   X23A:   .ASCII\R5"CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATI
ON"
     26 007357    104     071     042           .ASCII\D9" WORDS WERE TO BE CLEARED STARTING AT ADDRESS "O16N\
     27 007445    042     106     111           .ASCII\"FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):"N\
     28 007522    123     066     042           .ASCIZ\S6"ADDRESS"S4"CONTENTS"N\
     29 007553    123     067     117   X23B:   .ASCIZ\S7O16SSO16N\
     30 007567    122     065     042   X24:    .ASCII\R5"SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION"N\
     31 007677    042     120     125           .ASCIZ\"PURGE/POLE DIAGNOSTICS WERE REQUESTED"NR8R7\
     32 007754    122     065     042   X25:    .ASCII\R5"CONTROLLER DID NOT RETURN CORRECT DATA IN SA REGISTER DURING"N\
     33 010055    042     111     116           .ASCII\"INITIALIZATION"N\
     34 010076    042     040     040           .ASCIZ\"  SA EXPECTED  "O16NR8R7\
     35 010130    122     065     042   X30:    .ASCIZ\R5"CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE RUNNING FORMATTER"NR8
\
     36 010246    122     065     042   X31:    .ASCIZ\R5"FORMATTER PROGRAM IS HUNG"N\
     37 010305    122     065     042   X32:    .ASCIZ\R5"MESSAGE BUFFER RECEIVED FROM FORMATTER WITH UNKNOWN REQUEST NUMBER"N\
     38 010415    122     065     042   X36:    .ASCII\R5"NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS"N\
     39 010506    042     127     110           .ASCIZ\"WHILE LOADING FORMATTER"N\
     40 010541    122     065     042   X37:    .ASCIZ\R5"CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE LOADING FORMATTER"NR8
R7\
     41 010661    122     065     042   X100:   .ASCIZ\R5"FORMATTER ASKED UNEXPECTED QUESTION ("D12")"N\
     42 010742    122     065     042   X101:   .ASCIZ\R5"FORMATTER REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION"N\
```

GLOBAL TEXT SECTION

```
      1 011043     042     115     105  XMSG1:   .ASCIZ\"MESSAGE BUFFER CONTAINS:"N\
      2 011077     123     063     117  XMSG2:   .ASCIZ\S3016S1016S1016S1016S1016S1016S1016N\
      3 011144     122     065     042  XPKT1:   .ASCII\R5"RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA"N\
      4 011247     042     105     111           .ASCII\"EITHER CONTROLLER RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED"N\
      5 011354     042     103     117           .ASCII\"CORRECTLY"N\
      6 011370     123     063     042           .ASCIZ\S3"COMMAND PACKET SENT"S6"RESPONSE PACKET RECEIVED"N\
      7 011455     123     066     117  XPKT2:   .ASCIZ\S6016S1016S14016S1016N\
      8 011504     042     040     040  XSA:     .ASCIZ\"  SA CONTAINS  "016N\
      9 011532     042     122     105  XFRU:    .ASCIZ\"REPLACE CONTROLLER PROCESSOR MODULE"N\
     10
     11
     12 011601     045     101     111  SERNX:   .ASCIZ\%AINPUT ERROR. ANSWER WITH DECIMAL NUMBER LO= 0  HI= %T\
     13 011671     042     111     116  DATEX:   .ASCIZ\"INPUT ERROR."\
     14 011710     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                        SVCINS= -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

011772                          BGNMSG ERR001
                                        PNTB X1,#X1A
011772  012746  005043                                                  MOV #X1A,-(SP)
011776  004137  016422                                                  JSR R1,LPNTB
012002  005122                                                          .WORD X1
012004  000002                                                          .WORD PNT.CT
012006                          ENDMSG

012010                          BGNMSG ERR002
                                        PNTB X2,#X2A
012010  012746  005043                                                  MOV #X2A,-(SP)
012014  004137  016422                                                  JSR R1,LPNTB
012020  005225                                                          .WORD X2
012022  000002                                                          .WORD PNT.CT
012024                          ENDMSG

012026                          BGNMSG ERR003
                                        PNTB X3,#X3A
012026  012746  005043                                                  MOV #X3A,-(SP)
012032  004137  016422                                                  JSR R1,LPNTB
012036  005301                                                          .WORD X3
012040  000002                                                          .WORD PNT.CT
012042                          ENDMSG

012044                          BGNMSG ERR004
                                        PNTB X4
012044  004137  016422                                                  JSR R1,LPNTB
012050  005373                                                          .WORD X4
012052  000000                                                          .WORD PNT.CT
012054                          ENDMSG

012056                          BGNMSG ERR008
                                        PNTB X8,#X8A
012056  012746  005043                                                  MOV #X8A,-(SP)
012062  004137  016422                                                  JSR R1,LPNTB
012066  005564                                                          .WORD X8
012070  000002                                                          .WORD PNT.CT
012072                          ENDMSG

012074                          BGNMSG ERR009
                                        PNTB X9
012074  004137  016422                                                  JSR R1,LPNTB
012100  005637                                                          .WORD X9
012102  000000                                                          .WORD PNT.CT
```

GLOBAL ERROR REPORT SECTION

```
     36 012104                                  ENDMSG
     37
     38 012106                                  BGNMSG ERR010
     39 012106                                         PNTB X10
        012106  004137  016422                                          JSR R1,LPNTB
        012112  006031                                                  .WORD X10
        012114  000000                                                  .WORD PNT.CT
     40 012116                                  ENDMSG
     41
     42 012120                                  BGNMSG ERR014
     43 012120                                         PNTB X14,R2
        012120  010246                                                  MOV R2,-(SP)
        012122  004137  016422                                          JSR R1,LPNTB
        012126  006130                                                  .WORD X14
        012130  000002                                                  .WORD PNT.CT
     44 012132                                  ENDMSG
     45
     46 012134                                  BGNMSG ERR020
     47 012134                                         PNTB X20
        012134  004137  016422                                          JSR R1,LPNTB
        012140  006415                                                  .WORD X20
        012142  000000                                                  .WORD PNT.CT
     48 012144                                  ENDMSG
     49
     50 012146                                  BGNMSG ERR021
     51 012146  010201                                 MOV R2,R1
     52 012150  000301                                 SWAB R1
     53 012152                                         AND 2,R1
        012152  042701  177775                                          BIC #^C<2>,R1
     54 012156  001406                                 BEQ ERR21A
     55 012160                                         PNTB X21,R2
        012160  010246                                                  MOV R2,-(SP)
        012162  004137  016422                                          JSR R1,LPNTB
        012166  006661                                                  .WORD X21
        012170  000002                                                  .WORD PNT.CT
     56 012172  000405                                 BR EOFMSG
     57 012174                                  ERR21A:
     58 012174                                         PNTB X21A,R2
        012174  010246                                                  MOV R2,-(SP)
        012176  004137  016422                                          JSR R1,LPNTB
        012202  007011                                                  .WORD X21A
        012204  000002                                                  .WORD PNT.CT
     59 012206                                  EOFMSG:
     60 012206                                  ENDMSG
     61
     62 012210                                  BGNMSG ERR022
     63 012210  042737  100000  020356                BIC #SA.ERR,UDARSD
     64 012216                                         PNTB X22,UDARSD,R2
        012216  010246                                                  MOV R2,-(SP)
        012220  013746  020356                                          MOV UDARSD,-(SP)
        012224  004137  016422                                          JSR R1,LPNTB
        012230  007103                                                  .WORD X22
        012232  000004                                                  .WORD PNT.CT
     65 012234                                  ENDMSG
     66
     67 012236                                  BGNMSG ERR023
     68 012236                                         PNTB X23A,R3,R1
```

```
                012236  010146                                                  MOV R1,-(SP)
                012240  010346                                                  MOV R3,-(SP)
                012242  004137  016422                                          JSR R1,LPNTB
                012246  007236                                                  .WORD X23A
                012250  000004                                                  .WORD PNT.CT
            69  012252  005742                  TST -(R2)
            70  012254  005712          ERR23A: TST (R2)
            71  012256  001406                  BEQ ERR23B
            72  012260                          PNTB X23B,R2,(R2)
                012260  011246                                                  MOV (R2),-(SP)
                012262  010246                                                  MOV R2,-(SP)
                012264  004137  016422                                          JSR R1,LPNTB
                012270  007553                                                  .WORD X23B
                012272  000004                                                  .WORD PNT.CT
            73  012274  005722          ERR23B: TST (R2)+
            74  012276  005303                  DEC R3
            75  012300  001365                  BNE ERR23A
            76  012302                          ERR23C: PNTB XFRU
                012302  004137  016422                                          JSR R1,LPNTB
                012306  011532                                                  .WORD XFRU
                012310  000000                                                  .WORD PNT.CT
            77  012312                          ENDMSG
            78
            79  012314                          BGNMSG ERR024
            80  012314                                  PNTB X24,R2
                012314  010246                                                  MOV R2,-(SP)
                012316  004137  016422                                          JSR R1,LPNTB
                012322  007567                                                  .WORD X24
                012324  000002                                                  .WORD PNT.CT
            81  012326                          ENDMSG
            82
            83  012330                          BGNMSG ERR025
            84  012330                                  PNTB X25,R1,R2
                012330  010246                                                  MOV R2,-(SP)
                012332  010146                                                  MOV R1,-(SP)
                012334  004137  016422                                          JSR R1,LPNTB
                012340  007754                                                  .WORD X25
                012342  000004                                                  .WORD PNT.CT
            85  012344                          ENDMSG
            86
            87  012346                          BGNMSG ERR030
            88  012346                                  PNTB X30,R1
                012346  010146                                                  MOV R1,-(SP)
                012350  004137  016422                                          JSR R1,LPNTB
                012354  010130                                                  .WORD X30
                012356  000002                                                  .WORD PNT.CT
            89  012360                          ENDMSG
            90
            91  012362                          BGNMSG ERR031
            92  012362                                  PNTB X31
                012362  004137  016422                                          JSR R1,LPNTB
                012366  010246                                                  .WORD X31
                012370  000000                                                  .WORD PNT.CT
            93  012372                          ENDMSG
            94
            95  012374                          BGNMSG ERR032
            96  012374                                  PNTB X32
```

```
                012374  004137  016422                                                  JSR R1,LPNTB
                012400  010305                                                          .WORD X32
                012402  000000                                                          .WORD PNT.CT
     97         012404  004737  012574          CALL MSGPKT
     98         012410                  ENDMSG
     99
    100         012412                  BGNMSG ERR033
    101         012412  004737  012502          CALL PNTPKT
    102         012416                  ENDMSG
    103
    104         012420                  BGNMSG ERR034
    105         012420  004737  012502          CALL PNTPKT
    106         012424                  ENDMSG
    107
    108         012426                  BGNMSG ERR036
    109         012426                          PNTB X36
                012426  004137  016422                                                  JSR R1,LPNTB
                012432  010415                                                          .WORD X36
                012434  000000                                                          .WORD PNT.CT
    110         012436                  ENDMSG
    111
    112         012440                  BGNMSG ERR037
    113         012440                          PNTB X37,R1
                012440  010146                                                          MOV R1,-(SP)
                012442  004137  016422                                                  JSR R1,LPNTB
                012446  010541                                                          .WORD X37
                012450  000002                                                          .WORD PNT.CT
    114         012452                  ENDMSG
    115
    116         012454                  BGNMSG ERR100
    117         012454                          PNTB X100,(R4)
                012454  011446                                                          MOV (R4),-(SP)
                012456  004137  016422                                                  JSR R1,LPNTB
                012462  010661                                                          .WORD X100
                012464  000002                                                          .WORD PNT.CT
    118         012466                  ENDMSG
    119
    120         012470                  BGNMSG ERR101
    121         012470                          PNTB X101
                012470  004137  016422                                                  JSR R1,LPNTB
                012474  010742                                                          .WORD X101
                012476  000000                                                          .WORD PNT.CT
    122         012500                  ENDMSG
    123
    124         012502                  PNTPKT: PNTB XPKT1
                012502  004137  016422                                                  JSR R1,LPNTB
                012506  011144                                                          .WORD XPKT1
                012510  000000                                                          .WORD PNT.CT
    125         012512  010401                  MOV R4,R1
    126         012514  062701  000104          ADD #HC.CPK,R1
    127         012520  010402                  MOV R4,R2
    128         012522  062702  000020          ADD #HC.MPK,R2
    129         012526  012703  000014          MOV #12.,R3
    130         012532                  PNTPKL: PNTB XPKT2,2(R1),(R1),2(R2),(R2)
                012532  011246                                                          MOV (R2),-(SP)
                012534  016246  000002                                                  MOV 2(R2),-(SP)
                012540  011146                                                          MOV (R1),-(SP)
```

```
                012542  016146  000002                                                          MOV 2(R1),-(SP)
                012546  004137  016422                                                          JSR R1,LPNTB
                012552  011455                                                                  .WORD XPKT2
                012554  000010                                                                  .WORD PNT.CT
    131         012556  062701  000004          ADD #4,R1
    132         012562  062702  000004          ADD #4,R2
    133         012566  005303                  DEC R3
    134         012570  001360                  BNE PNTPKL
    135         012572  000207                  RETURN
    136
    137         012574                  MSGPKT: PNTB XMSG1
                012574  004137  016422                                                          JSR R1,LPNTB
                012600  011043                                                                  .WORD XMSG1
                012602  000000                                                                  .WORD PNT.CT
    138         012604  016504  000014          MOV C.RING(R5),R4
    139         012610  062704  000430          ADD #MC.BF2,R4
    140         012614  012703  000005          MOV #5,R3
    141         012620                  MSGPKL: PNTB XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)
                012620  016446  000014                                                          MOV 12.(R4),-(SP)
                012624  016446  000012                                                          MOV 10.(R4),-(SP)
                012630  016446  000010                                                          MOV 8.(R4),-(SP)
                012634  016446  000006                                                          MOV 6(R4),-(SP)
                012640  016446  000004                                                          MOV 4(R4),-(SP)
                012644  016446  000002                                                          MOV 2(R4),-(SP)
                012650  011446                                                                  MOV (R4),-(SP)
                012652  004137  016422                                                          JSR R1,LPNTB
                012656  011077                                                                  .WORD XMSG2
                012660  000016                                                                  .WORD PNT.CT
    142         012662  062704  000016          ADD #14.,R4
    143         012666  005303                  DEC R3
    144         012670  001353                  BNE MSGPKL
    145         012672  000207                  RETURN
```

```
                                SVCINS= 1                       ; LIST INSTRUCTIONS, SHIFTED RIGHT
                                SVCTST= 1                       ; LIST TEST TAGS, SHIFTED RIGHT
                                SVCSUB= 1                       ; LIST SUBTEST TAGS, SHIFTED RIGHT
                                SVCGBL= 1                       ; LIST GLOBAL TAGS, SHIFTED RIGHT
                                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

012674                          FMERR:  ERRSF 4,,ERR004
012674  104454                                                          TRAP    C$ERSF
012676  000004                                                          .WORD   4
012700  000000                                                          .WORD   0
012702  012044                                                          .WORD   ERR004
012704                                  DOCLN                   ;ABORT
012704  104444                                                          TRAP    C$DCLN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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.

 012706                                 ALOCM:  PUSH FFREE                      ;SAVE FFREE AT ENTRY
 012706  013746  002140                                                                 MOV FFREE,-(SP)
 012712  160137  002142                         SUB R1,FSIZE                   ;REDUCE SIZE OF FREE MEMORY
 012716  002766                                 BLT FMERR                      ;REPORT ERROR IF NOT ENOUGH MEMORY
 012720  060101                                 ADD R1,R1                      ;CHANGE WORDS TO BYTES
 012722  060137  002140                         ADD R1,FFREE                   ;CALCULATE NEW START OF FREE MEMORY
 012726                                         POP R1                         ;GET START OF ALLOCATED MEMORY
 012726  012601                                                                        MOV (SP)+,R1
 012730  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 TO HOST COMM AREA
                                        ;       R4 - ADDRESS OF HOST COMM AREA

012732  012701  000336          HCOMM:  MOV @HC.SIZ/2,R1                ;GET SIZE OF AREA TO ALLOCATE
012736  004737  012706                  CALL ALOCM                      ;ALLOCATE THE MEMORY
012742  010165  000014                  MOV R1,C.RING(R5)                       ;GET ADDRESS OF HOST COMM AREA
                                                                        ;PLACE IN CONTROLLER TABLE
012746  000207                          RETURN
```

```
                                        ;RESET
                                        ; RESET ALL UDA-50S IN THE CONTROLLER TABLES
                                        ;
                                        ; INPUTS:
                                        ;       IPADRS - CONTAINS ALL IP ADDRESSES
                                        ; OUTPUTS:
                                        ;       NONE
                                        ;
 9 012750                               RESET:  PUSH <R3,R4>                                    MOV R3,-(SP)
   012750  010346                                                                               MOV R4,-(SP)
   012752  010446
10 012754  005037  002172                       CLR     NXMAD
11 012760                                       SETVEC #4,#NXMI,#PRI07                          MOV     #PRI07,-(SP)
   012760  012746  000340                                                                       MOV     #NXMI,-(SP)
   012764  012746  017332                                                                       MOV     #4,-(SP)
   012770  012746  000004                                                                       MOV     #3,-(SP)
   012774  012746  000003                                                                       TRAP    C$SVEC
   013000  104437                                                                              ADD     #10,SP
   013002  062706  000010
12 013006                                       BREAK                                           TRAP    C$BRK
   013006  104422
13 013010  012703  000010                       MOV     #8.,R3          ; R3 = COUNTER OF ENTRIES
14 013014  012704  003434                       MOV     #IPADRS,R4      ; R4 -> IP ADDRESS
15 013020  005714                       1$:     TST     (R4)            ; IS THERE AN ENTRY?
16 013022  001406                               BEQ     2$              ; IF NOT, DONE
17 013024  005034                               CLR     @(R4)+          ; INIT UDA
18 013026  005737  002172                       TST     NXMAD           ; WAS THERE AN ERROR?
19 013032  001010                               BNE     3$              ; IF SO, EXIT
20 013034  005303                               DEC     R3              ; MAKE SURE WE DO NOT EXTEND OVER AREA
21 013036  001370                               BNE     1$              ; IF NOT DONE, BRANCH
22 013040                               2$:     CLRVEC  #4                                      MOV     #4,R0
   013040  012700  000004                                                                       TRAP    C$CVEC
   013044  104436
23 013046                                       POP     <R4,R3>                                 MOV (SP)+,R4
   013046  012604                                                                               MOV (SP)+,R3
   013050  012603
24 013052  000207                               RETURN
25
26 013054  005744                       3$:     TST     -(R4)           ; R4 -> UDAIP THAT FAILED
27 013056  010405                               MOV     R4,R5           ; SAVE IN R5 FOR REPORT
28 013060                                       ERRDF   20.,ERR020                              TRAP    C$ERDF
   013060  104455                                                                               .WORD   20
   013062  000024                                                                               .WORD   0
   013064  000000                                                                               .WORD   ERR020
   013066  012134
29 013070  005014                               CLR     (R4)            ; DESTROY ENTRY SO NOT TO FALL INTO RESET ERROR LOOP
30 013072                                       DOCLN                                           TRAP    C$DCLN
   013072  104444
```

```
                                        ;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.

013074  010137  002160          RUNDM:  MOV R1,URUN                     ;SAVE NUMBER OF UNITS TO RUN
013100  005037  002162                  CLR URNING                      ;CLEAR NUMBER OF UNITS RUNNING

                                        ;LOAD DM PROGRAM INTO EACH CONTROLLER

013104  013737  002160  002164          MOV URUN,UCNT                   ;SET COUNTER OF UNITS
013112  013705  002154                  MOV TSTTAB,R5                   ;GET FIRST CONTROLLER TABLE
013116                          LDDM:
013116  005065  000012                  CLR C.FLG(R5)                   ;CLEAR ALL FLAGS
013122  116537  000002  002074          MOVB C.UNIT(R5),L$LUN           ;SEE IF UNIT TO BE TESTED
013130  005765  000002                  TST C.UNIT(R5)
013134  100407                          BMI LDNEXT                      ;IF NOT, DON'T LOAD THIS UNIT
013136                                          ASSUME CT.AVL EQ BIT15
013136  004737  012732                  CALL MCOMM                      ;ALLOCATE SPACE FOR HOST COMM AREA
013142  004737  016554                  CALL LOADDM                     ;LOAD THE DM PROGRAM
013146  001402                          BEQ LDNEXT                      ;IF ERROR, GO TO NEXT CONTROLLER
013150  005237  002162                  INC URNING                      ;IF NO ERROR, COUNT UNIT RUNNING
013154  062705  000052          LDNEXT: ADD #C.SIZE,R5                  ;MOVE TO NEXT CONTROLLER TABLE
013160  005337  002164                  DEC UCNT                        ;CHECK IF MORE CONTROLLERS
013164  001354                          BNE LDDM                        ;LOAD NEXT
013166  005037  002170                  CLR UFREEZ                      ;CLEAR UNIT FREEZE FLAG
013172  012737  177777  003176          MOV #-1,FCTNUM  ;INVALIDATE FCT BLOCK NUMBER (BLOCK IN MEMORY)

                                        ;CHECK IF ANY CONTROLLERS LOADED

013200  005737  002162                  TST URNING                      ;ANY UNITS LOADED?

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

013204  000207                          RETURN
```

```
                                ;CLOSEF
                                ;
                                ;CLOSE DATA FILE FOR DM PROGRAMS
                                ;
                                ;INPUTS:
                                ;       FILOPN - ZERO IF FILE NOT OPEN
                                ;OUTPUTS:
                                ;       NONE
                                ;
013206  005737  002166          CLOSEF: TST FILOPN                      ;SEE IF FILE CURRENTLY OPEN
013212  001403                          BEQ 1$
013214                                  CLOSE                           ; IF SO, CLOSE IT
013214  104435                                                                  TRAP    C$CLOS
013216  005037  002166                  CLR FILOPN                      ;AND MARK AS SO
013222  000207                  1$:     RETURN
```

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

013224  013705  002154        RESPDM: MOV TSTTAB,R5                   ;GET CONTROLLER TABLE ADDRESS
013230  013737  002160  002164        MOV URUN,UCNT                   ;SET COUNTER OF UNITS
013236                      RESPCT: BREAK   ;ALLOW DRS TO SEE TERMINAL INPUT
013236  104422                                                              TRAP    C$BRK
013240  016504  000014              MOV C.RING(R5),R4                 ;GET HOST COMM AREA ADDRESS
013244  032765  000002  000012      BIT #CT.RN,C.FLG(R5)              ;CHECK IF PROGRAM RUNNING
013252  001502                      BEQ RSPNXT                        ;IF NOT, LOOK AT NEXT
013254  116537  000002  002074      MOVB C.UNIT(R5),L$LUN             ;STORE UNIT NUMBER UNDER TEST
013262  032765  000010  000012      BIT #CT.MSG,C.FLG(R5)             ;SEE IF INTERRUPT RECEIVED
013270  001150                      BNE RSPIN                         ;IF SO, LOOK AT PACKET
013272  032765  000004  000012      BIT #CT.CMD,C.FLG(R5)             ;SEE IF COMMAND HAS BEEN SENT
013300  001002                      BNE 1$                            ;IF NOT, SEND ONE
013302  000137  014050              JMP RSPOUT

                                        ;CHECK IF UDA STILL RUNNING

013306  011503              1$:     MOV (R5),R3                       ;GET ADDRESS OF UDAIP
013310  016301  000002              MOV 2(R3),R1                      ;LOOK AT UDASA REGISTER
013314  001405                      BEQ RSPTM                         ;IF ZERO, UDA STILL RUNNING
013316                              ERRDF 30.,ERR030                  ;REPORT UDA HAS FATAL ERROR
013316  104455                                                              TRAP    C$ERDF
013320  000036                                                              .WORD   30
013322  000000                                                              .WORD   0
013324  012346                                                              .WORD   ERR030
013326  000465                      BR RSPDRP                         ;DROP CONTROLLER FROM TESTING

                                        ;CHECK FOR TIMEOUT OF RESPONSE

013330  005765  000042      RSPTM:  TST C.TOT(R5)                     ;SEE IF DUP PROGRAM TO BE TIMED
013334  001451                      BEQ RSPNTO
013336  005737  003206              TST KW.CSR                        ;SEE IF A CLOCK ON SYSTEM
013342  001446                      BEQ RSPNTO                        ;DON'T TIME IF NO CLOCK
013344  023765  003220  000040      CMP KW.EL+2,C.TOH(R5)             ;COMPARE TO TIMEOUT COUNTER
013352  101005                      BMI RSPTMO
013354  001041                      BNE RSPNTO
013356  023765  003216  000036      CMP KW.EL,C.TO(R5)
013364  103435                      BLO RSPNTO                        ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
013366  032765  000040  000012  RSPTMO: BIT #CT.STA,C.FLG(R5)         ;SEE IF A GET DUST STATUS COMMAND OUTSTANDING
013374  001101                      BNE RSPTOE                        ;REPORT ERROR IF SO
013376  005764  000012              TST HC.CCT(R4)                    ;SEE IF UDA TOOK LAST COMMAND PACKET
013402  100476                      BMI RSPTOE                        ;REPORT ERROR IF NOT
013404  012700  000100              MOV #CT.TM1,R0                    ;SEE IF FIRST TIMEOUT ALREADY HAPPENED
013410  032765  000100  000012      BIT #CT.TM1,C.FLG(R5)
013416  001401                      BEQ 1$                            ;IF SO,
013420  006300                      ASL R0                            ;SET SECOND TIME OUT FLAG
013422  052700  000040      1$:     BIS #CT.STA,R0                    ;SET THE PROPER TIMEOUT BIT
013426  050065  000012              BIS R0,C.FLG(R5)                  ;  AND STATUS REQUESTED BIT
013432  012700  000001              MOV #OP.GDS,R0                    ;BUILD GET DUST STATUS COMMAND
013436  004737  016754              CALL BLDCMD
013442  012764  100000  000012      MOV #RG.OWN,HC.CCT(R4)            ;MARK COMMAND TO UDA
013450  005775  000000              TST @(R5)                         ;TELL UDA COMMAND IS THERE
013454  000137  014130              JMP RSPOU4
```

GLOBAL SUBROUTINES SECTION

53 013460 RSPNTO:

```
                                        ;SWITCH TO NEXT CONTROLLER
                                        
013460  005737  002170          RSPNXT: TST UFREEZ       ;FROZEN TO ONE UNIT?
013464  001264                          BNE RESPCT       ;STAY THERE IF SO
013466  062705  000052                  ADD #C.SIZE,R5                    ;MOVE TO NEXT TABLE
013472  005337  002164                  DEC UCNT                          ;CHECK IF MORE CONTROLLERS
013476  001257                          BNE RESPCT                        ;LOOK AT NEXT CONTROLLER
013500  000651                          BR RESPDM                         ;LOOK AT FIRST CONTROLLER AGAIN
                                        
                                        ;REMOVE A CONTROLLER FROM TESTING
                                        
013502  005065  000012          RSPDRP: CLR C.FLG(R5)    ;CLEAR PROGRAM RUNNING
013506  005037  002170                  CLR UFREEZ
013512  010504                          MOV R5,R4
013514  062704  000016                  ADD #C.DRO,R4
013520  012702  000010                  MOV #8.,R2
013524  012403                  1$:     MOV (R4)+,R3
013526  001420                          BEQ 3$
013530  005763  000002                  TST D.UNIT(R3)
013534                                          ASSUME DT.AVL EQ BIT15
013534  100003                          BPL 2$
013536  005302                          DEC R2
013540  001371                          BNE 1$
013542  000412                          BR 3$
013544  052763  100000  000002  2$:     BIS #DT.AVL,D.UNIT(R3)
013552  005302                          DEC R2
013554  001405                          BEQ 3$
013556  005714                          TST (R4)
013560  001403                          BEQ 3$
013562  004737  016554                  CALL LOADDM      ;START DM PROGRAM AGAIN
013566  001223                          BNE RESPCT
013570  005337  002162          3$:     DEC URNING                        ;REDUCE RUNNING CONTROLLERS COUNT
013574  001331                          BNE RSPNXT                        ;IF ANY STILL RUNNING, LOOK AT THEM
013576  000207                          RETURN                            ;ELSE RETURN TO TEST SECTION
                                        
013600                          RSPTOE: ERRDF 31.,ERR031                  ;REPORT TIMEOUT ERROR
                                                                                  TRAP    C$ERDF
                                                                                  .WORD   31
                                                                                  .WORD   0
                                                                                  .WORD   ERR031
013610  000734                          BR RSPDRP                         ;DROP CONTROLLER FROM TESTING
```

```
                                        ;CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
                                        ;
                                        ;CHECK FOR PROPER OPCODE IN END PACKET
                                        ;
013612  012700  000204          RSPIN:  MOV #OP.END+OP.SSD,R0           ;GET SEND DATA END PACKET OPCODE
013616  032765  000020  000012          BIT #CT.REQ,C.FLG(R5)           ;LOOK IF SEND DATA OR RECEIVE DATA
013624  001402                        BEQ RSPMWR
013626  012700  000205                MOV #OP.END+OP.RSD,R0           ;CHANGE TO RECEIVE DATA END PACKET OPCODE
013632  120064  000030          RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)       ;COMPARE TO OPCODE IN END PACKET
013636  001145                        BNE RSPERR
                                        ;
                                        ;LOOK AT STATUS CODE
                                        ;
013640  032764  000037  000032        BIT #ST.MSK,HC.MPK+P.STS(R4)    ;CHECK FOR STATUS CODE ST.SUC (ZERO)
013646  001004                        BNE RSPERW
                                        ;
                                        ;CHECK FOR EXPECTED REFERENCE NUMBER
                                        ;
013650  026564  000050  000020        CMP C.REF(R5),HC.MPK+P.CRF(R4)  ;CHECK IF CORRECT REF NUMBER
013656  001405                        BEQ RSPPTW
013660                          RSPERW: ERRDF 33,,ERR033
                                                                                TRAP    C$ERDF
013660  104455                                                                  .WORD   33
013662  000041                                                                  .WORD   0
013664  000000                                                                  .WORD   ERR033
013666  012412
013670  000704                        BR RSPDRP                       ;DROP UNIT FROM TESTING
                                        ;
                                        ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
                                        ;
013672  032765  000020  000012  RSPPTW: BIT #CT.REQ,C.FLG(R5)           ;CHECK IF RESPONSE FROM DM PROGRAM
013700  001463                  RSPOU:  BEQ RSPOUT                      ;LOOK AT REQUEST NUMBER IF SO
```

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

                                        RSPPT2: MOV HC.BF2(R4),R1               ;GET REQUEST NUMBER
013702  016401  000430                          BIC     #^C<DU.TYP>,R1          ;CHECK TYPE
013706  042701  007777                          BEQ 1$                          ;IF ZERO, ERROR
013712  001403                                  CMP R1,#DU.SPC                  ;CHECK IF IN EXPECTED RANGE
013714  020127  060000                          BLOS RSPPT3
013720  101405                                  ERRDF 32,,ERR032                ;BAD REQUEST NUMBER
013722                                  1$:
013722  104455                                                                          TRAP    C$ERDF
013724  000040                                                                          .WORD   32
013726  000000                                                                          .WORD   0
013730  012374                                                                          .WORD   ERR032
013732  000663                                  BR RSPDRP                       ;DROP UNIT FROM TESTING

                                        RSPPT3: MOV HC.MPK+P.BCNT(R4),R3        ;GET BYTE COUNT OF CHARACTERS RECEIVED IN R3
013734  016403  000034                          SUB #2,R3                       ;(FIRST TWO CHARACTERS ARE TYPE WORD)
013740  162703  000002                          MOV #OP.SSD,R0                  ;BUILD A SEND DATA COMMAND PACKET
013744  012700  000004                          CALL BLDCMD                     ;  FOR ANSWER TO DM PROGRAM
013750  004737  016754                          MOV #HC.BF1,R0                  ;POINT TO BUFFER IN PACKET
013754  012700  000164                          CALL CLRBUF                     ; AND CLEAR BUFFER
013760  004737  017116                          MOV R4,R2                       ;R2 POINTS TO SEND BUFFER
013764  010402                                  ADD #HC.BSZ,R4          ;R4 POINTS TO CHARACTERS IN RECEIVE BUFFER
013766  062704  000244                          BIC #DU.TYP,(R4)+       ;CLEAR TYPE FIELD IN BUFFER
013772  042724  170000                          SWAB R1                         ;GET TYPE RIGHT JUSTIFIED
013776  000301                                  ASR R1                          ;TIMES TWO
014000  006201                                  ASR R1
014002  006201                                  ASR R1
014004  006201                                  MOV R1,R0                       ;COPY MESSAGE TYPE TO R0
014006  010100                                  CLR R1                          ;R1 CONTAINS ZERO SEND BYTE COUNT
014010  005001                                  CALL @RSPDSP-2(R0)              ;CALL REQUESTED ROUTINE
014012  004770  014276                          BNE RSPDRP                      ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
014016  001231                                                                  ; Z SET IF UNIT TO CONTINUE RUNNING

014020  016504  000014                          MOV C.RING(R5),R4               ;GET RING ADDRESS
014024  032701  000001                          BIT #1,R1                       ;LOOK AT CHARACTER COUNT TO SEND TO DUP PROGRAM
014030  001401                                  BEQ 1$                          ;IF AN ODD COUNT
014032  005201                                  INC R1                          ;  INCREASE BY ONE
014034  010164  000120                  1$:     MOV R1,HC.CPK+P.BCNT(R4)        ;PUT CHARACTER COUNT IN COMMAND PACKET
014040  100003                                  BPL RSPOUT                      ;IF NEGATIVE BYTE COUNT RETURNED
014042  042765  000020  000012                  BIC #CT.REQ,C.FLG(R5)           ;  DON'T SEND ANY DATA TO UDA

                                        ;SEND COMMAND BACK TO UDA

014050  042765  000350  000012  RSPOUT: BIC #CT.MSG+CT.STA+CT.TM1+CT.TM2,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
014056  032765  000020  000012          BIT #CT.REQ,C.FLG(R5)           ;CHECK WHICH COMMAND TO SEND
014064  001014                          BNE RSPOU2                      ;BRANCH IF RESPONSE TO REQUEST

014066  012700  000005                  MOV #OP.RSD,R0                  ;BUILD RECEIVE DATA COMMAND
014072  004737  016754                  CALL BLDCMD
014076  012700  000430                  MOV #HC.BF2,R0                  ;POINT TO MESSAGE BUFFER
014102  004737  017116                  CALL CLRBUF                     ;  AND CLEAR IT
014106  052765  000020  000012          BIS #CT.REQ,C.FLG(R5)           ;SET REQUEST BIT
014114  000403                          BR RSPOU3

014116  042765  000020  000012  RSPOU2: BIC #CT.REQ,C.FLG(R5)           ;CLEAR REQUEST BIT
014124                          RSPOU3:
014124  004737  017040                  CALL SNDCMD                     ;SEND COMMAND TO UDA
014130  016500  000042          RSPOU4: MOV C.TOT(R5),R0                ;SET TIMEOUT
```

```
     54 014134  010501                          MOV R5,R1
     55 014136  062701  000036                  ADD #C.TO,R1                    ;PUT TIME IN CONTROLLER TABLE
     56 014142  004737  017352                  CALL SETTO
     57 014146  000137  013460                  JMP RSPNXT                      ;NOW WAIT FOR END PACKET
     58 014152  122764  000201  000030  RSPERR: CMPB #OP.END+OP.GDS,HC.MPK+P.OPCD(R4)   ;SEE IF GET DUST STATUS OPCODE
     59 014160  001237                          BNE RSPERW
     60 014162  132764  000010  000037          BITB #DF.ACT,HC.MPK+P.DFLG(R4)  ;IF DUST NO LONGER RUNNING
     61 014170  001603                          BEQ RSPTOE                      ;   REPORT ERROR
     62 014172  042765  000050  000012          BIC #CT.STA+CT.MSG,C.FLG(R5)    ;CLEAR CONTROL BITS
     63 014200  032765  000200  000012          BIT #CT.TM2,C.FLG(R5)           ;IF AT SECOND TIMEOUT
     64 014206  001413                          BEQ 1$
     65 014210  026465  000040  000044          CMP HC.MPK+P.DPI(R4),C.PRI(R5)  ;COMPARE PROGRESS INDICATOR
     66 014216  001004                          BNE 2$
     67 014220  026465  000042  000046          CMP HC.MPK+P.DPI+2(R4),C.PRI+2(R5)      ;COMPARE PROGRESS INDICATOR
     68 014226  001422                          BEQ 4$                          ;REPORT ERROR IF NOT CHANGED
     69 014230  042765  000200  000012  2$:     BIC #CT.TM2,C.FLG(R5)           ;CLEAR TIMEOUT 2 FLAG
     70 014236  032765  000100  000012  1$:     BIT #CT.TM1,C.FLG(R5)           ;IF AT FIRST TIMEOUT
     71 014244  001406                          BEQ 3$
     72 014246  016465  000040  000044          MOV HC.MPK+P.DPI(R4),C.PRI(R5)  ;GET COPY OF PROGRESS INDICATOR
     73 014254  016465  000042  000046          MOV HC.MPK+P.DPI+2(R4),C.PRI+2(R5)      ;GET COPY OF PROGRESS INDICATOR
     74 014262  012764  140000  000006  3$:     MOV #RG.OWN+RG.FLG,HC.MCT(R4)   ;GIVE MESSAGE BUFFER BACK TO UDA
     75 014270  000137  013460                  JMP RSPNXT
     76 014274  000137  013600          4$:     JMP RSPTOE
```

```
                                ;RESPONSE REQUEST DISPATCH TABLE

014300  014314                  RSPDSP: .WORD QUEST                     ;QUESTION
014302  014366                          .WORD DQUEST                    ;QUESTION WITH DEFAULT ANSWER
014304  014540                          .WORD INFO                      ;INFORMATION MESSAGE FOR OPERATOR
014306  014666                          .WORD TERM                      ;NORMAL TERMINATION
014310  014676                          .WORD ERRTRM                    ;FATAL ERROR TERMINATION
014312  014706                          .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
                                        
014314  004737  015040          QUEST:  CALL GTDRVT      ;GET POINTER TO DRIVE TABLE
014320  062700  000004                  ADD #D.SERN,R0   ;BUMP POINTER TO SERIAL NUMBER
014324  014403                          MOV -(R4),R3     ;GET QUESTION NUMBER
014326  001411                          BEQ QUE0         ;BRANCH IF QUESTION NUMBER 0
014330  020327  000007                  CMP R3,#7        ;IF NOT, SEE IF QUESTION NUMBER 7
014334  001410                          BEQ QUE7
014336                                  ERRDF 100,,ERR100       ;ANY OTHER NUMBER IS AN ERROR
                                                                        TRAP    C$ERDF
014336  104455                                                          .WORD   100
014340  000144                                                          .WORD   0
014342  000000                                                          .WORD   ERR100
014344  012454
014346  000244                          CLZ      ;CLEAR Z TO STOP DUP PROGRAM
014350  000207                          RETURN
                                        
014352  012700  003304          QUE0:   MOV #DATE0,R0    ;POINT TO DATE STRING
014356                           QUE7:
014356  005201                  QUEL:   INC R1   ;COUNT THE CHARACTERS
014360  112022                          MOVB (R0)+,(R2)+         ; AND PUT THEM IN OUTPUT BUFFER
014362  001375                          BNE QUEL         ;  UNTIL A NUL CHARACTER FOUND
014364  000207                          RETURN  ;RETURN WITH Z SET
```

```
                                        ;MESSAGE TYPE 2
                                        ;
                                        ;ANSWER QUESTION 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

014366  004737  015040        DQUEST: CALL GTDRVT     ;GET DRIVE TABLE ADDRESS INTO R0
014372  014403                        MOV -(R4),R3    ;GET QUESTION NUMBER
014374  020327  000006                CMP R3,#DQUESZ
014400  101035                        BHI DQUEX
014402  006303                        ASL R3
014404  000173  014410                JMP @DQUEJP(R3)
014410  014474                DQUEJP: .WORD DQUEX     ; 0  (NOT USED)
014412  014426                        .WORD DQUNIT    ; 1  ENTER UNIT NUMBER TO FORMAT
014414  014474                        .WORD DQUEX     ; 2  (NOT USED)
014416  014474                        .WORD DQUEX     ; 3  (NOT USED)
014420  014500                        .WORD DQRFMT    ; 4  USE EXISTING BAD SECTOR INFORMATION
014422  014520                        .WORD DQRSTR    ; 5  DOWN-LINE LOAD BAD SECTOR BLOCK INFORMATION
014424  014530                        .WORD DQCONT    ; 6  CONTINUE IF BAD BLOCK INFO INACCESSIBLE
        000006                              DQUESZ=<<.-DQUEJP>/2>-1

                                        ;ENTER UNIT NUMBER TO FORMAT

014426                          DQUNIT: PUSH R5                                 MOV R5,-(SP)
014426  010546
                                        CLR R4
014430  005004                          MOV (R0),R3     ;GET DRIVE NUMBER
014432  011003                                ASSUME D.DRV EQ 0
014436                                  MOV #10.,R0     ;RADIX 10.
014434  012700  000012
014440  004737  016516          DQUNL1: CALL DIVIDE
014444                                  PUSH R5                                 MOV R5,-(SP)
014444  010546
                                        INC R1
014446  005201                          TST R3
014450  005703                          BNE DQUNL1
014452  001372                          MOV R1,R0
014454  010100
014456                          DQUNL2: POP R5                                  MOV (SP)+,R5
014456  012605
                                        ADD #'0,R5
014460  062705  000060                  MOVB R5,(R2)+
014464  110522                          DEC R0
014466  005300                          BNE DQUNL2
014470  001372                          POP R5
014472                                                                          MOV (SP)+,R5
014472  012605
014474  000264                  DQUEX:  SEZ
014476  000207                          RETURN

014500  032737  000003  003200  DQRFMT: BIT #SO.FMT,MODE
```

```
     54 014506  001410                              BEQ DQNO
     55 014510  112712  000131          DQYES:      MOVB #'Y,(R2)
     56 014514  005201                              INC R1
     57 014516  000766                              BR DQUEX
     58
     59 014520  032737  000010  003200  DQRSTR:     BIT #S0.STR,MODE
     60 014526  001370                              BNE DQYES
     61 014530                          DQCONT:
     62 014530  112712  000116          DQNO:       MOVB #'N,(R2)
     63 014534  005201                              INC R1
     64 014536  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
                                        
014540  016400  177776          INFO:   MOV -2(R4),R0   ;GET MESSAGE NUMBER
014544  001434                          BEQ INFOB       ;IF ZERO, PRINT BEGUN MESSAGE
014546  020027  000100                  CMP R0,#100     ;IF OCTAL 100
014552  001423                          BEQ INFOE       ;  PRINT ERROR MESSAGE
014554  020027  000200                  CMP R0,#200     ;SEE IF 200 OR GREATER
014560  002005                          BGE INFOH       ;  IF SO, PRINT WITHOUT FREEZING
014562  005737  002170                  TST UFREEZ
014566  001007                          BNE INFOP
014570  005237  002170                  INC UFREEZ
014574  004737  015040          INFOH:  CALL GTDRVT
014600  010002                          MOV R0,R2
014602  004737  015064                  CALL HEADER
014606  004737  015004          INFOP:  CALL MESG       ;PRINT THE MESSAGE
014612  012701  100000          INFOX:  MOV #BIT15,R1                   ;RETURN A NEGATIVE BYTE COUNT
014616  000264                          SEZ
014620  000207                          RETURN                          ;RETURN WITH Z SET
                                        
014622                          INFOE:  ERRDF 101.,ERP101       ;ANSWER WAS REJECTED BY DUP PROGRAM
014622  104455                                                                  TRAP    C$ERDF
014624  000145                                                                  .WORD   101
014626  000000                                                                  .WORD   0
014630  012470                                                                  .WORD   ERR101
014632  000244                          CLZ     ;RETURN WITH Z CLEAR TO STOP DUP PROGRAM
014634  000207                          RETURN
                                        
014636  004737  015040          INFOB:  CALL GTDRVT     ;PRINT FORMAT BEGUN MESSAGE
014642  010002                          MOV R0,R2
014644  004737  015064                  CALL HEADER
014650  004737  015004                  CALL MESG
014654                                  PNT WNSTOP      ;PRINT WARNING NOT TO STOP NOW
014654  004137  016450                                                      JSR R1,LPNT
014660  004355                                                              .WORD WNSTOP
014662  000000                                                              .WORD PNT.CT
014664  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

014666  004737  014540          TERM:   CALL INFO       ;PRINT THE MESSAGE
014672  000244                          CLZ
014674  000207                          RETURN                          ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM
```

GLOBAL SUBROUTINES SECTION

```
                                ;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
                                
014676  004737  014540        ERRTRM: CALL INFO
014702  000244                        CLZ
014704  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

014706  023714  003176          SPECL:  CMP FCTNUM,(R4) ;SEE IF DESIRED BLOCK IS IN MEMORY
014712  001425                          BEQ SPECLX      ; IF SO, SEND TO DUP PROGRAM
014714  002407                          BLT SPECLR      ; IF LOWER NUMBERED BLOCK IN MEMORY,
                                        ;                 GO READ NEXT BLOCK
014716                          SPECLC:
014716                                  CLOSE   ;OTHERWISE, START READING FROM BEGINNING AGAIN
014716  104435                                                          TRAP    C$CLOS
014720                                  OPEN #FNAME
014720  012700  003230                                                  MOV     #FNAME,R0
014724  104434                                                          TRAP    C$OPEN
014726  012737  177777  003176                MOV #-1,FCTNUM
014734  012703  001000          SPECLR: MOV #512.,R3    ;GET BYTE COUNT IN A BLOCK
014740  012701  002176                  MOV #FCTBUF,R1  ;POINT TO STORAGE AREA
014744                          SPECLL: GETBYTE (R1)+   ;READ THE FILE
014744  104426                                                          TRAP    C$GETB
014746  110021                                                          MOVB    R0,(R1)+
014750                                  BNCOMPLETE SPECLE       ;PRINT ERROR IF NO MORE BYTES IN FILE
014750  103005                                                          BCC     SPECLE
014752  005303                          DEC R3  ;COUNT THE BYTES
014754  001373                          BNE SPECLL
014756  005237  003176                 INC FCTNUM      ;KEEP COUNT OF BLOCK IN MEMORY
014762  000751                          BR SPECL

014764  005212                  SPECLE: INC (R2)        ;TELL DUP PROGRAM DATA NOT AVAILABLE
014766  012762  002176  000002  SPECLX: MOV #FCTBUF,2(R2)       ;PUT ADDRESS OF DATA IN OUTPUT BUFFER
014774  012701  000006                  MOV #6,R1       ;SEND 3 WORDS TO DUP PROGRAM
015000  000264                          SEZ
015002  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

 015004                                 MESG:
 015004  112400                         1$:     MOVB (R4)+,R0                   ;PRINT CHARACTERS FROM DUP PROGRAM
 015006  001405                                 BEQ 2$                          ;  DISCARDING LF AND NULL CHARACTERS
 015010  020027  000012                         CMP R0,#12
 015014  001402                                 BEQ 2$
 015016                                         PRINT R0
 015016  004737  016240                                                         CALL CPNT
 015022  005303                         2$:     DEC R3                          ;COUNT THE CHARACTERS
 015024  003367                                 BGT 1$
 015026                                         PRINT #CR
 015026  112700  000015                                                         MOVB #CR,R0
 015032  004737  016240                                                         CALL CPNT
 015036  000207                                 RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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)
                                        ;
015040                                  GTDRVT: PUSH R5                                         MOV R5,-(SP)
015040  010546
015042  062705  000016                          ADD #C.DR0,R5
015046  012500                          GTDRVL: MOV (R5)+,R0
015050  016037  000002  002074                  MOV D.UNIT(R0),L$LUN
015056                                                  ASSUME DT.AVL EQ BIT15
015056  100773                                  BMI GTDRVL
015060                                          POP R5                                          MOV (SP)+,R5
015060  012605
015062  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
                                        ;
015064  022737  000001  002012  HEADER: CMP #1,L$UNIT                    ;IF MORE THAN ONE UNIT BEING TESTED
015072  001411                          BEQ 1$
015074                                  PNTF MESSG,D.UNIT(R2),(R5),(R2)         ;PRINT UDA ADDRESS
015074  011246                                                                  MOV (R2),-(SP)
015076  011546                                                                  MOV (R5),-(SP)
015100  016246  000002                                                          MOV D.UNIT(R2),-(SP)
015104  004137  016412                                                          JSR R1,LPNTF
015110  004035                                                                  .WORD MESSG
015112  000006                                                                  .WORD PNT.CT
015114                                          ASSUME C.UADR EQ 0
015114                                          ASSUME D.DRV EQ 0
015114  000407                          BR 2$
015116  005737  003206          1$:     TST KW.CSR                      ;IF NO CLOCK BEING USED
015122  001406                          BEQ 3$                          ;BYPASS RUNTIME MESSAGE
015124                                  PRINT #CR
015124  112700  000015                                                          MOVB #CR,R0
015130  004737  016240                                                          CALL CPNT
015134  004737  020402          2$:     CALL RNTIME                     ;PRINT RUNTIME IF A CLOCK IN USE
015140                          3$:     PRINT #CR
015140  112700  000015                                                          MOVB #CR,R0
015144  004737  016240                                                          CALL CPNT
015150  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
                                        ;
015152  112201          OSTRNG: MOVB (R2)+,R1           ;GET CONTROL CHARACTER
015154  001421                  BEQ OSTRE               ;EXIT IF NULL CHARACTER
015156  012700  015452          MOV #ERRC,R0            ;GET POINTER TO CHARACTER TABLE
015162  120110          NCONS:  CMPB R1,(R0)            ;COMPARE CHARACTER WITH TABLE ENTRY
015164  001407                  BEQ NCONF               ;BRANCH IF MATCH FOUND
015166  105720                  TSTB (R0)+              ;INCREMENT POINTER
015170  001374                  BNE NCONS               ;CONTINUE SEARCH IF NOT END OF TABLE
015172                          PNTF ERRME1             ;REPORT BAD CONTROL CHARACTER
015172  004137  016412                                                  JSR R1,LPNTF
015176  003746                                                          .WORD ERRME1
015200  000000                                                          .WORD PNT.CT
015202  000406                  BR OSTRE
015204  162700  015452  NCONF:  SUB #ERRC,R0            ;GET INCREMENT INTO TABLE
015210  006300                  ASL R0                  ;DOUBLE TO WORD COUNT
015212  004770  015464          CALL @ERRD(R0)          ;DISPATCH TO PRINT ROUTINE
015216  000755                  BR OSTRNG               ;GET NEXT
015220  000207          OSTRE:  RETURN
```

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

                                        CON.QU: MOVB (R2)+,R0                   ;GET CHARACTER
 015222  112200
 015224  120027  000042                        CMPB R0,#'"                    ;CHECK IF ENDING QUOTE
 015230  001403                                BEQ CON.QX                      ;IF SO, GO GET NEXT CONTROL CHARACTER
 015232                                         PRINT R0                       ;PRINT THE CHARACTER
 015232  004737  016240                                                         CALL CPNT
 015236  000771                                BR CON.QU                       ;CONTINUE PRINTING
 015240  000207                        CON.QX: RETURN

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

 015242  004737  015720                CON.A:  CALL GETCNT                     ;GET COUNT OF CHARACTERS
 015246                                CON.A1: PRINT (R4)+                     ;PRINT THE CHARACTER
 015246  112400                                                                 MOVB (R4)+,R0
 015250  004737  016240                                                         CALL CPNT
 015254  005301                                DEC R1                          ;COUNT THE CHARACTERS
 015256  001373                                BNE CON.A1                      ;PRINT UNTIL COUNT REACHES ZERO
 015260  032704  000001                        BIT #1,R4                       ;CHECK IF R4 NOW ODD
 015264  001401                                BEQ CON.A2
 015266  005204                                INC R4                          ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
 015270  000207                        CON.A2: RETURN                          ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.

 015272  012701  000012                CON.D:  MOV #10.,R1                     ;LOAD RADIX
 015276  004737  015776                        CALL PNTNUM                     ;PRINT NUMBER
 015302  000207                                RETURN                          ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.

 015304  012701  000020                CON.H:  MOV #16.,R1                     ;LOAD RADIX
 015310  004737  015776                        CALL PNTNUM                     ;PRINT NUMBER
 015314  000207                                RETURN                          ;NOW GET NEXT CONTROL CHARACTER
```

GLOBAL SUBROUTINES SECTION

```
                                        ;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.
 1
 2
 3 015316  012701  000010      CON.O:  MOV #8.,R1                      ;LOAD RADIX
 4 015322  004737  015776              CALL PNTNUM                     ;PRINT NUMBER
 5 015326  000207                      RETURN                          ;NOW GET NEXT CONTROL CHARACTER
 6
 7                                      ;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.
 8
 9 015330  004737  015720      CON.N:  CALL GETCNT                     ;GET COUNT
10 015334                      CON.N1: PRINT #CR                       ;PRINT NEW LINE SEQUENCE
   015334  112700  000015                                                      MOVB #CR,R0
   015340  004737  016240                                                      CALL CPNT
11 015344  005301                      DEC R1                          ;COUNT THE SEQUENCES
12 015346  001372                      BNE CON.N1
13 015350  000207                      RETURN                          ;NOW GET NEXT CONTROL CHARACTER
14
15                                      ;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.
16
17 015352  004737  015720      CON.R:  CALL GETCNT                     ;GET ROUTINE NUMBER
18 015356  020127  000010              CMP R1,#ERRRSZ                  ;CHECK IF DEFINED ROUTINE NUMBER
19 015362  101004                      BHI CON.R1
20 015364  060101                      ADD R1,R1                       ;DOUBLE COUNT TO GET WORD INDEX
21 015366  004771  015430              CALL @ERRRTB-2(R1)              ;CALL ROUTINE
22 015372  000207                      RETURN                          ;NOW GET NEXT CONTROL CHARACTER
23 015374                      CON.R1: PNTF ERRME1                     ;REPORT BAD MESSAGE STRING
   015374  004137  016412                                                      JSR R1,LPNTF
   015400  003746                                                             .WORD ERRME1
   015402  000000                                                             .WORD PNT.CT
24 015404                              POP R1                          ;FIX THE STACK
   015404  012601                                                              MOV (SP)+,R1
25 015406  000207                      RETURN
26
27                                      ;CONTROL CHARACTER WAS AN S. PRINT SPACES.
28
29 015410  004737  015720      CON.S:  CALL GETCNT                     ;GET COUNT
30 015414                      CON.S1: PRINT <#' '>                    ;PRINT A SPACE
   015414  112700  000040                                                      MOVB #' ,R0
   015420  004737  016240                                                      CALL CPNT
31 015424  005301                      DEC R1                          ;COUNT THE SPACES
32 015426  001372                      BNE CON.S1
33 015430  000207                      RETURN                          ;NOW GET NEXT CONTROL CHARACTER
```

GLOBAL SUBROUTINES SECTION

```
                                        ;ERROR ROUTINE DISPATCH TABLE

015432  015504                ERRRTB: .WORD CALRF                     ;NOT USED
015434  015504                        .WORD CALRE                     ;NOT USED
015436  015504                        .WORD CALRE                     ;NOT USED
015440  015516                        .WORD CALR4                     ;PRINT BASIC LINE WITHOUT UDA ADDRESS
015442  015572                        .WORD CALR5                     ;PRINT BASIC LINE WITH UDA ADDRESS
015444  015650                        .WORD CALR6                     ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
015446  015664                        .WORD CALR7                     ;PRINT  "REPLACE PROCESSOR MODULE"
015450  015702                        .WORD CALR8                     ;PRINT  "  UDASA CONTAINS  XXXXXX"
        000010                      ERRRSZ=<.-ERRRTB>/2

                                    ;BUILD TWO TABLES
                                    ;       FIRST CONTAINING CONTROL CHARACTERS
                                    ;       SECOND CONTAINING ROUTINE ADDRESSES

                                    .MACRO BUILD
                                            ENTRY ".CON.QU
                                            ENTRY A.CON.A
                                            ENTRY D.CON.D
                                            ENTRY H.CON.H
                                            ENTRY O.CON.O
                                            ENTRY N.CON.N
                                            ENTRY R.CON.R
                                            ENTRY S.CON.S
                                    .ENDM
```

```
                                        ;HERE IS FIRST TABLE

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

 015452                                 ERRC:   BUILD
 015452     042                                 .BYTE '"
 015453     101                                 .BYTE 'A
 015454     104                                 .BYTE 'D
 015455     110                                 .BYTE 'H
 015456     117                                 .BYTE 'O
 015457     116                                 .BYTE 'N
 015460     122                                 .BYTE 'R
 015461     123                                 .BYTE 'S
 015462     000                                 .BYTE 0                    ;FOLLOW WITH A NULL BYTE
                                                .EVEN

                                        ;HERE IS SECOND TABLE

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

 015464                                 ERRD:   BUILD
 015464  015222                                 .WORD CON.QU
 015466  015242                                 .WORD CON.A
 015470  015272                                 .WORD CON.D
 015472  015304                                 .WORD CON.H
 015474  015316                                 .WORD CON.O
 015476  015330                                 .WORD CON.N
 015500  015352                                 .WORD CON.R
 015502  015410                                 .WORD CON.S
```

GLOBAL SUBROUTINES SECTION

```
      1                                 ;PRE-PROGRAMMED ROUTINES 1, 2 AND 3
      2                                 ;NOT USED - PRINTS ERROR MESSAGE
      3
      4 015504                          CALRE:  PNTF ERRME1                     ;PRINT ERROR MESSAGE
        015504  004137  016412                                                            JSR R1,LPNTF
        015510  003746                                                                    .WORD ERRME1
        015512  000000                                                                    .WORD PNT.CT
      5 015514  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;PRE-PROGRAMMED ROUTINE 4
                                        ;PRINT BASIC LINE FOR HOST PROGRAM ERROR WITHOUT UDA ADDRESS
                                        ;THEN SWITCH TO EXTENDED FORMAT

015516                                  CALR4:  PNTB BASLN,@BASNO,@BAS,@BAS,@BAS
                                                                                    MOV @BAS,-(SP)
015516  012746  004247                                                              MOV @BAS,-(SP)
015522  012746  004247                                                              MOV @BAS,-(SP)
015526  012746  004247                                                              MOV @BASNO,-(SP)
015532  012746  004165                                                              JSR R1,LPNTB
015536  004137  016422                                                              .WORD BASLN
015542  004250                                                                      .WORD PNT.CT
015544  000010
015546  004737  020402                  CALL RNTIME
015552                                  PRINT @CR
                                                                                    MOVB @CR,R0
015552  112700  000015                                                              CALL CPNT
015556  004737  016240
015562  012737  016340  003222          MOV @PX,PTYPE
015570  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

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

                                CALR5:  PNTB BASLN,@BASNO,@BASL2,(R5),@BAS,@BAS
015572                                                                          MOV @BAS,-(SP)
015572  012746  004247                                                        MOV @BAS,-(SP)
015576  012746  004247                                                        MOV (R5),-(SP)
015602  011546                                                                MOV @BASL2,-(SP)
015604  012746  004204                                                        MOV @BASNO,-(SP)
015610  012746  004165                                                        JSR R1,LPNTB
015614  004137  016422                                                        .WORD BASLN
015620  004250                                                                .WORD PNT.CT
015622  000012
015624  004737  020402                  CALL RNTIME
015630                                  PRINT @CR
015630  112700  000015                                                        MOVB @CR,R0
015634  004737  016240                                                        CALL CPNT
015640  012737  016340  003222          MOV @PX,PTYPE
015646  000207                          RETURN
```

```
                                ;PRE-PROGRAMMED ROUTINE 6
                                ;CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
                                
                                CALR6:  PUSH R2                         ;SAVE CURRENT STRING POINTER
                                                                                MOV R2,-(SP)
                                        MOV (R4)+,R2                    ;GET NEW STRING POINTER
                                        CALL OSTRNG                     ;OUTPUT USING THIS STRING
                                        POP R2                          ;GET OLD POINTER BACK
                                                                                MOV (SP)+,R2
                                        RETURN                          ;NOW CONTINUE THE OLD STRING
```

GLOBAL SUBROUTINES SECTION

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

                                CALR7:  PUSH R2
015664                                                                  MOV R2,-(SP)
015664  010246
                                        MOV #XFRU,R2
015666  012702  011532
                                        CALL OSTRNG
015672  004737  015152
                                        POP R2
015676
                                                                        MOV (SP)+,R2
015676  012602
                                        RETURN
015700  000207
```

GLOBAL SUBROUTINES SECTION

```
                                ;PRE-PROGRAMMED ROUTINE 8
                                ;PRINT   "  UDASA CONTAINS  XXXXXX"
                                
                                CALR8:  PUSH R2                                 MOV R2,-(SP)
015702  010246
015704  012702  011504                  MOV #XSA,R2
015710  004737  015152                  CALL OSTRNG
015714                                  POP R2                                  MOV (SP)+,R2
015714  012602
015716  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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

015720                                  GETCNT: PUSH R0                                         MOV R0,-(SP)
015720  010046
                                                CLR R1                          ;START WITH ZERO COUNT
015722  005001
015724  121227  000060                  GETCNX: CMPB (R2),#'0                   ;CHECK IF CHARACTER A DIGIT
015730  103415                                  BLO GETCDN                      ;BRANCH IF LOWER THAN ZERO
015732  121227  000071                          CMPB (R2),#'9
015736  101012                                  BHI GETCDN                      ;BRANCH IF HIGHER THAN NINE
015740  006301                                  ASL R1                          ;MULTIPLY NUMBER BY 10
015742  010100                                  MOV R1,R0                       ; SAVE 2N
015744  006301                                  ASL R1                          ; COMPUTE 4N
015746  006301                                  ASL R1                          ; COMPUTE 8N
015750  060001                                  ADD R0,R1                       ; 8N + 2N = 10N
015752  112200                                  MOVB (R2)+,R0                   ;GET DIGIT FROM STING
015754  162700  000060                          SUB #'0,R0                      ;GET RID OF ASCII
015760  060001                                  ADD R0,R1                       ;ADD TO NUMBER
015762  000760                                  BR GETCNX                       ;GO TO NEXT CHARACTER
015764  005701                          GETCDN: TST R1                          ;CHECK IF NUMBER IS ZERO
015766  001001                                  BNE GETCXX                      ;IF ZERO, CHANGE
015770  005201                                  INC R1                          ; TO DEFAULT OF ONE
015772                                  GETCXX: POP R0                                          MOV (SP)+,R0
015772  012600
015774  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

016776  010100                  PNTNUM: MOV R1,R0                       ;SAVE RADIX
016000  004737  015720                  CALL GETCNT                     ;GET COUNT OF BITS
016004                          PNTNUS: PUSH <R2,R3,R5>
016004  010246                                                                  MOV R2,-(SP)
016006  010346                                                                  MOV R3,-(SP)
016010  010546                                                                  MOV R5,-(SP)
016012  012403                          MOV (R4)+,R3                    ;GET ONE PARAMETER WORD
016014  005005                          CLR R5                          ;CLEAR STORAGE FOR OTHER
016016  020127  000020                  CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
016022  003401                          BLE 1$
016024  012405                          MOV (R4)+,R5                    ;YES, GET SECOND PARAMETER WORD
016026                          1$:     PUSH R4
016026  010446                                                                  MOV R4,-(SP)
016030  010504                          MOV R5,R4                       ;PUT HIGH WORD IN R4
016032  012702  000020                  MOV #16.,R2                     ;COMPUTE BITS NOT WANTED
016036  160102                          SUB R1,R2                       ;BY SUBTRACTING BITS TO USE
016040  002002                          BGE 2$                          ;FROM 16.
016042  062702  000020                  ADD #16.,R2                     ;IF NEGATIVE, ADD 16 FOR FIRST WORD
016046  001414                  2$:     BEQ 6$                          ;IF ZERO, NO BITS NEED BE CLEARED
016050  012705  100000                  MOV #BIT15,R5                   ;START MASK WITH SIGN BIT SET
016054  005302                  3$:     DEC R2                          ;COUNT BITS IN MASK
016056  001402                          BEQ 4$
016060  006205                          ASR R5                          ;SHIFT MORE BITS TO RIGHT
016062  000774                          BR 3$
016064  020127  000020          4$:     CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
016070  003402                          BLE 5$
016072  040504                          BIC R5,R4                       ;YES, CLEAR IN HIGH WORD
016074  000401                          BR 6$
016076  040503                  5$:     BIC R5,R3                       ;NO, CLEAR IN LOW WORD
016100  004737  016516          6$:     CALL DIVIDE                     ;DIVIDE BY RADIX IN R0
016104                                  PUSH R5                         ;PUSH REMAINDER ON STACK
016104  010546                                                                  MOV R5,-(SP)
016106  005202                          INC R2                          ;COUNT DIGITS ON STACK
016110  005703                          TST R3                          ;CHECK IF QUOTIENT IS ZERO
016112  001372                          BNE 6$
016114  005704                          TST R4
016116  001370                          BNE 6$
```

```
016120  020027  000012              CMP R0,#10.             ;IF RADIX IS DECIMAL
016124  001423                      BEQ 10$                 ; JUST GO PRINT DIGITS ON STACK
016126  010103                      MOV R1,R3               ;OTHERWISE COMPUTE NUMBER OF LEADING ZEROS
016130  162700  000014              SUB #12.,R0             ;DIVIDEND IS BITS IN NUMBER
016134  003002                      BGT 7$                  ;DIVISOR IS BITS PER DIGIT PRINTED
016136  012700  000003              MOV #3,R0               ;    (3 OR 4)
016142  004737  016516      7$:     CALL DIVIDE
016146  005705                      TST R5                  ;IF REMAINDER NOT ZERO
016150  001401                      BEQ 8$                  ;INCREMENT QUOTIENT
016152  005203                      INC R3
016154  160203              8$:     SUB R2,R3               ;SUBTRACT DIGITS ON STACK
016156  001406                      BEQ 10$                 ;NO LEADING ZEROS IF ZERO
016160                      9$:     PRINT #'0               ;PRINT A ZERO
016160  112700  000060                                                      MOVB #'0,R0
016164  004737  016240                                                      CALL CPNT
016170  005303                      DEC R3
016172  001372                      BNE 9$                  ;REPEAT UNTIL COUNT REACHES ZERO

016174                      10$:    POP R5                  ;GET CHACACTER FROM STACK
016174  012605                                                              MOV (SP)+,R5
016176  062705  000060              ADD #'0,R5              ;CNVERT TO ASCII DIGIT
016202  020527  000071              CMP R5,#'9              ;IF GREATER THAN A 9
016206  003402                      BLE 11$                 ; CONVERT TO A OR HIGHER
016210  062705  000007              ADD #<'A-'9-1>,R5       ; FOR HEX DIGIT
016214                      11$:    PRINT R5                ;PRINT THE CHARACTER
016214  110500                                                              MOVB R5,R0
016216  004737  016240                                                      CALL CPNT
016222  005302                      DEC R2                  ;REPEAT FOR ALL DIGITS
016224  001363                      BNE 10$                 ; ON STACK
016226                              POP <R4,R5,R3,R2>
016226  012604                                                              MOV (SP)+,R4
016230  012605                                                              MOV (SP)+,R5
016232  012603                                                              MOV (SP)+,R3
016234  012602                                                              MOV (SP)+,R2
016236  000207                      RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;PRINT ONE CHARACTER
                                        ;
                                        ;CALL WITH MACRO PRINT
                                        ;
016240  110037  003224          CPNT:   MOVB R0,ERRCHR
                                        PUSH R1
016244  010146                                                  MOV R1,-(SP)
016246  012701  003704                  MOV #ERRONE,R1
016252  120027  000015                  CMPB R0,#CR
016256  001002                          BNE 1$
016260  012701  003707                  MOV #ERRNL,R1
016264  000177  164732          1$:     JMP @#TYPE
016270                          PF:     PRINTF R1,#ERRCHR
016270  012746  003224                                          MOV     #ERRCHR,-(SP)
016274  010146                                                  MOV     R1,-(SP)
016276  012746  000002                                          MOV     #2,-(SP)
016302  010600                                                  MOV     SP,R0
016304  104417                                                  TRAP    C$PNTF
016306  062706  000006                                          ADD     #6,SP
016312  000435                          BR CPNTX
016314                          PB:     PRINTB R1,#ERRCHR
016314  012746  003224                                          MOV     #ERRCHR,-(SP)
016320  010146                                                  MOV     R1,-(SP)
016322  012746  000002                                          MOV     #2,-(SP)
016326  010600                                                  MOV     SP,R0
016330  104414                                                  TRAP    C$PNTB
016332  062706  000006                                          ADD     #6,SP
016336  000423                          BR CPNTX
016340                          PX:     PRINTX R1,#ERRCHR
016340  012746  003224                                          MOV     #ERRCHR,-(SP)
016344  010146                                                  MOV     R1,-(SP)
016346  012746  000002                                          MOV     #2,-(SP)
016352  010600                                                  MOV     SP,R0
016354  104415                                                  TRAP    C$PNTX
016356  062706  000006                                          ADD     #6,SP
016362  000411                          BR CPNTX
016364                          PS:     PRINTS R1,#ERRCHR
016364  012746  003224                                          MOV     #ERRCHR,-(SP)
016370  010146                                                  MOV     R1,-(SP)
016372  012746  000002                                          MOV     #2,-(SP)
016376  010600                                                  MOV     SP,R0
016400  104416                                                  TRAP    C$PNTS
016402  062706  000006                                          ADD     #6,SP
016406                          CPNTX:  POP R1
016406  012601                                                  MOV (SP)+,R1
016410  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

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

016412  012737  016270  003222  LPNTF:  MOV #PF,PTYPE
016420  000413                        BR LPNT
016422  012737  016314  003222  LPNTB:  MOV #PB,PTYPE
016430  000407                        BR LPNT
016432  012737  016340  003222  LPNTX:  MOV #PX,PTYPE
016440  000403                        BR LPNT
016442  012737  016364  003222  LPNTS:  MOV #PS,PTYPE
016450                          LPNT:   PUSH <R2,R3,R4,R5>
016450  010246                                                          MOV R2,-(SP)
016452  010346                                                          MOV R3,-(SP)
016454  010446                                                          MOV R4,-(SP)
016456  010546                                                          MOV R5,-(SP)
016460  012102                        MOV (R1)+,R2              ;GET ADDRESS OF STRING
016462  010604                        MOV SP,R4                 ;COMPUTE ADDRESS OF ARGUMENTS
016464  062704  000012                ADD #10.,R4               ;  WHICH ARE NOW ON STACK (IF ANY)
016470                                PUSH R1                   ;SAVE RETURN ADDRESS
016470  010146                                                          MOV R1,-(SP)
016472  004737  015152                CALL OSTRNG               ;PRINT THE FORMATTED MESSAGE
016476                                POP <R0,R5,R4,R3,R2,R1>   ;RESTORE ALL REGISTERS
016476  012600                                                          MOV (SP)+,R0
016500  012605                                                          MOV (SP)+,R5
016502  012604                                                          MOV (SP)+,R4
016504  012603                                                          MOV (SP)+,R3
016506  012602                                                          MOV (SP)+,R2
016510  012601                                                          MOV (SP)+,R1
016512  062006                        ADD (R0)+,SP              ;ADJUST STACK POINTER OVER ARGUMENTS
016514  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
                                        ;
016516                                  DIVIDE: PUSH R2                                 MOV R2,-(SP)
016516  010246
016520  012702  000040                          MOV #32.,R2             ;SET UP SHIFT COUNT
016524  005005                                  CLR R5                  ;START WITH ZERO REMAINDER
016526  006303                        1$:       ASL R3                  ;SHIFT LEFT INTO R5
016530  006104                                  ROL R4
016532  006105                                  ROL R5
016534  020005                                  CMP R0,R5               ;WILL DIVISOR GO INTO REMAINDER
016536  101002                                  BMI 2$                  ;ONLY SUBTRACT IF IT WILL
016540  160005                                  SUB R0,R5               ;SUBTRACT DIVISOR
016542  005203                                  INC R3                  ;PUT A ONE INTO QUOTIENT
016544  005302                        2$:       DEC R2                  ;COUNT THE SHIFTS
016546  001367                                  BNE 1$
016550                                          POP R2                                  MOV (SP)+,R2
016550  012602
016552  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
                                        ;
016554  013701  002156          LOADDM: MOV DMPROG,R1                   ;GET STORAGE ADDRESS OF DM PROGRAM
016560  116165  000021  000042          MOVB DMTMO(R1),C.TOT(R5)        ;GET TIMEOUT VALUE
016566  105065  000043                  CLRB C.TOT+1(R5)
016572  016504  000004                  MOV C.VEC(R5),R4                ;GET VECTOR OF UDA
016576                                  AND CT.VEC,R4
016576  042704  177000                                                          BIC #^C<CT.VEC>,R4
016602  010501                          MOV R5,R1                       ;GET INTERRUPT SERVICE LINK
016604  062701  000006                  ADD #C.JSR,R1
016610                                  SETVEC R4,R1,#PRIO7             ;SET UP INTERRUPT VECTOR
016610  012746  000340                                                          MOV     #PRIO7,-(SP)
016614  010146                                                                  MOV     R1,-(SP)
016616  010446                                                                  MOV     R4,-(SP)
016620  012746  000003                                                          MOV     #3,-(SP)
016624  104437                                                                  TRAP    C$SVEC
016626  062706  000010                                                          ADD     #10,SP
                                                                        ;INITIALIZE UDA WITH SMALLEST
016632  004737  017434                  CALL UDAINT                     ; RING BUFFER AND INTERRUPTS ENABLED
016636  001444                          BEQ LOADER                      ;BRANCH IF AN ERROR
```

GLOBAL SUBROUTINES SECTION

```
      1 016640  012700  000002                MOV #OP.ESP,R0                  ;BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET
      2 016644  004737  016754                CALL BLDCMD
      3 016650  013764  002156  000124        MOV DMPROG,MC.CPK+P.UADR(R4)    ;LOAD MAIN PROGRAM ADDRESS
      4 016656  017764  163274  000120        MOV @DMPROG,MC.CPK+P.BCNT(R4)   ;  AND SIZE
      5 016664  013764  002156  000140        MOV DMPROG,MC.CPK+P.OVRL(R4)    ;LOAD OVERLAY ADDRESS
      6 016672  067764  163260  000140        ADD @DMPROG,MC.CPK+P.OVRL(R4)
      7 016700  004737  017040                CALL SNDCMD                     ;SEND COMMAND TO UDA
      8 016704  004737  017160                CALL WAITMS                     ;WAIT FOR MESSAGE RESPONSE
      9 016710  001417                        BEQ LOADER      ;ABORT IF NO RESPONSE
     10 016712  032764  000037  000032        BIT #ST.MSK,MC.MPK+P.STS(R4)    ;CHECK FOR ERRORS
     11 016720  001007                        BNE LOADE1
     12 016722  042765  000024  000012        BIC #CT.CMD+CT.REQ,C.FLG(R5)    ;CLEAR COMMAND OUTSTANDING FLAG
     13 016730  052765  000002  000012        BIS #CT.RN,C.FLG(R5)            ;SET DM PROGRAM RUNNING FLAG
     14 016736  000207                        RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;UDA FAILED TO DOWNLINE LOAD DM PROGRAM
                                ;
                                LOADE1: ERRDF 34,,FRR034
                                                                        TRAP    C$ERDF
016740  104455                                                         .WORD   34
016742  000042                                                         .WORD   0
016744  000000                                                         .WORD   ERR034
016746  012420
016750  000264                LOADER: SEZ                             ;SET Z TO INDICATE ERROR OCCURRED
016752  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
                                        ;
016754                                  BLDCMD: PUSH <R1,R0>
016754  010146                                                                    MOV R1,-(SP)
016756  010046                                                                    MOV R0,-(SP)
016760  016504  000014                        MOV C.RING(R5),R4       ;GET ADDRESS OF HOST COMM AREA
016764  010400                                MOV R4,R0               ;COPY TO R0
016766  062700  000100                        ADD #HC.CEV,R0          ;COMPUTE ADDRESS OF COMMAND ENVELOPE
016772  012720  000060                        MOV #HC.PSZ,(R0)+       ;LOAD PACKET LENGTH
016776  012701  001000                        MOV #DUP,R1             ;LOAD DIAG CIRCUIT IDENTIFIER
017002  022716  000031                        CMP #OP.MWR,(SP)        ;IF CODE IS MAINTENANCE WRITE
017006  001002                                BNE BLDC0               ;  GET OTHER CIRCUIT IDENTIFIER
017010  012701  177777                        MOV #DIAG,R1
017014  010120                        BLDC0:  MOV R1,(R0)+            ;PUT IDENTIFIER INTO PACKET
017016  012701  000030                        MOV #<HC.PSZ>/2,R1      ;GET WORDS TO CLEAR
017022  005020                        BLDC1:  CLR (R0)+               ;CLEAR PACKET
017024  005301                                DEC R1
017026  001375                                BNE BLDC1
017030                                        POP HC.CPK+P.OPCD(R4)   ;PUT OPCODE IN PACKET
017030  012664  000114                                                                MOV (SP)+,HC.CPK+P.OPCD(R4)
017034                                        POP R1                  ;RESTORE R1
017034  012601                                                                    MOV (SP)+,R1
017036  000207                                RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;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


017040                          SNDCMD: PUSH <R0,R1>
017040  010046                                                                 MOV R0,-(SP)
017042  010146                                                                 MOV R1,-(SP)
017044  016504  000014                MOV C.RING(R5),R4           ;LOAD R4 WITH HOST COMM AREA ADDRESS
017050  005265  000050                INC C.REF(R5)               ;INCREMENT CMD REFERENCE NUMBER
017054  016564  000050  000104        MOV C.REF(R5),HC.CPK+P.CRF(R4)  ;PUT IN PACKET
017062  012764  140000  000006        MOV #RG.OWN+RG.FLG,HC.MCT(R4)   ;MARK MESSAGE PACKET AVAILABLE
017070  012764  100000  000012        MOV #RG.OWN,HC.CCT(R4)      ;MARK COMMAND TO UDA
017076  005775  000000                TST @(R5)                   ;TELL UDA COMMAND IS THERE
017102  052765  000004  000012        BIS #CT.CMD,C.FLG(R5)       ;MARK COMMAND ISSUED
017110                                POP <R1,R0>
017110  012601                                                                 MOV (SP)+,R1
017112  012600                                                                 MOV (SP)+,R0
017114  000207                        RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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
                                        ;
017116                                  CLRBUF: PUSH <R0,R1>
017116  010046                                                                 MOV R0,-(SP)
017120  010146                                                                 MOV R1,-(SP)
017122  060400                                  ADD R4,R0                     ;ADD START OF HOST COMM AREA TO OFFSET
017124  010064  000124                          MOV R0,HC.CPK+P.UADR(R4)      ;PUT BUFFER ADDRESS IN COMMAND PACKET
017130  012764  000244  000120                  MOV #HC.BSZ,HC.CPK+P.BCNT(R4) ;PUT SIZE OF BUFFER IN COMMAND PACKET
017136  010004                                  MOV R0,R4                     ;PUT BUFFER ADDRESS IN R4
017140  012701  000122                          MOV #HC.BSZ/2,R1              ;GET SIZE OF BUFFER IN WORDS
017144  005020                          CLRBFL: CLR (R0)+                     ;CLEAR ALL THE WORDS
017146  005301                                  DEC R1
017150  001375                                  BNE CLRBFL
017152                                          POP <R1,R0>
017152  012601                                                                 MOV (SP)+,R1
017154  012600                                                                 MOV (SP)+,R0
017156  000207                                  RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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
                                        ;
017160                                  WAITMS: PUSH <R0,R1>
017160  010046                                                                 MOV R0,-(SP)
017162  010146                                                                 MOV R1,-(SP)
017164  012700  000036                          MOV #30.,R0                   ;SET TIME OUT VALUE OF 30 SECONDS
017170  010501                                  MOV R5,R1                     ;POINT TO TIME OUT COUNTER
017172  062701  000036                          ADD #C.TO,R1
017176  004737  017352                          CALL SETTO
017202  011500                          1$:     MOV (R5),R0                   ;GET ADDRESS OF UDAIP REGISTER
017204  032765  000010  000012                  BIT #CT.MSG,C.FLG(R5)         ;LOOK IF INTERRUPT OCCURRED
017212  001030                                  BNE 3$                        ;BRANCH IF SO
017214  016001  000002                          MOV 2(R0),R1                  ;LOOK AT UDASA REGISTER
017220  001034                                  BNE 4$                        ;BRANCH IF ERROR CODE PRESENT
017222                                          BREAK
017222  104422                                                                 TRAP    C$BRK
017224  005737  003206                          TST KW.CSR                    ;SEE IF A CLOCK ON SYSTEM
017230  001764                                  BEQ 1$
017232  023765  003220  000040                  CMP KW.EL+2,C.TOH(R5)         ;CHECK IF TIMEOUT HAS HAPPENED
017240  101005                                  BHI 2$
017242  001357                                  BNE 1$
017244  023765  003216  000036                  CMP KW.EL,C.TO(R5)
017252  103753                                  BLO 1$
017254                                  2$:     ERRDF 36,,ERR036
017254  104455                                                                 TRAP    C$ERDF
017256  000044                                                                 .WORD   36
017260  000000                                                                 .WORD   0
017262  012426                                                                 .WORD   ERR036
017264                                          POP <R1,R0>
017264  012601                                                                 MOV (SP)+,R1
017266  012600                                                                 MOV (SP)+,R0
017270  000264                                  SEZ
017272  000207                                  RETURN
```

```
      1 017274  042765  000010  000012  3$:     BIC #CT.MSG,C.FLG(R5)          ;CLEAR MESSAGE RECEIVED FLAG
      2 017302                                  POP <R1,R0>
        017302  012601                                                          MOV (SP)+,R1
        017304  012600                                                          MOV (SP)+,R0
      3 017306  000244                          CLZ                           ;GIVE NO ERROR RETURN
      4 017310  000207                          RETURN
      5 017312                            4$:   ERRDF 37,,ERR037
        017312  104455                                                         TRAP    C$ERDF
        017314  000045                                                         .WORD   37
        017316  000000                                                         .WORD   0
        017320  012440                                                         .WORD   ERR037
      6 017322                                  POP <R1,R0>
        017322  012601                                                         MOV (SP)+,R1
        017324  012600                                                         MOV (SP)+,R0
      7 017326  000264                          SEZ
      8 017330  000207                          RETURN
```

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

017332                          BGNSRV NXMI
017332                                                                  NXMI::
017332  012737  177777  002172          MOV #-1,NXMAD
017340                          ENDSRV
017340                                                                  L10031:
017340  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

                                BGNSRV UDASRV
017342                                                                  UDASRV::
017342
017342  052710  000010                  BIS #CT.MSG,(R0)                ;SET CT.MSG
017346                                  POP R0                          ;RESTORE R0
017346  012600                                                                  MOV (SP)+,R0
017350                          ENDSRV
017350                                                                  L10032:
017350  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

017352                          SETTO:  PUSH <R2,R3>
017352  010246                                                                    MOV R2,-(SP)
017354  010346                                                                    MOV R3,-(SP)
017356  005002                          CLR R2                  ;CLEAR PRODUCT
017360  013703  003214                  MOV KW.HZ,R3            ;GET MULTIPLICAND
017364  006200                  SETO0:  ASR R0                  ;SHIFT MULTIPLIER TO RIGHT
017366  103001                          BCC SETO1               ;IF A ONE BIT SHIFTED OUT
017370  060302                          ADD R3,R2               ;  ADD MULTIPLICAND TO PRODUCT
017372  006303                  SETO1:  ASL R3                  ;DOUBLE THE MULTIPLICAND
017374  005700                          TST R0
017376  001372                          BNE SETO0               ;CONTINUE UNTIL MULTIPLIER IS ZERO

                                ;GET CURRENT TIME

017400  013700  003216          SETO2:  MOV KW.EL,R0            ;GET TIME
017404  013703  003220                  MOV KW.EL+2,R3
017410  020037  003216                  CMP R0,KW.EL            ;IF CHANGED DURING RETRIEVAL
017414  001371                          BNE SETO2               ;  GET IT AGAIN

                                ;ADD TIME TIL TIMEOUT

017416  060200                          ADD R2,R0               ;ADD
017420  005503                          ADC R3

                                ;PUT RESULT IN STORAGE

017422  010021                          MOV R0,(R1)+
017424  010311                          MOV R3,(R1)

017426                                  POP <R3,R2>
017426  012603                                                                    MOV (SP)+,R3
017430  012602                                                                    MOV (SP)+,R2
017432  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

017434  016502  000014        UDAINT: MOV C.RING(R5),R2               ;GET FIRST ADDRESS OF RING BUFFER
017440  012703  000006                MOV #<HC.RSZ*2*HC.ISZ>/2,R3       ;GET SIZE OF RING BUFFER
017444  012722  177777        UDAI1L: MOV #-1,(R2)+                   ;WRITE ONES TO BUFFER
017450  005303                        DEC R3                          ;COUNT THE WORDS IN BUFFER
017452  003374                        BGT UDAI1L                      ;LOOP UNTIL ENTIRE BUFFER WRITTEN

                                ;DO THE INITIALIZATION

017454  004737  017702                CALL UDAIST                     ;DO FIRST THREE STEPS
017460  103506                        BCS UDAIEX                      ;GET OUT IF UDA MICROCODE REPORTED FAILURE
017462  012364  000002                MOV (R3)+,2(R4)                 ;WRITE NEXT WORD TO UDASA REGISTER
017466  012703  000310                MOV #200.,R3                    ;GET TRY COUNTER
017472  016402  000002        UDAI1A: MOV 2(R4),R2                    ;LOOK AT UDASA
017476  001407                        BEQ UDAI1C
017500  005303                        DEC R3
017502  001373                        BNE UDAI1A
017504                                ERRDF 24.,ERR024
017504  104455                                                          TRAP    C$ERDF
017506  000030                                                          .WORD   24
017510  000000                                                          .WORD   0
017512  012314                                                          .WORD   ERR024
017514  000470                        BR UDAIEX
017516  010264  000002        UDAI1C: MOV R2,2(R4)                    ;WRITE 0 TO UDASA (PURGE)
017522  011402                        MOV (R4),R2                     ;READ FROM UDAIP (POLL)
017524  004737  020220                CALL UDARSP                     ;WAIT FOR STEP OR ERROR BIT
017530  103462                        BCS UDAIEX                      ;GET OUT IF UDA MICROCODE REPORTED FAILURE
017532  042702  174017                BIC #^C<SA.CNT>,R2              ;CLEAR OTHER BITS
017536  006202                        ASR R2                          ;MOVE TO RIGHT OF REGISTER
017540  006202                        ASR R2
017542  006202                        ASR R2
017544  006202                        ASR R2
017546  020227  000006                CMP R2,#6                       ;CONTROLLER MODEL MUST BE 6
017552  001410                        BEQ UDAI2
017554  020227  000015                CMP R2,#13.                     ;  OR 13
017560  001405                        BEQ UDAI2
017562                                ERRDF 14.,ERR014                ;REPORT CONTROLLER NEEDS NEW REVISION
017562  104455                                                          TRAP    C$ERDF
017564  000016                                                          .WORD   14
```

GLOBAL SUBROUTINES SECTION

```
        017566  000000                                          .WORD   0
        017570  012120                                          .WORD   ERR014
     52 017572  000441          BR UDAIEx
```

```
                                        ;CHECK HOST COMMUNICATION AREA FOR ALL ZEROS

                                        UDAI2:  MOV C.RING(R5),R2               ;GET FIRST ADDRESS OF RING BUFFER
                                                MOV R2,R1                       ;SAVE FOR ERROR MESSAGE
                                                MOV #<HC.RSZ*2*HC.ISZ>/2.R3     ;GET SIZE OF RING BUFFER
                                        UDAI2L: TST (R2)+                       ;CHECK WORD IN BUFFER
                                                BNE UDAI2E                      ;GO TO ERROR REPORTER IF NOT ZERO
                                                DEC R3                          ;COUNT THE WORDS IN BUFFER
                                                BGT UDAI2L                      ;LOOP UNTIL ALL WORDS CHECKED
                                                BR UDAI3

                                        UDAI2E: ERRDF 23.,ERR023                ;REPORT BUFFER NOT CLEARED
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   23
                                                                                        .WORD   0
                                                                                        .WORD   ERR023
                                                BR UDAIEX

                                        ;SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION

                                        UDAI3:
                                                MOV #SA.GO,R0
                                                MOV R0,2(R4)                    ;SEND TO UDA
                                                MOV C.RING(R5),R1
                                                MOV R1,HC.MSG(R1)
                                                ADD #HC.MPK,HC.MSG(R1)
                                                MOV R1,HC.CMD(R1)
                                                ADD #HC.CPK,HC.CMD(R1)
                                                CLZ                             ;CLEAR Z AS NO ERROR INDICATION
                                                RETURN

                                        ;ERROR RETURN

                                        UDAIEX: SEZ                             ;SET Z TO INDICATE ERROR OCCURRED
                                                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
                                ;
017702                          UDAIST: BREAK                                                   TRAP    C$BRK
017702  104422
017704                                  PUSH R1                                                 MOV R1,-(SP)
017704  010146
017706  016504  000004                  MOV C.VEC(R5),R4
017712                                  AND CT.VEC,R4                                           BIC #^C<CT.VEC>,R4
017712  042704  177000
017716  006204                          ASR R4
017720  006204                          ASR R4
017722  052704  100000                  BIS #SA.STP,R4          ;SET STEP BIT IN DATA WORD
017726  010437  020120                  MOV R4,UDAID1           ;LOAD INTERRUPT VECTOR
017732  016537  000014  020124          MOV C.RING(R5),UDAID2   ;LOAD MEMORY ADDRESS
017740  062737  000004  020124          ADD #HC.MSG,UDAID2      ;  OF FIRST RESPONSE RING
                                ;
                                ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER
                                ;
017746  016504  000000          MOV C.UADR(R5),R4       ;GET ADDRESS OF UDAIP REGISTER
017752  005037  002172          CLR NXMAD               ;CLEAR MEMORY ERROR FLAG
017756                          SETVEC #4,#NXMI,#PRI07  ;SET UP VECTOR 4
017756  012746  000340                                                  MOV     #PRI07,-(SP)
017762  012746  017332                                                  MOV     #NXMI,-(SP)
017766  012746  000004                                                  MOV     #4,-(SP)
017772  012746  000003                                                  MOV     #3,-(SP)
017776  104437                                                          TRAP    C$SVEC
020000  062706  000010                                                  ADD     #10,SP
020004  005764  000002          TST 2(R4)               ;ACCESS UDASA REGISTER
020010  005014                  CLR (R4)                ;WRITE TO UDAIP
020012                          CLRVEC #4               ;GIVE UP THE VECTOR
020012  012700  000004                                                  MOV     #4,R0
020016  104436                                                          TRAP    C$CVEC
020020  005737  002172          TST NXMAD               ;SEE IF A MEMORY ERROR OCCURRED
020024  001406                  BEQ UDAISG
020026                          ERRDF 20.,ERR020                                                TRAP    C$ERDF
020026  104455                                                          .WORD   20
020030  000024                                                          .WORD   0
020032  000000                                                          .WORD   ERR020
020034  012134
020036  000261          SEC
020040  000424          BR UDAISE
```

GLOBAL SUBROUTINES SECTION

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

020042  012737  004000  020356  UDAISG: MOV #SA.S1,UDARSD               ;STORE RESPONSE MASK
020050  012703  020116                  MOV #UDAIDT,R3                  ;AND INDEX TO TABLE

                                        ;WAIT FOR AND CHECK RESPONSE DATA

020054  004737  020220          UDAISL: CALL UDARSP                     ;WAIT FOR STEP OR ERROR BITS
020060  103414                          BCS UDAISE                      ;EXIT IF ERROR
020062  004733                          CALL @(R3)+                     ;CALL RESPONSE CHECKER FOR STEP
020064  103412                          BCS UDAISE                      ;GET OUT IF ERROR
020066  006337  020356                  ASL UDARSD                      ;SHIFT TO NEXT STEP BIT
020072  032737  040000  020356          BIT #SA.S4,UDARSD               ;CHECK IF NOW AT STEP 4
020100  001003                          BNE UDAISX                      ;GET OUT IF SO
020102  012364  000002                  MOV (R3)+,2(R4)                 ;WRITE DATA TO UDASA REGISTER
020106  000762                          BR UDAISL                       ;STAY IN LOOP

020110  000241          UDAISX: CLC                                     ;CLEAR CARRY FOR NO ERROR INDICATION
020112          UDAISE: POP R1
020112  012601                                                          MOV (SP)+,R1
020114  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION
                                
020116  020132                  UDAIDT: .WORD UDAIR1                    ;FIRST WORD RESPONSE CHECK ROUTINE
020120  000000                  UDAID1: .WORD 0                         ;FIRST WORD TO SEND TO UDASA
020122  020144                          .WORD UDAIR2                    ;SECOND WORD RESPONSE CHECK ROUTINE
020124  000000                  UDAID2: .WORD 0                         ;SECOND WORD TO SEND TO UDASA
020126  020164                          .WORD UDAIR3                    ;THIRD WORD RESPONSE CHECK ROUTINE
020130  100000                  UDAID3: .WORD SA.TST                    ;THIRD WORD TO SEND TO UDASA
                                
                                ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
                                ;CHECK FOR PROPER CONTROLLER TYPE
                                
020132  012701  004400          UDAIR1: MOV #SA.S1+SA.DI,R1             ;SET STEP ONE BIT
020136  042702  001140                  BIC #<SA.QB+SA.MP+SA.SM>,R2     ;MASK OFF UNWANTED BITS
020142  000416                          BR UDAIRC                       ;NOW COMPARE
                                
                                ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
                                ;CHECK FOR ECHO OF INTI AND VECTOR
                                
020144  013701  020120          UDAIR2: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
020150  000301                          SWAB R1                         ;GET HIGH 8 BITS
020152  042701  177400                  BIC #177400,R1
020156  052701  010000                  BIS #SA.S2,R1                   ;SET STEP 2 BIT
020162  000406                          BR UDAIRC                       ;NOW COMPARE
                                
                                ;RESPONSE CHECK FOR THIRD WORD FROM UDASA
                                ;CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
                                
020164  013701  020120          UDAIR3: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
020170  042701  177400                  BIC #177400,R1                  ;JUST LOW 8 BITS
020174  052701  020000                  BIS #SA.S3,R1                   ;SET STEP 3 BIT
                                
                                ;COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
                                
020200  020102                  UDAIRC: CMP R1,R2                       ;COMPARE THE DATA
020202  001405                          BEQ UDAIRX                      ;EXIT IF COMPARED CORRECTLY
020204                                  ERRDF 25.,ERR025                ;REPORT ERROR
020204  104455                                                                  TRAP    C$ERDF
020206  000031                                                                  .WORD   25
020210  000000                                                                  .WORD   0
020212  012330                                                                  .WORD   ERR025
020214  000261                          SEC
020216  000207                  UDAIRX: RETURN
```

```
                                        ;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

020220                          UDARSP: PUSH R1                                         MOV R1,-(SP)
020220  010146
020222  052737  100000  020356          BIS #SA.ERR,UDARSD              ;SET ERROR BIT IN MASK WORD
020230  012700  000012                  MOV #10.,R0                     ;SET UP FOR 10 SECOND TIMEOUT
020234  010501                          MOV R5,R1                       ;POINT TO COUNTER IN CONTROLLER TABLE
020236  062701  000036                  ADD #C.TO,R1
020242  004737  017352                  CALL SETTO
020246                                  POP R1                                          MOV (SP)+,R1
020246  012601
020250  033764  020356  000002  UDARS1: BIT UDARSD,2(R4)                 ;LOOK AT ERROR AND STEP BIT
020256  001024                          BNE UDARS2                      ;BRANCH IF EITHER SET
020260                                  BREAK                                           TRAP    C$BRK
020260  104422
020262  005737  003206                  TST KW.CSR                      ;SEE IF CLOCK ON SYSTEM
020266  001770                          BEQ UDARS1
020270  023765  003220  000040          CMP KW.EL+2,C.TOH(R5)           ;CHECK IF TIME OUT OCCURRED
020276  101005                          BHI 1$
020300  001363                          BNE UDARS1
020302  023765  003216  000036          CMP KW.EL,C.TO(R5)
020310  103757                          BLO UDARS1
020312  016402  000002          1$:     MOV 2(R4),R2                    ;GET REGISTER CONTENTS
020316                                  ERRDF 22.,ERR022                ;REPORT TIME OUT ERROR
                                                                                        TRAP    C$ERDF
020316  104455                                                                          .WORD   22
020320  000026                                                                          .WORD   0
020322  000000                                                                          .WORD   ERR022
020324  012210
020326  000407                          BR UDARSE
```

GLOBAL SUBROUTINES SECTION

```
                                        ;CHECK IF ERROR BIT SET
 1
 2
 3 020330  016402  000002      UDARS2: MOV 2(R4),R2                    ;GET REGISTER CONTENTS
 4 020334  100006                      BPL UDARSX                      ;EXIT IF ERROR NOT SET
 5 020336                              ERRDF 21,,ERR021                ;REPORT ERROR INFO
                                                                               TRAP    C$ERDF
   020336  104455                                                              .WORD   21
   020340  000025                                                              .WORD   0
   020342  000000                                                              .WORD   ERR021
   020344  012146
 6 020346  000261              UDARSE: SEC
 7 020350  000207                      RETURN
 8
 9                                      ;NORMAL EXIT
10
11 020352  000241              UDARSX: CLC                             ;CLEAR CARRY AS NO ERROR INDICATION
12 020354  000207                      RETURN
13
14                                     ;LOCATION FOR STEP BIT MASK
15
16 020356  000000              UDARSD: .WORD 0                         ;LOAD BY CALLING ROUTINE
```

GLOBAL SUBROUTINES SECTION

```
                                        ;KW11I
                                        ;
                                        ;CLOCK INTERRUPT SERVICE ROUTINE
                                        
 020360                                 BGNSRV KW11I
                                                                                KW11I::
 020360
 020360  062737  000001  003216                ADD #1,KW.EL                  ;COUNT THE INTERRUPT
 020366  005537  003220                        ADC KW.EL+2
 020372  012777  000105  162606                MOV #KWOUT.,@KW.CSR           ;RESTART THE CLOCK
 020400                                 ENDSRV
                                                                                L10033:
 020400                                                                                 RTI
 020400  000002
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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
                                        
020402  005737  003206          RNTIME: TST KW.CSR                      ;CHECK IF A CLOCK PRESENT
020406  001465                          BEQ RNTIMX                      ;BRANCH IF NOT
020410                                  PUSH <R0,R3,R4,R5>
020410  010046                                                                  MOV R0,-(SP)
020412  010346                                                                  MOV R3,-(SP)
020414  010446                                                                  MOV R4,-(SP)
020416  010546                                                                  MOV R5,-(SP)
020420  013703  003216                  MOV KW.EL,R3                    ;GET ELAPSED TIME
020424  013704  003220                  MOV KW.EL+2,R4
020430  013700  003214                  MOV KW.HZ,R0                    ;GET SPEED OF CLOCK
020434  004737  016516                  CALL DIVIDE                     ;COMPUTE SECONDS OF ELAPSED TIME
020440  012700  000074                  MOV #60.,R0                     ;NOW DIVIDE BY 60
020444  004737  016516                  CALL DIVIDE                     ; TO COMPUTE MINUTES
020450                                  PUSH R5                         ;SAVE REMAINDER AS SECONDS
020450  010546                                                                  MOV R5,-(SP)
020452  004737  016516                  CALL DIVIDE                     ;DIVIDE BY 60 AGAIN
020456                                  PNT RNTIM,R3                    ;PRINT HOURS
020456  010346                                                                  MOV R3,-(SP)
020460  004137  016450                                                          JSR R1,LPNT
020464  003712                                                                  .WORD RNTIM
020466  000002                                                                  .WORD PNT.CT
020470  020527  000011                  CMP R5,#9.                      ;IF MINUTES 9 OR LESS
020474  003004                          BGT 1$
020476                                  PRINT #'0                       ;PRINT A LEADING ZERO
020476  112700  000060                                                          MOVB #'0,R0
020502  004737  016240                                                          CALL CPNT
020506                          1$:     PNT RNTIM1,R5                   ;NOW PRINT MINUTES
020506  010546                                                                  MOV R5,-(SP)
020510  004137  016450                                                          JSR R1,LPNT
020514  003735                                                                  .WORD RNTIM1
020516  000002                                                                  .WORD PNT.CT
020520                                  POP R5                          ;GET SECONDS
020520  012605                                                                  MOV (SP)+,R5
020522  020527  000011                  CMP R5,#9.                      ;IF 9 OR LESS
020526  003004                          BGT 2$
020530                                  PRINT #'0                       ;PRINT A LEADING ZERO
020530  112700  000060                                                          MOVB #'0,R0
020534  004737  016240                                                          CALL CPNT
020540                          2$:     PNT RNTIM2,R5                   ;NOW PRINT SECONDS
020540  010546                                                                  MOV R5,-(SP)
020542  004137  016450                                                          JSR R1,LPNT
020546  003743                                                                  .WORD RNTIM2
020550  000002                                                                  .WORD PNT.CT
020552                                  POP <R5,R4,R3,R0>               ;HOURS IN R3
020552  012605                                                                  MOV (SP)+,R5
```

GLOBAL SUBROUTINES SECTION

```
                                                                                MOV (SP)+,R4
                                                                                MOV (SP)+,R3
                                                                                MOV (SP)+,R0
    020554  012604
    020556  012603
    020560  012600
 35 020562                          RNTIMX: PRINT <#' >                         ;PRINT A SPACE
    020562  112700  000040                                                      MOVB #' ,R0
    020566  004737  016240                                                      CALL CPNT
 36 020572  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                DATE:   GMANID DATEQ,DATEI,A,-1,1,11.,YES       ;GET DATE
020574                                                                                  TRAP    C$GMAN
020574  104443                                                                          BR      10000$
020576  000406                                                                          .WORD   DATEI
020600  003270                                                                          .WORD   T*CODE
020602  000152                                                                          .WORD   DATEQ
020604  003544                                                                          .WORD   -1
020606  177777                                                                          .WORD   T$LOLIM
020610  000001                                                                          .WORD   T$HILIM
020612  000013
020614                                                                          10000$:
020614  012705  003270                  MOV #DATEI,R5   ;GET POINTER TO ANSWER
020620  121527  000060                  CMPB (R5),#'0
020624  103443                          BLO DERR
020626  122527  000071          DAY:    CMPB (R5)+,#'9
020632  101040                          BHI DERR
020634  121527  000055                  CMPB (R5),#'-
020640  001406                          BEQ DAS1
020642  121527  000060                  CMPB (R5),#'0
020646  103432                          BLO DERR
020650  122527  000071                  CMPB (R5)+,#'9
020654  101027                          BHI DERR
020656  122527  000055          DAS1:   CMPB (R5)+,#'-
020662  001024                          BNE DERR
020664  012704  000014                  MOV #12.,R4     ;GET NUMBER OF MONTH
020670  012703  003345                  MOV #MONTHS,R3  ;GET POINTER TO MONTH NAMES
020674  005000          MON1:   CLR R0
020676  121523                  CMPB (R5),(R3)+
020700  001401                  BEQ MON2
020702  005200                  INC R0
020704  126523  000001          MON2:   CMPB 1(R5),(R3)+
020710  001401                  BEQ MON3
020712  005200                  INC R0
020714  126523  000002          MON3:   CMPB 2(R5),(R3)+
020720  001401                  BEQ MON4
020722  005200                  INC R0
020724  005700          MON4:   TST R0
020726  001407                  BEQ MON5
020730  005304                  DEC R4
020732  001360                  BNE MON1
020734                  DERR:   PNTF DATEX
020734  004137  016412                                                                  JSR R1,LPNTF
020740  011671                                                                          .WORD DATEX
020742  000000                                                                          .WORD PNT.CT
020744  000713                  BR DATE
020746  012701  003304          MON5:   MOV #DATEO,R1   ;GET POINTER TO DATE FOR FORMATTER
020752  010403                  MOV R4,R3       ;GET COPY OF MONTH NUMBER
020754  020327  000012                  CMP R3,#10.     ;  IF 10 OR GREATER
020760  103404                  BLO MON6
020762  112721  000061                  MOVB #'1,(R1)+  ;PUT A "1" IN OUTPUT
020766  162703  000012                  SUB #10.,R3
020772  062703  000060          MON6:   ADD #'0,R3      ;CONVERT MONTH NUMBER TO ASCII
020776  110321                  MOVB R3,(R1)+   ;PUT A NUMBER IN OUTPUT
021000  112721  000055                  MOVB #'-,(R1)+  ;PUT A "-" IN OUTPUT
021004  062704  003410                  ADD #DAYS-1,R4  ;GET POINTER TO DAYS IN MONTH
                                                        ;INDEXED BY NUMBER OF MONTH
021010  012703  003270                  MOV #DATEI,R3   ;GET POINTER TO DATE INPUT
021014  005000                  CLR R0
```

GLOBAL SUBROUTINES SECTION

```
 46 021016  121327  000055          DAY1:   CMPB (R3),#'-
 47 021022  001413                          BEQ DAY2
 48 021024  111321                          MOVB (R3),(R1)+ ;PUT DAY CHARACTER IN OUTPUT
 49 021026  006300                          ASL R0
 50 021030  010002                          MOV R0,R2
 51 021032  006300                          ASL R0
 52 021034  006300                          ASL R0
 53 021036  060200                          ADD R2,R0
 54 021040  112302                          MOVB (R3)+,R2
 55 021042  162702  000060                  SUB #'0,R2
 56 021046  060200                          ADD R2,R0
 57 021050  000762                          BR DAY1
 58 021052  120014                  DAY2:   CMPB R0,(R4)
 59 021054  101327                          BHI DERR
 60 021056  005700                          TST R0  ;SEE IF DATE IS ZERO
 61 021060  001725                          BEQ DERR        ;ERROR IF SO
 62 021062  062705  000003                  ADD #3,R5
 63 021066  121527  000055                  CMPB (R5),#'-   ;CHECK FOR "-" BETWEEN DAY
 64 021072  001320                          BNE DERR        ;  AND YEAR IN OUTPUT
 65 021074  112521                          MOVB (R5)+,(R1)+        ;PUT "-" IN OUTPUT
 66 021076  010504                          MOV R5,R4       ;GET COPY OF INPUT STRING POINTER
 67 021100  005000                          CLR R0
 68 021102  005002                          CLR R2
 69 021104  121427  000060          YER1:   CMPB (R4),#'0
 70 021110  103416                          BLO YER2
 71 021112  121427  000071                  CMPB (R4),#'9
 72 021116  101013                          BHI YER2
 73 021120  006300                          ASL R0
 74 021122  010003                          MOV R0,R3
 75 021124  006300                          ASL R0
 76 021126  006300                          ASL R0
 77 021130  060300                          ADD R3,R0
 78 021132  112403                          MOVB (R4)+,R3
 79 021134  162703  000060                  SUB #'0,R3
 80 021140  060300                          ADD R3,R0
 81 021142  005202                          INC R2
 82 021144  000757                          BR YER1
 83 021146  105714                  YER2:   TSTB (R4)
 84 021150  001271                          BNE DERR
 85 021152  020227  000002                  CMP R2,#2
 86 021156  001407                          BEQ YER3
 87 021160  020227  000004                  CMP R2,#4
 88 021164  001263                          BNE DERR
 89 021166  020027  003554                  CMP R0,#1900.
 90 021172  103660                          BLO DERR
 91 021174  000413                          BR YER5
 92 021176  012702  003425          YER3:   MOV #YEAR19,R2
 93 021202  020027  000106                  CMP R0,#70.
 94 021206  103002                          BHIS YER4
 95 021210  012702  003430                  MOV #YEAR20,R2
 96 021214  105712                  YER4:   TSTB (R2)
 97 021216  001402                          BEQ YER5
 98 021220  112221                          MOVB (R2)+,(R1)+
 99 021222  000774                          BR YER4
100 021224  112521                  YER5:   MOVB (R5)+,(R1)+
101 021226  001376                          BNE YER5
102 021230  000207                          RETURN
```

```
    103
    104 021232  000000                BRSAV:  .WORD   0                       ;DEFAULT BR LEVEL AND VECTOR
    105
    106 021234                                ENDMOD
```

```
                                .SBTTL  PROTECTION TABLE

                                        BGNMOD

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

                                        BGNPROT
021234                                                                  L$PROT::

021234  177777                          -1                      ;OFFSET INTO P-TABLE FOR CSR ADDRESS
021236  177777                          -1                      ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
021240  177777                          -1                      ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

021242                                  ENDPROT
```

```
                                        .SBTTL  INITIALIZE SECTION
                                        ;
                                        ;**
                                        ;       THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                        ;       AT THE BEGINNING OF EACH PASS. THIS CODE IS EXECUTED UNDER FIVE
                                        ;       CONDITIONS.  THERE
                                        ;       ARE SUPERVISOR EVENT FLAGS THAT ARE USED TO LET THE
                                        ;       DIAGNOSTIC KNOW UNDER WHICH CONDITION THE EXECUTION IS TAKING
                                        ;       PLACE.  THE EVENT FLAGS ARE READ USING THE "READEF" MACRO.
                                        ;       THE CONDITIONS UNDER WHICH THE INIT CODE IS EXECUTED AND THE
                                        ;       CORRRESPONDING EVENT FLAGS ARE:
                                        ;               START COMMAND            EF.START
                                        ;               RESTART COMMAND          EF.RESTART
                                        ;               CONTINUE COMMAND         EF.CONTINUE
                                        ;               POWERDOWN/POWERUP        EF.PWR
                                        ;               NEW PASS                 EF.NEW
                                        ;
                                        ;       IF HERE FROM START COMMAND THEN
                                        ;               SET ISTRT BIT & CLEAR OTHER BITS IN FLAG
                                        ;
                                        ;       IF HERE FROM RESTART COMMAND THEN
                                        ;               SET IREST BIT IN IFLAGS
                                        ;
                                        ;       IF HERE FROM START OR RESTART COMMAND THEN
                                        ;               RESET ALL UNITS
                                        ;               ESTABLISH FREE MEMORY
                                        ;               CLEAR TNUM
                                        ;               INITIALIZE CLOCK
                                        ;               BUILD CONTROLLER & DRIVES TABLES IN MEMORY
                                        ;               EXIT INIT SECTION
                                        ;
                                        ;       IF HERE FROM CONTINUE COMMAND THEN
                                        ;               SET ICONT BIT IN IFLAGS
                                        ;               EXIT INIT SECTION
                                        ;
                                        ;       IF HERE FROM POWER FAIL RESTART THEN
                                        ;               EXIT INIT SECTION
                                        ;
                                        ;       IF HERE FROM NEW PASS OR SUB-PASS THEN
                                        ;               LOOK FOR ANY ADDED OR DROPPED UNITS
                                        ;               EXIT INIT SECTION
                                        ;--
                                        BGNINIT
                                                                                L$INIT::
                                                                        ;HERE FROM START COMMAND?
                                        READEF  #EF.STA
021242  012700  000040                                                    MOV     #EF.STA,R0
021246  104447                                                          TRAP    C$REFG
                                                                        ;BRANCH TO 1$ IF NOT, ELSE
                                        BNCOMPLETE      1$
                                                                                BCC     1$
021252  012737  000010  003204          MOV     #ISTRT,IFLAGS           ;SET START BIT IN FLAG.
021260  000432                          BR      INIT1
                                1$:                                     ;HERE FROM RESTART COMMAND?
                                        READEF  #EF.RES
```

```
                021262  012700  000037                                          MOV     #EF.RES,R0
                021266  104447                                                  TRAP    C$REFG
                                                        ;BRANCH TO 2$ IF NOT, ELSE
54
55              021270                          BNCOMPLETE      2$
                021270  103004                                                  BCC     2$
56              021272  052737  000004  003204  BIS     #IREST,IFLAGS           ;SET RESTART BIT IN FLAG.
57              021300  000422                  BR      INIT1
58              021302                  2$:                                     ;HERE FROM CONTINUE COMMAND?
59              021302                          READEF  #EF.CON
                021302  012700  000036                                          MOV     #EF.CON,R0
                021306  104447                                                  TRAP    C$REFG
                                                        ;BRANCH TO 3$ IF NOT, ELSE
60
61              021310                          BNCOMPLETE      3$
                021310  103007                                                  BCC     3$
62              021312  042737  000020  003204  BIC     #ISTRTH,IFLAGS          ;CLEAR 1ST TIME THRU FLAG AND
63              021320  052737  000002  003204  BIS     #ICONT,IFLAGS           ;SET CONTINUE BIT IN FLAG.
64              021326  000405                  BR      INIT0
65              021330                  3$:                                     ;HERE FROM POWER FAIL?
66              021330                          READEF  #EF.PWR
                021330  012700  000034                                          MOV     #EF.PWR,R0
                021334  104447                                                  TRAP    C$REFG
                                                        ;BRANCH TO INIT0 IF POWER FAIL, ELSE
67
68              021336                          BCOMPLETE       INIT0
                021336  103401                                                  BCS     INIT0
69              021340                  INITQT: DOCLN                           ; ABORT PROGRAM ON NEW PASS
                021340  104444                                                  TRAP    C$DCLN
70
71              021342  000137  022126  INIT0:  JMP     INITXX                  ; EXIT THE INITIALIZE SECTION.
72
73                                      ;
74                                      ;       INITIALIZE KW11 CLOCK, FREE MEMORY AND IP ADDRESS TABLE
75                                      ;       DURING START OR RESTART COMMAND ONLY
76                                      ;
77
78              021346  012700  000003  INIT1:  MOV     #S0.FMT,R0              ; GET BITS FOR REFORMAT MODE FLAG
79              021352  030037  002136          BIT     R0,SFPTBL               ; CHECK IF REFORMAT
80              021356  001011                  BNE     1$                      ; IF SO, CONTINUE
81              021360  012700  000004          MOV     #S0.CNS,R0              ; GET BIT FOR RECONSTRUCT FLAG
82              021364  030037  002136          BIT     R0,SFPTBL               ; CHECK IF RECONSTRUCT MODE
83              021370  001004                  BNE     1$                      ; IF SO, CONTINUE
84              021372  006300                  ASL     R0                      ; GET BIT FOR RESTORE MODE
85              021374  030037  002136          BIT     R0,SFPTBL               ; CHECK IF RESTORE MODE
86              021400  001757                  BEQ     INITQT                  ; IF NONE OF ABOVE, ABORT TEST
87              021402  010037  003200  1$:     MOV     R0,MODE                 ; SAVE MODE FLAGS
88
89                      000105                  KWOUT.=105                      ; DATA TO START CLOCK
90
91              021406  005037  003216          CLR     KW.EL                   ;CLEAR ELAPSED TIME
92              021412  005037  003220          CLR     KW.EL+2
93              021416                          CLOCK   L,R0                    ;SEE IF L-CLOCK PRESENT
                021416  012700  000114                                          MOV     #'L,R0
                021422  104462                                                  TRAP    C$CLCK
94              021424                          BCOMPLETE       2$
                021424  103413                                                  BCS     2$
95              021426                          CLOCK   P,R0                    ;SEE IF P-CLOCK PRESENT
                021426  012700  000120                                          MOV     #'P,R0
                021432  104462                                                  TRAP    C$CLCK
```

```
021434                                  BCOMPLETE       2$
021434  103407                                                                                  BCS     2$
021436  005037  003206                  CLR     KW.CSR                  ;IF NEITHER, CLEAR CSR STORAGE WORD
021442                                  PNTF    NOCLOCK
021442  004137  016412                                                                          JSR R1,LPNTF
021446  004110                                                                                  .WORD NOCLOCK
021450  000000                                                                                  .WORD PNT.CT
021452  000426                          BR      3$

021454  012037  003206          2$:     MOV     (R0)+,KW.CSR            ;STORE DATA RETURNED
021460  012037  003210                  MOV     (R0)+,KW.BRL
021464  012037  003212                  MOV     (R0)+,KW.VEC
021470  012037  003214                  MOV     (R0)+,KW.HZ
                                                                        ;SETUP KW11 VECTOR ADDRESS
021474                                  SETVEC  KW.VEC,#KW11I,#PRI07
021474  012746  000340                                                                          MOV     #PRI07,-(SP)
021500  012746  020360                                                                          MOV     #KW11I,-(SP)
021504  013746  003212                                                                          MOV     KW.VEC,-(SP)
021510  012746  000003                                                                          MOV     #3,-(SP)
021514  104437                                                                                  TRAP    C$SVEC
021516  062706  000010                                                                          ADD     #10,SP
021522  012777  000105  161456          MOV     #KWOUT.,@KW.CSR         ;START THE CLOCK
021530  004737  012750          3$:     CALL    RESET                   ;RESET ALL CONTROLLERS
021534                                  MEMORY  FFREE                   ;RESET START OF FREE MEMORY
021534  104431                                                                                  TRAP    C$MEM
021536  010037  002140                                                                          MOV     R0,FFREE
021542  017737  160372  002142          MOV     @FFREE,FSIZE            ;RESET SIZE OF FREE MEMORY


                                ;
                                ;       ALLOCATE DRIVE TABLES TO MEMORY
                                ;

021550  013737  002140  003202  INIT2:  MOV     FFREE,DTABS             ;STORE START OF DRIVE TABLES AND
021556  005077  161420                  CLR     @DTABS                  ;MARK ZERO END.
021562  013700  002012                  MOV     L$UNIT,R0               ;GET NUMBER OF LOGICAL UNITS TO RUN.
021566  012701  000001                  MOV     #1,R1                   ;GET INITIAL SIZE OF DRIVE TABLE AND
021572  062701  000015          1$:     ADD     #<D.SIZE>/2,R1          ;ACCUMULATE DRIVE TABLE SIZE.
021576  005300                          DEC     R0                      ;SEE IF ANY MORE LOGICAL UNITS.
021600  001374                          BNE     1$                      ;BRANCH IF NOT, ELSE
021602  004737  012706                  CALL    ALOCM                   ;ALLOCATE ALL DRIVE TABLES TO MEMORY.
                                                                        ; R1 POINTS TO 1ST WORD IN DRIVE TABLE


                                ;
                                ;       INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
                                ;

021606  013737  002140  002150  INIT3:  MOV     FFREE,CTABS             ; STORE START OF CONTROLLER TABLES AND
021614  005077  160330                  CLR     @CTABS                  ; MARK ZEROS END.
021620  005037  002152                  CLR     CTRLRS                  ; CLEAR CONTROLLER COUNT
021624  012701  003434                  MOV     #IPADRS,R1              ; R1 -> IP ADDRESS
021630  012702  000010                  MOV     #8.,R2                  ; GET MAXIMUM # OF CONTROLLERS
021634  005021                  1$:     CLR     (R1)+                   ; CLEAR ENTRY
021636  005302                          DEC     R2                      ; DONE?
021640  001375                          BNE     1$                      ; IF NOT, BRANCH

                                ;
                                ;       BUILD CONTROLLER TABLES
```

```
                                        ;
                                        ;
                                INIT4:  CLR     R5                      ;CLEAR CUSTOMER DATA FLAG
                                        CLR     R2                      ;START WITH LOGICAL UNIT 0
021642  005005
021644  005002
021646  012737  005160  021232            MOV     #5160,BRSAV             ; SAVE DEFAULT FOR BR LEVEL & VECTOR
                                1$:     GPHARD  R2,R0                   ; GET POINTER TO IT'S P-TABLE
                                                                                MOV     R2,R0
021654  010200                                                                 TRAP    C$GPHRD
021656  104442
                                        BNCOMPLETE      16$             ; BRANCH TO 16$ IF NOT AVAILABLE
021660                                                                          BCC     16$
021660  103104
021662  013703  002150                    MOV     CTABS,R3                ; GET ADDRESS OF 1ST CONTROLLER TABLE
021666  005713                  2$:       TST     (R3)                    ; CHECK IF ANY MORE TABLES
021670  001405                            BEQ     6$                      ; BUILD NEW TABLE IF FOUND ZERO WORD
021672  021013                            CMP     (R0),(R3)               ; CHECK IF SAME CSR ADDRESS.
021674                                            ASSUME  C.UADR EQ 0
021674                                            ASSUME  H0.UBA EQ 0
021674  001444                            BEQ     11$                     ; BRANCH IF SO

021676  062703  000052                  5$:       ADD     #C.SIZE,R3              ;POINT TO BEGINNING OF NEXT CONTROLLER
021702  000771                            BR      2$                      ;TABLE IN MEMORY.

                                        ;
                                        ;       BUILD NEW CONTROLLER TABLE
                                        ;

021704  012704  003434                  6$:       MOV     #IPADRS,R4              ;GET BEGINNING OF IP ADDRESS TABLE
021710  020427  003444                  7$:       CMP     R4,#IPADRS+8.           ;SEE IF END OF IP ADDRESS TABLE.
021714  101004                            BHI     9$                      ;BRANCH IF SO, ELSE
021716  005724                            TST     (R4)+                   ;DID WE FIND AN OPEN ENTRY ?
021720  001401                            BEQ     8$                      ;BRANCH IF SO, ELSE
021722  000772                            BR      7$                      ;LOOK AGAIN.

021724  011044                  8$:       MOV     (R0),-(R4)              ;TAKE CSR ADDRESS FROM P-TABLE
                                                                          ;AND STORE IT IN THE IP ADDRESS TABLE.
021726  012701  000025                  9$:       MOV     #<C.SIZE>/2,R1          ;GET # OF ENTRIES IN CONTROLLER TABLE
021732  004737  012706                            CALL    ALOCM                   ;AND ALLOCATE A TABLE TO MEMORY.
                                                                                  ; R0 => 1ST WORD P-TABLE
                                                                                  ; R1 => 1ST WORD IN CONTROLLER TABLE
021736  011021                            MOV     (R0),(R1)+              ; STORE CSR ADDRESS AND
021740  010221                            MOV     R2,(R1)+                ; UNIT NUMBER IN THE CONTROLLER TABLE.
021742  013704  021232                    MOV     BRSAV,R4                ; GET DEFAULT VECTOR & BR LEVEL
021746  162704  000004                    SUB     #4,R4                   ; GET NEXT VECTOR
021752  010437  021232                    MOV     R4,BRSAV                ; SAVE NEXT VECTOR
021756  010421                            MOV     R4,(R1)+                ;STORE IT IN THE CONTROLLER TABLE.
021760  012721  004037                    MOV     #4037,(R1)+             ;THE 'JSR R0' INSTRUCION AND
021764  012721  017342                    MOV     #UDASRV,(R1)+           ;THE ADDRESS OF THE INTERRUPT SERVICE
                                                                          ;ROUTINE IN THE CONTROLLER TABLE.
021770  012704  000020                    MOV     #<C.SIZE-C.FLG>/2,R4    ;GET # OF ENTRIES TO END OF TABLE.
021774  005021                  10$:      CLR     (R1)+                   ;CLEAR REST OF TABLE AND
021776  005304                            DEC     R4                      ;ADD ZERO WORD AT END.
022000  002375                            BGE     10$                     ;LOOP TIL ALL CLEARED
022002  005237  002152                    INC     CTRLRS                  ;KEEP TRACK OF CONTROLLER COUNT

                                        ;
                                        ;       BUILD DRIVE TABLES
                                        ;
```

```
                                                11$:    MOV     DTABS,R1                ;GET ADDRESS OF CURRENT DRIVE TABLE
                                                        ADD     #C.DRO,R3               ; INDEX TO 1ST DRIVE IN TABLE
                                                        MOV     #8.,R4                  ; GET # OF DRIVES PER CONTROLLER
                                                12$:    TST     (R3)                    ;ANY ENTRY TO DRIVE TABLE,
                                                        BEQ     14$                     ;BRANCH IF NOT, ELSE
                                                        CMP     M0.LDR(R0),8(R3)+       ;COMPARE DRIVE NUMBER IN DRIVE TABLE,
                                                        BNE     13$                     ;BRANCH IF DIFFERENT, ELSE
                                                        JMP     MLDRER                  ;FOUND TWO P-TABLES WITH SAME DRIVE.

                                                13$:    DEC     R4                      ; COUNT DRIVES
                                                        BNE     12$                     ; IF FOUR DRIVE TABLES ALREADY EXIST,
                                                        JMP     TOOMER                  ;  THEN REPORT ERROR

                                                14$:    MOV     R1,(R3)                 ;STORE ADDRESS OF DRIVE TABLE IN
                                                                                        ;CONTROLLER TABLE.
                                                        MOV     M0.LDR(R0),(R1)+        ;STORE DRIVE NUMBER AND
                                                        MOV     R2,(R1)+                ;LOGICAL UNIT NUMBER IN DRIVE TABLE.

                                        003202          ADD     #D.SIZE,DTABS           ;NEXT DRIVE TABLE ADDRESS AND
                                                        CLR     @DTABS                  ;MARK ZERO END.
                                                16$:    INC     R2                      ;INCREMENT LOGICAL UNIT NUMBER
                                                        CMP     R2,L$UNIT               ;CHECK IF GOT ALL TABLES
                                                        BLT     1$                      ;IF NOT, GO BACK FOR NEXT, ELSE
                                                        MOV     #1,R1                   ;GET 1 WORD TO TERMINATE ALL CONTROLLER
                                                        CALL    ALOCM                   ;TABLES AND ALLOCATE IT TO MEMORY.

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

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

                                                ;
                                                ;       EXIT INITIALIZE SECTION
                                                ;

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

                                                        EXIT    INIT
                                                                                                TRAP    C$EXIT
                                                                                                .WORD   L10035-.

                                                ;TWO P-TABLES FOR SAME DRIVE
                                003242          MLDRER: MOV TEMP,R5             ;GET CONTROLLER ADDRESS
                                                        ERRSF 2,,ERR002
                                                                                                TRAP    C$ERSF
                                                                                                .WORD   2
                                                                                                .WORD   0
                                                                                                .WORD   ERR002
                                                        DOCLN
                                                                                                TRAP    C$DCLN

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

INITIALIZE SECTION

```
    243
    244 022156  013705  003242          TOOMER: MOV TEMP,R5              ;GET CONTROLLER ADDRESS
    245 022162                                  ERRSF 3.,ERR003
        022162  104454                                                    TRAP    C$ERSF
        022164  000003                                                    .WORD   3
        022166  000000                                                    .WORD   0
        022170  012026                                                    .WORD   ERR003
    246 022172                                  DOCLN
        022172  104444                                                    TRAP    C$DCLN
    247
    248
    249 022174                                  ENDINIT
        022174                                                  L10035:
        022174  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.
                                ;--

                                        BGNAUTO
022176                                                          L$AUTO::
022176
                                        ENDAUTO
022176                                                          L10036:
022176                                                                  TRAP    C$AUTO
022176  104461
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

                                        BGNCLN
022200                                                                  L$CLEAN::

022200  004737  013206                  CALL CLOSEF                    ;CLOSE DATA FILE
022204                                  SETVEC #4,#NXMI,#PRIO7
022204  012746  000340                                                  MOV     #PRIO7,-(SP)
022210  012746  017332                                                  MOV     #NXMI,-(SP)
022214  012746  000004                                                  MOV     #4,-(SP)
022220  012746  000003                                                  MOV     #3,-(SP)
022224  104437                                                          TRAP    C$SVEC
022226  062706  000010                                                  ADD     #10,SP
022232  012703  000010                  MOV     #8.,R3          ; R3 = COUNTER OF ENTRIES
022236  012704  003434                  MOV     #IPADRS,R4      ; R4 -> IP ADDRESS
022242  005714          1$:             TST     (R4)            ; IS THERE AN ENTRY?
022244  001403                          BEQ     2$              ; IF NOT, DONE
022246  005034                          CLR     @(R4)+          ; INIT UDA
022250  005303                          DEC     R3              ; MAKE SURE WE DO NOT EXTEND OVER AREA
022252  001373                          BNE     1$              ; IF NOT DONE, BRANCH
022254                          2$:     CLRVEC  #4
022254  012700  000004                                                  MOV     #4,R0
022260  104436                                                          TRAP    C$CVEC

022262                                  ENDCLN
022262                                                          L10037:
022262  104412                                                          TRAP    C$CLEAN

022264                                  ENDMOD
```

```
                                .SBTTL  TEST 1: DUP PROGRAM DRIVER
                                
022264                                  BGNMOD
                                
022264                                  BGNTST
                                                                        T1::
022264                                  PNTX WNSTRT             ;PRINT WARNING MESSAGE
022264  004137  016432                                                  JSR R1,LPNTX
022270  004565                                                          .WORD WNSTRT
022272  000000                                                          .WORD PNT.CT
022274                                  MANUAL                  ;SEE IF MANUAL INTERVENTION ALLOWED
022274  104450                                                          TRAP    C$MANI
022276                                  BNCOMPLETE T1MODE       ;IF NOT, JUST RUN THE PROGRAM
022276  103020                                                          BCC     T1MODE
022300  005037  003242                  CLR TEMP                ;CLEAR WORD FOR ANSWER
022304                                  GMANIL WNQUES,TEMP,1,YES        ;ASK IF STILL WANT TO RUN
022304  104443                                                          TRAP    C$GMAN
022306  000404                                                          BR      10000$
022310  003242                                                          .WORD   TEMP
022312  000130                                                          .WORD   T$CODE
022314  003630                                                          .WORD   WNQUES
022316  000001                                                          .WORD   1
022320                                                          10000$:
022320  005737  003242                  TST TEMP                        ;LOOK AT ANSWER
022324  001417                          BEQ T1QUIT              ;IF NO, QUIT NOW
022326  005737  003304                  TST DATEO       ;SEE IF ALREADY ASKED FOR DATE
022332  001002                          BNE T1MODE
022334  004737  020574                  CALL DATE       ;IF NOT, GET IT NOW
                                
022340  032737  000003  003200  T1MODE: BIT #SO.FMT,MODE
022346  001164                          BNE T1FMT
022350                                  MANUAL
022350  104450                                                  TRAP    C$MANI
022352                                  BCOMPLETE T1GO
022352  103406                                                  BCS     T1GO
022354                                  ERRSF 10.,ERR010
022354  104454                                                  TRAP    C$ERSF
022356  000012                                                  .WORD   10
022360  000000                                                  .WORD   0
022362  012106                                                  .WORD   ERR010
022364                          T1QUIT: EXIT TST
022364  104432                                                  TRAP    C$EXIT
022366  000362                                                  .WORD   L10040-.
022370  032737  000010  003200  T1GO:   BIT #SO.STR,MODE
022376  001435                          BEQ T1CNS
022400  023727  002012  000001          CMP L$UNIT,#1
022406  001406                          BEQ T1RST
022410                                  ERRSF 9.,.ERR009
022410  104454                                                  TRAP    C$ERSF
022412  000011                                                  .WORD   9
022414  000000                                                  .WORD   0
022416  012074                                                  .WORD   ERR009
022420                                  EXIT TST
022420  104432                                                  TRAP    C$EXIT
022422  000326                                                  .WORD   L10040-.
                                
022424                          T1RST:  PNTF FILNAM
```

```
                                                                                    JSR R1,LPNTF
                                                                                    .WORD FILNAM
                                                                                    .WORD PNT.CT
                        GMANID FILNAQ,FNAME,A,-1,1,10.,,NO      ;GET FILE NAME
                                                                                    TRAP    C$GMAN
                                                                                    BR      10001$
                                                                                    .WORD   FNAME
                                                                                    .WORD   T$CODE
                                                                                    .WORD   FILNAQ
                                                                                    .WORD   -1
                                                                                    .WORD   T$LOLIM
                                                                                    .WORD   T$HILIM
                                                                            10001$:
                        OPEN #FNAME
                                                                                    MOV     #FNAME,R0
                                                                                    TRAP    C$OPEN
                        MOV #-1,FILOPN  ;MARK FLAG AS FILE OPEN
                        BR T1FMT
                T1CNS:  MOV CTABS,R5
                T1SER1: MOV R5,R4
                        ADD #C.DRO,R4
                        MOV #8.,R3
                T1SER2: MOV (R4),R2     ;GET DRIVE TABLE POINTER
                        BEQ T1SERN
                        PNTF SERNUM,D.UNIT(R2),(R5),(R2)
                                                                                    MOV (R2),-(SP)
                                                                                    MOV (R5),-(SP)
                                                                                    MOV D.UNIT(R2),-(SP)
                                                                                    JSR R1,LPNTF
                                                                                    .WORD SERNUM
                                                                                    .WORD PNT.CT
                                ASSUME C.UADR EQ 0
                                ASSUME D.DRV EQ 0
                T1SER3: GMANID SERNQ,TEMP,A,-1,1,20.,,NO ;GET SERIAL NUMBER
                                                                                    TRAP    C$GMAN
                                                                                    BR      10002$
                                                                                    .WORD   TEMP
                                                                                    .WORD   T$CODE
                                                                                    .WORD   SERNQ
                                                                                    .WORD   -1
                                                                                    .WORD   T$LOLIM
                                                                                    .WORD   T$HILIM
                                                                            10002$:
                        MOV #TEMP,R1
                        CLR R0
                T1SER4: TSTB (R1)
                        BEQ T1SER5
                        INC R0
                        CMPB (R1),#'0
                        BLO T1SER7
                        CMPB (R1)+,#'9
                        BLOS T1SER4
                        BR T1SER7
                T1SER5: CMP R0,#20.
                        BLO T1SER8
                        MOV #TEMP,R1
                        MOV #HIGHEST,R0
```

```
     59 022624  105710                  T1SER6: TSTB (R0)
     60 022626  001416                          BEQ T1SER8
     61 022630  122120                          CMPB (R1)+,(R0)+
     62 022632  001774                          BEQ T1SER6
     63 022634  103413                          BLO T1SER8
     64 022636                          T1SER7: PRINTF #SERNX,#HIGHEST
        022636  012746  003320                                          MOV     #HIGHEST,-(SP)
        022642  012746  011601                                          MOV     #SERNX,-(SP)
        022646  012746  000002                                          MOV     #2,-(SP)
        022652  010600                                                  MOV     SP,R0
        022654  104417                                                  TRAP    C$PNTF
        022656  062706  000006                                          ADD     #6,SP
     65 022662  000724                          BR T1SER3
     66 022664  062702  000004          T1SER8: ADD #D.SERN,R2  ;PUT ANSWER INTO DRIVE TABLE
     67 022670  012701  003242                  MOV #TEMP,R1
     68 022674  112122                 T1SER9: MOVB (R1)+,(R2)+
     69 022676  001376                          BNE T1SER9
     70 022700  005303                          DEC R3
     71 022702  001402                          BEQ T1SERN
     72 022704  005724                          TST (R4)+
     73 022706  000700                          BR T1SER2
     74 022710  062705  000052          T1SERN: ADD #C.SIZE,R5
     75 022714  005715                          TST (R5)
     76 022716  001267                          BNE T1SER1
     77 022720  013737  002150  002154  T1FMT:  MOV CTABS,TSTTAB                ;GET FIRST TABLE ADDRESS
     78 022726  013701  002152                  MOV CTRLRS,R1                   ;RUN DM PROGRAM ON ALL CONTROLLERS
     79 022732  004737  013074                  CALL RUNDM                      ;  RUN ALL CONTROLLERS OF ONE TYPE AT ONCE
     80 022736  001402                          BEQ 6$
     81 022740  004737  013224                  CALL RESPDM
     82 022744                          6$:     EXIT TST
        022744  104432                                                  TRAP    C$EXIT
        022746  000002                                                  .WORD   L10040-.
     83 022750                                  ENDTST
        022750                                                  L10040:
        022750  104401                                                  TRAP    C$ETST
     84 022752                                  ENDMOD
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION

                                        BGNMOD

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

                                        BGNHRD
022752                                                                  .WORD L10041-L$HARD/2
022752  000011                                                  L$HARD::
022754

                                ;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:

                                TABLE                                   ;START A TEBLE DEFINITION

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

HARDWARE PARAMETER CODING SECTION

```
      1 022754                                GPRMA   M.UBA,MO.UBA,0,160000,177774,YES        ;BUS ADDRESS
        022754  000031                                                                            .WORD   T$CODE
        022756  022776                                                                            .WORD   M.UBA
        022760  160000                                                                            .WORD   T$LOLIM
        022762  177774                                                                            .WORD   T$HILIM
      2 022764                                GPRMD   M.LDR,MO.LDR,D,-1,0..255..YES   ; DRIVE SELECT NUMBER
        022764  001052                                                                            .WORD   T$CODE
        022766  023012                                                                            .WORD   M.LDR
        022770  177777                                                                            .WORD   -1
        022772  000000                                                                            .WORD   T$LOLIM
        022774  000377                                                                            .WORD   T$HILIM
      3 022776                                ENDMRD
                                                                                                  .EVEN
        022776                                                                          L10041:
      4
      5 022776      103     123     122 M.UBA:  .ASCIZ  \CSR ADDRESS\
      6 023012      104     122     111 M.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.
                                ;--

                                        BGNSFT
023030                                                                  .WORD L10042-L$SOFT/2
023030  000022                                                    L$SOFT::
023032

                                ;FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:

023032                          TABLE                                   ;START A TABLE DEFINITION

023032                          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

023032                          END
```

```
                                        GPRML S.FMT,SO.BIT,SO.FM1,YES   ;REFORMAT?
023032  000130                                                                  .WORD   T$CODE
023034  023247                                                                  .WORD   S.FMT
023036  000001                                                                  .WORD   SO.FM1
                                        XFERT SWEND
023040  017024                                                                  .WORD   T$CODE
                                        GPRML S.NRF,SO.BIT,SO.FM2,YES   ;AGAIN - REFORMAT?
023042  000130                                                                  .WORD   T$CODE
023044  023076                                                                  .WORD   S.NRF
023046  000002                                                                  .WORD   SO.FM2
                                        XFERT SWEND
023050  013024                                                                  .WORD   T$CODE
                                        GPRML S.CNS,SO.BIT,SO.CNS,YES   ;RECONSTRUCT
023052  000130                                                                  .WORD   T$CODE
023054  023326                                                                  .WORD   S.CNS
023056  000004                                                                  .WORD   SO.CNS
                                        XFERT SWEND
023060  007024                                                                  .WORD   T$CODE
                                        GPRML S.RST,SO.BIT,SO.STR,YES   ;RESTORE?
023062  000130                                                                  .WORD   T$CODE
023064  023371                                                                  .WORD   S.RST
023066  000010                                                                  .WORD   SO.STR
                                        XFERT SWEND
023070  003024                                                                  .WORD   T$CODE
                                        DISPLAY S.NOF   ;WARNING
023072  000003                                                                  .WORD   T$CODE
023074  023512                                                                  .WORD   S.NOF
023076                          SWEND:  ENDSFT                                  .EVEN
023076                                                                  L10042:

023076  015     012             S.NRF:  .BYTE 15,12
023100  116     117     124             .ASCII\NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR\
023202  015     012                     .BYTE 15,12
023204  111     116     106             .ASCII\INFORMATION ON THE DISKS.\
023235  015     012                     .BYTE 15,12
023237  101     107     101             .ASCII\AGAIN - \
023247  122     105     106     S.FMT:  .ASCIZ\REFORMAT USING EXISTING BAD SECTOR INFORMATION\
023326  122     105     103     S.CNS:  .ASCIZ\RECONSTRUCT BAD SECTOR INFORMATION\
023371  104     117     040     S.RST:  .ASCII\DO YOU HAVE A FILE ON THE SYSTEM LOAD DEVICE\
023445  015     012                     .BYTE 15,12
023447  040     103     117             .ASCIZ\ CONTAINING BAD SECTOR INFORMATION\
023512  131     117     125     S.NOF:  .ASCIZ\YOU CANNOT PROCEED WITHOUT SUCH A FILE.\
023562  122     105     123             .ASCIZ\RESTART PROGRAM AND SELECT TO REFORMAT OR RECONSTRUCT DISK.\
023656  000                             .BYTE 0
                                        .EVEN

                                        .DSABL AMA
000000                                  .PSECT END
```

```
                                        .SBTTL  PATCH AREA

                                $PATCH::
000000                                  .REPT 40.
        000050                          .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: 589
Work  file writes: 519
Size of work file: 29208 Words  ( 115 Pages)
Size of core pool: 14336 Words  ( 56 Pages)
Operating  system: RT-11 (Under RTEM-11)

Elapsed time: 00:07:56.00
ZUDKA0,ZUDKA0/C=SVC34R.MLB/P:1,ZUDKA0.DOC,ZUDKA0
```

Cross reference table (CREF V05.01)

```
$PATCH 124-3#
ADR      30-10#
ALOCM    56-16#   57-14    116-123  116-174  116-220
ASSEMB   26-8     26-8
BAS      50-14#   83-5      83-5     83-5     84-5     84-5
BASL2    50-12#   84-5
BASL3    50-13#
BASLN    50-16#   83-5      84-5
BASNO    50-11#   83-5      84-5
BIT0     30-10#  122-19
BIT00    30-10    30-10#
BIT01    30-10    30-10#
BIT02    30-10    30-10#
BIT03    30-10    30-10#
BIT04    30-10    30-10#
BIT05    30-10    30-10#
BIT06    30-10    30-10#
BIT07    30-10    30-10#
BIT08    30-10    30-10#
BIT09    30-10    30-10#
BIT1     30-10#   33-26    41-22    122-20
BIT10    30-10#
BIT11    30-10#
BIT12    30-10#
BIT13    30-10#
BIT14    30-10#
BIT15    30-10#   41-15    42-12     59-27    62-20    70-28    75-15    89-29
BIT2     30-10#   33-27    41-23    122-22
BIT3     30-10#   33-28    41-24    122-23
BIT4     30-10#   33-29    41-26
BIT5     30-10#   41-29
BIT6     30-10#   41-30
BIT7     30-10#   41-32
BIT8     30-10#
BIT9     30-10#
BLDC0    97-22    97-24#
BLDC1    97-26#   97-28
BLDCMD   61-49    64-14     64-44     95-2     97-15#
BOE      30-10#
BRSAV   114-104#  116-145*  116-179  116-181*
C$AU     26-8#
C$AUTO   26-8#   117-12
C$BRK    26-8#    58-12     61-8     100-21   107-12   110-27
C$BSEG   26-8#
C$BSUB   26-8#
C$CEFG   26-8#
C$CLCK   26-8#   116-93    116-95
C$CLEA   26-8#   118-21
C$CLOS   26-8#    60-12     73-19
C$CLP1   26-8#
C$CVEC   26-8#    58-22    107-30   118-19
C$DCLN   26-8#    55-8      58-30    116-69   116-240  116-246
C$DODU   26-8#
C$DRPT   26-8#
C$DU     26-8#
C$EDIT   26-8#    26-34
```

Cross reference table (CREF V05.01)

```
C$ERDF  26-80   58-28   61-24   62-36   63-21   64-8    68-22   70-32   96-3    100-29  101-5   105-36  105-51  106-12
        107-33  109-37  110-36  111-5
C$ERHR  26-80
C$ERRO  26-80
C$ERSF  26-80   55-7    116-239 116-245 119-21  119-27
C$ERSO  26-80
C$ESCA  26-80
C$ESEG  26-80
C$ESUB  26-80
C$ETST  26-80   119-83
C$EXIT  26-80   116-235 119-22  119-28  119-82
C$GETB  26-80   73-24
C$GETW  26-80
C$GMAN  26-80   114-1   119-10  119-31  119-44
C$GPHR  26-80   116-146
C$GPLO  26-80
C$GPRI  26-80
C$INIT  26-80   116-249
C$INLP  26-80
C$MANI  26-80   119-7   119-19
C$MEM   26-80   116-109
C$MSG   26-80   53-16   53-20   53-24   53-28   53-32   53-36   53-40   53-44   53-48   53-60   53-65   53-77   53-81
        53-85   53-89   53-93   53-98   53-102  53-106  53-110  53-114  53-118  53-122
C$OPEN  26-80   73-20   119-32
C$PNTB  26-80   91-14
C$PNTF  26-80   91-12   119-64
C$PNTS  26-80   91-18
C$PNTX  26-80   91-16
C$QIO   26-80
C$RDBU  26-80
C$REFG  26-80   116-47  116-53  116-59  116-66
C$RESE  26-80   26-80
C$REVI  26-80   26-34
C$RFLA  26-80
C$RPT   26-80
C$SEFG  26-80
C$SPRI  26-80   116-233
C$SVEC  26-80   58-11   94-20   107-27  116-106 118-11
C$TPRI  26-80
C.DR0   41-340  62-15   75-12   116-197 119-37
C.DR1   41-350
C.DR2   41-360
C.DR3   41-370
C.DR4   41-380
C.DR5   41-390
C.DR6   41-400
C.DR7   41-410
C.FLG   41-210  59-23*  61-10   61-13   61-15   61-38   61-43   61-47*  62-12*  63-6    63-26   64-35*  64-39*  64-40
        64-47*  64-50*  64-62*  64-63   64-69*  64-70   95-12*  95-13*  98-21*  100-17  101-1*  116-186
C.JAD   41-200
C.JSR   41-190  94-19
C.PRI   41-450  64-65   64-67   64-72*  64-73*
C.REF   41-460  63-19   98-16*  98-17
C.RING  41-330  53-138  57-15*  61-9    64-29   97-16   98-15   105-20  106-3   106-20  107-20
C.SIZE  41-480  59-32   62-5    116-157 116-173 116-186 119-74
C.TO    41-420  61-36   64-55   100-14  100-27  110-22  110-33
```

Cross reference table (CREF V05.01)

```
C.TOH    41-43#   61-33    100-24   110-30
C.TOT    41-44#   61-29     64-53    94-14*   94-15*
C.UADR   41-12#   76-16    107-25   116-152  119-42
C.UNIT   41-13#   59-24     59-25    61-12
C.VEC    41-16#   94-16    107-14
CALR4    80-6     83-5#
CALR5    80-7     84-5#
CALR6    80-8     85-4#
CALR7    80-9     86-4#
CALR8    80-10    87-4#
CALRE    80-3     80-4     80-5     82-4#
CF.576   37-39#
CF.ATN   37-34#
CF.MSC   37-35#
CF.OTH   37-36#
CF.SHD   37-38#
CF.THS   37-37#
CLOSEF   60-10#  118-10
CLR3FL   99-22#   99-24
CLRBUF   64-16    64-46    99-16#
CON.A    78-12#   81-21
CON.A1   78-13#   78-15
CON.A2   78-17    78-19#
CON.D    78-23#   81-21
CON.H    78-29#   81-21
CON.N    79-9#    81-21
CON.N1   79-10#   79-12
CON.O    79-3#    81-21
CON.QU   78-3#    78-7     81-21
CON.QX   78-5     78-8#
CON.R    79-17#   81-21
CON.R1   79-19    79-23#
CON.S    79-29#   81-21
CON.S1   79-30#   79-32
CPNT     74-18    74-21    76-21    76-23    78-6     78-13    79-10    79-30    83-7     84-7     90-13    90-22    91-5#    113-27
        113-32   113-35
CPNTX    91-13    91-15    91-17    91-19#
CR       30-12#   74-21    76-21    76-23    79-10    83-7     84-7     91-8
CT.AVL   41-15#   59-27
CT.BRL   41-18#
CT.CMD   41-23#   61-15    95-12    98-21
CT.MSG   41-24#   61-13    64-39    64-62   100-17   101-1    103-19
CT.REQ   41-26#   63-6     63-26    64-35    64-40    64-47    64-50    95-12
CT.RN    41-22#   61-10    95-13
CT.STA   41-29#   61-38    61-46    64-39    64-62
CT.TM1   41-30#   61-42    61-43    64-39    64-70
CT.TM2   41-32#   64-39    64-63    64-69
CT.UNT   41-14#
CT.VEC   41-17#   94-17   107-15
CTABS    46-12#  116-130* 116-131* 116-148  119-35   119-77
CTRLRS   46-13#  116-132* 116-190* 119-78
D.DRV    42-9#    69-36    76-17    119-43
D.SERN   42-13#   68-17   119-66
D.SIZE   42-15#  116-120  116-214
D.UNIT   42-10#   62-19    62-25*   75-14    76-15    119-41
DAS1    114-8    114-13#
```

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
F$MW    26-8#    28-10    28-14
F$INIT  26-8#   116-45   116-249
F$JMP   26-8#   116-235  119-22   119-28   119-82
F$MOD   26-8#    26-26    29-16    30-3    114-106  115-3    118-23   119-3    119-84   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    119-83
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#  119-30
FILNAQ  49-4#   119-31
FILOPN  46-20#   60-10    60-13*   119-33*
FMEM    46-10#  116-226*
FMEMS   46-11#  116-227*
FMERR   55-7#    56-18
FNAME   46-44#   73-20    119-31   119-32
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$NO    26-8#   119-31   119-44
G$OFFS  26-8#   114-1    119-10   119-31   119-44   121-1    121-2    123-1    123-3    123-5    123-7
G$OFSI  26-8#   114-1    119-10   119-31   119-44   121-1    121-2    123-1    123-3    123-5    123-7
G$PRMA  26-8#   121-1
G$PRMD  26-8#   114-1    119-31   119-44   121-2
G$PRML  26-8#   119-10   123-1    123-3    123-5    123-7
G$RADA  26-8#   114-1    119-31   119-44
G$RADB  26-8#
G$RADD  26-8#   121-2
G$RADL  26-8#   119-10   123-1    123-3    123-5    123-7
G$RADO  26-8#   121-1
G$XFER  26-8#   123-2    123-4    123-6    123-8
G$YES   26-8#   114-1    119-10   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 V05.01)

Cross reference table (CREF V05.01)

```
INFOX    70-28#   70-41
INITO   116-64    116-68   116-71#
INIT1   116-51    116-57   116-78#
INIT2   116-116#
INIT3   116-130#
INIT4   116-143#
INIT6   116-226#
INITQT  116-69#   116-86
INITXX  116-71    116-233#
IPADRS   47-34#   58-14    116-133  116-164  116-165  118-13
IREST    33-27#  116-56
ISR      30-10#
ISTRT    33-28#  116-50
ISTRTM   33-29#  116-62
IXE      30-10#
J$JMP    26-8#
KW.BRL   46-36#  116-102*
KW.CSR   46-35#   61-31    76-19   100-22   110-28   112-8*   113-13   116-97*  116-101* 116-107*
KW.EL    46-39#   61-33    61-36   100-24   100-27   104-26   104-27   104-28   110-30   110-33   112-6*   112-7*   113-16   113-17
        116-91*  116-92*
KW.MZ    46-38#  104-16   113-18   116-104*
KW.VEC   46-37#  116-103* 116-106
KW11I   112-5#   116-106
KWOUT.  112-8    116-89#  116-107
L$ACP    26-34#
L$APT    26-34#
L$AUT    26-34#
L$AUTO   26-34   117-10#
L$CCP    26-34#
L$CLEA   26-34   118-8#
L$CO     26-34#
L$DEPO   26-34#
L$DESC   25-34    48-16#
L$DESP   26-34#
L$DEVP   26-34#
L$DISP   26-34    27-9#
L$DLY    26-34#
L$DTP    26-34#
L$DTYP   26-34#
L$DUT    26-34#
L$DVTY   26-34    48-12#
L$EF     26-34#
L$ENVI   26-34#
L$ETP    26-34#
L$EXP1   26-34#
L$EXP4   26-34#
L$EXP5   26-34#
L$HARD   26-34   120-14   120-14#
L$HIME   26-34#
L$HPCP   26-34#
L$HPTP   26-34#
L$HW     26-34    28-10    28-10#
L$ICP    26-34#
L$INIT   26-34   116-45#
L$LADP   26-34#
L$LAST   26-34   124-8#   125-10
```

Cross reference table (CREF V05.01)

```
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  119-25
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  119-22   119-28   119-82   119-83#
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 V05.01)

```
LOADDM  59-29   62-30   94-13#
LOADE1  95-11   96-3#
LOADER  94-23   95-9    96-4#
LOE     30-10#
LOG     34-32#
LOT     30-10#
LPNT    70-40   92-6    92-8    92-10   92-12#  113-24  113-28  113-33
LPNTB   53-15   53-19   53-23   53-27   53-31   53-35   53-39   53-43   53-47   53-55   53-58   53-64   53-68   53-72
        53-76   53-80   53-84   53-88   53-92   53-96   53-109  53-113  53-117  53-121  53-124  53-130  53-137  53-141
        83-5    84-5    92-7#
LPNTF   76-15   77-41   79-23   82-4    92-5#   114-31  116-98  119-30  119-41
LPNTS   92-11#
LPNTX   92-9#   119-6
MD.CMP  37-4#
MD.CMB  37-22#
MD.ERR  37-6#
MD.EXP  37-5#
MD.FEU  37-16#
MD.IMF  37-20#
MD.NXU  37-18#
MD.PRI  37-23#
MD.RIP  37-19#
MD.SCH  37-7#
MD.SCL  37-8#
MD.SEC  37-9#
MD.SEQ  37-14#
MD.SER  37-10#
MD.SPD  37-15#
MD.SSH  37-11#
MD.SMP  37-21#
MD.VOL  37-17#
MD.WBN  37-12#
MD.WBV  37-13#
MESG    70-27   70-39   74-13#
MESSG   50-9#   76-15
MLDRER  116-203 116-238#
MODE    46-26#  69-53   69-59   116-87*  119-17  119-23
MON1    114-17# 114-30
MON2    114-19  114-21#
MON3    114-22  114-24#
MON4    114-25  114-27#
MON5    114-28  114-33#
MON6    114-36  114-39#
MONTHS  47-7#   114-16
MSCP    34-31#
MSGPKL  53-141# 53-144
MSGPKT  53-97   53-137#
NCONF   77-38   77-43#
NCONS   77-37#  77-40
NOCLOC  50-10#  116-98
NULL    46-43#
NXMAD   46-22#  58-10*  58-18   102-12*  107-26*  107-31
NXMI    58-11   102-10# 107-27  118-11
O$APTS  26-8#   26-34
O$AU    26-8#   26-34
O$BGNR  26-8#   26-34
```

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
PRIO6    30-10#
PRIO7    26-34    30-10#   58-11    94-20    107-27   116-106  118-11
PS       91-18#   92-11
PTYPE    46-41#   83-8*    84-8*    91-11    92-5*    92-7*    92-9*    92-11*
PX       83-8     84-8     91-16#   92-9
QUE0     68-19    68-26#
QUE7     68-21    68-27#
QUEL     68-28#   68-30
QUEST    65-3     68-16#
RAFMT    46-15    46-16
RESET    58-9#    116-108
RESPCT   61-8#    62-4     62-7     62-31
RESPDM   61-6#    62-8     119-81
RG.FLG   34-4#    64-74    98-18
RG.OWN   34-3#    61-50    64-74    98-18    98-19
RNTIM    50-5#    113-24
RNTIM1   50-6#    113-28
RNTIM2   50-7#    113-33
RNTIME   76-22    83-6     84-6     113-13#
RNTIMX   113-14   113-35#
RSPORP   61-25    62-12#   62-37    63-22    64-9     64-27
RSPOSP   64-26    65-3#    65-9
RSPERR   63-10    64-58#
RSPERW   63-15    63-21#   64-59
RSPIN    61-14    63-5#
RSPMAR   63-7     63-9#
RSPNTO   61-30    61-32    61-35    61-37    61-53#
RSPNXT   61-11    62-3#    62-33    64-57    64-75
RSPOU    63-27#
RSPOU2   64-41    64-50#
RSPOU3   64-48    64-51#
RSPOU4   61-52    64-53#
RSPOUT   61-17    63-27    64-34    64-39#
RSPPT2   64-3#
RSPPT3   64-7     64-11#
RSPPTW   63-20    63-26#
RSPTM    61-23    61-29#
RSPTMO   61-34    61-38#
RSPTOE   61-39    61-41    62-36#   64-61    64-76
RUNDM    59-15#   119-79
S$LSYM   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#   114-1    114-1    114-1    114-1#   116-249#  117-12#  118-21#  119-10   119-10   119-10   119-10#  119-31   119-31
         119-31   119-31#  119-44   119-44   119-44   119-44#  119-83#  121-3#   123-10#
S.CNS    123-5    123-19#
S.FMT    123-1    123-18#
S.NOF    123-9    123-23#
S.NRF    123-3    123-12#
S.RST    123-7    123-20#
SA.A2    32-21#
SA.BST   33-22#
SA.CM1   32-35#
SA.CMD   32-30#
SA.CME   32-40#
SA.CNT   33-16#   105-42
SA.CTP   32-43#
```

Cross reference table (CREF V05.01)

```
ST.MST  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
```

Cross reference table (CREF V05.01)

```
        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
        116-109 116-109 116-146 116-146 116-146 116-146 116-146 116-146 116-147 116-147 116-147 116-233 116-233 116-233
        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  119-7   119-7   119-7   119-8   119-8   119-8   119-10  119-10  119-10  119-10  119-10  119-10
        119-10  119-10  119-10  119-10  119-10  119-10  119-10  119-10  119-10  119-10  119-10  119-10  119-19  119-19
        119-19  119-20  119-20  119-20  119-21  119-21  119-21  119-21  119-21  119-21  119-21  119-21  119-21  119-21
        119-21  119-21  119-22  119-22  119-22  119-22  119-22  119-22  119-27  119-27  119-27  119-27  119-27  119-27
        119-27  119-27  119-27  119-27  119-27  119-27  119-28  119-28  119-28  119-28  119-28  119-28  119-31  119-31
        119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31
        119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-31  119-32  119-32  119-32  119-32  119-32  119-32
        119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44
        119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-44  119-64  119-64  119-64  119-64
        119-64  119-64  119-64  119-64  119-64  119-64  119-64  119-64  119-64  119-64  119-64  119-64  119-64  119-64
        119-82  119-82  119-82  119-82  119-82  119-82  119-83  119-83  119-83  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   123-5   123-6   123-6   123-6   123-7   123-7   123-7
        123-7   123-7   123-7   123-7   123-7   123-7   123-8   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  119-10  119-31
        119-44  119-83  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
         53-116#  53-118  53-120#  53-122
```

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
GPHARD  116-146
GPRMA   121-1
GPRMD   114-1     114-1#    119-31    119-31#   119-44    119-44#   121-2
GPRML   119-10    119-10#   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#  119-22    119-22#   119-28    119-28#   119-82    119-82#
M$CNTO  114-1     114-1#    119-10    119-10#   119-31    119-31#   119-44    119-44#   121-1     121-1#    121-2     121-2#    123-1     123-1#
        123-3     123-3#    123-5     123-5#    123-7     123-7#
M$COUN   91-12    91-12#    91-14     91-14#    91-16     91-16#    91-18     91-18#    119-64    119-64#
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#
        119-83    119-83#   119-84    119-84#   121-3     121-3#    123-10    123-10#   124-10    124-10#   125-3     125-3#
M$DEFA  114-1     114-1#    119-10    119-10#   119-31    119-31#   119-44    119-44#   121-1     121-1#    121-2     121-2#    123-1     123-1#
        123-3     123-3#    123-5     123-5#    123-7     123-7#
M$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#   119-83#   119-84#   121-3#    123-10#   124-10#
M$ERRI   55-7     55-7#     58-28     58-28#    61-24     61-24#    62-36     62-36#    63-21     63-21#    64-8      64-8#     68-22     68-22#
         70-32    70-32#    96-3      96-3#     100-29    100-29#   101-5     101-5#    105-36    105-36#   105-51    105-51#   106-12    106-12#
        107-33    107-33#   109-37    109-37#   110-36    110-36#   111-5     111-5#    116-239   116-239#  116-245   116-245#  119-21    119-21#
        119-27    119-27#
M$EXCP  114-1     114-1     114-1#    119-31    119-31    119-31#   119-44    119-44    119-44#   121-1     121-1     121-1#    121-2     121-2
        121-2#
M$EXIT  116-235   116-235#  119-22    119-22#   119-28    119-28#   119-82    119-82#
M$EXSE  116-235#  119-22#   119-28#   119-82#
M$EXTJ  116-235#  119-22#   119-28#   119-82#
M$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#    119-10    119-10#   119-31    119-31#   119-44    119-44#   119-83    119-83#
        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$GENB  114-1     114-1#    119-10    119-10#   119-31    119-31#   119-44    119-44#
M$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#
```

Cross reference table (CREF V05.01)

	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#
	119-83	119-83#	119-84	119-84#	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$GETT	116-235#	119-22#	119-28#	119-82#	123-2	123-2#	123-4	123-4#	123-6	123-6#	123-8	123-8#		
M$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#	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#

Cross reference table (CREF V05.01)

	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#	119-7	119-7#	119-8	119-8#	119-10	119-10	119-10	119-10	119-10	119-10	119-10#
	119-10#	119-10#	119-10#	119-19	119-19#	119-20	119-20#	119-21	119-21	119-21	119-21	119-21#	119-21#	119-21#
	119-21#	119-21#	119-22	119-22	119-22#	119-22#	119-27	119-27	119-27	119-27	119-27#	119-27#	119-27#	119-27#
	119-27#	119-28	119-28	119-28#	119-28#	119-31	119-31	119-31	119-31	119-31	119-31	119-31	119-31	119-31#
	119-31#	119-31#	119-31#	119-32	119-32	119-32#	119-32#	119-44	119-44	119-44	119-44	119-44	119-44	119-44
	119-44	119-44#	119-44#	119-44#	119-44#	119-64	119-64	119-64	119-64	119-64	119-64	119-64#	119-64#	119-64#
	119-64#	119-64#	119-82	119-82	119-82#	119-82#	119-83	119-83#	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$GNLS	114-1	114-1#	119-10	119-10#	119-31	119-31#	119-44	119-44#						
M$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#	119-83	119-83#	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#	119-7#	119-10	119-10#	119-10#	119-19#	119-21#	119-22#	119-27#	119-28#	119-31	119-31#
	119-31#	119-32#	119-44	119-44#	119-44#	119-64#	119-82#	119-83#	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$LDRO	58-22	58-22#	73-20	73-20#	107-30	107-30#	116-47	116-47#	116-53	116-53#	116-59	116-59#	116-66	116-66#
	116-93	116-93#	116-95	116-95#	116-146	116-146#	116-233	116-233#	118-19	118-19#	119-32	119-32#		
M$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#
	119-83	119-83#	119-84	119-84#	121-3	121-3#	123-10	123-10#	124-10	124-10#				
M$PRIN	91-12	91-12#	91-14	91-14#	91-16	91-16#	91-18	91-18#	119-64	119-64#				
M$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 V05.01)

Cross reference table (CREF V05.01)