RA80

PDP-11 UDA DRV FMTR
CZUDEBO

AH-S837B-MC
FICHE 1 OF 1

```
.REM ~
```

IDENTIFICATION

PRODUCT CODE: AC-S836B-MC

PRODUCT NAME: CZUDEBO UDA DISK FORMATTER

PRODUCT DATE: 14-APR-82

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: DALE KECK

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

This program will format any disk drive connected to a UDA-50 disk controller. There are three ways to format a disk with this program:

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

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

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

The format operation is performed by a Diagnostic Utilities and Protocol (DUP) program loaded into the UDA-50 disk controller. The host program simply downline loads the DUP program in the UDA-50 and monitors its execution. The DUP program obtains parameters from the host program (eg. drive number and format mode) and requests the host program to print error and summary messages. The DUP program is also commonly called a "diagnostic machine" (DM) program.

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

In REFORMAT and RECONSTRUCT modes, any number of disk drives may be selected. A UDA-50 can only format one disk at a time, so each disk on a UDA-50 will be selected sequentially. If the disk drives to be formatted are connected to different UDA-50s, all UDA-50s will be run simultaneously. For example, lets assume three units are selected for formatting in the hardware questions, units 1 and 2 are connected to one UDA-50 and unit 3 is connected to a different UDA-50 (Unibus addresses are different). This program will automatically start simultaneous format operations on units 1 and 3. When unit 1 finishes (or errors), unit 2 will be started. After units 2 and 3 are finished, the program stops.

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

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

1.2 SYSTEM REQUIREMENTS

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

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

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

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

2.2 SWITCHES

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

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

Example of switch usage:

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

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

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

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

2.3 FLAGS

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

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

*Error messages are described in section 3.1

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

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

2.4 HARDWARE QUESTIONS

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

UNIBUS ADDRESS OF UDA (O) 172150 ?

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

VECTOR (O) 154 ?

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

BR LEVEL (D) 5 ?

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

UNIBUS BURST RATE (D) 63 ?

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

DRIVE NUMBER (D) 0 ?

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

2.5 SOFTWARE QUESTIONS

After you have answered the hardware questions or after a RESTART or CONTINUE command, the Runtime Services will ask for software parameters. You will be prompted by "CHANGE SW (L) ?" If you wish to change any parameters, answer by typing "Y". The software questions and the default values are described in the next paragraphs. You may change the default values with the SETUP utility.

REFORMAT USING EXISTING BAD SECTOR INFORMATION (L) Y ?

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

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

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

RECONSTRUCT BAD SECTOR INFORMATION (L) Y ?

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

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

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

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

2.6 MANUAL INTERVENTION QUESTIONS

Once the program is started, the date will be asked for in the format used by the XXDP+ system.

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

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

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

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

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

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

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

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

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

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

2.7 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.8 QUICK START-UP PROCEDURE

To start-up this program:

1. Boot XXDP+
2. Give the date and answer the LSI and 50HZ (if there is a clock) questions
3. Type "R ZUDEBO"
4. Type "START"
5. Answer the "CHANGE HW" question with "Y"
6. Answer all the hardware questions
7. Answer the "CHANGE SW" question with "N"

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

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

```
DR>STA

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
UNIBUS ADDRESS OF UDA (O)  172150 ?
VECTOR (O)  154 ?
BR LEVEL (D) 5 ?
UNIBUS BURST RATE (D)  63 ?
DRIVE NUMBER (D)  0 ? 0,1

CHANGE SW (L)  ? N
```

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

UNIT 0 UDA AT 172150 DRIVE 0   RUNTIME 0:12:20
Format completed
     2 Revectored LBNS
     2 Primary revectored LBNS
     0 Secondary/tertiary revectored LBNS
     0 Bad blocks in the RCT area due to data errors
     0 Bad blocks in the DBN area due to data errors
     0 Bad blocks in the XBN area due to data errors
     2 Blocks retried on the check pass
FCT used successfully
UNIT 1 UDA AT 172150 DRIVE 1   RUNTIME 0:25:18
Format completed
     6 Revectored LBNS
     5 Primary revectored LBNS
     1 Secondary/tertiary revectored LBNS
     0 Bad blocks in the RCT area due to data errors
     0 Bad blocks in the DBN area due to data errors
     0 Bad blocks in the XBN area due to data errors
     4 Blocks retried on the check pass
FCT used successfully
CZUDE EOP    1
    0 CUMULATIVE ERRORS
DR>
```

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

```
DR>STA

CHANGE HW (L)  ? N

CHANGE SW (L)  ? Y

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

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

  RUNTIME 0:12:45
Format completed
     2 Revectored LBNS
     2 Primary revectored LBNS
     0 Secondary/tertiary revectored LBNS
     0 Bad blocks in the RCT area due to data errors
     0 Bad blocks in the DBN area due to data errors
     0 Bad blocks in the XBN area due to data errors
     2 Blocks retried on the check pass
FCT used successfully
CZUDE EOP    1
    0 CUMULATIVE ERRORS
DR>
```

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

There are three levels of error messages that may be issued by a diagnostic: general, basic and extended. General error messages are always printed unless the "IER" flag is set (section 2.3). The general error message is of the form:

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

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

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

Device fatal errors (DVC FTL ERR) are used to report errors that are fatal to the device (may be either a UDA-50 or disk drive). Testing stops on that device for the remainder of the current test.

Basic error messages are messages that contain some additional information about the error. These are always printed unless the "IER" or "IBE" flags are set (section 2.3). These messages are printed after the associated general message.

Extended error messages contain supplementary error information such as register contents or good/bad data. These are always printed unless the "IER", "IBE" or "IXE" flags are set (section 2.3). These messages are printed after the associated general error message and any associated basic error messages.

The general and basic error messages from this diagnostic are always one line each. The basic message defines what program detected the error, the UDA-50 being used and the time of the error:

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

The host program (PDP-11) detected the error. UDA AT xxxxx identifies the address of the UDA-50 being tested. It may be omitted if the error is not specific to one UDA-50.

Sample error message:

```
CZUDE DVC FTL ERR  00021 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx      -  general message
HOST PROGRAM  UDA AT 172150  RUNTIME 0:00:12                        -  basic message
UDA RESIDENT DIAGNOSTICS DETECTED FAILURE                          \
  UDASA CONTAINS 104041                                             :-  extended message
REPLACE UDA MODULE M7161                                           /
```

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

3.2 SPECIFIC ERROR MESSAGES

3.2.1 HOST PROGRAM ERROR MESSAGES

Following is a list of the error messages that may be printed by the diagnostic program. In the list, some of the numbers that may vary with execution or program version are shown as "xxx". These include program counters and runtime. Other numbers, such as unit number, drive number, UDA-50 address and data in registers are filled with sample numbers. Additional information about the error may follow the error message.

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

When the hardware questions were answered, two units were selected with the same UNIBUS address but with a different vector, BR level or burst rate. A single UDA-50 can have only one vector, BR level or burst rate. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

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

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

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

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

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

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

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

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

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

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

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

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

```
00020   CZUDE DVC FTL ERR  00020 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        MEMORY ERROR TRYING TO READ UDA REGISTERS
        CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7161
        OR UNIBUS
        OR REPLACE UDA MODULE M7161

                A non-existant memory error occurred when the host program
                tried to access the UDAIP and UDASA registers.  The UDA
                is at another address (check the UNIBUS selection switches)
                or module M7161 is broken or the UNIBUS is broken.

00021   CZUDE DVC FTL ERR  00021 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        UDA RESIDENT DIAGNOSTICS DETECTED FAILURE
          UDASA CONTAINS  105154
        REPLACE UDA MODULE M7162

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

                        104000 - Fatal sequncer 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.  M7161 is the Unibus interface board.
                M7162 is the SDI interface board.
```

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

                The UDA did not respond as expected during the
                initialization sequence which communicates using data
                in the UDASA register. A normal response from the UDA
                contains either a STEP bit or an ERROR bit defined as
                follows:
                        Bit 15 (100000)   Error bit
                        Bit 14 (040000)   Step 4 bit
                        Bit 13 (020000)   Step 3 bit
                        Bit 12 (010000)   Step 2 bit
                        bit 11 (004000)   Step 1 bit
                The expected step bit nor the error bit set within the
                expected time.

00023   CZUDE DVC FTL ERR  00023 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION
        6 WORDS WERE TO BE CLEARED STARTING AT ADDRESS 040644
        FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):
              ADDRESS    CONTENTS
               040644     000010
               040650     000010
               040652     000010
        REPLACE UDA MODULE M7161

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

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

                For better testing, the host can test the PURGE and POLE
                mechanism of the UDA.  To do so the host sets bit15 of the
                step 3 data and sends the data to the UDA.  The UDA must
                go to zero and wait for the purge and pole.  If the UDA
                never went to zero, then error message 00024 is displayed.
                The UDA may have a bad M7161 module or the UNIBUS may
                be broken.
```

00025 CZUDE DVC FTL ERR 00025 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION
UDASA EXPECTED 004400
UDASA CONTAINS 004000
REPLACE UDA MODULE M7161

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3.2.2 DUP PROGRAM ERROR MESSAGES

Error messages returned by the UDA formatter are as follows:

GET STATUS failure

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

SDI send error

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

Unsuccessful SDI command

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

SDI receive error

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

UNIBUS read error

This is caused by one of two problems. While trying to read an overlay into the UDA buffer memory, the formatter came across a nonexistent memory error. Or, there was a failure while downline loading the bad block information. There may be something wrong with the UNIBUS or the UDA module M7161.

Formatter initialization error

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

Nonexistent unit number

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

DBN/XBN format error (drive FORMAT command failed)

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

FCT does not have enough good copies of each block

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

SEEK error

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

RCT does not have enough good copies of each block

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

LBN format error (drive FORMAT command failed)

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

FCT write error

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

RCT read error

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

RCT write error

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

RCT full

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

FCT read error

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

FCT downline-load error

The formatter was led to believe that a bad block information file was larger than it really was. There may be a UNIBUS or M7161 problem.

Drive init timeout

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

Illegal response to start-up question

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

An example of how the errors are presented is below:

```
RUNTIME  0:00:18
Nonexistant unit number
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

The DUP program issues the following messages upon normal completion:

Format completed

n 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 in message #2 which were primary revectors.

n Secondary/teritary revectored LBNS

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

n Bad blocks in the RCT area due to data errors

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

n Bad blocks in the DBN area due to data errors

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

n Bad blocks in the XBN area due to data errors

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

n Blocks retried on the check pass

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

FCT used successfully or
FCT was not used

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

An example of how the messages are presented is below.

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

5.0 TEST SUMMARIES

There is only one test in this program - Test #1. Its only purpose is to load and run the format program in a UDA-50.

```
                                .SBTTL  PROGRAM HEADER

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

                                        POINTER BGNSW, BGNSFT, BGNSETUP

                                        HEADER  CZUDE,B,P,0,1,PRI07     ;FIELD SERVICE
002000                                                                  L$NAME::
002000  103                                                                     .ASCII /C/
002001  132                                                                     .ASCII /Z/
002002  125                                                                     .ASCII /U/
002003  104                                                                     .ASCII /D/
002004  105                                                                     .ASCII /E/
002005  000                                                                     .BYTE   0
002006  000                                                                     .BYTE   0
002007  000                                                                     .BYTE   0
002010                                                                  L$REV::
002010  102                                                                     .ASCII  /B/
002011                                                                  L$DEPO::
002011  120                                                                     .ASCII  /P/
002012                                                                  L$UNIT::
002012  000001                                                                  .WORD   T$PTHV
002014                                                                  L$TIML::
002014  000000                                                                  .WORD   0
002016                                                                  L$HPCP::
002016  021064                                                                  .WORD   L$HARD
002020                                                                  L$SPCP::
002020  021252                                                                  .WORD   L$SOFT
002022                                                                  L$HPT?::
002022  002130                                                                  .WORD   L$HW
002024                                                                  L$SPTP::
002024  002144                                                                  .WORD   L$SW
002026                                                                  L$LADP::
002026  000124'                                                                 .WORD   L$LAST
002030                                                                  L$STA::
002030  000000                                                                  .WORD   0
002032                                                                  L$CO::
002032  000000                                                                  .WORD   0
002034                                                                  L$DTYP::
002034  000001                                                                  .WORD   1
002036                                                                  L$APT::
002036  000000                                                                  .WORD   0
002040                                                                  L$DTP::
002040  002124                                                                  .WORD   L$DISPATCH
002042                                                                  L$PRIO::
002042  000340                                                                  .WORD   PRI07
002044                                                                  L$ENVI::
002044  000000                                                                  .WORD   0
002046                                                                  L$EXP1::
002046  000000                                                                  .WORD   0
002050                                                                  L$MREV::
002050      003                                                                 .BYTE   C$REVISION
002051      003                                                                 .BYTE   C$EDIT
```

PROGRAM HEADER

```
002052
002052  000000                  L$EF::
                                        .WORD   0
002054  000000                  L$SPC::
                                        .WORD   0
002056
002056  000000                  L$DEVP::
                                        .WORD   L$DVTYP
002060
002060  003442                  L$REPP::
                                        .WORD   0
002062
002062  000000                  L$EXP4::
                                        .WORD   0
002064
002064  000000                  L$EXP5::
                                        .WORD   0
002066
002066  000000                  L$AUT::
                                        .WORD   0
002070
002070  000000                  L$DUT::
                                        .WORD   0
002072
002072  000000                  L$LUN::
                                        .WORD   0
002074
002074  000000                  L$DESP::
                                        .WORD   L$DESC
002076
002076  003464                  L$LOAD::
                                        EMT     E$LOAD
002100
002100  104035                  L$ETP::
                                        .WORD   0
002102
002102  000000                  L$ICP::
                                        .WORD   L$INIT
002104
002104  017036                  L$CCP::
                                        .WORD   L$CLEAN
002106
002106  020454                  L$ACP::
                                        .WORD   L$AUTO
002110
002110  020452                  L$PRT::
                                        .WORD   L$PROT
002112
002112  017030                  L$TEST::
                                        .WORD   0
002114
002114  000000                  L$DLY::
                                        .WORD   0
002116
002116  000000                  L$HIME::
                                        .WORD   0
002120
002120  000000
```

```
                                .SBTTL  DISPATCH TABLE

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

002122                          DISPATCH 1
002122  000001                                                 .WORD   1
002124                                                  L$DISPATCH::
002124  020462                                                  .WORD   T1
```

DEFAULT HARDWARE P-TABLE

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

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

002126                                  BGNHW   DFPTBL
002126  000005                                                         .WORD   L10000-L$HW/2
002130                                                                  L$HW::
002130                                                                  DFPTBL::

002130  172150                          .WORD   172150          ; UNIBUS ADDRESS
002132  000154                          .WORD   154             ; VECTOR ADDRESS
002134  000005                          .WORD   5.              ; BR LEVEL
002136  000077                          .WORD   63.             ; UNIBUS BURST RATE
002140  000000                          .WORD   0.              ; LOGICAL DRIVE NUMBER
002142                                  ENDHW
002142                                                          L10000:
```

```
                        .SBTTL  SOFTWARE P-TABLE

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

002142                          BGNSW   SFPTBL
002142  000001                                                 .WORD   L10001-L$SW/2
002144                                                      L$SW::
002144                                                      SFPTBL::

                                                    ;OFFSET         USE
002144  000007                  .WORD   7          ; 0.    YES/NO ANSWERS
002146                          ENDSW
002146                                                      L10001:

002146                          ENDMOD
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

002146                                  BGNMOD

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

002146                                  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
```

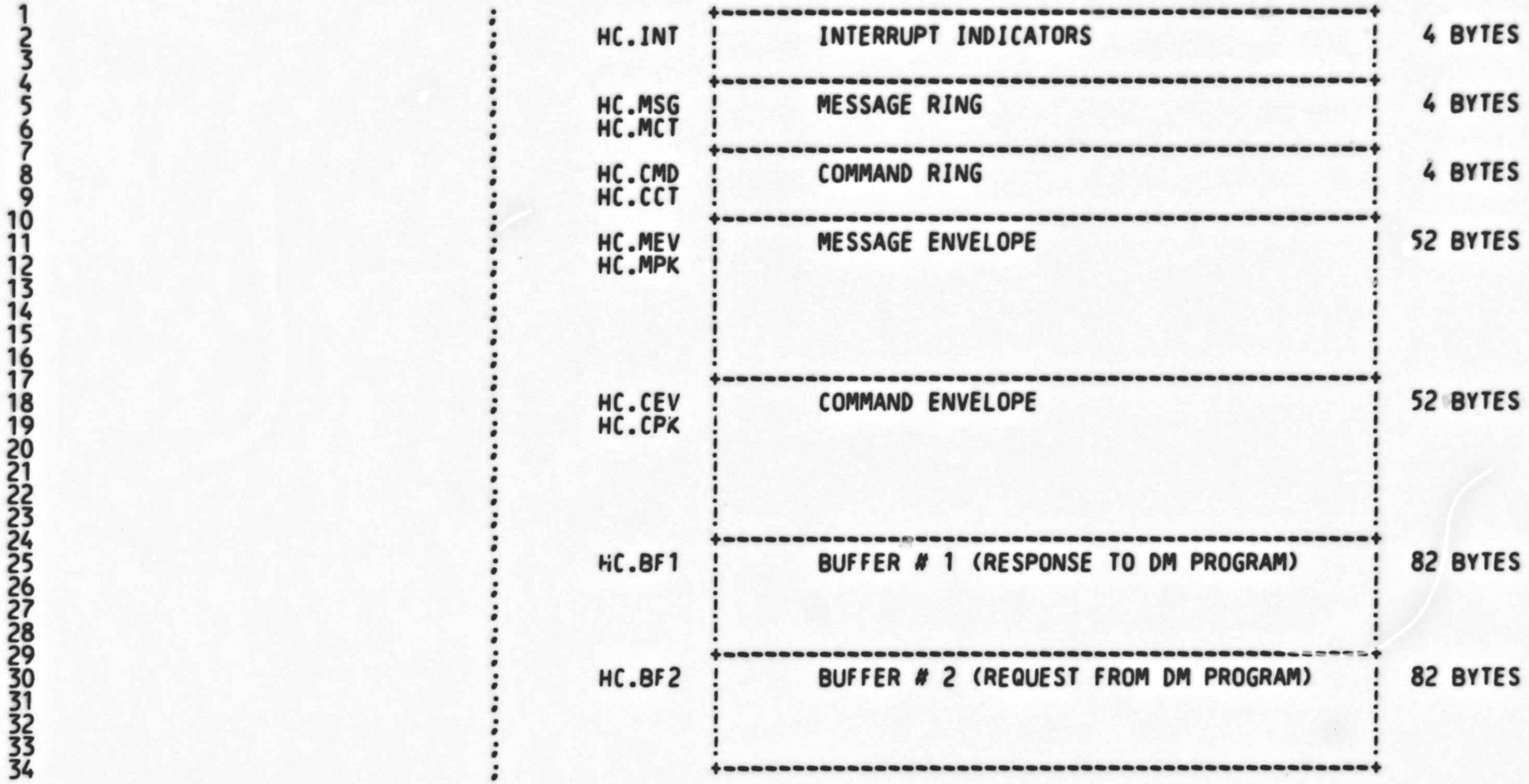
GLOBAL EQUATES SECTION

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

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

                                ;UDASA REGISTER STEP THREE WRITE BITS

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

                                ;UDASA REGISTER STEP FOUR READ BITS

000377                          SA.MCV= 000377          ;UDA MICROCODE VERSION
                                ;       003400          ;RESERVED

                                ;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
```

```
                        ;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
000122                  HC.BSZ= 82.                     ;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
000306                  HC.BF2= HC.BF1+HC.BSZ           ;SECOND BUFFER

000430                  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
```

GLOBAL EQUATES SECTION



GLOBAL EQUATES SECTION

```
                        ;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
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
```

GLOBAL EQUATES SECTION

```
                                ;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
```

GLOBAL EQUATES SECTION

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

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.

002146                          TABLE                                   ;START A TABLE DEFINITION

002146                          ITEM C.UADR     2                       ;UNIBUS ADDRESS OF UDAIP REGISTER
002146                          ITEM C.UNIT     2
        000077                          CT.UNT= 000077                  ; LOGICAL UNIT NUMBER (FIRST)
        100000                          CT.AVL= BIT15                   ; SET WHEN NOT AVAILABLE FOR TESTING
002146                          ITEM C.VEC      2
        000777                          CT.VEC= 000777                  ; VECTOR ADDRESS
        007000                          CT.BRL= 007000                  ; BR LEVEL
002146                          ITEM C.BST      2                       ; BURST LEVEL
002146                          ITEM C.JSR      2                       ;INTERRUPT SERVICE ROUTINE FOR CONTROLLER
002146                          ITEM C.JAD      2                       ;  THESE TWO WORDS LOADED WITH [JSR R0,UDASRV]
002146                          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
002146                          ITEM C.RING     2                       ;RING BUFFER ADDRESS
002146                          ITEM C.DR0      2                       ;POINTER TO DRIVE TABLES
002146                          ITEM C.DR1      2                       ;  IF ZERO, NO DRIVE TABLE EXISTS
002146                          ITEM C.DR2      2
002146                          ITEM C.DR3      2
002146                          ITEM C.DR4      2
002146                          ITEM C.DR5      2
002146                          ITEM C.DR6      2
002146                          ITEM C.DR7      2
002146                          ITEM C.TO       2                       ;TIMEOUT COUNTER
002146                          ITEM C.TOH      2                       ;  (TWO WORDS)
002146                          ITEM C.TOT      2                       ;DUP PROGRAM TIMEOUT VALUE IN SECONDS
002146                          ITEM C.PRI      4                       ;DUP PROGRAM PROGRESS INDICATOR
002146                          ITEM C.REF      2                       ;COMMAND REFERENCE NUMBER

002146                          END C.SIZE                              ;SIZE OF CONTROLLER TABLE IN BYTES
```

GLOBAL EQUATES SECTION

```
                                ;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.
                                
002146                          TABLE                           ;START A TABLE DEFINITION
                                
002146                          ITEM D.DRV      2                       ;DRIVE NUMBER
002146                          ITEM D.UNIT     2
        000077                          DT.UNT= 000077                  ; LOGICAL UNIT NUMBER OF DRIVE
        100000                          DT.AVL= BIT15                   ; SET WHEN NOT AVAILABLE FOR TESTING
002146                          ITEM D.SERN     22.                     ;DISK SERIAL NUMBER
                                
002146                          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.
                                ;--

002146                          FFREE:: .BLKW 1                         ;FIRST FREE WORD IN MEMORY
002150                          FSIZE:: .BLKW 1                         ;SIZE OF FREE MEMORY IN WORDS
002152                          FMEM:   .BLKW 1                 ;COPY OF FFREE AT END OF INIT SECTION
002154                          FMEMS:  .BLKW 1                 ;COPY OF FSIZE AT END OF INIT SECTION
002156                          CTABS:: .BLKW 1                         ;START OF CONTROLLER TABLE STORAGE
002160                          CTRLRS: .BLKW 1                         ;COUNT OF UDA CONTROLLERS IN PTABLES
002162                          TSTTAB: .BLKW 1                         ;POINTER TO FIRST CONTROLLER TABLE UNDER TEST
                                        .GLOBL UDAFM
002164  000000G                 DMPROG: .WORD UDAFM                     ;START ADDRESS OF DM PROGRAM
002166                          DMEND:  .BLKW 1                         ;END ADDRESS OF DM PROGRAM(FIRST FREE MEMORY ADR)
002170                          DMENDS: .BLKW 1                         ;FREE MEMORY SIZE FROM END OF DM PROGRAM
002172                          URUN:   .BLKW 1                         ;NUMBER OF UNITS TO RUN AT ONE TIME
002174                          URNING: .BLKW 1                         ;NUMBER OF UNITS STILL RUNNING
002176                          UCNT:   .BLKW 1                         ;COUNTER OF UNITS UNDER TEST
002200                          UFREEZ: .BLKW 1                         ;FREEZE ON UNIT WHEN NOT ZERO
002202                          NXMAD:  .BLKW 1                         ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
002204  000000                  FDATA:  .WORD 0
002206                          FCTBUF: .BLKB 512.              ;STORAGE FOR FCT BLOCK
003206                          FCTNUM: .BLKW 1         ;FCT BLOCK NUMBER
003210                          MODE:   .BLKW 1 ;MODE WORD, SAME BIT DEFS AS SO.BIT

                                ;CLOCK CONTROL

003212  000000                  KW.CSR: .WORD 0                         ;CSR OF CLOCK
003214                          KW.BRL: .BLKW 1                         ;BR LEVEL
003216                          KW.VEC: .BLKW 1                         ;VECTOR
003220                          KW.HZ:  .BLKW 1                         ;HERTZ (50. OR 60.)
003222                          KW.EL:  .BLKW 2                         ;ELAPSED TIME

003226  014526                  PTYPE:  .WORD PF                        ;PRINT TYPE
003230    000   000             ERRCHR: .BYTE 0,0                       ;FIRST BYTE LOADED WITH OUTPUT CHARACTER
003232  000000                  NULL:   .WORD 0                         ;USED TO PRINT A NULL CHARACTER
003234  000000                  FNAME:  .WORD 0                 ;SPACE FOR DATA FILE NAME
003236                                  .BLKB 10.
```

GLOBAL DATA SECTION

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

```
                                .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
                                ;
 003442                                 DEVTYP <UDA-50 CONTROLLER>
 003442                                                                 L$DVTYP::
 003442  125     104     101                                                    .ASCIZ  /UDA-50 CONTROLLER/
                                                                                .EVEN

                                ; TEST DESCRIPTION
                                ;
 003464                                 DESCRIPT <CZUDEBP UDA-50 DISK DRV FORMATTER>
 003464                                                                 L$DESC::
 003464  103     132     125                                                    .ASCIZ  /CZUDEBP UDA-50 DISK
                                                                                .EVEN
```

GLOBAL TEXT SECTION

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

GLOBAL TEXT SECTION

```
                                        ; FORMAT STATEMENTS USED IN PRINT CALLS
                                        ;
003612      045     124     000     ERRONE: .ASCIZ\%T\
003615      045     116     000     ERRNL:  .ASCIZ\%N\
003620      042     040     040     RNTIM:  .ASCIZ\"  RUNTIME "D16":"\
003643      104     071     042     RNTIM1: .ASCIZ\D9":"\
003651      104     071     000     RNTIM2: .ASCIZ\D9\
003654      042     040     040     ERRME1: .ASCIZ\"  *  *  *  ERROR PROCESSING MESSAGE STRING  *  *  *"N\
003743      116     042     125     MESSG:  .ASCIZ\N"UNIT "D6" UDA AT "O16" DRIVE "D9S\
004007      042     116     117     NOCLOCK:.ASCIZ\"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\
                                        ;
004064      042     110     117     BASN0:  .ASCIZ\"HOST PROGRAM"\
004103      042     040     040     BASL2:  .ASCIZ\"  UDA AT "O16\
004122      042     040     040     BASL3:  .ASCIZ\"  DRIVE "D9\
004137      000                     BAS:    .BYTE 0                         ;NULL TO PRINT NOTHING
                                        ;
004140      122     066     122     BASLN:  .ASCIZ\R6R6R6R6\                ;USED TO PRINT BASIC LINE OF ERROR MESSAGE
004151      116     042     123     SERNUM: .ASCIZ\N"SERIAL NUMBER FOR UNIT "D6" UDA AT "O16" DRIVE "D9\
```

GLOBAL TEXT SECTION

```
                                        X1A:
                                        X2A:
                                        X3A:
004236  042     111     040     X8A:    .ASCIZ\''I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS'N\
004335  122     065     122     X1:     .ASCIZ\R5R6''UDA HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE'N\
004431  122     065     122     X2:     .ASCIZ\R5R6''TWO UNITS SELECT THE SAME DRIVE'N\
004500  122     065     122     X3:     .ASCIZ\R5R6''MORE THAN EIGHT DRIVES SELECTED ON THIS UDA'N\
004563  122     064     042     X4:     .ASCII\R4''NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED'N\
004654  042     120     114             .ASCIZ\''PLEASE START PROGRAM OVER AND TEST FEWER UNITS AT A TIME'N\
004750  122     065     122     X8:     .ASCIZ\R5R6''TWO UDA'S USE THE SAME VECTOR'N\
005015  122     064     042     X9:     .ASCII\R4''ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE.'N\
005120  042     120     114             .ASCIZ\''PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK.'N\
005207  122     064     042     X10:    .ASCIZ\R4''THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE.'N\
005306  122     065     042     X20:    .ASCII\R5''MEMORY ERROR TRYING TO READ UDA REGISTERS'N\
005364  042     103     110             .ASCII\''CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7161'N\
005452  042     117     122             .ASCII\''OR UNIBUS'N\
005466  042     117     122             .ASCIZ\''OR ''R7\
005476  122     065     042     X21:    .ASCII\R5''UDA RESIDENT DIAGNOSTICS DETECTED FAILURE'NR8\
005556  042     122     105             .ASCIZ\''REPLACE UDA MODULE M716''02N\
005613  122     065     042     X22:    .ASCII\R5''STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION'N\
005714  042     123     124             .ASCIZ\''STEP BIT EXPECTED ''016NR8R7\
005751  122     065     042     X23A:   .ASCII\R5''UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION'N\
006063  104     071     042             .ASCII\D9'' WORDS WERE TO BE CLEARED STARTING AT ADDRESS ''016N\
006151  042     106     111             .ASCII\''FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):'N\
006226  123     066     042             .ASCIZ\S6''ADDRESS''S4''CONTENTS'N\
006257  123     067     117     X23B:   .ASCIZ\S7016S5016N\
006273  122     065     042     X24:    .ASCII\R5''UDASA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION'N\
006406  042     120     125             .ASCIZ\''PURGE/POLE DIAGNOSTICS WERE REQUESTED'NR8R7\
006463  122     065     042     X25:    .ASCII\R5''UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION'N\
006577  042     040     040             .ASCIZ\''  UDASA EXPECTED  ''016NR8R7\
006634  122     065     042     X30:    .ASCIZ\R5''UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE RUNNING DM PROGRAM'NR8\
006747  122     065     042     X31:    .ASCIZ\R5''DUP PROGRAM IS HUNG'N\
007000  122     065     042     X32:    .ASCIZ\R5''MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER'N\
007111  122     065     042     X36:    .ASCII\R5''NO INTERRUPT RECEIVED FROM UDA FOR 30 SECONDS'N\
007173  042     127     110             .ASCIZ\''WHILE LOADING DM PROGRAM'N\
007227  122     065     042     X37:    .ASCIZ\R5''UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM'NR8R7\
007344  122     065     042     X100:   .ASCIZ\R5''DUP PROGRAM ASKED UNEXPECTED QUESTION (''D12'')'N\
007427  122     065     042     X101:   .ASCIZ\R5''DUP PROGRAM REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION'N\
```

GLOBAL TEXT SECTION

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

010430                          BGNMSG ERR001
010430                                  PNTB X1,#X1A
010430  012746  004236                                          MOV #X1A,-(SP)
010434  004137  014660                                          JSR R1,LPNTB
010440  004335                                                  .WORD X1
010442  000002                                                  .WORD PNT.CT
010444                          ENDMSG

010446                          BGNMSG ERR002
010446                                  PNTB X2,#X2A
010446  012746  004236                                          MOV #X2A,-(SP)
010452  004137  014660                                          JSR R1,LPNTB
010456  004431                                                  .WORD X2
010460  000002                                                  .WORD PNT.CT
010462                          ENDMSG

010464                          BGNMSG ERR003
010464                                  PNTB X3,#X3A
010464  012746  004236                                          MOV #X3A,-(SP)
010470  004137  014660                                          JSR R1,LPNTB
010474  004500                                                  .WORD X3
010476  000002                                                  .WORD PNT.CT
010500                          ENDMSG

010502                          BGNMSG ERR004
010502                                  PNTB X4
010502  004137  014660                                          JSR R1,LPNTB
010506  004563                                                  .WORD X4
010510  000000                                                  .WORD PNT.CT
010512                          ENDMSG

010514                          BGNMSG ERR009
010514                                  PNTB X9
010514  004137  014660                                          JSR R1,LPNTB
010520  005015                                                  .WORD X9
010522  000000                                                  .WORD PNT.CT
010524                          ENDMSG

010526                          BGNMSG ERR010
010526                                  PNTB X10
010526  004137  014660                                          JSR R1,LPNTB
010532  005207                                                  .WORD X10
010534  000000                                                  .WORD PNT.CT
010536                          ENDMSG
```

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

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

GLOBAL ERROR REPORT SECTION

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

```
        011234  004137  014660                                                  JSR R1,LPNTB
        011240  007532                                                          .WORD XMSG1
        011242  000000                                                          .WORD PNT.CT
    134 011244  016504  000016              MOV C.RING(R5),R4
    135 011250  062704  000306              ADD #HC.BF2,R4
    136 011254  012703  000005              MOV #5,R3
    137 011260                      MSGPKL: PNTB XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)
        011260  016446  000014                                                  MOV 12.(R4),-(SP)
        011264  016446  000012                                                  MOV 10.(R4),-(SP)
        011270  016446  000010                                                  MOV 8.(R4),-(SP)
        011274  016446  000006                                                  MOV 6(R4),-(SP)
        011300  016446  000004                                                  MOV 4(R4),-(SP)
        011304  016446  000002                                                  MOV 2(R4),-(SP)
        011310  011446                                                          MOV (R4),-(SP)
        011312  004137  014660                                                  JSR R1,LPNTB
        011316  007566                                                          .WORD XMSG2
        011320  000016                                                          .WORD PNT.CT
    138 011322  062704  000016              ADD #14.,R4
    139 011326  005303                      DEC R3
    140 011330  001353                      BNE MSGPKL
    141 011332  000207                      RETURN
```

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

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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

011334                          FMERR:  ERRSF 4,,ERR004
011334  104454                                                          TRAP    C$ERSF
011336  000004                                                          .WORD   4
011340  000000                                                          .WORD   0
011342  010502                                                          .WORD   ERR004
011344                                  DOCLN                   ;ABORT
011344  104444                                                          TRAP    C$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.
                                        
011346                  ALOCM:  PUSH FFREE                      ;SAVE FFREE AT ENTRY
011346  013746  002146                                                  MOV FFREE,-(SP)
011352  160137  002150                  SUB R1,FSIZE            ;REDUCE SIZE OF FREE MEMORY
011356  002766                          BLT FMERR               ;REPORT ERROR IF NOT ENOUGH MEMORY
011360  060101                          ADD R1,R1               ;CHANGE WORDS TO BYTES
011362  060137  002146                  ADD R1,FFREE            ;CALCULATE NEW START OF FREE MEMORY
011366                                  POP R1                  ;GET START OF ALLOCATED MEMORY
011366  012601                                                          MOV (SP)+,R1
011370  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

011372  012701  000214          HCOMM:  MOV #HC.SIZ/2,R1                ;GET SIZE OF AREA TO ALLOCATE
011376  004737  011346                  CALL ALOCM                      ;ALLOCATE THE MEMORY
011402  010165  000016                  MOV R1,C.RING(R5)                       ;GET ADDRESS OF HOST COMM AREA
                                                                        ;PLACE IN CONTROLLER TABLE
011406  000207                          RETURN
```

```
                                        ;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.

011410  010137  002172          RUNDM:  MOV R1,URUN                     ;SAVE NUMBER OF UNITS TO RUN
011414  005037  002174                  CLR URNING                      ;CLEAR NUMBER OF UNITS RUNNING

                                        ;LOAD DM PROGRAM INTO EACH CONTROLLER

011420  013737  002172  002176          MOV URUN,UCNT                   ;SET COUNTER OF UNITS
011426  013705  002162                  MOV TSTTAB,R5                   ;GET FIRST CONTROLLER TABLE
011432                          LDDM:
011432  005065  000014                  CLR C.FLG(R5)                   ;CLEAR ALL FLAGS
011436  116537  000002  002074          MOVB C.UNIT(R5),L$LUN           ;SEE IF UNIT TO BE TESTED
011444  005765  000002                  TST C.UNIT(R5)
011450  100407                          BMI LDNEXT                      ;IF NOT, DON'T LOAD THIS UNIT
011452                                          ASSUME CT.AVL EQ BIT15
011452  004737  011372                  CALL HCOMM      ;ALLOCATE SPACE FOR HOST COMM AREA
011456  004737  015012                  CALL LOADDM                     ;LOAD THE DM PROGRAM
011462  001402                          BEQ LDNEXT                      ;IF ERROR, GO TO NEXT CONTROLLER
011464  005237  002174                  INC URNING                      ;IF NO ERROR, COUNT UNIT RUNNING
011470  062705  000054          LDNEXT: ADD #C.SIZE,R5                  ;MOVE TO NEXT CONTROLLER TABLE
011474  005337  002176                  DEC UCNT                        ;CHECK IF MORE CONTROLLERS
011500  001354                          BNE LDDM                        ;LOAD NEXT
011502  005037  002200                  CLR UFREEZ                  ;CLEAR UNIT FREEZE FLAG
011506  012737  177777  003206          MOV #-1,FCTNUM  ;INVALIDATE FCT BLOCK NUMBER (BLOCK IN MEMORY)

                                        ;CHECK IF ANY CONTROLLERS LOADED

011514  005737  002174                  TST URNING                      ;ANY UNITS LOADED?

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

011520  000207                          RETURN
```

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

011522  013705  002162          RESPDM: MOV TSTTAB,R5                   ;GET CONTROLLER TABLE ADDRESS
011526  013737  002172  002176          MOV URUN,UCNT                   ;SET COUNTER OF UNITS
011534                          RESPCT: BREAK   ;ALLOW DRS TO SEE TERMINAL INPUT
011534  104422                                                                  TRAP    C$BRK
011536  016504  000016                  MOV C.RING(R5),R4               ;GET HOST COMM AREA ADDRESS
011542  032765  000002  000014          BIT #CT.RN,C.FLG(R5)            ;CHECK IF PROGRAM RUNNING
011550  001502                          BEQ RSPNXT                      ;IF NOT, LOOK AT NEXT
011552  116537  000002  002074          MOVB C.UNIT(R5),L$LUN           ;STORE UNIT NUMBER UNDER TEST
011560  032765  000010  000014          BIT #CT.MSG,C.FLG(R5)           ;SEE IF INTERRUPT RECEIVED
011566  001150                          BNE RSPIN                       ;IF SO, LOOK AT PACKET
011570  032765  000004  000014          BIT #CT.CMD,C.FLG(R5)           ;SEE IF COMMAND HAS BEEN SENT
011576  001002                          BNE 1$                          ;IF NOT, SEND ONE
011600  000137  012346                  JMP RSPOUT

                                        ;CHECK IF UDA STILL RUNNING

011604  011503                  1$:     MOV (R5),R3                     ;GET ADDRESS OF UDAIP
011606  016301  000002                  MOV 2(R3),R1                    ;LOOK AT UDASA REGISTER
011612  001405                          BEQ RSPTM                       ;IF ZERO, UDA STILL RUNNING
011614                                  ERRDF 30,,ERR030                ;REPORT UDA HAS FATAL ERROR
011614  104455                                                                  TRAP    C$ERDF
011616  000036                                                                  .WORD   30
011620  000000                                                                  .WORD   0
011622  011006                                                                  .WORD   ERR030
011624  000465                          BR RSPDRP                       ;DROP CONTROLLER FROM TESTING

                                        ;CHECK FOR TIMEOUT OF RESPONSE

011626  005765  000044          RSPTM:  TST C.TOT(R5)                   ;SEE IF DUP PROGRAM TO BE TIMED
011632  001451                          BEQ RSPNTO
011634  005737  003212                  TST KW.CSR                      ;SEE IF A CLOCK ON SYSTEM
011640  001446                          BEQ RSPNTO                      ;DON'T TIME IF NO CLOCK
011642  023765  003224  000042          CMP KW.EL+2,C.TOH(R5)           ;COMPARE TO TIMEOUT COUNTER
011650  101005                          BHI RSPTMO
011652  001041                          BNE RSPNTO
011654  023765  003222  000040          CMP KW.EL,C.TO(R5)
011662  103435                          BLO RSPNTO                      ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
011664  032765  000040  000014  RSPTMO: BIT #CT.STA,C.FLG(R5)           ;SEE IF A GET DUST STATUS COMMAND OUTSTANDING
011672  001101                          BNE RSPTOE                      ;REPORT ERROR IF SO
011674  005764  000012                  TST HC.CCT(R4)                  ;SEE IF UDA TOOK LAST COMMAND PACKET
011700  100476                          BMI RSPTOE                      ;REPORT ERROR IF NOT
011702  012700  000100                  MOV #CT.TM1,R0                  ;SEE IF FIRST TIMEOUT ALREADY HAPPENED
011706  032765  000100  000014          BIT #CT.TM1,C.FLG(R5)
011714  001401                          BEQ 1$                          ;IF SO,
011716  006300                          ASL R0                          ;SET SECOND TIME OUT FLAG
011720  052700  000040          1$:     BIS #CT.STA,R0                  ;SET THE PROPER TIMEOUT BIT
011724  050065  000014                  BIS R0,C.FLG(R5)                ;  AND STATUS REQUESTED BIT
011730  012700  000001                  MOV #OP.GDS,R0                  ;BUILD GET DUST STATUS COMMAND
011734  004737  015214                  CALL BLDCMD
011740  012764  100000  000012          MOV #RG.OWN,HC.CCT(R4)          ;MARK COMMAND TO UDA
011746  005775  000000                  TST @(R5)                       ;TELL UDA COMMAND IS THERE
011752  000137  012426                  JMP RSPOU4
```

```
     53 011756                          RSPNTO:
```

```
                                        ;SWITCH TO NEXT CONTROLLER
                                        
011756  005737  002200          RSPNXT: TST UFREEZ      ;FROZEN TO ONE UNIT?
011762  001264                          BNE RESPCT      ;STAY THERE IF SO
011764  062705  000054                  ADD #C.SIZE,R5                  ;MOVE TO NEXT TABLE
011770  005337  002176                  DEC UCNT                        ;CHECK IF MORE CONTROLLERS
011774  001257                          BNE RESPCT                      ;LOOK AT NEXT CONTROLLER
011776  000651                          BR RESPDM                       ;LOOK AT FIRST CONTROLLER AGAIN
                                        
                                        ;REMOVE A CONTROLLER FROM TESTING
                                        
012000  005065  000014          RSPDRP: CLR C.FLG(R5)   ;CLEAR PROGRAM RUNNING
012004  005037  002200                  CLR UFREEZ
012010  010504                          MOV R5,R4
012012  062704  000020                  ADD #C.DRO,R4
012016  012702  000010                  MOV #8.,R2
012022  012403                  1$:     MOV (R4)+,R3
012024  001420                          BEQ 3$
012026  005763  000002                  TST D.UNIT(R3)
012032                                          ASSUME DT.AVL EQ BIT15
012032  100003                          BPL 2$
012034  005302                          DEC R2
012036  001371                          BNE 1$
012040  000412                          BR 3$
012042  052763  100000  000002  2$:     BIS #DT.AVL,D.UNIT(R3)
012050  005302                          DEC R2
012052  001405                          BEQ 3$
012054  005714                          TST (R4)
012056  001403                          BEQ 3$
012060  004737  015012                  CALL LOADDM     ;START DM PROGRAM AGAIN
012064  001223                          BNE RESPCT
012066  005337  002174          3$:     DEC URNING                      ;REDUCE RUNNING CONTROLLERS COUNT
012072  001331                          BNE RSPNXT                      ;IF ANY STILL RUNNING, LOOK AT THEM
012074  000207                          RETURN                          ;ELSE RETURN TO TEST SECTION
                                        
012076                          RSPTOE: ERRDF 31,,ERR031                ;REPORT TIMEOUT ERROR
012076  104455                                                                  TRAP    C$ERDF
012100  000037                                                                  .WORD   31
012102  000000                                                                  .WORD   0
012104  011022                                                                  .WORD   ERR031
012106  000734                          BR RSPDRP                       ;DROP CONTROLLER FROM TESTING
```

```
                                        ;CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
                                        ;
                                        ;CHECK FOR PROPER OPCODE IN END PACKET
                                        ;
012110  012700  000204          RSPIN:  MOV #OP.END+OP.SSD,R0           ;GET SEND DATA END PACKET OPCODE
012114  032765  000020  000014          BIT #CT.REQ,C.FLG(R5)           ;LOOK IF SEND DATA OR RECEIVE DATA
012122  001402                          BEQ RSPMWR
012124  012700  000205                  MOV #OP.END+OP.RSD,R0           ;CHANGE TO RECEIVE DATA END PACKET OPCODE
012130  120064  000030          RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)       ;COMPARE TO OPCODE IN END PACKET
012134  001145                          BNE RSPERR
                                        ;
                                        ;LOOK AT STATUS CODE
                                        ;
012136  032764  000037  000032          BIT #ST.MSK,HC.MPK+P.STS(R4)    ;CHECK FOR STATUS CODE ST.SUC (ZERO)
012144  001004                          BNE RSPERW
                                        ;
                                        ;CHECK FOR EXPECTED REFERENCE NUMBER
                                        ;
012146  026564  000052  000020          CMP C.REF(R5),HC.MPK+P.CRF(R4)  ;CHECK IF CORRECT REF NUMBER
012154  001405                          BEQ RSPPTW
012156                          RSPERW: ERRDF 33,,ERR033
012156  104455                                                          TRAP    C$ERDF
012160  000041                                                          .WORD   33
012162  000000                                                          .WORD   0
012164  011052                                                          .WORD   ERR033
012166  000704                          BR RSPDRP                       ;DROP UNIT FROM TESTING
                                        ;
                                        ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
                                        ;
012170  032765  000020  000014  RSPPTW: BIT #CT.REQ,C.FLG(R5)           ;CHECK IF RESPONSE FROM DM PROGRAM
012176  001463                  RSPOU:  BEQ RSPOUT                      ;LOOK AT REQUEST NUMBER IF SO
```

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

012200  016401  000306        RSPPT2: MOV HC.BF2(R4),R1               ;GET REQUEST NUMBER
012204  042701  007777                BIC     #^C<DU.TYP>,R1          ;CHECK TYPE
012210  001403                        BEQ 1$                          ;IF ZERO, ERROR
012212  020127  060000                CMP R1,#DU.SPC                  ;CHECK IF IN EXPECTED RANGE
012216  101405                        BLOS RSPPT3
012220                          1$:     ERRDF 32,,ERR032                ;BAD REQUEST NUMBER
012220  104455                                                                  TRAP    C$ERDF
012222  000040                                                                  .WORD   32
012224  000000                                                                  .WORD   0
012226  011034                                                                  .WORD   ERR032
012230  000663                        BR RSPDRP                       ;DROP UNIT FROM TESTING

012232  016403  000034        RSPPT3: MOV HC.MPK+P.BCNT(R4),R3        ;GET BYTE COUNT OF CHARACTERS RECEIVED IN R3
012236  162703  000002                SUB #2,R3                       ;(FIRST TWO CHARACTERS ARE TYPE WORD)
012242  012700  000004                MOV #OP.SSD,R0                  ;BUILD A SEND DATA COMMAND PACKET
012246  004737  015214                CALL BLDCMD                     ;  FOR ANSWER TO DM PROGRAM
012252  012700  000164                MOV #HC.BF1,R0                  ;POINT TO BUFFER IN PACKET
012256  004737  015356                CALL CLRBUF                     ; AND CLEAR BUFFER
012262  010402                        MOV R4,R2                       ;R2 POINTS TO SEND BUFFER
012264  062704  000122                ADD #HC.BSZ,R4          ;R4 POINTS TO CHARACTERS IN RECEIVE BUFFER
012270  042724  170000                BIC #DU.TYP,(R4)+       ;CLEAR TYPE FIELD IN BUFFER
012274  000301                        SWAB R1                         ;GET TYPE RIGHT JUSTIFIED
012276  006201                        ASR R1                          ;TIMES TWO
012300  006201                        ASR R1
012302  006201                        ASR R1
012304  010100                        MOV R1,R0                       ;COPY MESSAGE TYPE TO R0
012306  005001                        CLR R1                          ;R1 CONTAINS ZERO SEND BYTE COUNT
012310  004770  012574                CALL @RSPDSP-2(R0)              ;CALL REQUESTED ROUTINE
012314  001231                        BNE RSPDRP                      ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
                                                                        ;  Z SET IF UNIT TO CONTINUE RUNNING
012316  016504  000016                MOV C.RING(R5),R4               ;GET RING ADDRESS
012322  032701  000001                BIT #1,R1                       ;LOOK AT CHARACTER COUNT TO SEND TO DUP PROGRAM
012326  001401                        BEQ 1$                          ;IF AN ODD COUNT
012330  005201                        INC R1                          ;  INCREASE BY ONE
012332  010164  000120          1$:   MOV R1,HC.CPK+P.BCNT(R4)        ;PUT CHARACTER COUNT IN COMMAND PACKET
012336  100003                        BPL RSPOUT                      ;IF NEGATIVE BYTE COUNT RETURNED
012340  042765  000020  000014        BIC #CT.REQ,C.FLG(R5)           ;  DON'T SEND ANY DATA TO UDA

                                        ;SEND COMMAND BACK TO UDA

012346  042765  000350  000014  RSPOUT: BIC #CT.MSG+CT.STA+CT.TM1+CT.TM2,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
012354  032765  000020  000014        BIT #CT.REQ,C.FLG(R5)           ;CHECK WHICH COMMAND TO SEND
012362  001014                        BNE RSPOU2                      ;BRANCH IF RESPONSE TO REQUEST

012364  012700  000005                MOV #OP.RSD,R0                  ;BUILD RECEIVE DATA COMMAND
012370  004737  015214                CALL BLDCMD
012374  012700  000306                MOV #HC.BF2,R0                  ;POINT TO MESSAGE BUFFER
012400  004737  015356                CALL CLRBUF                     ;  AND CLEAR IT
012404  052765  000020  000014        BIS #CT.REQ,C.FLG(R5)           ;SET REQUEST BIT
012412  000403                        BR RSPOU3

012414  042765  000020  000014  RSPOU2: BIC #CT.REQ,C.FLG(R5)         ;CLEAR REQUEST BIT
012422                          RSPOU3:
012422  004737  015300                CALL SNDCMD                     ;SEND COMMAND TO UDA
012426  016500  000044          RSPOU4: MOV C.TOT(R5),R0              ;SET TIMEOUT
```

GLOBAL SUBROUTINES SECTION

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

```
                                        ;RESPONSE REQUEST DISPATCH TABLE
                                        
012576  012612                  RSPDSP: .WORD QUEST                     ;QUESTION
012600  012664                          .WORD DQUEST                    ;QUESTION WITH DEFAULT ANSWER
012602  013036                          .WORD INFO                      ;INFORMATION MESSAGE FOR OPERATOR
012604  013126                          .WORD TERM                      ;NORMAL TERMINATION
012606  013136                          .WORD ERRTRM                    ;FATAL ERROR TERMINATION
012610  013146                          .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          +-------------------------------+
                        ;
                        ;         .                        .
                        ;         .                        .
                        ;         .                        .
                        ;                    +-------------------------------+
                        ;        80          !          DATA BYTES           !
                        ;                    +-------------------------------+
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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
                                        
012612  004737  013300          QUEST:  CALL GTDRVT      ;GET POINTER TO DRIVE TABLE
012616  062700  000004                  ADD #D.SERN,R0   ;BUMP POINTER TO SERIAL NUMBER
012622  014403                          MOV -(R4),R3    ;GET QUESTION NUMBER
012624  001411                          BEQ QUE0        ;BRANCH IF QUESTION NUMBER 0
012626  020327  000007                  CMP R3,#7       ;IF NOT, SEE IF QUESTION NUMBER 7
012632  001410                          BEQ QUE7
012634                                  ERRDF 100,,ERR100       ;ANY OTHER NUMBER IS AN ERROR
012634  104455                                                          TRAP    C$ERDF
012636  000144                                                          .WORD   100
012640  000000                                                          .WORD   0
012642  011114                                                          .WORD   ERR100
012644  000244                          CLZ     ;CLEAR Z TO STOP DUP PROGRAM
012646  000207                          RETURN
                                        
012650  012700  003312          QUE0:   MOV #DATE0,R0   ;POINT TO DATE STRING
012654                          QUE7:
012654  005201                  QUEL:   INC R1  ;COUNT THE CHARACTERS
012656  112022                          MOVB (R0)+,(R2)+        ; AND PUT THEM IN OUTPUT BUFFER
012660  001375                          BNE QUEL        ;  UNTIL A NUL CHARACTER FOUND
012662  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

012664  004737  013300        DQUEST: CALL GTDRVT     ;GET DRIVE TABLE ADDRESS INTO R0
012670  014403                        MOV -(R4),R3    ;GET QUESTION NUMBER
012672  020327  000006                CMP R3,#DQUESZ
012676  101035                        BHI DQUEX
012700  006303                        ASL R3
012702  000173  012706                JMP @DQUEJP(R3)
012706  012772                DQUEJP: .WORD DQUEX     ; 0  (NOT USED)
012710  012724                        .WORD DQUNIT    ; 1  ENTER UNIT NUMBER TO FORMAT
012712  012772                        .WORD DQUEX     ; 2  (NOT USED)
012714  012772                        .WORD DQUEX     ; 3  (NOT USED)
012716  012776                        .WORD DQRFMT    ; 4  USE EXISTING BAD SECTOR INFORMATION
012720  013016                        .WORD DQRSTR    ; 5  DOWN-LINE LOAD BAD SECTOR BLOCK INFORMATION
012722  013026                        .WORD DQCONT    ; 6  CONTINUE IF BAD BLOCK INFO INACCESSIBLE
        000006                                  DQUESZ=<<.-DQUEJP>/2>-1

                                ;ENTER UNIT NUMBER TO FORMAT

012724                          DQUNIT: PUSH R5
012724  010546                                                          MOV R5,-(SP)
012726  005004                        CLR R4
012730  011003                        MOV (R0),R3     ;GET DRIVE NUMBER
012732                                        ASSUME D.DRV EQ 0
012732  012700  000012                MOV #10.,R0     ;RADIX 10.
012736  004737  014754        DQUNL1: CALL DIVIDE
012742                                PUSH R5
012742  010546                                                          MOV R5,-(SP)
012744  005201                        INC R1
012746  005703                        TST R3
012750  001372                        BNE DQUNL1
012752  010100                        MOV R1,R0
012754                          DQUNL2: POP R5
012754  012605                                                          MOV (SP)+,R5
012756  062705  000060                ADD #'0,R5
012762  110522                        MOVB R5,(R2)+
012764  005300                        DEC R0
012766  001372                        BNE DQUNL2
012770                                POP R5
012770  012605                                                          MOV (SP)+,R5
012772  000264                DQUEX:  SEZ
012774  000207                        RETURN

012776  032737  000003  003210  DQRFMT: BIT #SO.FMT,MODE
```

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

```
                                        ;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
                                        
013036  016400  177776          INFO:   MOV -2(R4),R0   ;GET MESSAGE NUMBER
013042  001417                          BEQ INFOX       ;IF ZERO, IGNORE IT
013044  020027  000100                  CMP R0,#100     ;IF OCTAL 100
013050  001420                          BEQ INFOE       ;  PRINT ERROR MESSAGE
013052  005737  002200                  TST UFREEZ
013056  001007                          BNE INFOP
013060  005237  002200                  INC UFREEZ
013064  004737  013300                  CALL GTDRVT
013070  010002                          MOV R0,R2
013072  004737  013324                  CALL HEADER
013076  004737  013244          INFOP:  CALL MESG       ;PRINT THE MESSAGE
013102  012701  100000          INFOX:  MOV #BIT15,R1                   ;RETURN A NEGATIVE BYTE COUNT
013106  000264                          SEZ
013110  000207                          RETURN                          ;RETURN WITH Z SET
                                        
013112                          INFOE:  ERRDF 101,,ERR101       ;ANSWER WAS REJECTED BY DUP PROGRAM
013112  104455                                                                  TRAP    C$ERDF
013114  000145                                                                  .WORD   101
013116  000000                                                                  .WORD   0
013120  011130                                                                  .WORD   ERR101
013122  000244                          CLZ     ;RETURN WITH Z CLEAR TO STOP DUP PROGRAM
013124  000207                          RETURN
```

```
                                        ;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
                
013126  004737  013036          TERM:   CALL INFO       ;PRINT THE MESSAGE
013132  000244                          CLZ
013134  000207                          RETURN                          ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM
```

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

013136  004737  013036          ERRTRM: CALL INFO
013142  000244                          CLZ
013144  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
                                        
013146  023714  003206          SPECL:  CMP FCTNUM,(R4) ;SEE IF DESIRED BLOCK IS IN MEMORY
013152  001425                          BEQ SPECLX      ; IF SO, SEND TO DUP PROGRAM
013154  002407                          BLT SPECLR      ; IF LOWERED NUMBERED BLOCK IN MEMORY,
                                        ;                 GO READ NEXT BLOCK
013156                          SPECLC: CLOSE   ;OTHERWISE, START READING FROM BEGINNING AGAIN
013156  104435                                                                  TRAP    C$CLOS
013160                                  OPEN #FNAME
013160  012700  003234                                                          MOV     #FNAME,R0
013164  104434                                                                  TRAP    C$OPEN
013166  012737  177777  003206          MOV #-1,FCTNUM
013174  012703  001000          SPECLR: MOV #512.,R3    ;GET BYTE COUNT IN A BLOCK
013200  012701  002206                  MOV #FCTBUF,R1  ;POINT TO STORAGE AREA
013204                          SPECLL: GETBYTE (R1)+   ;READ THE FILE
013204  104426                                                                  TRAP    C$GETB
013206  110021                                                                  MOVB    R0,(R1)+
013210                                  BNCOMPLETE SPECLE       ;PRINT ERROR IF NO MORE BYTES IN FILE
013210  103005                                                                  BCC     SPECLE
013212  005303                          DEC R3  ;COUNT THE BYTES
013214  001373                          BNE SPECLL
013216  005237  003206                  INC FCTNUM      ;KEEP COUNT OF BLOCK IN MEMORY
013222  000751                          BR SPECL
                                        
013224  005212                  SPECLE: INC (R2)        ;TELL DUP PROGRAM DATA NOT AVAILABLE
013226  012762  002206  000002  SPECLX: MOV #FCTBUF,2(R2)       ;PUT ADDRESS OF DATA IN OUTPUT BUFFER
013234  012701  000006                  MOV #6,R1       ;SEND 3 WORDS TO DUP PROGRAM
013240  000264                          SEZ
013242  000207                          RETURN                  ;RETURN WITH Z SET TO SEND DATA TO DUP PROGRAM
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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

013244                                  MESG:   PRINT #CR
013244  112700  000015                                                  MOVB #CR,R0
013250  004737  014476                                                  CALL CPNT
013254  112400                          1$:     MOVB (R4)+,R0           ;PRINT CHARACTERS FROM DUP PROGRAM
013256  001405                                  BEQ 2$                  ;  DISCARDING LF AND NULL CHARACTERS
013260  020027  000012                          CMP R0,#12
013264  001402                                  BEQ 2$
013266                                          PRINT R0
013266  004737  014476                                                  CALL CPNT
013272  005303                          2$:     DEC R3                  ;COUNT THE CHARACTERS
013274  003367                                  BGT 1$
013276  000207                                  RETURN
```

```
                                        ;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)
                                        ;
013300                                  GTDRVT: PUSH R5
013300  010546                                                                         MOV R5,-(SP)
013302  062705  000020                          ADD #C.DR0,R5
013306  012500                          GTDRVL: MOV (R5)+,R0
013310  016037  000002  002074                  MOV D.UNIT(R0),L$LUN
013316                                                  ASSUME DT.AVL EQ BIT15
013316  100773                                  BMI GTDRVL
013320                                          POP R5
013320  012605                                                                         MOV (SP)+,R5
013322  000207                                  RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;HEADER
                                        ;
                                        ;PRINT A HEADER IN FRONT OF EACH MESSAGE FROM DUP PROGRAM.
                                        ;A UDA ADDRESS IS PRINTED IF MORE THAN ONE JDA 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

013324  022737  000001  002012  HEADER: CMP #1,L$UNIT                  ;IF MORE THAN ONE UNIT BEING TESTED
013332  001411                          BEQ 1$
013334                                  PNTF MESSG,D.UNIT(R2),(R5),(R2)         ;PRINT UDA ADDRESS
013334  011246                                                                  MOV (R2),-(SP)
013336  011546                                                                  MOV (R5),-(SP)
013340  016246  000002                                                          MOV D.UNIT(R2),-(SP)
013344  004137  014650                                                          JSR R1,LPNTF
013350  003743                                                                  .WORD MESSG
013352  000006                                                                  .WORD PNT.CT
013354                                          ASSUME C.UADR EQ 0
013354                                          ASSUME D.DRV EQ 0
013354  000407                          BR 2$
013356  005737  003212          1$:     TST KW.CSR                      ;IF NO CLOCK BEING USED
013362  001406                          BEQ 3$                          ;BYPASS RUNTIME MESSAGE
013364                                  PRINT #CR
013364  112700  000015                                                          MOVB #CR,R0
013370  004737  014476                                                          CALL CPNT
013374  004737  016636          2$:     CALL RNTIME                     ;PRINT RUNTIME IF A CLOCK IN USE
013400  000207                  3$:     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
                                
013402  112201          OSTRNG: MOVB (R2)+,R1           ;GET CONTROL CHARACTER
013404  001421                  BEQ OSTRE               ;EXIT IF NULL CHARACTER
013406  012700  013704          MOV #ERRC,R0            ;GET POINTER TO CHARACTER TABLE
013412  120110          NCONS:  CMPB R1,(R0)            ;COMPARE CHARACTER WITH TABLE ENTRY
013414  001407                  BEQ NCONF               ;BRANCH IF MATCH FOUND
013416  105720                  TSTB (R0)+              ;INCREMENT POINTER
013420  001374                  BNE NCONS               ;CONTINUE SEARCH IF NOT END OF TABLE
013422                          PNTF ERRME1             ;REPORT BAD CONTROL CHARACTER
013422  004137  014650                                                  JSR R1,LPNTF
013426  003654                                                          .WORD ERRME1
013430  000000                                                          .WORD PNT.CT
013432  000406                  BR OSTRE
013434  162700  013704  NCONF:  SUB #ERRC,R0            ;GET INCREMENT INTO TABLE
013440  006300                  ASL R0                  ;DOUBLE TO WORD COUNT
013442  004770  013716          CALL @ERRD(R0)          ;DISPATCH TO PRINT ROUTINE
013446  000755                  BR OSTRNG               ;GET NEXT
013450  000207          OSTRE:  RETURN
```

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

013452  112200          CON.QU: MOVB (R2)+,R0           ;GET CHARACTER
013454  120027  000042          CMPB R0,#''             ;CHECK IF ENDING QUOTE
013460  001403                  BEQ CON.QX              ;IF SO, GO GET NEXT CONTROL CHARACTER
013462                          PRINT R0                ;PRINT THE CHARACTER
013462  004737  014476                                                  CALL CPNT
013466  000771                  BR CON.QU               ;CONTINUE PRINTING
013470  000207          CON.QX: RETURN

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

013472  004737  014156  CON.A:  CALL GETCNT              ;GET COUNT OF CHARACTERS
013476          CON.A1: PRINT (R4)+                     ;PRINT THE CHARACTER
013476  112400                                                          MOVB (R4)+,R0
013500  004737  014476                                                  CALL CPNT
013504  005301                  DEC R1                  ;COUNT THE CHARACTERS
013506  001373                  BNE CON.A1              ;PRINT UNTIL COUNT REACHES ZERO
013510  032704  000001          BIT #1,R4               ;CHECK IF R4 NOW ODD
013514  001401                  BEQ CON.A2
013516  005204                  INC R4                  ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
013520  000207          CON.A2: RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.

013522  012701  000012  CON.D:  MOV #10.,R1             ;LOAD RADIX
013526  004737  014234          CALL PNTNUM             ;PRINT NUMBER
013532  000207                  RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.

013534  012701  000020  CON.H:  MOV #16.,R1             ;LOAD RADIX
013540  004737  014234          CALL PNTNUM             ;PRINT NUMBER
013544  000207                  RETURN                  ;NOW GET NEXT CONTROL CHARACTER
```

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

GLOBAL SUBROUTINES SECTION

```
                                ;ERROR ROUTINE DISPATCH TABLE
                                
013662  013736                  ERRRTB: .WORD CALRE                     ;NOT USED
013664  013736                          .WORD CALRE                     ;NOT USED
013666  013736                          .WORD CALRE                     ;NOT USED
013670  013750                          .WORD CALR4                     ;PRINT BASIC LINE WITHOUT UDA ADDRESS
013672  014024                          .WORD CALR5                     ;PRINT BASIC LINE WITH UDA ADDRESS
013674  014102                          .WORD CALR6                     ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
013676  014116                          .WORD CALR7                     ;PRINT  "REPLACE UDA MODULE M7161"
013700  014134                          .WORD CALR8                     ;PRINT  "  UDASA CONTAINS  XXXXXX"
013702  014152                          .WORD CALR9                     ;REPRINT LAST NUMBER
        000011                          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
```

GLOBAL SUBROUTINES SECTION

```
                                ;HERE IS FIRST TABLE

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

013704                          ERRC:   BUILD
013704  042                             .BYTE '"
013705  101                             .BYTE 'A
013706  104                             .BYTE 'D
013707  110                             .BYTE 'H
013710  117                             .BYTE 'O
013711  116                             .BYTE 'N
013712  122                             .BYTE 'R
013713  123                             .BYTE 'S
013714  000                             .BYTE 0                 ;FOLLOW WITH A NULL BYTE
                                        .EVEN

                                ;HERE IS SECOND TABLE

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

013716                          ERRD:   BUILD
013716  013452                          .WORD CON.QU
013720  013472                          .WORD CON.A
013722  013522                          .WORD CON.D
013724  013534                          .WORD CON.H
013726  013546                          .WORD CON.O
013730  013560                          .WORD CON.N
013732  013602                          .WORD CON.R
013734  013640                          .WORD CON.S
```

GLOBAL SUBROUTINES SECTION

```
                                ;PRE-PROGRAMMED ROUTINES 1, 2 AND 3
                                ;NOT USED - PRINTS ERROR MESSAGE

013736                          CALRE:  PNTF ERRME1                     ;PRINT ERROR MESSAGE
013736  004137  014650                                                          JSR R1,LPNTF
013742  003654                                                                  .WORD ERRME1
013744  000000                                                                  .WORD PNT.CT
013746  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

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

013750                                  CALR4:  PNTB BASLN,#BASN0,#BAS,#BAS,#BAS
013750  012746  004137                                                          MOV #BAS,-(SP)
013754  012746  004137                                                          MOV #BAS,-(SP)
013760  012746  004137                                                          MOV #BAS,-(SP)
013764  012746  004064                                                          MOV #BASN0,-(SP)
013770  004137  014660                                                          JSR R1,LPNTB
013774  004140                                                                  .WORD BASLN
013776  000010                                                                  .WORD PNT.CT
014000  004737  016636                          CALL RNTIME
014004                                          PRINT #CR
014004  112700  000015                                                          MOVB #CR,R0
014010  004737  014476                                                          CALL CPNT
014014  012737  014576  003226                  MOV #PX,PTYPE
014022  000207                                  RETURN
```

GLOBAL SUBROUTINES SECTION

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

 5 014024                               CALR5:  PNTB BASLN,#BASN0,#BASL2,(R5),#BAS,#BAS
   014024  012746  004137                                                       MOV #BAS,-(SP)
   014030  012746  004137                                                       MOV #BAS,-(SP)
   014034  011546                                                               MOV (R5),-(SP)
   014036  012746  004103                                                       MOV #BASL2,-(SP)
   014042  012746  004064                                                       MOV #BASN0,-(SP)
   014046  004137  014660                                                       JSR R1,LPNTB
   014052  004140                                                              .WORD BASLN
   014054  000012                                                              .WORD PNT.CT
 6 014056  004737  016636                       CALL RNTIME
 7 014062                                       PRINT #CR
   014062  112700  000015                                                       MOVB #CR,R0
   014066  004737  014476                                                       CALL CPNT
 8 014072  012737  014576  003226               MOV #PX,PTYPE
 9 014100  000207                              RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;PRE-PROGRAMMED ROUTINE 6
                                ;CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
                                
014102                          CALR6:  PUSH R2                 ;SAVE CURRENT STRING POINTER
014102  010246                                                          MOV R2,-(SP)
014104  012402                          MOV (R4)+,R2            ;GET NEW STRING POINTER
014106  004737  013402                  CALL OSTRNG             ;OUTPUT USING THIS STRING
014112                                  POP R2                  ;GET OLD POINTER BACK
014112  012602                                                          MOV (SP)+,R2
014114  000207                          RETURN                  ;NOW CONTINUE THE OLD STRING
```

GLOBAL SUBROUTINES SECTION

```
                                        ;PRE-PROGRAMMED ROUTINE 7
                                        ;PRINT "REPLACE UDA MODULE M7161"

4 014116                                CALR7:  PUSH R2
  014116  010246                                                        MOV R2,-(SP)
5 014120  012702  010204                        MOV #XFRU,R2
6 014124  004737  013402                        CALL OSTRNG
7 014130                                        POP R2
  014130  012602                                                        MOV (SP)+,R2
8 014132  000207                                RETURN
```

GLOBAL SUBROUTINES SECTION

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

014134                                  CALR8:  PUSH R2                         MOV R2,-(SP)
014134  010246
014136  012702  010153                          MOV #XSA,R2
014142  004737  013402                          CALL OSTRNG
014146                                          POP R2                          MOV (SP)+,R2
014146  012602
014150  000207                                  RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;       REPRINT LAST NUMBER
                                ;               R4 -> TABLE
                                ;
 014152  005744                 CALR9:  TST     -(R4)
 014154  000207                         RETURN
```

```
                                ;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

014156                          GETCNT: PUSH R0                                 MOV R0,-(SP)
014156  010046
014160  005001                          CLR R1                          ;START WITH ZERO COUNT
014162  121227  000060          GETCNX: CMPB (R2),#'0                   ;CHECK IF CHARACTER A DIGIT
014166  103415                          BLO GETCDN                      ;BRANCH IF LOWER THAN ZERO
014170  121227  000071                  CMPB (R2),#'9
014174  101012                          BHI GETCDN                      ;BRANCH IF HIGHER THAN NINE
014176  006301                          ASL R1                          ;MULTIPLY NUMBER BY 10
014200  010100                          MOV R1,R0                       ; SAVE 2N
014202  006301                          ASL R1                          ; COMPUTE 4N
014204  006301                          ASL R1                          ; COMPUTE 8N
014206  060001                          ADD R0,R1                       ; 8N + 2N = 10N
014210  112200                          MOVB (R2)+,R0                   ;GET DIGIT FROM STING
014212  162700  000060                  SUB #'0,R0                      ;GET RID OF ASCII
014216  060001                          ADD R0,R1                       ;ADD TO NUMBER
014220  000760                          BR GETCNX                       ;GO TO NEXT CHARACTER
014222  005701                  GETCDN: TST R1                          ;CHECK IF NUMBER IS ZERO
014224  001001                          BNE GETCXX                      ;IF ZERO, CHANGE
014226  005201                          INC R1                          ; TO DEFAULT OF ONE
014230                          GETCXX: POP R0                                  MOV (SP)+,R0
014230  012600
014232  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
                                ;
014234  010100                  PNTNUM: MOV R1,R0                       ;SAVE RADIX
014236  004737  014156                  CALL GETCNT                     ;GET COUNT OF BITS
014242                          PNTNUS: PUSH <R2,R3,R5>
014242  010246                                                                  MOV R2,-(SP)
014244  010346                                                                  MOV R3,-(SP)
014246  010546                                                                  MOV R5,-(SP)
014250  012403                          MOV (R4)+,R3                    ;GET ONE PARAMETER WORD
014252  005005                          CLR R5                          ;CLEAR STORAGE FOR OTHER
014254  020127  000020                  CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
014260  003401                          BLE 1$
014262  012405                          MOV (R4)+,R5                    ;YES, GET SECOND PARAMETER WORD
014264                          1$:     PUSH R4
014264  010446                                                                  MOV R4,-(SP)
014266  010504                          MOV R5,R4                       ;PUT HIGH WORD IN R4
014270  012702  000020                  MOV #16.,R2                     ;COMPUTE BITS NOT WANTED
014274  160102                          SUB R1,R2                       ;BY SUBTRACTING BITS TO USE
014276  002002                          BGE 2$                          ;FROM 16.
014300  062702  000020                  ADD #16.,R2                     ;IF NEGATIVE, ADD 16 FOR FIRST WORD
014304  001414                  2$:     BEQ 6$                          ;IF ZERO, NO BITS NEED BE CLEARED
014306  012705  100000                  MOV #BIT15,R5                   ;START MASK WITH SIGN BIT SET
014312  005302                  3$:     DEC R2                          ;COUNT BITS IN MASK
014314  001402                          BEQ 4$
014316  006205                          ASR R5                          ;SHIFT MORE BITS TO RIGHT
014320  000774                          BR 3$
014322  020127  000020          4$:     CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
014326  003402                          BLE 5$
014330  040504                          BIC R5,R4                       ;YES, CLEAR IN HIGH WORD
014332  000401                          BR 6$
014334  040503                  5$:     BIC R5,R3                       ;NO, CLEAR IN LOW WORD
014336  004737  014754          6$:     CALL DIVIDE                     ;DIVIDE BY RADIX IN R0
014342                                  PUSH R5                         ;PUSH REMAINDER ON STACK
014342  010546                                                                  MOV R5,-(SP)
014344  005202                          INC R2                          ;COUNT DIGITS ON STACK
014346  005703                          TST R3                          ;CHECK IF QUOTIENT IS ZERO
014350  001372                          BNE 6$
014352  005704                          TST R4
014354  001370                          BNE 6$
```

GLOBAL SUBROUTINES SECTION

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

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

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

014650  012737  014526  003226  LPNTF:  MOV #PF,PTYPE
014656  000413                          BR LPNT
014660  012737  014552  003226  LPNTB:  MOV #PB,PTYPE
014666  000407                          BR LPNT
014670  012737  014576  003226  LPNTX:  MOV #PX,PTYPE
014676  000403                          BR LPNT
014700  012737  014622  003226  LPNTS:  MOV #PS,PTYPE
014706                          LPNT:   PUSH <R2,R3,R4,R5>
014706  010246                                                  MOV R2,-(SP)
014710  010346                                                  MOV R3,-(SP)
014712  010446                                                  MOV R4,-(SP)
014714  010546                                                  MOV R5,-(SP)
014716  012102                          MOV (R1)+,R2            ;GET ADDRESS OF STRING
014720  010604                          MOV SP,R4               ;COMPUTE ADDRESS OF ARGUMENTS
014722  062704  000012                  ADD #10.,R4             ;  WHICH ARE NOW ON STACK (IF ANY)
014726                                  PUSH R1                 ;SAVE RETURN ADDRESS
014726  010146                                                  MOV R1,-(SP)
014730  004737  013402                  CALL OSTRNG             ;PRINT THE FORMATTED MESSAGE
014734                                  POP <R0,R5,R4,R3,R2,R1> ;RESTORE ALL REGISTERS
014734  012600                                                  MOV (SP)+,R0
014736  012605                                                  MOV (SP)+,R5
014740  012604                                                  MOV (SP)+,R4
014742  012603                                                  MOV (SP)+,R3
014744  012602                                                  MOV (SP)+,R2
014746  012601                                                  MOV (SP)+,R1
014750  062006                          ADD (R0)+,SP            ;ADJUST STACK POINTER OVER ARGUMENTS
014752  000110                          JMP @R0                 ;RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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
                                        
014754                                  DIVIDE: PUSH R2                                 MOV R2,-(SP)
014754  010246
014756  012702  000040                         MOV #32.,R2                     ;SET UP SHIFT COUNT
014762  005005                                 CLR R5                          ;START WITH ZERO REMAINDER
014764  006303                          1$:    ASL R3                          ;SHIFT LEFT INTO R5
014766  006104                                 ROL R4
014770  006105                                 ROL R5
014772  020005                                 CMP R0,R5                       ;WILL DIVISOR GO INTO REMAINDER
014774  101002                                 BHI 2$                          ;ONLY SUBTRACT IF IT WILL
014776  160005                                 SUB R0,R5                       ;SUBTRACT DIVISOR
015000  005203                                 INC R3                          ;PUT A ONE INTO QUOTIENT
015002  005302                          2$:    DEC R2                          ;COUNT THE SHIFTS
015004  001367                                 BNE 1$
015006                                         POP R2                                  MOV (SP)+,R2
015006  012602
015010  000207                                 RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;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
                                        ;
015012  013701  002164          LOADDM: MOV DMPROG,R1                   ;GET STORAGE ADDRESS OF DM PROGRAM
015016  116165  000021  000044          MOVB DMTMO(R1),C.TOT(R5)        ;GET TIMEOUT VALUE
015024  105065  000045                  CLRB C.TOT+1(R5)
015030  016504  000004                  MOV C.VEC(R5),R4                ;GET VECTOR OF UDA
015034                                  AND CT.VEC,R4
015034  042704  177000                                                  BIC #^C<CT.VEC>,R4
015040  010501                          MOV R5,R1                       ;GET INTERRUPT SERVICE LINK
015042  062701  000010                  ADD #C.JSR,R1
015046                                  SETVEC R4,R1,#PRI07             ;SET UP INTERRUPT VECTOR
015046  012746  000340                                                  MOV     #PRI07,-(SP)
015052  010146                                                          MOV     R1,-(SP)
015054  010446                                                          MOV     R4,-(SP)
015056  012746  000003                                                  MOV     #3,-(SP)
015062  104437                                                          TRAP    C$SVEC
015064  062706  000010                                                  ADD     #10,SP
                                                                ;INITIALIZE UDA WITH SMALLEST
015070  004737  015674                  CALL UDAINT                     ; RING BUFFER AND INTERRUPTS ENABLED
015074  001445                          BEQ LOADER                      ;BRANCH IF AN ERROR
```

GLOBAL SUBROUTINES SECTION

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

GLOBAL SUBROUTINES SECTION

```
                                ;UDA FAILED TO DOWNLINE LOAD DM PROGRAM

                                LOADE1: ERRDF 34,,ERR034
015200                                                          TRAP    C$ERDF
015200  104455                                                 .WORD   34
015202  000042                                                 .WORD   0
015204  000000                                                 .WORD   ERR034
015206  011060
015210  000264                  LOADER: SEZ                   ;SET Z TO INDICATE ERROR OCCURRED
015212  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
                                        ;
015214                                  BLDCMD: PUSH <R1,R0>
015214  010146                                                                          MOV R1,-(SP)
015216  010046                                                                          MOV R0,-(SP)
015220  016504  000016                          MOV C.RING(R5),R4           ;GET ADDRESS OF HOST COMM AREA
015224  010400                                  MOV R4,R0                   ;COPY TO R0
015226  062700  000100                          ADD #HC.CEV,R0              ;COMPUTE ADDRESS OF COMMAND ENVELOPE
015232  012720  000060                          MOV #HC.PSZ,(R0)+           ;LOAD PACKET LENGTH
015236  012701  001000                          MOV #DUP,R1                 ;LOAD DIAG CIRCUIT IDENTIFIER
015242  022716  000031                          CMP #OP.MWR,(SP)            ;IF CODE IS MAINTENANCE WRITE
015246  001002                                  BNE BLDC0                   ;  GET OTHER CIRCUIT IDENTIFIER
015250  012701  177777                          MOV #DIAG,R1
015254  010120                          BLDC0:  MOV R1,(R0)+                ;PUT IDENTIFIER INTO PACKET
015256  012701  000030                          MOV #<HC.PSZ>/2,R1          ;GET WORDS TO CLEAR
015262  005020                          BLDC1:  CLR (R0)+                   ;CLEAR PACKET
015264  005301                                  DEC R1
015266  001375                                  BNE BLDC1
015270                                          POP HC.CPK+P.OPCD(R4)       ;PUT OPCODE IN PACKET
015270  012664  000114                                                              MOV (SP)+,HC.CPK+P.OPCD(R4)
015274                                          POP R1                      ;RESTORE R1
015274  012601                                                                      MOV (SP)+,R1
015276  000207                                  RETURN
```

```
                                        ;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


 015300                                 SNDCMD: PUSH <R0,R1>
 015300  010046                                                                        MOV R0,-(SP)
 015302  010146                                                                        MOV R1,-(SP)
 015304  016504  000016                        MOV C.RING(R5),R4               ;LOAD R4 WITH HOST COMM AREA ADDRESS
 015310  005265  000052                        INC C.REF(R5)                   ;INCREMENT CMD REFERENCE NUMBER
 015314  016564  000052  000104                MOV C.REF(R5),HC.CPK+P.CRF(R4)  ;PUT IN PACKET
 015322  012764  140000  000006                MOV #RG.OWN+RG.FLG,HC.MCT(R4)   ;MARK MESSAGE PACKET AVAILABLE
 015330  012764  100000  000012                MOV #RG.OWN,HC.CCT(R4)          ;MARK COMMAND TO UDA
 015336  005775  000000                        TST @(R5)                       ;TELL UDA COMMAND IS THERE
 015342  052765  000004  000014                BIS #CT.CMD,C.FLG(R5)           ;MARK COMMAND ISSUED
 015350                                        POP <R1,R0>
 015350  012601                                                                        MOV (SP)+,R1
 015352  012600                                                                        MOV (SP)+,R0
 015354  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
                                        ;
015356                                  CLRBUF: PUSH <R0,R1>
015356  010046                                                                 MOV R0,-(SP)
015360  010146                                                                 MOV R1,-(SP)
015362  060400                                  ADD R4,R0                       ;ADD START OF HOST COMM AREA TO OFFSET
015364  010064  000124                         MOV R0,HC.CPK+P.UADR(R4)        ;PUT BUFFER ADDRESS IN COMMAND PACKET
015370  012764  000122  000120                 MOV #HC.BSZ,HC.CPK+P.BCNT(R4)   ;PUT SIZE OF BUFFER IN COMMAND PACKET
015376  010004                                  MOV R0,R4                       ;PUT BUFFER ADDRESS IN R4
015400  012701  000051                         MOV #HC.BSZ/2,R1                ;GET SIZE OF BUFFER IN WORDS
015404  005020                          CLRBFL: CLR (R0)+                       ;CLEAR ALL THE WORDS
015406  005301                                  DEC R1
015410  001375                                  BNE CLRBFL
015412                                          POP <R1,R0>
015412  012601                                                                 MOV (SP)+,R1
015414  012600                                                                 MOV (SP)+,R0
015416  000207                                  RETURN
```

```
                                        ;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
                                        ;
015420                                  WAITMS: PUSH <R0,R1>
015420  010046                                                                    MOV R0,-(SP)
015422  010146                                                                    MOV R1,-(SP)
015424  012700  000036                       MOV #30.,R0             ;SET TIME OUT VALUE OF 30 SECONDS
015430  010501                               MOV R5,R1               ;POINT TO TIME OUT COUNTER
015432  062701  000040                       ADD #C.TO,R1
015436  004737  015612                       CALL SETTO
015442  011500                          1$:  MOV (R5),R0             ;GET ADDRESS OF UDAIP REGISTER
015444  032765  000010  000014               BIT #CT.MSG,C.FLG(R5)   ;LOOK IF INTERRUPT OCCURRED
015452  001030                               BNE 3$                  ;BRANCH IF SO
015454  016001  000002                       MOV 2(R0),R1            ;LOOK AT UDASA REGISTER
015460  001034                               BNE 4$                  ;BRANCH IF ERROR CODE PRESENT
015462                                       BREAK
015462  104422                                                                    TRAP    C$BRK
015464  005737  003212                       TST KW.CSR              ;SEE IF A CLOCK ON SYSTEM
015470  001764                               BEQ 1$
015472  023765  003224  000042               CMP KW.EL+2,C.TOH(R5)   ;CHECK IF TIMEOUT HAS HAPPENED
015500  101005                               BHI 2$
015502  001357                               BNE 1$
015504  023765  003222  000040               CMP KW.EL,C.TO(R5)
015512  103753                               BLO 1$
015514                                  2$:  ERRDF 36,,ERR036
015514  104455                                                                    TRAP    C$ERDF
015516  000044                                                                    .WORD   36
015520  000000                                                                    .WORD   0
015522  011066                                                                    .WORD   ERR036
015524                                       POP <R1,R0>
015524  012601                                                                    MOV (SP)+,R1
015526  012600                                                                    MOV (SP)+,R0
015530  000264                               SEZ
015532  000207                               RETURN
```

```
      1 015534  042765  000010  000014  3$:     BIC #CT.MSG,C.FLG(R5)          ;CLEAR MESSAGE RECEIVED FLAG
      2 015542                                  POP <R1,R0>
        015542  012601                                                          MOV (SP)+,R1
        015544  012600                                                          MOV (SP)+,R0
      3 015546  000244                          CLZ                           ;GIVE NO ERROR RETURN
      4 015550  000207                          RETURN
      5 015552                            4$:   ERRDF 37,,ERR037
        015552  104455                                                        TRAP    C$ERDF
        015554  000045                                                        .WORD   37
        015556  000000                                                        .WORD   0
        015560  011100                                                        .WORD   ERR037
      6 015562                                  POP <R1,R0>
        015562  012601                                                          MOV (SP)+,R1
        015564  012600                                                          MOV (SP)+,R0
      7 015566  000264                          SEZ
      8 015570  000207                          RETURN
```

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

015602                          BGNSRV UDASRV
015602                                                                  UDASRV::
015602  052710  000010                  BIS #CT.MSG,(R0)                ;SET CT.MSG
015606                                   POP R0                         ;RESTORE R0
015606  012600                                                                  MOV (SP)+,R0
015610                          ENDSRV
015610                                                                  L10031:
015610  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

015612                                  SETTO:  PUSH <R2,R3>
015612  010246                                                                    MOV R2,-(SP)
015614  010346                                                                    MOV R3,-(SP)
015616  005002                                  CLR R2                  ;CLEAR PRODUCT
015620  013703  003220                         MOV KW.HZ,R3            ;GET MULTIPLICAND
015624  006200                          SETO0:  ASR R0                  ;SHIFT MULTIPLIER TO RIGHT
015626  103001                                  BCC SETO1               ;IF A ONE BIT SHIFTED OUT
015630  060302                                  ADD R3,R2               ;  ADD MULTIPLICAND TO PRODUCT
015632  006303                          SETO1:  ASL R3                  ;DOUBLE THE MULTIPLICAND
015634  005700                                  TST R0
015636  001372                                  BNE SETO0               ;CONTINUE UNTIL MULTIPLIER IS ZERO

                                        ;GET CURRENT TIME

015640  013700  003222                  SETO2:  MOV KW.EL,R0            ;GET TIME
015644  013703  003224                          MOV KW.EL+2,R3
015650  020037  003222                          CMP R0,KW.EL            ;IF CHANGED DURING RETRIEVAL
015654  001371                                  BNE SETO2               ;  GET IT AGAIN

                                        ;ADD TIME TIL TIMEOUT

015656  060200                                  ADD R2,R0               ;ADD
015660  005503                                  ADC R3

                                        ;PUT RESULT IN STORAGE

015662  010021                                  MOV R0,(R1)+
015664  010311                                  MOV R3,(R1)

015666                                          POP <R3,R2>
015666  012603                                                                    MOV (SP)+,R3
015670  012602                                                                    MOV (SP)+,R2
015672  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

015674  016502  000016  UDAINT: MOV C.RING(R5),R2                    ;GET FIRST ADDRESS OF RING BUFFER
015700  012703  000006          MOV #<HC.RSZ*2+HC.ISZ>/2,R3                ;GET SIZE OF RING BUFFER
015704  012722  177777  UDAI1L: MOV #-1,(R2)+                  ;WRITE ONES TO BUFFER
015710  005303                  DEC R3                          ;COUNT THE WORDS IN BUFFER
015712  003374                  BGT UDAI1L                      ;LOOP UNTIL ENTIRE BUFFER WRITTEN

                                ;DO THE INITIALIZATION

015714  004737  016110          CALL UDAIST                     ;DO FIRST THREE STEPS
015720  103471                  BCS UDAIEX                      ;GET OUT IF UDA MICROCODE REPORTED FAILURE
015722  012364  000002          MOV (R3)+,2(R4)                 ;WRITE NEXT WORD TO UDASA REGISTER
015726  012703  000310          MOV #200.,R3                    ;GET TRY COUNTER
015732  016402  000002  UDAI1A: MOV 2(R4),R2                    ;LOOK AT UDASA
015736  001410                  BEQ UDAI1C
015740  100005                  BPL UDAI1B
015742                          ERRDF 24,,ERR024
015742  104455                                                          TRAP    C$ERDF
015744  000030                                                          .WORD   24
015746  000000                                                          .WORD   0
015750  010754                                                          .WORD   ERR024
015752  000454                  BR UDAIEX
015754  005303          UDAI1B: DEC R3
015756  001365                  BNE UDAI1A
015760  010264  000002  UDAI1C: MOV R2,2(R4)                  ;WRITE 0 TO UDASA (PURGE)
015764  011402                  MOV (R4),R2                     ;READ FROM UDAIP (POLL)
015766  004737  016434          CALL UDARSP                     ;WAIT FOR STEP OR ERROR BIT
015772  103444                  BCS UDAIEX                      ;GET OUT IF UDA MICROCODE REPORTED FAILURE
```

```
                                        ;CHECK HOST COMMUNICATION AREA FOR ALL ZEROS

                                        UDAI2:  MOV C.RING(R5),R2               ;GET FIRST ADDRESS OF RING BUFFER
 015774  016502  000016                       MOV #<HC.RSZ*2+HC.ISZ>/2,R3     ;GET SIZE OF RING BUFFER
 016000  012703  000006                 UDAI2L: TST (R2)+                       ;CHECK WORD IN BUFFER
 016004  005722                                 BNE UDAI2E                      ;GO TO ERROR REPORTER IF NOT ZERO
 016006  001003                                 DEC R3                          ;COUNT THE WORDS IN BUFFER
 016010  005303                                 BGT UDAI2L                      ;LOOP UNTIL ALL WORDS CHECKED
 016012  003374                                 BR UDAI3
 016014  000405

 016016                                 UDAI2E: ERRDF 23,,ERR023                ;REPORT BUFFER NOT CLEARED
 016016  104455                                                                         TRAP    C$ERDF
 016020  000027                                                                         .WORD   23
 016022  000000                                                                         .WORD   0
 016024  010670                                                                         .WORD   ERR023
 016026  000426                                 BR UDAIEX

                                        ;SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION

 016030                                 UDAI3:
 016030  016500  000006                       MOV C.BST(R5),R0                ;GET BURST VALUE
 016034  006300                               ASL R0                          ;SHIFT TO POSITION
 016036  006300                               ASL R0
 016040  052700  000001                       BIS #SA.GO,R0                   ;SET THE GO BIT
 016044  010064  000002                       MOV R0,2(R4)                    ;SEND TO UDA
 016050  016501  000016                       MOV C.RING(R5),R1
 016054  010161  000004                       MOV R1,HC.MSG(R1)
 016060  062761  000020  000004              ADD #HC.MPK,HC.MSG(R1)
 016066  010161  000010                       MOV R1,HC.CMD(R1)
 016072  062761  000104  000010              ADD #HC.CPK,HC.CMD(R1)
 016100  000244                               CLZ                             ;CLEAR Z AS NO ERROR INDICATION
 016102  000207                               RETURN

                                        ;ERROR RETURN

 016104  000264                       UDAIEX: SEZ                             ;SET Z TO INDICATE ERROR OCCURRED
 016106  000207                               RETURN
```

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

                                        ;LOAD TABLE OF DATA TO SEND TO UDASA REGISTER

016110                                  UDAIST: BREAK
016110  104422                                                                  TRAP    C$BRK
016112                                          PUSH R1
016112  010146                                                                  MOV R1,-(SP)
016114  016504  000004                        MOV C.VEC(R5),R4
016120                                          AND CT.VEC,R4
016120  042704  177000                                                          BIC #^C<CT.VEC>,R4
016124  006204                                  ASR R4
016126  006204                                  ASR R4
016130  052704  100000                        BIS #SA.STP,R4          ;SET STEP BIT IN DATA WORD
016134  010437  016326                        MOV R4,UDAID1           ;LOAD INTERRUPT VECTOR
016140  016537  000016  016332                MOV C.RING(R5),UDAID2   ;LOAD MEMORY ADDRESS
016146  062737  000004  016332                ADD #HC.MSG,UDAID2      ;  OF FIRST RESPONSE RING

                                        ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER

016154  016504  000000                        MOV C.UADR(R5),R4       ;GET ADDRESS OF UDAIP REGISTER
016160  005037  002202                        CLR NXMAD               ;CLEAR MEMORY ERROR FLAG
016164                                          SETVEC #4,#NXMI,#PRI07  ;SET UP VECTOR 4
016164  012746  000340                                                          MOV     #PRI07,-(SP)
016170  012746  015572                                                          MOV     #NXMI,-(SP)
016174  012746  000004                                                          MOV     #4,-(SP)
016200  012746  000003                                                          MOV     #3,-(SP)
016204  104437                                                                  TRAP    C$SVEC
016206  062706  000010                                                          ADD     #10,SP
016212  005764  000002                TST 2(R4)               ;ACCESS UDASA REGISTER
016216  005014                          CLR (R4)              ;WRITE TO UDAIP
016220                                  CLRVEC #4             ;GIVE UP THE VECTOR
016220  012700  000004                                                          MOV     #4,R0
016224  104436                                                                  TRAP    C$CVEC
016226  005737  002202                TST NXMAD               ;SEE IF A MEMORY ERROR OCCURRED
016232  001406                          BEQ UDAISG
016234                                  ERRDF 20,,ERR020
016234  104455                                                                  TRAP    C$ERDF
016236  000024                                                                  .WORD   20
016240  000000                                                                  .WORD   0
016242  010556                                                                  .WORD   ERR020
016244  000261                          SEC
016246  000424                          BR UDAISE
```

```
                                        ;SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
                                        
016250  012737  004000  016572  UDAISG: MOV #SA.S1,UDARSD               ;STORE RESPONSE MASK
016256  012703  016324                  MOV #UDAIDT,R3                  ;AND INDEX TO TABLE
                                        
                                        ;WAIT FOR AND CHECK RESPONSE DATA
                                        
016262  004737  016434          UDAISL: CALL UDARSP                     ;WAIT FOR STEP OR ERROR BITS
016266  103414                          BCS UDAISE                      ;EXIT IF ERROR
016270  004733                          CALL @(R3)+                     ;CALL RESPONSE CHECKER FOR STEP
016272  103412                          BCS UDAISE                      ;GET OUT IF ERROR
016274  006337  016572                  ASL UDARSD                      ;SHIFT TO NEXT STEP BIT
016300  032737  040000  016572          BIT #SA.S4,UDARSD               ;CHECK IF NOW AT STEP 4
016306  001003                          BNE UDAISX                      ;GET OUT IF SO
016310  012364  000002                  MOV (R3)+,2(R4)                 ;WRITE DATA TO UDASA REGISTER
016314  000762                          BR UDAISL                       ;STAY IN LOOP
                                        
016316  000241          UDAISX: CLC                                     ;CLEAR CARRY FOR NO ERROR INDICATION
016320                  UDAISE: POP R1
016320  012601                                                          MOV (SP)+,R1
016322  000207                          RETURN
```

```
                                        ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION

016324  016342                  UDAIDT: .WORD UDAIR1                    ;FIRST WORD RESPONSE CHECK ROUTINE
016326  000000                  UDAID1: .WORD 0                         ;FIRST WORD TO SEND TO UDASA
016330  016350                          .WORD UDAIR2                    ;SECOND WORD RESPONSE CHECK ROUTINE
016332  000000                  UDAID2: .WORD 0                         ;SECOND WORD TO SEND TO UDASA
016334  016370                          .WORD UDAIR3                    ;THIRD WORD RESPONSE CHECK ROUTINE
016336  100000                  UDAID3: .WORD SA.TST                    ;THIRD WORD TO SEND TO UDASA
016340  016406                          .WORD UDAIR4                    ;FOURTH WORD RESPONSE CHECK ROUTINE

                                        ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
                                        ;CHECK FOR PROPER CONTROLLER TYPE

016342  012701  004400          UDAIR1: MOV #SA.S1+SA.DI,R1             ;SET STEP ONE BIT
016346  000422                          BR UDAIRC                       ;NOW COMPARE

                                        ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
                                        ;CHECK FOR ECHO OF INTI AND VECTOR

016350  013701  016326          UDAIR2: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
016354  000301                          SWAB R1                         ;GET HIGH 8 BITS
016356  042701  177400                  BIC #177400,R1
016362  052701  010000                  BIS #SA.S2,R1                   ;SET STEP 2 BIT
016366  000412                          BR UDAIRC                       ;NOW COMPARE

                                        ;RESPONSE CHECK FOR THIRD WORD FROM UDASA
                                        ;CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS

016370  013701  016326          UDAIR3: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
016374  042701  177400                  BIC #177400,R1                  ;JUST LOW 8 BITS
016400  052701  020000                  BIS #SA.S3,R1                   ;SET STEP 3 BIT
016404  000403                          BR UDAIRC                       ;NOW COMPARE

                                        ;RESPONSE CHECK FOR FOURTH WORD FROM UDASA
                                        ;CHECK FOR ECHO OF PURGE AND LFAIL BITS

016406  010201                  UDAIR4: MOV R2,R1                       ;GET RESPONSE FROM UDA
016410  042701  137400                  BIC #^C<SA.S4+SA.MCV>,R1        ;KEEP MICROCODE VERSION AND STEP 4

                                        ;COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2

016414  020102                  UDAIRC: CMP R1,R2                       ;COMPARE THE DATA
016416  001405                          BEQ UDAIRX                      ;EXIT IF COMPARED CORRECTLY
016420                                  ERRDF 25,,ERR025                ;REPORT ERROR
016420  104455                                                                  TRAP    C$ERDF
016422  000031                                                                  .WORD   25
016424  000000                                                                  .WORD   0
016426  010770                                                                  .WORD   ERR025
016430  000261                          SEC
016432  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
                                        ;
016434                          UDARSP: PUSH R1
016434  010146                                                                  MOV R1,-(SP)
016436  052737  100000  016572          BIS #SA.ERR,UDARSD                      ;SET ERROR BIT IN MASK WORD
016444  012700  000012                  MOV #10.,R0                             ;SET UP FOR 10 SECOND TIMEOUT
016450  010501                          MOV R5,R1                               ;POINT TO COUNTER IN CONTROLLER TABLE
016452  062701  000040                  ADD #C.TO,R1
016456  004737  015612                  CALL SETTO
016462                                  POP R1
016462  012601                                                                  MOV (SP)+,R1
016464  033764  016572  000002  UDARS1: BIT UDARSD,2(R4)                       ;LOOK AT ERROR AND STEP BIT
016472  001024                          BNE UDARS2                              ;BRANCH IF EITHER SET
016474                                  BREAK
016474  104422                                                                  TRAP    C$BRK
016476  005737  003212                  TST KW.CSR                              ;SEE IF CLOCK ON SYSTEM
016502  001770                          BEQ UDARS1
016504  023765  003224  000042          CMP KW.EL+2,C.TOH(R5)                   ;CHECK IF TIME OUT OCCURRED
016512  101005                          BHI 1$
016514  001363                          BNE UDARS1
016516  023765  003222  000040          CMP KW.EL,C.TO(R5)
016524  103757                          BLO UDARS1
016526  016402  000002          1$:     MOV 2(R4),R2                            ;GET REGISTER CONTENTS
016532                                  ERRDF 22,,ERR022                        ;REPORT TIME OUT ERROR
016532  104455                                                                  TRAP    C$ERDF
016534  000026                                                                  .WORD   22
016536  000000                                                                  .WORD   0
016540  010622                                                                  .WORD   ERR022
016542  000407                          BR UDARSE
```

GLOBAL SUBROUTINES SECTION

```
                                        ;CHECK IF ERROR BIT SET
                                        
016544  016402  000002        UDARS2: MOV 2(R4),R2                    ;GET REGISTER CONTENTS
016550  100006                        BPL UDARSX                      ;EXIT IF ERROR NOT SET
016552                                ERRDF 21,,ERR021                ;REPORT ERROR INFO
016552  104455                                                                TRAP    C$ERDF
016554  000025                                                                .WORD   21
016556  000000                                                                .WORD   0
016560  010570                                                                .WORD   ERR021
016562  000261                UDARSE: SEC
016564  000207                        RETURN
                                        
                                        ;NORMAL EXIT
                                        
016566  000241                UDARSX: CLC                             ;CLEAR CARRY AS NO ERROR INDICATION
016570  000207                        RETURN
                                        
                                        ;LOCATION FOR STEP BIT MASK
                                        
016572  000000                UDARSD: .WORD 0                         ;LOAD BY CALLING ROUTINE
```

GLOBAL SUBROUTINES SECTION

```
                                ;CLOG
                                ;
                                ;COMPUTE LOGARITHMIC VALUE OF NUMBER TO BASE 2.
                                ;
                                ;INPUTS:
                                ;       R0 - LOGARITHM TO BE CONVERTED
                                ;OUTPUTS:
                                ;       R1 - VALUE OF 2 RAISED TO POWER OF INPUT NUMBER
                                ;
 016574                         CLOG:   PUSH R0
 016574  010046                                                                  MOV R0,-(SP)
 016576  005001                         CLR R1                 ;SET UP ZERO START VALUE
 016600  000261                         SEC                    ;WITH CARRY READY TO SHIFT IN
 016602  006101                 CLOGLP: ROL R1                 ;SHIFT TO LEFT
 016604  005300                         DEC R0                 ; UNTIL R0
 016606  100375                         BPL CLOGLP             ; GOES NEGATIVE
 016610                                 POP R0
 016610  012600                                                                  MOV (SP)+,R0
 016612  000207                         RETURN
```

```
                                ;KW11I
                                ;
                                ;CLOCK INTERRUPT SERVICE ROUTINE
                                ;
016614                          BGNSRV KW11I                                    KW11I::
016614
016614  062737  000001  003222          ADD #1,KW.EL                  ;COUNT THE INTERRUPT
016622  005537  003224                  ADC KW.EL+2
016626  012777  000105  164356          MOV #KWOUT.,@KW.CSR           ;RESTART THE CLOCK
016634                          ENDSRV                                          L10032:
016634
016634  000002                                                                  RTI
```

```
                                ;RNTIME
                                ;
                                ;PRINT RUNTIME
                                ;
                                ;INPUTS:
                                ;       KW.EL - CONTAINS ELAPSED TIME
                                ;       KW.HZ - HERTZ OF CLOCK
                                ;OUTPUTS:
                                ;       IF CLOCK ON SYSTEM:
                                ;          '' RUNTIME HH:MM:SS '' PRINTED
                                ;       IF NO CLOCK: ONE SPACE IS PRINTED
                                ;
016636  005737  003212          RNTIME: TST KW.CSR                      ;CHECK IF A CLOCK PRESENT
016642  001465                          BEQ RNTIMX                      ;BRANCH IF NOT
016644                                  PUSH <R0,R3,R4,R5>
016644  010046                                                                  MOV R0,-(SP)
016646  010346                                                                  MOV R3,-(SP)
016650  010446                                                                  MOV R4,-(SP)
016652  010546                                                                  MOV R5,-(SP)
016654  013703  003222                  MOV KW.EL,R3                    ;GET ELAPSED TIME
016660  013704  003224                  MOV KW.EL+2,R4
016664  013700  003220                  MOV KW.HZ,R0                    ;GET SPEED OF CLOCK
016670  004737  014754                  CALL DIVIDE                     ;COMPUTE SECONDS OF ELAPSED TIME
016674  012700  000074                  MOV #60.,R0                     ;NOW DIVIDE BY 60
016700  004737  014754                  CALL DIVIDE                     ; TO COMPUTE MINUTES
016704                                  PUSH R5                         ;SAVE REMAINDER AS SECONDS
016704  010546                                                                  MOV R5,-(SP)
016706  004737  014754                  CALL DIVIDE                     ;DIVIDE BY 60 AGAIN
016712                                  PNT RNTIM,R3                    ;PRINT HOURS
016712  010346                                                                  MOV R3,-(SP)
016714  004137  014706                                                          JSR R1,LPNT
016720  003620                                                                  .WORD RNTIM
016722  000002                                                                  .WORD PNT.CT
016724  020527  000011                  CMP R5,#9.                      ;IF MINUTES 9 OR LESS
016730  003004                          BGT 1$
016732                                  PRINT #'0                       ;PRINT A LEADING ZERO
016732  112700  000060                                                          MOVB #'0,R0
016736  004737  014476                                                          CALL CPNT
016742                          1$:     PNT RNTIM1,R5                   ;NOW PRINT MINUTES
016742  010546                                                                  MOV R5,-(SP)
016744  004137  014706                                                          JSR R1,LPNT
016750  003643                                                                  .WORD RNTIM1
016752  000002                                                                  .WORD PNT.CT
016754                                  POP R5                          ;GET SECONDS
016754  012605                                                                  MOV (SP)+,R5
016756  020527  000011                  CMP R5,#9.                      ;IF 9 OR LESS
016762  003004                          BGT 2$
016764                                  PRINT #'0                       ;PRINT A LEADING ZERO
016764  112700  000060                                                          MOVB #'0,R0
016770  004737  014476                                                          CALL CPNT
016774                          2$:     PNT RNTIM2,R5                   ;NOW PRINT SECONDS
016774  010546                                                                  MOV R5,-(SP)
016776  004137  014706                                                          JSR R1,LPNT
017002  003651                                                                  .WORD RNTIM2
017004  000002                                                                  .WORD PNT.CT
017006                                  POP <R5,R4,R3,R0>               ;HOURS IN R3
017006  012605                                                                  MOV (SP)+,R5
```

GLOBAL SUBROUTINES SECTION

```
                                                                        MOV (SP)+,R4
                                                                        MOV (SP)+,R3
                                                                        MOV (SP)+,R0
017010  012604
017012  012603
017014  012600
35 017016                       RNTIMX: PRINT <#' >             ;PRINT A SPACE
017016  112700  000040                                          MOVB #' ,R0
017022  004737  014476                                          CALL CPNT
36 017026  000207                       RETURN
37
38 017030                               ENDMOD
```

```
                                .SBTTL  PROTECTION TABLE

017030                                  BGNMOD

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

017030                                  BGNPROT
017030                                                                  L$PROT::

017030  177777                          -1                      ;OFFSET INTO P-TABLE FOR CSR ADDRESS
017032  177777                          -1                      ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
017034  177777                          -1                      ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

017036                                  ENDPROT
```

```
                        .SBTTL  INITIALIZE SECTION

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

017036                          BGNINIT
017036                                                          L$INIT::

                                READEF #EF.START                ;CHECK IF STARTED BY OPERATOR
017036  012700  000040                                          MOV     #EF.START,R0
017042  104447                                                 TRAP    C$REFG
                                BCOMPLETE INIT1         ;  IF NOT,
017044  103415                                                  BCS     INIT1
                                READEF #EF.RESTART
017046  012700  000037                                          MOV     #EF.RESTART,R0
017052  104447                                                 TRAP    C$REFG
                                BCOMPLETE INIT1
017054  103411                                                  BCS     INIT1
                                READEF #EF.CONTINUE
017056  012700  000036                                          MOV     #EF.CONTINUE,R0
017062  104447                                                 TRAP    C$REFG
                                BCOMPLETE INIT1
017064  103405                                                  BCS     INIT1
                                READEF #EF.PWR
017066  012700  000034                                          MOV     #EF.PWR,R0
017072  104447                                                 TRAP    C$REFG
                                BCOMPLETE INIT1
017074  103401                                                  BCS     INIT1
INITQT: DOCLN                                   ;  ABORT PROGRAM
017076  104444                                                 TRAP    C$DCLN
INIT1:  MOV #SO.FMT,R0  ;BUILD MODE WORD FROM SOFTWARE QUESTIONS
        BIT R0,SFPTBL   ;SEE IF REFORMAT
        BNE INIT2       ;  BRANCH IF SO
        MOV #SO.CNS,R0  ;SEE IF RECONSTRUCT
        BIT R0,SFPTBL
        BNE INIT2       ;  BRANCH IF SO
        ASL R0  ;SEE IF RESTORE
                ASSUME SO.STR EQ SO.CNS*2
        BIT R0,SFPTBL
        BEQ INITQT      ;IF NOT, ABORT PROGRAM
INIT2:  MOV R0,MODE     ;SAVE MODE FLAGS
        TST DATEQ       ;SEE IF ALREADY ASKED FOR DATE
        BNE INIT3
        CALL DATE       ;IF NOT, GET IT NOW
INIT3:  BRESET                                  ;RESET ALL UNITS
017152  104433                                                 TRAP    C$RESET
                        MEMORY FFREE                    ;RESET START OF FREE MEMORY
017154  104431                                                 TRAP    C$MEM
017156  010037  002146                                          MOV     R0,FFREE
017162  017737  162760  002150  MOV @FFREE,FSIZE               ;RESET SIZE OF FREE MEMORY

;INITIALIZE CLOCK

000105           KWOUT.=105                     ;DATA TO SEND TO KW11 TO START CLOCK
017170  005037  003212          CLR KW.CSR      ;MARK CLOCK AS NOT ON SYSTEM
```

INITIALIZE SECTION

```
     41 017174  005037  003222                    CLR KW.EL                       ;CLEAR ELAPSED TIME
     42 017200  005037  003224                    CLR KW.EL+2
     43 017204                                    CLOCK L,R0                      ;SEE IF AN L CLOCK PRESENT
        017204  012700  000114                                                        MOV     #'L,R0
        017210  104462                                                               TRAP    C$CLCK
     44 017212                                    BCOMPLETE KYES
        017212  103413                                                               BCS     KYES
     45 017214                                    CLOCK P,R0                      ;SEE IF A P CLOCK PRESENT
        017214  012700  000120                                                        MOV     #'P,R0
        017220  104462                                                               TRAP    C$CLCK
     46 017222                                    BCOMPLETE KYES
        017222  103407                                                               BCS     KYES
     47 017224  005037  003212                    CLR KW.CSR                      ;IF NEITHER, CLEAR CSR STORAGE WORD
     48 017230                                    PNTF NOCLOCK
        017230  004137  014650                                                        JSR R1,LPNTF
        017234  004007                                                                .WORD NOCLOCK
        017236  000000                                                               .WORD PNT.CT
     49 017240  000426                           BR KNO
     50 017242  012037  003212          KYES:     MOV (R0)+,KW.CSR                ;STORE DATA RETURNED
     51 017246  012037  003214                    MOV (R0)+,KW.BRL
     52 017252  012037  003216                    MOV (R0)+,KW.VEC
     53 017256  012037  003220                    MOV (R0)+,KW.HZ
     54 017262                                    SETVEC KW.VEC,#KW11I,KW.BRL     ;SET THE VECTOR
        017262  013746  003214                                                        MOV     KW.BRL,-(SP)
        017266  012746  016614                                                        MOV     #KW11I,-(SP)
        017272  013746  003216                                                        MOV     KW.VEC,-(SP)
        017276  012746  000003                                                        MOV     #3,-(SP)
        017302  104437                                                               TRAP    C$SVEC
        017304  062706  000010                                                        ADD     #10,SP
     55 017310  012777  000105  163674             MOV #KWOUT.,@KW.CSR             ;START THE CLOCK
     56 017316                          KNO:
```

INITIALIZE SECTION

```
                                        ;INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
                                        ;
017316  013737  002146  002156            MOV FFREE,CTABS             ;STORE START OF CONTROLLER TABLES
017324  005077  162626                    CLR @CTABS                  ;ZEROS MARKS END CONTROLLER TABLES
017330  005037  002160                    CLR CTRLRS                  ;CLEAR CONTROLLER COUNT
                                        ;
                                        ;GET A P-TABLE FROM DRS
                                        ;
017334  005002                    INIT4:  CLR R2                      ;LOGICAL UNIT NUMBER IN R2
017336                                    GPHARD R2,R0                ;GET POINTER TO A P-TABLE
017336  010200                                                            MOV     R2,R0
017340  104442                                                            TRAP    C$GPHRD
017342                                    BNCOMPLETE NXTTAB           ;IGNORE IF NO TABLE RETURNED
017342  103075                                                            BCC     NXTTAB
                                        ;
                                        ;SEE IF A CONTROLLER TABLE ALREADY EXISTS FOR CONTROLLER IN P-TABLE
                                        ;
017344  013703  002156            INIT5:  MOV CTABS,R3                ;GET ADDRESS OF CONTROLLER TABLES
017350  005713                            TST (R3)                    ;CHECK IF ANY MORE TABLES
017352  001416                            BEQ NEWTAB                  ;BUILD NEW TABLE IF FOUND ZERO WORD
017354  021013                            CMP (R0),(R3)               ;CHECK IF SAME UNIBUS ADDRESS
017356                                            ASSUME C.UADR EQ 0
017356                                            ASSUME HO.UBA EQ 0
017356  001450                            BEQ SAMTAB                  ;CHECK TABLE IF ALREADY EXISTS
017360  016301  000004                    MOV C.VEC(R3),R1            ;GET VECTOR FROM EXISTING CONTROLLER TABLE
017364  042701  177000                    BIC #^C<CT.VEC>,R1
017370  026001  000002                    CMP HO.VEC(R0),R1           ;SEE IF DIFFERENT VECTOR
017374  001002                            BNE 1$
017376  000137  017776                    JMP SAMVEC                  ;ERROR, CAN'T HAVE TWO UDA'S WITH SAME VECTOR
017402  062703  000054            1$:     ADD #C.SIZE,R3              ;MOVE TO NEXT TABLE
017406  000760                            BR INIT5
```

INITIALIZE SECTION

```
                                        ;BUILD A CONTROLLER TABLE
                                        
017410  012701  000026    NEWTAB: MOV #C.SIZE/2,R1        ;GET WORDS IN CONTROLLER TABLE
017414  004737  011346            CALL ALOCM              ;ALLOCATE SPACE FOR IT
017420  011021                    MOV (R0),(R1)+          ;STORE UNIBUS ADDRESS
017422  010221                    MOV R2,(R1)+            ;UNIT NUMBER
017424  016004  000004            MOV HO.BRL(R0),R4       ;GET BR LEVEL
017430  000304                    SWAB R4                 ;SWAP TO HIGH BYTE
017432  006104                    ROL R4                  ;SHIFT ONE MORE TO LEFT
017434  056004  000002            BIS HO.VEC(R0),R4       ;ADD VECTOR ADDRESS
017440  010421                    MOV R4,(R1)+            ;  TO TABLE
017442  016021  000006            MOV HO.BST(R0),(R1)+
017446  012721  004037            MOV #4037,(R1)+         ;PUT [JSR R0,UDASRV]
017452  012721  015602            MOV #UDASRV,(R1)+       ;  INTO TABLE
017456  012703  000020            MOV #16.,R3             ;CLEAR POINTERS TO DRIVE TABLES,
                                                        ; TIMEOUT COUNTER, FLAGS, REF. NUMBER
017462  005021            INIT7:  CLR (R1)+
017464  005303                    DEC R3
017466  001375                    BNE INIT7               ;LOOP TIL ALL CLEARED
017470  005237  002160            INC CTRLRS              ;COUNT THE CONTROLLER
017474  005011                    CLR (R1)                ;CLEAR TABLE END MARKER
017476  000417                    BR NXTTAB               ;NOW GO TO NEXT P-TABLE
```

INITIALIZE SECTION

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

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

INITIALIZE SECTION

```
                                ;BUILD DRIVE TABLE
                                ;
017604  012701  000015  INIT11: MOV #D.SIZE/2,R1          ;GET SIZE OF DRIVE TABLE
017610  004737  011346          CALL ALOCM                      ;ALLOCATE SPACE FROM FREE MEMORY
                        ;       R0 POINTS TO P-TABLE
                        ;       R1 POINTS TO DRIVE TABLE
                        ;       R3 POINTS TO CONTROLLER TABLE
                        ;       R2 IS UNIT NUMBER
017614  010337  003250          MOV R3,TEMP                     ;SAVE CONTROLLER TABLE ADDRESS
                                                                ;IN CASE AN ERROR IS DETECTED
017620  062703  000020          ADD #C.DR0,R3                   ;BUILD POINTER TO C.DR ENTRY IN CONTROLLER TABLE
017624  012704  000010          MOV #8.,R4                      ;GET MAX COUNT OF DRIVES ON ONE CONTROLLER
017630  005713          INIT12: TST (R3)                ;CHECK IF ENTRY CONTAINS POINTER TO DRIVE TABLE
017632  001411                  BEQ INIT13
017634  026033  000010          CMP H0.LDR(R0),@(R3)+           ;CHECK DRIVE NUMBER IN DRIVE TABLE
017640  001002                  BNE 1$
017642  000137  017742          JMP MLDRER                      ;IF SAME, TWO P-TABLES POINT TO SAME DRIVE
017646  005304          1$:     DEC R4                          ;COUNT DRIVES
017650  001367                  BNE INIT12                      ;IF EIGHT DRIVE TABLES EXIST,
017652  000137  017760          JMP TOOMER                      ;  THEN REPORT ERROR
017656  010113          INIT13: MOV R1,(R3)             ;LOAD DRIVE TABLE POINTER
017660  016021  000010          MOV H0.LDR(R0),(R1)+            ;LOAD DRIVE NUMBER
017664  010221                  MOV R2,(R1)+                    ;LOAD UNIT NUMBER
```

INITIALIZE SECTION

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

```
                                        ;DIFFERENT VECTORS, BR LEVELS OR BURST RATES FOR ONE CONTROLLER
017726  010305                  CTABER: MOV R3,R5                       ;GET CONTROLLER ADDRESS
017730                                  ERRSF 1,,ERR001
017730  104454                                                          TRAP    C$ERSF
017732  000001                                                          .WORD   1
017734  000000                                                          .WORD   0
017736  010430                                                          .WORD   ERR001
017740                                  DOCLN
017740  104444                                                          TRAP    C$DCLN

                                        ;TWO P-TABLES FOR SAME DRIVE
017742  013705  003250          MLDRER: MOV TEMP,R5             ;GET CONTROLLER ADDRESS
017746                                  ERRSF 2,,ERR002
017746  104454                                                          TRAP    C$ERSF
017750  000002                                                          .WORD   2
017752  000000                                                          .WORD   0
017754  010446                                                          .WORD   ERR002
017756                                  DOCLN
017756  104444                                                          TRAP    C$DCLN

                                        ;MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER

017760  013705  003250          TOOMER: MOV TEMP,R5             ;GET CONTROLLER ADDRESS
017764                                  ERRSF 3,,ERR003
017764  104454                                                          TRAP    C$ERSF
017766  000003                                                          .WORD   3
017770  000000                                                          .WORD   0
017772  010464                                                          .WORD   ERR003
017774                                  DOCLN
017774  104444                                                          TRAP    C$DCLN

                                        ;TWO UDA'S USE THE SAME VECTOR

017776  010305                  SAMVEC: MOV R3,R5                       ;GET CONTROLLER ADDRESS
020000                                  ERRSF 8,ERR008
020000  104454                                                          TRAP    C$ERSF
020002  000010                                                          .WORD   8
020004  010540                                                          .WORD   ERR008
020006  000000                                                          .WORD   0
020010                                  DOCLN
020010  104444                                                          TRAP    C$DCLN
```

INITIALIZE SECTION

```
      1 020012                          DATE:   GMANID DATEQ,DATEI,A,-1,1,11.,YES       ;GET DATE
        020012  104443                                                                    TRAP    C$GMAN
        020014  000406                                                                    BR      10000$
        020016  003276                                                                    .WORD   DATEI
        020020  000152                                                                    .WORD   T$CODE
        020022  003526                                                                    .WORD   DATEQ
        020024  177777                                                                    .WORD   -1
        020026  000001                                                                    .WORD   T$LOLIM
        020030  000013                                                                    .WORD   T$HILIM
        020032                                                                  10000$:
      2 020032  012705  003276                    MOV #DATEI,R5   ;GET POINTER TO ANSWER
      3 020036  121527  000060                    CMPB (R5),#'0
      4 020042  103443                           BLO DERR
      5 020044  122527  000071            DAY:    CMPB (R5)+,#'9
      6 020050  101040                           BHI DERR
      7 020052  121527  000055                    CMPB (R5),#'-
      8 020056  001406                           BEQ DAS1
      9 020060  121527  000060                    CMPB (R5),#'0
     10 020064  103432                           BLO DERR
     11 020066  122527  000071                    CMPB (R5)+,#'9
     12 020072  101027                           BHI DERR
     13 020074  122527  000055            DAS1:   CMPB (R5)+,#'-
     14 020100  001024                           BNE DERR
     15 020102  012704  000014                    MOV #12.,R4     ;GET NUMBER OF MONTH
     16 020106  012703  003353                    MOV #MONTHS,R3  ;GET POINTER TO MONTH NAMES
     17 020112  005000                    MON1:   CLR R0
     18 020114  121523                           CMPB (R5),(R3)+
     19 020116  001401                           BEQ MON2
     20 020120  005200                           INC R0
     21 020122  126523  000001            MON2:   CMPB 1(R5),(R3)+
     22 020126  001401                           BEQ MON3
     23 020130  005200                           INC R0
     24 020132  126523  000002            MON3:   CMPB 2(R5),(R3)+
     25 020136  001401                           BEQ MON4
     26 020140  005200                           INC R0
     27 020142  005700                    MON4:   TST R0
     28 020144  001407                           BEQ MON5
     29 020146  005304                           DEC R4
     30 020150  001360                           BNE MON1
     31 020152                          DERR:   PNTF DATEX
        020152  004137  014650                                                            JSR R1,LPNTF
        020156  010330                                                                    .WORD DATEX
        020160  000000                                                                    .WORD PNT.CT
     32 020162  000713                          BR DATE
     33 020164  012701  003312            MON5:   MOV #DATEO,R1   ;GET POINTER TO DATE FOR FORMATTER
     34 020170  010403                          MOV R4,R3       ;GET COPY OF MONTH NUMBER
     35 020172  020327  000012                   CMP R3,#10.     ;  IF 10 OR GREATER
     36 020176  103404                          BLO MON6
     37 020200  112721  000061                   MOVB #'1,(R1)+  ;PUT A "1" IN OUTPUT
     38 020204  162703  000012                   SUB #10.,R3
     39 020210  062703  000060            MON6:   ADD #'0,R3      ;CONVERT MONTH NUMBER TO ASCII
     40 020214  110321                          MOVB R3,(R1)+   ;PUT A NUMBER IN OUTPUT
     41 020216  112721  000055                   MOVB #'-,(R1)+  ;PUT A "-" IN OUTPUT
     42 020222  062704  003416                   ADD #DAYS-1,R4  ;GET POINTER TO DAYS IN MONTH
     43                                                                 ;INDEXED BY NUMBER OF MONTH
     44 020226  012703  003276                   MOV #DATEI,R3   ;GET POINTER TO DATE INPUT
     45 020232  005000                          CLR R0
```

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

INITIALIZE SECTION

```
    103
    104 020450                          ENDINIT
         020450                                    L10034:
         020450  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.
                                ;--

020452                                  BGNAUTO
020452                                                                  L$AUTO::

020452                                  ENDAUTO
020452                                                                  L10035:
020452  104461                                                                  TRAP    C$AUTO
```

CLEANUP CODING SECTION

```
                        .SBTTL  CLEANUP CODING SECTION

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

020454                          BGNCLN
020454                                                          L$CLEAN::

020454                          CLOSE                   ;CLOSE DATA FILE
020454  104435                                                  TRAP    C$CLOS
020456                          BRESET                  ;RESET ALL UDAS
020456  104433                                                  TRAP    C$RESET

020460                          ENDCLN
020460                                                  L10036:
020460  104412                                                  TRAP    C$CLEAN

020462                          ENDMOD
```

```
                                .SBTTL  TEST 1: DUP PROGRAM DRIVER

                                        BGNMOD

                                        BGNTST
                                                                        T1::
                                        BIT #SO.FMT,MODE
                                        BNE T1FMT
                                        MANUAL
                                                                        TRAP    C$MANI
                                        BCOMPLETE T1GO
                                                                        BCS     T1GO
                                        ERRSF 10,,ERR010
                                                                        TRAP    C$ERSF
                                                                        .WORD   10
                                                                        .WORD   0
                                                                        .WORD   ERR010
                                        EXIT TST
                                                                        TRAP    C$EXIT
                                                                        .WORD   L10037-.
T1GO:   BIT #SO.STR,MODE
                                        BEQ T1CNS
                                        CMP L$UNIT,#1
                                        BEQ T1RST
                                        ERRSF 9,,ERR009
                                                                        TRAP    C$ERSF
                                                                        .WORD   9
                                                                        .WORD   0
                                                                        .WORD   ERR009
                                        EXIT TST
                                                                        TRAP    C$EXIT
                                                                        .WORD   L10037-.

T1RST:  PNTF FILNAM
                                                                        JSR R1,LPNTF
                                                                        .WORD FILNAM
                                                                        .WORD PNT.CT
                                        GMANID FILNAQ,FNAME,A,-1,1,10.,NO       ;GET FILE NAME
                                                                        TRAP    C$GMAN
                                                                        BR      10000$
                                                                        .WORD   FNAME
                                                                        .WORD   T$CODE
                                                                        .WORD   FILNAQ
                                                                        .WORD   -1
                                                                        .WORD   T$LOLIM
                                                                        .WORD   T$HILIM
                                                                10000$:
                                        OPEN #FNAME
                                                                        MOV     #FNAME,R0
                                                                        TRAP    C$OPEN
                                        BR T1FMT
T1CNS:  MOV CTABS,R5
T1SER1: MOV R5,R4
                                        ADD #C.DR0,R4
                                        MOV #8.,R3
T1SER2: MOV (R4),R2     ;GET DRIVE TABLE POINTER
                                        BEQ T1SERN
```

```
                                        PNTF SERNUM,D.UNIT(R2),(R5),(R2)
020630  011246                                                          MOV (R2),-(SP)
020632  011546                                                          MOV (R5),-(SP)
020634  016246  000002                                                  MOV D.UNIT(R2),-(SP)
020640  004137  014650                                                  JSR R1,LPNTF
020644  004151                                                          .WORD SERNUM
020646  000006                                                          .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
020650  104443                                                          TRAP    C$GMAN
020652  000406                                                          BR      10001$
020654  003250                                                          .WORD   TEMP
020656  000142                                                          .WORD   T$CODE
020660  003610                                                          .WORD   SERNQ
020662  177777                                                          .WORD   -1
020664  000001                                                          .WORD   T$LOLIM
020666  000024                                                          .WORD   T$HILIM
020670                                                          10001$:
020670  012701  003250                MOV #TEMP,R1
020674  005000                        CLR R0
020676  105711                T1SER4: TSTB (R1)
020700  001410                        BEQ T1SER5
020702  005200                        INC R0
020704  121127  000060                CMPB (R1),#'0
020710  103416                        BLO T1SER7
020712  122127  000071                CMPB (R1)+,#'9
020716  101767                        BLOS T1SER4
020720  000412                        BR T1SER7
020722  020027  000024        T1SER5: CMP R0,#20.
020726  103422                        BLO T1SER8
020730  012701  003250                MOV #TEMP,R1
020734  012700  003326                MOV #HIGHEST,R0
                                T1SER6: CMPB (R1)+,(R0)+
                                        BEQ T1SER6
                                        BLO T1SER8
                                T1SER7: PRINTF #SERNX,#HIGHEST
020746  012746  003326                                                  MOV     #HIGHEST,-(SP)
020752  012746  010240                                                  MOV     #SERNX,-(SP)
020756  012746  000002                                                  MOV     #2,-(SP)
020762  010600                                                          MOV     SP,R0
020764  104417                                                          TRAP    C$PNTF
020766  062706  000006                                                  ADD     #6,SP
                                        BR T1SER3
020774  062702  000004        T1SER8: ADD #D.SERN,R2  ;PUT ANSWER INTO DRIVE TABLE
021000  012701  003250                MOV #TEMP,R1
021004  112122                T1SER9: MOVB (R1)+,(R2)+
021006  001376                        BNE T1SER9
021010  005303                        DEC R3
021012  001402                        BEQ T1SERN
021014  005724                        TST (R4)+
021016  000702                        BR T1SER2
021020  062705  000054        T1SERN: ADD #C.SIZE,R5
021024  005715                        TST (R5)
021026  001271                        BNE T1SER1
021030  013737  002156  002162 T1FMT: MOV CTABS,TSTTAB              ;GET FIRST TABLE ADDRESS
021036  013701  002160                MOV CTRLRS,R1                 ;RUN DM PROGRAM ON ALL CONTROLLERS
```

```
     65 021042  004737  011410                CALL RUNDM                      ;  AT ONCE
     66 021046  001402                        BEQ 6$
     67 021050  004737  011522                CALL RESPDM
     68 021054                          6$:   EXIT TST
        021054  104432                                                        TRAP    C$EXIT
        021056  000002                                                        .WORD   L10037-.
     69 021060                                ENDTST
        021060                                                          L10037:
        021060  104401                                                        TRAP    C$ETST
     70 021062                                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
021062  000027                                                            .WORD L10040-L$HARD/2
021064                                                                  L$HARD::

                                ;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:

                                TABLE                                   ;START A TEBLE DEFINITION

                                ITEM HO.UBA     2                       ; UNIBUS ADDRESS
                                ITEM HO.VEC     2                       ; UDA VECTOR
                                ITEM HO.BRL     2                       ; BR LEVEL
                                ITEM HO.BST     2                       ; BURST RATE
                                ITEM HO.LDR     2                       ; DRIVE NUMBER
                                END
```

HARDWARE PARAMETER CODING SECTION

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

SOFTWARE PARAMETER CODING SECTION

```
                                .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.
                                ;--

021250                                  BGNSFT
021250  000022                                                          .WORD L10041-L$SOFT/2
021252                                                          L$SOFT::

                                ;FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:

021252                          TABLE                                   ;START A TABLE DEFINITION

021252                          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

021252                          END
```

```
                                                GPRML S.FMT,SO.BIT,SO.FM1,YES   ;REFORMAT?
                                                                                        .WORD   T$CODE
                                                                                        .WORD   S.FMT
                                                                                        .WORD   SO.FM1
                                                XFERT SWEND
                                                                                        .WORD   T$CODE
                                                GPRML S.NRF,SO.BIT,SO.FM2,YES   ;AGAIN - REFORMAT?
                                                                                        .WORD   T$CODE
                                                                                        .WORD   S.NRF
                                                                                        .WORD   SO.FM2
                                                XFERT SWEND
                                                                                        .WORD   T$CODE
                                                GPRML S.CNS,SO.BIT,SO.CNS,YES   ;RECONSTRUCT
                                                                                        .WORD   T$CODE
                                                                                        .WORD   S.CNS
                                                                                        .WORD   SO.CNS
                                                XFERT SWEND
                                                                                        .WORD   T$CODE
                                                GPRML S.RST,SO.BIT,SO.STR,YES   ;RESTORE?
                                                                                        .WORD   T$CODE
                                                                                        .WORD   S.RST
                                                                                        .WORD   SO.STR
                                                XFERT SWEND
                                                                                        .WORD   T$CODE
                                                DISPLAY S.NOF   ;WARNING
                                                                                        .WORD   T$CODE
                                                                                        .WORD   S.NOF
                                        SWEND:  ENDSFT
                                                                                        .EVEN
                                                                                L10041:

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


                                                .DSABL AMA
000000                                          .PSECT END
```

PATCH AREA

```
                                        .SBTTL  PATCH AREA
                                
000000                          $PATCH::
        000050                          .REPT 40.
                                        .WORD 0
                                        .ENDR
                                
000120                                  LASTAD
                                                                .EVEN
000120  000142'                                                 .WORD T$FREE
000122  000007                                                 .WORD T$SIZE
000124                          L$LAST::
                                
000124                                  ENDMOD
```

PATCH AREA

```
      1 000124                                  BGNSETUP        1
      2
      3 000124                                  BGNPTAB
        000124  000000                                                                  .WORD   0
        000126  000005                                                                  .WORD   L10044-./2-1
        000130                                                                L10042:
      4
      5 000130  172150                          .WORD   172150              ; UNIBUS ADDRESS
      6 000132  000154                          .WORD   154                 ; VECTOR ADDRESS
      7 000134  000005                          .WORD   5.                  ; BR LEVEL
      8 000136  000077                          .WORD   63.                 ; UNIBUS BURST RATE
      9 000140  000000                          .WORD   0.                  ; LOGICAL DRIVE NUMBER
     10
     11 000142                                  ENDPTAB
        000142                                                                L10044:
     12
     13 000142                                  ENDSETUP
     14
     15
     16
     17
     18
     19
     20
     21         000001                  .END
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  29184 WORDS  ( 114 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  70 PAGES
ZUDEB0,A:ZUDEB0/C=[20,0]SVC34R.MLB/P:1,ZUDEB0.DOC,ZUDEB0.MAC
```

CROSS REFERENCE TABLE (CREF V04.00)

```
$PATCH 132-3#
ADR      31-10#
ALOCM    57-16#   58-14   118-4    119-20   121-4
ASSEMB   27-8     27-8
BAS      51-15#   82-5     82-5     82-5     83-5     83-5
BASL2    51-13#   83-5
BASL3    51-14#
BASLN    51-17#   82-5     83-5
BASNO    51-12#   82-5     83-5
BIT0     31-10#  130-19
BIT00    31-10    31-10#
BIT01    31-10    31-10#
BIT02    31-10    31-10#
BIT03    31-10    31-10#
BIT04    31-10    31-10#
BIT05    31-10    31-10#
BIT06    31-10    31-10#
BIT07    31-10    31-10#
BIT08    31-10    31-10#
BIT09    31-10    31-10#
BIT1     31-10#   42-23   130-20
BIT10    31-10#
BIT11    31-10#
BIT12    31-10#
BIT13    31-10#
BIT14    31-10#
BIT15    31-10#   42-15    43-12    59-27    61-20    69-26    74-15    89-29
BIT2     31-10#   42-24   130-22
BIT3     31-10#   42-25   1?' -23
BIT4     31-10#   42-27
BIT5     31-10#   42-30
BIT6     31-10#   42-31
BIT7     31-10#   42-33
BIT8     31-10#
BIT9     31-10#
BLDC0    97-22    97-24#
BLDC1    97-26#   97-28
BLDCMD   60-49    63-14    63-44    95-3     97-15#
BOE      31-10#
C$AU     27-8#
C$AUTO   27-8#   125-12
C$BRK    27-8#    60-8    100-21   107-12   110-27
C$BSEG   27-8#
C$BSUB   27-8#
C$CEFG   27-8#
C$CLCK   27-8#   116-43   116-45
C$CLEA   27-8#   126-13
C$CLOS   27-8#    72-18   122-13   126-10
C$CLP1   27-8#
C$CVEC   27-8#   107-30
C$DCLN   27-8#    56-8    116-18   123-4    123-9    123-15   123-21
C$DODU   27-8#
C$DRPT   27-8#
C$DU     27-8#
C$EDIT   27-8#   27-34
C$ERDF   27-8#   60-24    61-36    62-21    63-8     67-22    69-30    96-3     100-29   101-5    105-35   106-11   107-33   109-44
```

CROSS REFERENCE TABLE (CREF V04.00)

```
        110-36  111-5
C$ERHR  27-8#
C$ERRO  27-8#
C$ERSF  27-8#    56-7    123-3    123-8    123-14   123-20   127-10   127-16
C$ERSO  27-8#
C$ESCA  27-8#
C$ESEG  27-8#
C$ESUB  27-8#
C$ETST  27-8#    127-69
C$EXIT  27-8#    122-14   127-11   127-17   127-68
C$GETB  27-8#     72-23
C$GETW  27-8#
C$GMAN  27-8#    124-1    127-20   127-32
C$GPHR  27-8#    117-10   120-4
C$GPLO  27-8#
C$GPRI  27-8#
C$INIT  27-8#    124-104
C$INLP  27-8#
C$MANI  27-8#    127-8
C$MEM   27-8#    116-34
C$MSG   27-8#     54-16    54-20    54-24    54-28    54-32    54-36    54-40    54-44    54-53    54-59    54-73    54-77    54-81
         54-85    54-89    54-94    54-98    54-102   54-106   54-110   54-114   54-118
C$OPEN  27-8#     72-19   127-21
C$PNTB  27-8#     91-14
C$PNTF  27-8#     91-12   127-50
C$PNTS  27-8#     91-18
C$PNTX  27-8#     54-58    91-16
C$QIO   27-8#
C$RDBU  27-8#
C$REFG  27-8#    116-10   116-12   116-14   116-16
C$RESE  27-8#     27-8#   116-33   126-11
C$REVI  27-8#     27-34
C$RFLA  27-8#
C$RPT   27-8#
C$SEFG  27-8#
C$SPRI  27-8#    122-12
C$SVEC  27-8#     94-20   107-27   116-54
C$TPRI  27-8#
C.BST   42-19#   106-17   119-9
C.DR0   42-35#    61-15    74-12   121-11   127-25
C.DR1   42-36#
C.DR2   42-37#
C.DR3   42-38#
C.DR4   42-39#
C.DR5   42-40#
C.DR6   42-41#
C.DR7   42-42#
C.FLG   42-22#   59-23*   60-10    60-13    60-15    60-38    60-43    60-47*   61-12*   62-6     62-26    63-35*   63-39*   63-40
         63-47*   63-50*   63-62*   63-63    63-69*   63-70    95-13*   95-14*   98-21*   100-17   101-1*
C.JAD   42-21#
C.JSR   42-20#    94-19
C.PRI   42-46#    63-65    63-67    63-72*   63-73*
C.REF   42-47#    62-19    98-16*   98-17
C.RING  42-34#    54-134   58-15*   60-9     63-29    97-16    98-15    105-20   106-3    106-22   107-20
C.SIZE  42-49#    59-32    61-5     117-27   118-3    120-12   127-60
C.TO    42-43#    60-36    63-55    100-14   100-27   110-22   110-33
```

CROSS REFERENCE TABLE (CREF V04.00)

```
C.TOH    42-44#   60-33   100-24   110-30
C.TOT    42-45#   60-29    63-53    94-14*   94-15*
C.UADR   42-12#   75-16   107-25   117-19   127-30
C.UNIT   42-13#   59-24    59-25    60-12
C.VEC    42-16#   94-16   107-14   117-22   119-7
CALR4    79-6     82-5#
CALR5    79-7     83-5#
CALR6    79-8     84-4#
CALR7    79-9     85-4#
CALR8    79-10    86-4#
CALR9    79-11    87-3#
CALRE    79-3     79-4     79-5     81-4#
CF.576   38-39#
CF.ATN   38-34#
CF.MSC   38-35#
CF.OTH   38-36#
CF.SHD   38-38#
CF.THS   38-37#
CLOG    112-10#
CLOGLP  112-13#  112-15
CLRBFL   99-22#   99-24
CLRBUF   63-16    63-46    99-16#
CON.A    77-12#   80-21
CON.A1   77-13#   77-15
CON.A2   77-17    77-19#
CON.D    77-23#   80-21
CON.H    77-29#   80-21
CON.N    78-9#    80-21
CON.N1   78-10#   78-12
CON.O    78-3#    80-21
CON.QU   77-3#    77-7     80-21
CON.QX   77-5     77-8#
CON.R    78-17#   80-21
CON.R1   78-19    78-23#
CON.S    78-29#   80-21
CON.S1   78-30#   78-32
CPNT     73-13    73-18    75-21    77-6     77-13    78-10    78-30    82-7     83-7     90-13    90-22    91-5#    114-27   114-32
        114-35
CPNTX    91-13    91-15    91-17    91-19#
CR       31-12#   73-13    75-21    78-10    82-7     83-7     91-8
CT.AVL   42-15#   59-27
CT.BRL   42-18#
CT.CMD   42-24#   60-15    95-13    98-21
CT.MSG   42-25#   60-13    63-39    63-62    100-17   101-1    103-19
CT.REQ   42-27#   62-6     62-26    63-35     63-40    63-47    63-50    95-13
CT.RN    42-23#   60-10    95-14
CT.STA   42-30#   60-38    60-46    63-39    63-62
CT.TM1   42-31#   60-42    60-43    63-39    63-70
CT.TM2   42-33#   63-39    63-63    63-69
CT.UNT   42-14#
CT.VEC   42-17#   94-17   107-15   117-23
CTABER  119-11   123-2#
CTABS    47-12#  117-3*   117-4*   117-15   120-9    127-23   127-63
CTRLRS   47-13#  117-5*   118-20*  127-64
D.DRV    43-9#    68-36    75-17   127-31
D.SERN   43-13#   67-17   127-52
```

CROSS REFERENCE TABLE (CREF V04.00)

```
D.SIZE  43-15#  121-3
D.UNIT  43-10#   61-19    61-25*   74-14    75-15    127-29
DAS1   124-8    124-13#
DATE   116-32   124-1#   124-32
DATEI   48-2#   124-1    124-2    124-44
DATEO   48-4#    67-26   116-30   124-33
DATEQ   50-3#   124-1
DATEX   53-12#  124-31
DAY    124-5#
DAY1   124-46#  124-57
DAY2   124-47   124-58#
DAYS    48-19#  124-42
DERR   124-4    124-6    124-10   124-12   124-14   124-31#  124-59   124-61   124-64   124-84   124-88   124-90
DF.ACT  41-21#   63-60
DF.LCL  41-24#
DF.NES  41-22#
DF.SA   41-26#
DFPTBL  29-10#
DIAG    35-33#   97-23
DIAGMC  27-8     27-8
DIVIDE  68-38    89-39    90-7     93-16#  114-19   114-21   114-23
DMEND   47-17#
DMENDS  47-18#
DMFRST  41-47#
DMMAIN  41-46#
DMOVRL  41-44#
DMPROG  47-16#   94-13    95-1     95-4     95-6     95-7
DMTMO   41-45#   94-14
DMTRLN  41-43#
DQCONT  68-28    68-61#
DQNO    68-54    68-62#
DQRFMT  68-26    68-53#
DQRSTR  68-27    68-59#
DQUEJP  68-21    68-22#   68-29
DQUEST  64-4     68-16#
DQUESZ  68-18    68-29#
DQUEX   68-19    68-22    68-24    68-25    68-50#   68-57    68-64
DQUNIT  68-23    68-33#
DQUNL1  68-38#   68-42
DQUNL2  68-44#   68-48
DQYES   68-55#   68-60
DSPSIZ  64-9#
DT.AVL  43-12#   61-20    61-25    74-15
DT.UNT  43-11#
DU.DFL  41-33#
DU.FTL  41-36#
DU.INF  41-34#
DU.QUE  41-32#
DU.SPC  41-37#   63-6
DU.TER  41-35#
DU.TYP  41-39#   63-4     63-19
DUP     35-34#   97-20
E$END   27-8#
E$LOAD  27-8#    27-34
EF.BBR  38-27#
EF.BBU  38-28#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
EF.CON   31-10#  116-14
EF.LOG   38-29#
EF.NEW   31-10#
EF.PWR   31-10#  116-16
EF.RES   31-10#  116-12
EF.SEX   38-30#
EF.STA   31-10#  116-10
ERR001   54-14#  123-3
ERR002   54-18#  123-8
ERR003   54-22#  123-14
ERR004   54-26#   56-7
ERR008   54-38#  123-20
ERR009   54-30#  127-16
ERR010   54-34#  127-10
ERR020   54-42#  107-33
ERR021   54-46#  111-5
ERR022   54-55#  110-36
ERR023   54-61#  106-11
ERR024   54-75#  105-35
ERR025   54-79#  109-44
ERR030   54-83#   60-24
ERR031   54-87#   61-36
ERR032   54-91#   63-8
ERR033   54-96#   62-21
ERR034   54-100#  96-3
ERR036   54-104# 100-29
ERR037   54-108# 101-5
ERR100   54-112#  67-22
ERR101   54-116#  69-30
ERR23A   54-64#   54-71
ERR23B   54-65    54-69#
ERR23C   54-68    54-72#
ERRC     76-36    76-43    80-9#
ERRCHR   47-38#   91-5*    91-12    91-14    91-16    91-18
ERRD     76-45    80-21#
ERRME1   51-8#    76-41    78-23    81-4
ERRNL    51-4#    91-10
ERRONE   51-3#    91-7
ERRRSZ   78-18    79-12#
ERRRTB   78-21    79-3#    79-12
ERRTRM   64-7     71-14#
EVL      31-10#
F$AU     27-8#
F$AUTO   27-8#   125-10   125-12
F$BGN    27-8#    27-26    30-16    31-3     54-14    54-18    54-22    54-26    54-30    54-34    54-38    54-42    54-46    54-55
          54-61    54-75    54-79    54-83    54-87    54-91    54-96    54-100   54-104   54-108   54-112   54-116   102-10   103-18
         113-5    114-38   115-3    115-10   116-8    122-14   125-10   126-8    126-15   127-3    127-5    127-11   127-17   127-68
         127-69   127-70   128-3    128-14   130-12   131-9    131-9    132-10   133-1    133-3    133-3    133-11   133-13
F$CLEA   27-8#    126-8    126-13
F$DU     27-8#
F$END    27-8     27-8     27-8     27-8     27-8     27-8     27-8     27-8     27-8     27-8     27-8     27-8     27-8     27-8
          27-8     27-8     27-8#    27-26    30-16    31-3     54-16    54-20    54-24    54-28    54-32    54-36    54-40    54-44
          54-53    54-59    54-73    54-77    54-81    54-85    54-89    54-94    54-98    54-102   54-106   54-110   54-114   54-118
         102-14   103-21   113-9    114-38   115-3    122-14   124-104  125-12   126-13   126-15   127-3    127-5    127-5    127-5
         127-11   127-17   127-68   127-69   127-69   127-70   128-3    129-6    131-10   132-10   133-1    133-3    133-11   133-13
F$HARD   27-8#    128-14   129-6    131-2    131-4    131-6    131-8
```

CROSS REFERENCE TABLE (CREF V04.00)

```
F$HW     27-8#    29-10    29-17
F$INIT   27-8#   116-8    124-104
F$JMP    27-8#   122-14   127-11   127-17   127-68
F$MOD    27-8#    27-26    30-16    31-3    114-38   115-3    126-15   127-3    127-70   128-3    132-10
F$MSG    27-8#    54-14    54-16    54-18    54-20    54-22    54-24    54-26    54-28    54-30    54-32    54-34    54-36    54-38
          54-40    54-42    54-44    54-46    54-53    54-55    54-59    54-61    54-73    54-75    54-77    54-79    54-81    54-83
          54-85    54-87    54-89    54-91    54-94    54-96    54-98    54-100   54-102   54-104   54-106   54-108   54-110   54-112
          54-114   54-116   54-118
F$PROT   27-8#   115-10   115-16
F$PWR    27-8#
F$RPT    27-8#
F$SEG    27-8#
F$SOFT   27-8#   130-12   131-2    131-4    131-6    131-8    131-10
F$SRV    27-8#   102-10   102-14   103-18   103-21   113-5    113-9
F$SUB    27-8#
F$SW     27-8#    30-10    30-14
F$TEST   27-8#   127-5    127-69
FCTBUF   47-25#   72-22    72-31
FCTNUM   47-26#   59-36*   72-14    72-20*   72-27*
FDATA    47-24#
FFREE    47-8#    54-62    57-16    57-20*   116-34*  116-35   117-3    122-9
FILNAM   53-13#  127-19
FILNAQ   50-4#   127-20
FMEM     47-10#  122-9*
FMEMS    47-11#  122-10*
FMERR    56-7#    57-18
FNAME    47-40#   72-19    127-20   127-21
FSIZE    47-9#    57-17*   116-35*  122-10
G$CNTO   27-8#
G$DELM   27-8#
G$DISP   27-8#   131-9
G$EXCP   27-8#
G$HILI   27-8#
G$LOLI   27-8#
G$NO     27-8#   127-20   127-32
G$OFFS   27-8#   124-1    127-20   127-32   129-1    129-2    129-3    129-4    129-5    131-1    131-3    131-5    131-7
G$OFSI   27-8#   124-1    127-20   127-32   129-1    129-2    129-3    129-4    129-5    131-1    131-3    131-5    131-7
G$PRMA   27-8#   129-1    129-2
G$PRMD   27-8#   124-1    127-20   127-32   129-3    129-4    129-5
G$PRML   27-8#   131-1    131-3    131-5    131-7
G$RADA   27-8#   124-1    127-20   127-32
G$RADB   27-8#
G$RADD   27-8#   129-3    129-4    129-5
G$RADL   27-8#   131-1    131-3    131-5    131-7
G$RADO   27-8#   129-1    129-2
G$XFER   27-8#   131-2    131-4    131-6    131-8
G$YES    27-8#   124-1    129-1    129-2    129-3    129-4    129-5    131-1    131-3    131-5    131-7
GETCDN   88-16    88-18    88-28#
GETCNT   77-12    78-9     78-17    78-29    88-13#   89-15
GETCNX   88-15#   88-27
GETCXX   88-29    88-31#
GTDRVL   74-13#   74-16
GTDRVT   67-16    68-16    69-22    74-11#
H.BRL   129-3    129-10#
H.BST   129-4    129-11#
H.LDR   129-5    129-12#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
H.UBA   129-1    129-8#
H.VEC   129-2    129-9#
HC.BF1   35-24#   35-25    63-15
HC.BF2   35-25#   35-27    54-135   63-3     63-45
HC.BSZ   35-13#   35-25    35-27    63-18    99-19    99-21
HC.CCT   35-19#   60-40    60-50*   98-19*
HC.CEV   35-22#   35-23    97-18
HC.CMD   35-18#   35-19    35-20    106-25*  106-26*
HC.CPK   35-23#   35-24    54-122   63-33*   95-4*    95-5*    95-6*    95-7*    97-29*   98-17*   99-18*   99-19*   106-26
HC.ESZ   35-11#   35-21    35-23
HC.INT   35-15#   35-16
HC.ISZ   35-9#    35-16    105-21   106-4
HC.MCT   35-17#   63-74*   98-18*
HC.MEV   35-20#   35-21
HC.MPK   35-21#   35-22    54-124   62-9     62-14    62-19    63-11    63-58    63-60    63-65    63-67    63-72    63-73    95-11
         106-24
HC.MSG   35-16#   35-17    35-18    106-23*  106-24*  107-21
HC.PSZ   35-12#   35-22    35-24    97-19    97-25
HC.RSZ   35-10#   35-18    35-20    105-21   106-4
HC.SIZ   35-27#   58-13
HCOMM    58-13#   59-28
HEADER   69-24    75-13#
HICHES   48-6#    127-46   127-50
HO.CRL  118-7     119-3    128-22#  129-3    129-3    129-3
HO.BST  118-12    119-9    128-23#  129-4    129-4    129-4
HO.LDR  121-15    121-22   128-24#  129-5    129-5    129-5
HO.UBA  117-20    128-20#  129-1    129-1    129-1
HO.VEC  117-24    118-10   119-6    128-21#  129-2    129-2    129-2
HOE      31-10#
I$AU     27-8#
I$AUTO   27-8#    125-10#  125-12#
I$CLN    27-8#    126-8#   126-13#
I$DU     27-8#
I$HRD   128-14#   129-6#   131-9
I$INIT   27-8#    116-8#   122-14   124-104#
I$MOD    27-8#     27-26    27-26#   30-16    30-16#   31-3     31-3#    114-38   114-38#  115-3    115-3#   126-15   126-15#  127-3
         127-3#   127-70   127-70#  128-3    128-3#   132-10   132-10#
I$MSG    27-8#    54-14#   54-16#   54-18#   54-20#   54-22#   54-24#   54-26#   54-28#   54-30#   54-32#   54-34#   54-36#   54-38#
         54-40#   54-42#   54-44#   54-46#   54-53#   54-55#   54-59#   54-61#   54-73#   54-75#   54-77#   54-79#   54-81#   54-83#
         54-85#   54-87#   54-89#   54-91#   54-94#   54-96#   54-98#   54-100#  54-102#  54-104#  54-106#  54-108#  54-110#  54-112#
         54-114#  54-116#  54-118#
I$PROT   27-8#    115-10#
I$PTAB   27-8#    133-3    133-3#   133-11   133-11#
I$PWR    27-8#
I$RPT    27-8#
I$SEG    27-8#    127-5
I$SETU   27-8#    133-1    133-1#   133-3    133-13   133-13#
I$SFT   130-12#   131-9    131-10#
I$SRV    27-8#    102-10#  102-14#  103-18#  103-21#  113-5#   113-9#
I$SUB    27-8#    127-5
I$TST    27-8#    127-5    127-5#   127-11   127-17   127-68   127-69   127-69#  127-69#
IBE      31-10#
IDU      31-10#
IER      31-10#
INFO     64-5     69-15#   70-14    71-14
INFOE    69-18    69-30#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
INFOP    69-20    69-25#
INFOX    69-16    69-26#
INIT1   116-11   116-13   116-15   116-17   116-19#
INIT10  120-10#  120-13
INIT11  120-11   121-3#
INIT12  121-13#  121-19
INIT13  121-14   121-21#
INIT14  120-5    122-3#
INIT15  122-9#
INIT2   116-21   116-24   116-29#
INIT3   116-31   116-33#
INIT4   117-10#  119-17
INIT5   117-16#  117-28
INIT7   118-17#  118-19
INIT8   120-4#   122-5
INITQT  116-18#  116-28
INITXX  122-12#
ISR      31-10#
IXE      31-10#
J$JMP    27-8#
KNO     116-49   116-56#
KW.BRL   47-32#  116-51*  116-54
KW.CSR   47-31#   60-31    75-19   100-22   110-28   113-8*   114-13   116-40*  116-47*  116-50*  116-55*
KW.EL    47-35#   60-33    60-36   100-24   100-27   104-26   104-27   104-28   110-30   110-33   113-6*   113-7*   114-16   114-17
        116-41*  116-42*
KW.HZ    47-34#  104-16   114-18   116-53*
KW.VEC   47-33#  116-52*  116-54
KW11I   113-5#   116-54
KWOUT.  113-8    116-39#  116-55
KYES    116-44   116-46   116-50#
L$ACP    27-34#
L$APT    27-34#
L$AUT    27-34#
L$AUTO   27-34   125-10#
L$CCP    27-34#
L$CLEA   27-34   126-8#
L$CO     27-34#
L$DEPO   27-34#
L$DESC   27-34    49-16#
L$DESP   27-34#
L$DEVP   27-34#
L$DISP   27-34    28-9#
L$DLY    27-34#
L$DTP    27-34#
L$DTYP   27-34#
L$DUT    27-34#
L$DVTY   27-34    49-12#
L$EF     27-34#
L$ENVI   27-34#
L$ETP    27-34#
L$EXP1   27-34#
L$EXP4   27-34#
L$EXP5   27-34#
L$HARD   27-34   128-14   128-14#
L$HIME   27-34#
L$HPCP   27-34#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
L$HPTP  27-34#
L$HW    27-34    29-10    29-10#
L$ICP   27-34#
L$INIT  27-34   116-8#
L$LADP  27-34#
L$LAST  27-34   132-8#   133-13
L$LOAD  27-34#
L$LUN   27-34#   59-24*   60-12*   74-14*
L$MREV  27-34#
L$NAME  27-34#
L$PRIO  27-34#
L$PROT  27-34   115-10#
L$PRT   27-34#
L$REPP  27-34#
L$REV   27-34#
L$SOFT  27-34   130-12   130-12#
L$SPC   27-34#
L$SPCP  27-34#
L$SPTP  27-34#
L$STA   27-34#
L$SW    27-34    30-10    30-10#
L$TEST  27-34#
L$TIML  27-34#
L$UNIT  27-34#   75-13   119-16   122-4    127-14
L10000  29-10    29-17#
L10001  30-10    30-14#
L10002  54-16#
L10003  54-20#
L10004  54-24#
L10005  54-28#
L10006  54-32#
L10007  54-36#
L10010  54-40#
L10011  54-44#
L10012  54-53#
L10013  54-59#
L10014  54-73#
L10015  54-77#
L10016  54-81#
L10017  54-85#
L10020  54-89#
L10021  54-94#
L10022  54-98#
L10023  54-102#
L10024  54-106#
L10025  54-110#
L10026  54-114#
L10027  54-118#
L10030 102-14#
L10031 103-21#
L10032 113-9#
L10034 122-14    124-104#
L10035 125-12#
L10036 126-13#
L10037 127-11    127-17    127-68    127-69#
L10040 128-14    129-6#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
L10041 130-12   131-10#
L10042 133-3#
L10044 133-3    133-11#
LDDM    59-22#   59-34
LDNEXT  59-26    59-30    59-32#
LOADB   95-2#
LOADDM  59-29    61-30    94-13#
LOADE1  95-12    96-3#
LOADER  94-23    95-10    96-4#
LOE     31-10#
LOG     35-32#
LOT     31-10#
LPNT    92-6     92-8     92-10    92-12#  114-24   114-28   114-33
LPNTB   54-15    54-19    54-23    54-27    54-31    54-35    54-39    54-43    54-52    54-57    54-62    54-66    54-72    54-76
        54-80    54-84    54-88    54-92    54-105   54-109   54-113   54-117   54-120   54-126   54-133   54-137   82-5     83-5
        92-7#
LPNTF   75-15    76-41    78-23    81-4     92-5#    116-48   124-31   127-19   127-29
LPNTS   92-11#
LPNTX   92-9#
MD.CMP  38-4#
MD.CWB  38-22#
MD.ERR  38-6#
MD.EXP  38-5#
MD.FEU  38-16#
MD.IMF  38-20#
MD.NXU  38-18#
MD.PRI  38-23#
MD.RIP  38-19#
MD.SCH  38-7#
MD.SCL  38-8#
MD.SEC  38-9#
MD.SEQ  38-14#
MD.SER  38-10#
MD.SPD  38-15#
MD.SSH  38-11#
MD.SWP  38-21#
MD.VOL  38-17#
MD.WBN  38-12#
MD.WBV  38-13#
MESG    69-25    73-13#
MESSG   51-9#    75-15
MLDRER 121-17   123-7#
MODE    47-27#   68-53    68-59    116-29*  127-6    127-12
MON1   124-17#  124-30
MON2   124-19   124-21#
MON3   124-22   124-24#
MON4   124-25   124-27#
MON5   124-28   124-33#
MON6   124-36   124-39#
MONTHS  48-7#   124-16
MSCP    35-31#
MSGPKL  54-137#  54-140
MSGPKT  54-93    54-133#
NCONF   76-38    76-43#
NCONS   76-37#   76-40
NEWTAB 117-17   118-3#
```

CROSS REFERENCE TABLE (CREF V04.00)

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

CROSS REFERENCE TABLE (CREF V04.00)

```
P.DMDT   39-50#
P.DPI    40-54#   63-65    63-67    63-72    63-73
P.DTO    40-55#
P.ELGF   39-32#
P.FBBK   40-10#
P.FLGS   40-7#
P.GRP    40-25#
P.HSTI   39-31#   40-19#   40-35#
P.HTMO   39-41#
P.LBN    39-24#
P.MEDI   40-21#   40-37#
P.MLUN   40-17#   40-33#
P.MOD    39-20#
P.OPCD   39-19#   40-6#    62-9     63-58    97-29*
P.OTRF   39-27#   40-13#
P.OVRL   39-51#   95-6*    95-7*
P.RBN    39-36#
P.RBNS   40-28#
P.RCTC   40-29#
P.RCTS   40-27#
P.RGID   39-46#
P.RGOF   39-47#
P.SHST   40-23#   40-39#
P.SHUN   39-33#   40-22#   40-38#
P.STS    40-8#    62-14    95-11
P.TIME   39-43#
P.TRCK   40-24#
P.UADR   39-23#   95-4*    99-18*
P.UNCL   40-40#
P.UNFL   39-30#   40-18#   40-34#
P.UNIT   39-18#   40-5#
P.UNSZ   40-41#
P.UNTI   40-20#   40-36#
P.USEF   39-42#
P.VRSN   39-39#   40-45#
P.VSER   40-42#
PB       91-14#   92-7
PF       47-37    91-12#   92-5
PNT      31-10#
PNT.CT   54-15    54-15    54-15#   54-15#   54-19    54-19    54-19#   54-19#   54-23    54-23    54-23#   54-23#   54-27    54-27#
         54-31    54-31#   54-35    54-35#   54-39    54-39    54-39#   54-39#   54-43    54-43#   54-52    54-52    54-52    54-52#
         54-52#   54-52#   54-57    54-57    54-57    54-57#   54-57#   54-57#   54-62    54-62    54-62    54-62#   54-62#   54-62#
         54-66    54-66    54-66    54-66#   54-66#   54-66#   54-72    54-72#   54-76    54-76    54-76#   54-76#   54-80    54-80
         54-80    54-80#   54-80#   54-80#   54-84    54-84    54-84#   54-84#   54-88    54-88#   54-92    54-92#   54-105   54-105#
         54-109   54-109   54-109#  54-109#  54-113   54-113   54-113#  54-113#  54-117   54-117#  54-120   54-120#  54-126   54-126
         54-126   54-126   54-126   54-126#  54-126#  54-126#  54-126#  54-126#  54-133   54-133#  54-137   54-137   54-137   54-137
         54-137   54-137   54-137   54-137   54-137#  54-137#  54-137#  54-137#  54-137#  54-137#  54-137#  54-137#  75-15    75-15
         75-15    75-15    75-15#   75-15#   75-15#   75-15#   76-41    76-41#   78-23    78-23#   81-4     81-4#    82-5     82-5
         82-5     82-5     82-5     82-5#    82-5#    82-5#    82-5#    82-5#    83-5     83-5     83-5     83-5     83-5     83-5
         83-5#    83-5#    83-5#    83-5#    83-5#    83-5#    114-24   114-24   114-24#  114-24#  114-28   114-28   114-28#  114-28#
         114-33   114-33   114-33#  114-33#  116-48   116-48#  124-31   124-31#  127-19   127-19#  127-29   127-29   127-29   127-29
         127-29#  127-29#  127-29#  127-29#
PNTNUM   77-24    77-30    78-4     89-14#
PNTNUS   89-16#
PNTPKL   54-126#  54-130
PNTPKT   54-97    54-101   54-120#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
PRI      31-10#
PRI00    31-10#  122-12
PRI01    31-10#
PRI02    31-10#
PRI03    31-10#
PRI04    31-10#
PRI05    31-10#
PRI06    31-10#
PRI07    27-34    31-10#   94-20   107-27
PS       91-18#   92-11
PTYPE    47-37#   82-8*    83-8*   91-11   92-5*   92-7*   92-9*   92-11*
PX       82-8     83-8     91-16#  92-9
QUE0     67-19    67-26#
QUE7     67-21    67-27#
QUEL     67-28#   67-30
QUEST    64-3     67-16#
RESPCT   60-8#    61-4     61-7    61-31
RESPDM   60-6#    61-8    127-67
RG.FLG   35-4#    63-74    98-18
RG.OWN   35-3#    60-50    63-74   98-18   98-19
RNTIM    51-5#   114-24
RNTIM1   51-6#   114-28
RNTIM2   51-7#   114-33
RNTIME   75-22    82-6     83-6   114-13#
RNTIMX  114-14   114-35#
RSPDRP   60-25    61-12#   61-37   62-22   63-9    63-27
RSPDSP   63-26    64-3#    64-9
RSPERR   62-10    63-58#
RSPERW   62-15    62-21#   63-59
RSPIN    60-14    62-5#
RSPMWR   62-7     62-9#
RSPNTO   60-30    60-32    60-35   60-37   60-53#
RSPNXT   60-11    61-3#    61-33   63-57   63-75
RSPOU    62-27#
RSPOU2   63-41    63-50#
RSPOU3   63-48    63-51#
RSPOU4   60-52    63-53#
RSPOUT   60-17    62-27    63-34   63-39#
RSPPT2   63-3#
RSPPT3   63-7     63-11#
RSPPTW   62-20    62-26#
RSPTM    60-23    60-29#
RSPTMO   60-34    60-38#
RSPTOE   60-39    60-41    61-36#  63-61   63-76
RUNDM    59-15#  127-65
S$LSYM   27-8#    29-17#   30-14#  54-16#  54-20#  54-24#  54-28#  54-32#  54-36#  54-40#  54-44#  54-53#  54-59#  54-73#
         54-77#   54-81#   54-85#  54-89#  54-94#  54-98#  54-102# 54-106# 54-110# 54-114# 54-118# 102-14# 103-21# 113-9#
        124-1    124-1    124-1   124-1#  124-104# 125-12# 126-13# 127-20  127-20  127-20  127-20#  127-32  127-32  127-32
        127-32#  127-69#  129-6#  131-10#
S.CNS    131-5   131-19#
S.FMT    131-1   131-18#
S.NOF    131-9   131-23#
S.NRF    131-3   131-12#
S.RST    131-7   131-20#
SA.A2     33-18#
SA.BST    34-22#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
SA.CM1   33-32#
SA.CMD   33-27#
SA.CME   33-37#
SA.CTP   33-40#
SA.DI    33-19#  109-14
SA.ERC   33-13#
SA.ERR   33-9#    54-56   110-19
SA.GO    34-20#  106-20
SA.INE   34-4#
SA.INT   33-25#
SA.LFC   34-21#
SA.MCV   34-15#  109-38
SA.MS1   33-31#
SA.MSE   33-36#
SA.MSG   33-26#
SA.NV    33-17#
SA.NVE   34-5#
SA.PRG   33-44#
SA.S1    33-5#   108-3    109-14
SA.S2    33-6#   109-23
SA.S3    33-7#   109-31
SA.S4    33-8#   108-13   109-38
SA.STE   33-39#
SA.STP   33-29#  107-18
SA.TST   34-11#  109-8
SA.VCE   34-3#
SA.VEC   33-24#
SA.WRP   33-28#
SAMTAB  117-21   119-3#
SAMVEC  117-26   123-19#
SERNQ    50-5#   127-32
SERNUM   51-18#  127-29
SERNX    53-11#  127-50
SETOO   104-17#  104-22
SETO1   104-18   104-20#
SETO2   104-26#  104-29
SETTO    63-56   100-15   104-14#  110-23
SFPTBL   30-10#  116-20   116-23   116-27
SNDCMD   63-52    95-8     98-14#
SO.BIT  130-18#  131-1    131-1    131-1    131-3    131-3    131-3    131-5    131-5    131-5    131-7    131-7    131-7
SO.CNS  116-22   116-26   130-22#  131-5
SO.FM1  130-19#  130-21   131-1
SO.FM2  130-20#  130-21   131-3
SO.FMT   68-53   116-19   127-6    130-21#
SO.STR   68-59   116-26   127-12   130-23#  131-7
SPECL    64-8     72-14#   72-28
SPECLC   72-18#
SPECLE   72-24    72-30#
SPECLL   72-23#   72-26
SPECLR   72-16    72-21#
SPECLX   72-15    72-31#
ST.ABO   41-7#
ST.AVL   41-9#
ST.CMD   41-6#
ST.CMP   41-12#
ST.CNT   41-15#
```

```
ST.DAT   41-13#
ST.DIA   41-17#
ST.DRV   41-16#
ST.HST   41-14#
ST.MFE   41-10#
ST.MSK   41-3#    62-14    95-11
ST.OFL   41-8#
ST.SUB   41-4#
ST.SUC   41-5#
ST.WPR   41-11#
SVCGBL   27-8#    27-13#   27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    28-9     29-10
         29-10    30-10    30-10    49-12    49-16    54-11#   54-14    54-14    54-14    54-18    54-18    54-18    54-22    54-22
         54-22    54-26    54-26    54-26    54-30    54-30    54-30    54-34    54-34    54-34    54-38    54-38    54-38    54-42
         54-42    54-42    54-46    54-46    54-46    54-55    54-55    54-55    54-61    54-61    54-61    54-75    54-75    54-75
         54-79    54-79    54-79    54-83    54-83    54-83    54-87    54-87    54-87    54-91    54-91    54-91    54-96    54-96
         54-96    54-100   54-100   54-100   54-104   54-104   54-104   54-108   54-108   54-108   54-112   54-112   54-112   54-116
         54-116   54-116   55-4#    102-10   103-18   113-5    115-10   116-8    125-10   126-8    128-14   130-12   132-8    132-8
         132-8    132-8#
SVCINS   27-8#    27-10#   27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34
         27-34    27-34    27-34    28-9     28-9     28-9     28-9     28-9     28-9     29-10    29-10    29-10    30-10    30-10
         30-10    49-12    49-12    49-12    49-12    49-12    49-12    49-16    49-16    49-16    49-16    49-16    49-16    54-8#
         54-16    54-20    54-24    54-28    54-32    54-36    54-40    54-44    54-53    54-58    54-58    54-58    54-58    54-58
         54-59    54-73    54-77    54-81    54-85    54-89    54-94    54-98    54-102   54-106   54-110   54-114   54-118   55-1#
         56-7     56-7     56-7     56-7     56-7     56-7     56-7     56-7     56-7     56-7     56-7     56-7     56-8     56-8
         56-8     60-8     60-8     60-8     60-24    60-24    60-24    60-24    60-24    60-24    60-24    60-24    60-24    60-24
         60-24    60-24    61-36    61-36    61-36    61-36    61-36    61-36    61-36    61-36    61-36    61-36    61-36    61-36
         62-21    62-21    62-21    62-21    62-21    62-21    62-21    62-21    62-21    62-21    62-21    62-21    63-8     63-8
         63-8     63-8     63-8     63-8     63-8     63-8     63-8     63-8     63-8     63-8     67-22    67-22    67-22    67-22
         67-22    67-22    67-22    67-22    67-22    67-22    67-22    67-22    69-30    69-30    69-30    69-30    69-30    69-30
         69-30    69-30    69-30    69-30    69-30    69-30    72-18    72-18    72-18    72-19    72-19    72-19    72-19    72-19
         72-19    72-23    72-23    72-23    72-23    72-23    72-23    72-24    72-24    72-24    91-12    91-12    91-12    91-12
         91-12    91-12    91-12    91-12    91-12    91-12    91-12    91-12    91-12    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   105-35   105-35   105-35   105-35   105-35   105-35   105-35   105-35   105-35
         105-35   105-35   105-35   106-11   106-11   106-11   106-11   106-11   106-11   106-11   106-11   106-11   106-11   106-11
         106-11   107-12   107-12   107-12   107-27   107-27   107-27   107-27   107-27   107-27   107-27   107-27   107-27   107-27
         107-27   107-27   107-27   107-27   107-27   107-27   107-27   107-27   107-30   107-30   107-30   107-30   107-30   107-30
         107-33   107-33   107-33   107-33   107-33   107-33   107-33   107-33   107-33   107-33   107-33   107-33   109-44   109-44
```

CROSS REFERENCE TABLE (CREF V04.00)

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

CROSS REFERENCE TABLE (CREF V04.00)

```
        54-87#  54-89    54-91#  54-94    54-96#  54-98    54-100#  54-102   54-104#  54-106   54-108#  54-110   54-112#  54-114
        54-116#  54-118
T$$PC   133-1#   133-13
T$$PRO  115-10#
T$$PTA  133-1#   133-3    133-3#
T$$SOF  130-12   130-12#  131-10
T$$SRV  102-10#  102-14   103-18#  103-21   113-5#   113-9
T$$SW    30-10    30-10#   30-14
T$$TES  127-5#   127-11   127-17   127-68   127-69
T$ARGC   27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34    27-34#   27-34#   27-34#
         27-34#   27-34#   27-34#   54-58    54-58    54-58#   91-12    91-12    91-12    91-12#   91-12#   91-14    91-14    91-14
         91-14#   91-14#   91-16    91-16    91-16    91-16#   91-16#   91-18    91-18    91-18    91-18#   91-18#   127-50   127-50
        127-50   127-50#  127-50#
T$CODE  124-1    124-1    124-1    124-1#   124-1#   124-1#   127-20   127-20   127-20   127-20#  127-20#  127-20#  127-32   127-32
        127-32   127-32#  127-32#  127-32#  129-1    129-1    129-1    129-1#   129-1#   129-1#   129-2    129-2    129-2    129-2#
        129-2#   129-2#   129-3    129-3    129-3    129-3#   129-3#   129-3#   129-4    129-4    129-4    129-4#   129-4#   129-4#
        129-5    129-5    129-5    129-5#   129-5#   129-5#   131-1    131-1    131-1    131-1#   131-1#   131-1#   131-2    131-2
        131-2    131-2    131-2    131-2    131-2#   131-2#   131-2#   131-2#   131-3    131-3    131-3    131-3#   131-3#   131-3#
        131-4    131-4    131-4    131-4    131-4    131-4    131-4#   131-4#   131-4#   131-4#   131-5    131-5    131-5    131-5#
        131-5#   131-5#   131-6    131-6    131-6    131-6    131-6    131-6    131-6#   131-6#   131-6#   131-6#   131-7    131-7
        131-7    131-7#   131-7#   131-7#   131-8    131-8    131-8    131-8    131-8    131-8    131-8#   131-8#   131-8#   131-8#
        131-9    131-9#
T$ERRN   27-8#    56-7     56-7#    60-24    60-24#   61-36    61-36#   62-21    62-21#   63-8     63-8#    67-22    67-22#   69-30
         69-30#   96-3     96-3#   100-29   100-29#  101-5    101-5#   105-35   105-35#  106-11   106-11#  107-33   107-33#  109-44
        109-44#  110-36   110-36#  111-5    111-5#   123-3    123-3#   123-8    123-8#   123-14   123-14#  123-20   123-20#  127-10
        127-10#  127-16   127-16#
T$EXCP  124-1    124-1#   127-20   127-20#  127-32   127-32#  129-1    129-1#   129-2    129-2#   129-3    129-3#   129-4    129-4#
        129-5    129-5#
T$FLAG  122-14   122-14   122-14#  122-14#  127-11   127-11   127-11#  127-11#  127-17   127-17   127-17#  127-17#  127-68   127-68
        127-68#  127-68#
T$FREE  132-8    133-13#
T$GMAN   27-8#   124-1    124-1#   124-1#   127-20   127-20#  127-20#  127-32   127-32#  127-32#
T$HILI  124-1    124-1#   127-20   127-20#  127-32   127-32#  129-1    129-1#   129-2    129-2#   129-3    129-3#   129-4    129-4#
        129-5    129-5#
T$LAST   27-8#   132-8#   133-1
T$LOLI  124-1    124-1#   127-20   127-20#  127-32   127-32#  129-1    129-1#   129-2    129-2#   129-3    129-3#   129-4    129-4#
        129-5    129-5#
T$LSYM   27-8     27-8#    29-17    30-14    54-16    54-20    54-24    54-28    54-32    54-36    54-40    54-44    54-53    54-59
         54-73    54-77    54-81    54-85    54-89    54-94    54-98    54-102   54-106   54-110   54-114   54-118   102-14   103-21
        113-9    124-104  125-12   126-13   127-69   129-6    131-10
T$LTNO  132-8#
T$NEST   27-8#    27-26    27-26    27-26#   29-10    29-10    29-10#   29-17    29-17    29-17    29-17#   30-10    30-10    30-10#
         30-14    30-14    30-14    30-14#   30-16    30-16    30-16    30-16#   31-3     31-3     31-3#    54-14    54-14    54-14#
         54-16    54-16    54-16    54-16#   54-18    54-18    54-18#   54-20    54-20    54-20    54-20#   54-22    54-22    54-22#
         54-24    54-24    54-24    54-24#   54-26    54-26    54-26#   54-28    54-28    54-28    54-28#   54-30    54-30    54-30#
         54-32    54-32    54-32    54-32#   54-34    54-34    54-34#   54-36    54-36    54-36    54-36#   54-38    54-38    54-38#
         54-40    54-40    54-40    54-40#   54-42    54-42    54-42#   54-44    54-44    54-44    54-44#   54-46    54-46    54-46#
         54-53    54-53    54-53    54-53#   54-55    54-55    54-55#   54-59    54-59    54-59    54-59#   54-61    54-61    54-61#
         54-73    54-73    54-73    54-73#   54-75    54-75    54-75#   54-77    54-77    54-77    54-77#   54-79    54-79    54-79#
         54-81    54-81    54-81    54-81#   54-83    54-83    54-83#   54-85    54-85    54-85    54-85#   54-87    54-87    54-87#
         54-89    54-89    54-89    54-89#   54-91    54-91    54-91#   54-94    54-94    54-94    54-94#   54-96    54-96    54-96#
         54-98    54-98    54-98    54-98#   54-100   54-100   54-100#  54-102   54-102   54-102   54-102#  54-104   54-104   54-104#
         54-106   54-106   54-106   54-106#  54-108   54-108   54-108#  54-110   54-110   54-110   54-110#  54-112   54-112   54-112#
         54-114   54-114   54-114   54-114#  54-116   54-116   54-116#  54-118   54-118   54-118   54-118#  102-10   102-10   102-10#
        102-14   102-14   102-14   102-14#  103-18   103-18   103-18#  103-21   103-21   103-21   103-21#  113-5    113-5    113-5#
        113-9    113-9    113-9    113-9#   114-38   114-38   114-38   114-38#  115-3    115-3    115-3#   115-10   115-10   115-10#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
        115-16  115-16  115-16  115-16# 116-8   116-8   116-8#  124-104 124-104 124-104 124-104# 125-10  125-10  125-10#
        125-12  125-12  125-12  125-12# 126-8   126-8   126-8#  126-13  126-13  126-13  126-13#  126-15  126-15  126-15
        126-15# 127-3   127-3   127-3#  127-5   127-5   127-5#  127-69  127-69  127-69  127-69#  127-70  127-70  127-70
        127-70# 128-3   128-3   128-3#  128-14  128-14  128-14# 129-6   129-6   129-6   129-6#   130-12  130-12  130-12#
        131-2   131-4   131-6   131-8   131-10  131-10  131-10  131-10# 132-10  132-10  132-10   132-10#
T$NSO    27-26#  30-16   31-3#  114-38  115-3#  126-15  127-3#  127-70  128-3#  132-10
T$NS1    29-10#  29-17   30-10#  30-14   54-14#  54-16   54-18#  54-20   54-22#  54-24   54-26#  54-28   54-30#  54-32
         54-34#  54-36   54-38#  54-40   54-42#  54-44   54-46#  54-53   54-55#  54-59   54-61#  54-73   54-75#  54-77
         54-79#  54-81   54-83#  54-85   54-87#  54-89   54-91#  54-94   54-96#  54-98   54-100#  54-102  54-104#  54-106
         54-108# 54-110  54-112#  54-114  54-116#  54-118  102-10#  102-14  103-18#  103-21  113-5#  113-9  115-10#  115-16
        116-8#  124-104 125-10#  125-12  126-8#  126-13  127-5#  127-69  128-14#  129-6   130-12#  131-2   131-4   131-6
        131-8   131-10
T$PCNT  133-1#  133-3   133-3   133-3#
T$PTAB  133-3   133-3#
T$PTHV   27-34  133-13#
T$PTNU   27-8#  133-3   133-3#  133-13  133-13
T$SAVL   27-8#
T$SEGL   27-8#
T$SIZE  132-8   133-13#
T$SUBN   27-8#  127-5#
T$TAGL   27-8#
T$TAGN   27-8#   29-10   29-10   29-10#  30-10   30-10   30-10#  54-14   54-14   54-14#  54-18   54-18   54-18#  54-22
         54-22   54-22#  54-26   54-26   54-26#  54-30   54-30   54-30#  54-34   54-34   54-34#  54-38   54-38   54-38#
         54-42   54-42   54-42#  54-46   54-46   54-46#  54-55   54-55   54-55#  54-61   54-61   54-61#  54-75   54-75
         54-75#  54-79   54-79   54-79#  54-83   54-83   54-83#  54-87   54-87   54-87#  54-91   54-91   54-91#  54-96
         54-96   54-96#  54-100  54-100  54-100# 54-104  54-104  54-104# 54-108  54-108  54-108# 54-112  54-112  54-112#
         54-116  54-116  54-116# 102-10  102-10  102-10# 103-18  103-18  103-18# 113-5   113-5   113-5#  115-10  115-10
        115-10# 116-8   116-8   116-8#  125-10  125-10  125-10# 126-8   126-8   126-8#  127-5   127-5   127-5#  128-14
        128-14  128-14# 130-12  130-12  130-12# 133-1   133-1   133-1#  133-3   133-3   133-3   133-3   133-3#  133-3#
T$TEMP   28-9    28-9    28-9#   28-9#   29-17   29-17#  30-14   30-14#  30-16   30-16#  54-16   54-16#  54-20   54-20#
         54-24   54-24#  54-28   54-28#  54-32   54-32#  54-36   54-36#  54-40   54-40#  54-44   54-44#  54-53   54-53#
         54-59   54-59#  54-73   54-73#  54-77   54-77#  54-81   54-81#  54-85   54-85#  54-89   54-89#  54-94   54-94#
         54-98   54-98#  54-102  54-102# 54-106  54-106# 54-110  54-110# 54-114  54-114# 54-118  54-118# 102-14  102-14#
        103-21  103-21# 113-9   113-9#  114-38  114-38# 115-16  115-16# 122-14  122-14# 124-1   124-1   124-1   124-1#
        124-1#  124-1#  124-104 124-104# 125-12  125-12# 126-13  126-13# 126-15  126-15# 127-11  127-11# 127-17  127-17#
        127-20  127-20  127-20  127-20# 127-20# 127-20# 127-32  127-32  127-32  127-32# 127-32# 127-32# 127-68  127-68#
        127-69  127-69# 127-70  127-70# 129-1   129-1   129-1   129-1#  129-1#  129-1#  129-2   129-2   129-2   129-2#
        129-2#  129-2#  129-3   129-3   129-3   129-3#  129-3#  129-3#  129-4   129-4   129-4   129-4#  129-4#  129-4#
        129-5   129-5   129-5   129-5#  129-5#  129-5#  129-6   129-6#  131-1   131-1   131-1   131-1#  131-1#  131-1#
        131-3   131-3   131-3   131-3#  131-3#  131-3#  131-5   131-5   131-5   131-5#  131-5#  131-5#  131-7   131-7
        131-7   131-7#  131-7#  131-7#  131-10  131-10# 132-10  132-10#
T$TEST   27-8#  127-5   127-5   127-5#  132-8
T$TSTM   27-8#   54-16   54-20   54-24   54-28   54-32   54-36   54-40   54-44   54-53   54-58   54-59   54-73   54-77
         54-81   54-85   54-89   54-94   54-98   54-102  54-106  54-110  54-114  54-118  56-7    56-8    60-8    60-24
         61-36   62-21   63-8    67-22   69-30   72-18   72-19   72-23   91-12   91-14   91-16   91-18   94-20   96-3
        100-21  100-29  101-5   105-35  106-11  107-12  107-27  107-30  107-33  109-44  110-27  110-36  111-5   116-10
        116-12  116-14  116-16  116-18  116-33  116-34  116-43  116-45  116-54  117-10  120-4   122-12  122-13  122-14
        123-3   123-4   123-8   123-9   123-14  123-15  123-20  123-21  124-1   124-104 125-12  126-10  126-11  126-13
        127-8   127-10  127-11  127-16  127-17  127-20  127-21  127-32  127-50  127-68  127-69
T$TSTS   27-8#  127-5#
T1       28-9   127-5#
T1CNS   127-13  127-23#
T1FMT   127-7   127-22  127-63#
T1GO    127-9   127-12#
T1RST   127-15  127-19#
T1SER1  127-24# 127-62
```

CROSS REFERENCE TABLE (CREF V04.00)

```
T1SER2 127-27#  127-59
T1SER3 127-32#  127-51
T1SER4 127-35#  127-41
T1SER5 127-36   127-43#
T1SER6 127-47#  127-48
T1SER7 127-39   127-42   127-50#
T1SER8 127-44   127-49   127-52#
T1SER9 127-54#  127-55
T1SERN 127-28   127-57   127-60#
TEMP    48-1#   121-9*   123-7    123-13   127-32   127-33   127-45   127-53
TERM    64-6     70-14#
TINDEX  42-10#   42-12    42-12    42-12#   42-13    42-13    42-13#   42-16    42-16    42-16#   42-19    42-19    42-19#   42-20
        42-20    42-20#   42-21    42-21    42-21#   42-22    42-22    42-22#   42-34    42-34    42-34#   42-35    42-35    42-35#
        42-36    42-36    42-36#   42-37    42-37    42-37#   42-38    42-38    42-38#   42-39    42-39    42-39#   42-40    42-40
        42-40#   42-41    42-41    42-41#   42-42    42-42    42-42#   42-43    42-43    42-43#   42-44    42-44    42-44#   42-45
        42-45    42-45#   42-46    42-46    42-46#   42-47    42-47    42-47#   42-49    43-7#    43-9     43-9     43-9#    43-10
        43-10    43-10#   43-13    43-13    43-13#   43-15    128-18#  128-20   128-20   128-20#  128-21   128-21   128-21#  128-22
       128-22   128-22#  128-23   128-23   128-23#  128-24   128-24   128-24#  130-16#  130-18   130-18   130-18#
TOOMER 121-20   123-13#
TSTTAB  47-14#   59-21    60-6     127-63*
UAM     31-10#
UCNT    47-21#   59-20*   59-33*   60-7*    61-6*
UDAFM   47-15    47-16
UDAI1A 105-32#  105-38
UDAI1B 105-34   105-37#
UDAI1C 105-33   105-39#
UDAI1L 105-22#  105-24
UDAI2  106-3#
UDAI2E 106-6    106-11#
UDAI2L 106-5#   106-8
UDAI3  106-9    106-16#
UDAID1 107-19*  109-4#   109-20   109-29
UDAID2 107-20*  107-21*  109-6#
UDAID3 109-8#
UDAIDT 108-4    109-3#
UDAIEX 105-29   105-36   105-42   106-12   106-32#
UDAINT  94-22   105-20#
UDAIR1 109-3    109-14#
UDAIR2 109-5    109-20#
UDAIR3 109-7    109-29#
UDAIR4 109-9    109-37#
UDAIRC 109-15   109-24   109-32   109-42#
UDAIRX 109-43   109-46#
UDAISE 107-35   108-9    108-11   108-19#
UDAISG 107-32   108-3#
UDAISL 108-8#   108-16
UDAIST 105-28   107-12#
UDAISX 108-14   108-18#
UDARS1 110-25#  110-29   110-32   110-34
UDARS2 110-26   111-3#
UDARSD  54-56*   54-57   108-3*   108-12*  108-13   110-19*  110-25   111-16#
UDARSE 110-37   111-6#
UDARSP 105-41   108-8    110-18#
UDARSX 111-4    111-11#
UDASRV 103-18#  118-14
UF.576  39-12#
```

CROSS REFERENCE TABLE (CREF V04.00)

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

CROSS REFERENCE TABLE (CREF V04.00)

```
.BR      44-31#
AND      44-3#    54-49    94-17    107-15
ASSUME   44-39#   59-27    61-20    68-36    74-15    75-16    75-17    116-26   117-19   117-20   127-30   127-31
BCOMPL   116-11   116-13   116-15   116-17   116-44   116-46   127-9
BGNAUT   125-10
BGNCLN   126-8
BGNHRD   128-14
BGNHW    29-10
BGNINI   116-8
BGNMOD   27-26    31-3     115-3    127-3    128-3
BGNMSG   54-14    54-18    54-22    54-26    54-30    54-34    54-38    54-42    54-46    54-55    54-61    54-75    54-79    54-83
         54-87    54-91    54-96    54-100   54-104   54-108   54-112   54-116
BGNPRO   115-10
BGNPTA   133-3
BGNSET   133-1
BGNSFT   130-12
BGNSRV   102-10   103-18   113-5
BGNSW    30-10
BGNTST   127-5
BNCOMP   72-24    117-11   120-5
BREAK    60-8     100-21   107-12   110-27
BRESET   116-33   126-11
BUILD    79-18#   80-9     80-21
CLOCK    116-43   116-45
CLOSE    72-18    122-13   126-10
CLRVEC   107-30
DESCRI   49-16
DEVTYP   49-12
DISPAT   28-9
DISPLA   131-9
DOCLN    56-8     116-18   123-4    123-9    123-15   123-21
END      32-29#   42-49    43-15    128-25   130-25
ENDAUT   125-12
ENDCLN   126-13
ENDHRD   129-6
ENDHW    29-17
ENDINI   124-104
ENDMOD   30-16    114-38   126-15   127-70   132-10
ENDMSG   54-16    54-20    54-24    54-28    54-32    54-36    54-40    54-44    54-53    54-59    54-73    54-77    54-81    54-85
         54-89    54-94    54-98    54-102   54-106   54-110   54-114   54-118
ENDPRO   115-16
ENDPTA   133-11
ENDSET   133-13
ENDSFT   131-10
ENDSRV   102-14   103-21   113-9
ENDSW    30-14
ENDTST   127-69
ENTRY    80-3#    80-9     80-9     80-9     80-9     80-9     80-9     80-9     80-9     80-15#   80-21    80-21    80-21    80-21
         80-21    80-21    80-21    80-21
EQUALS   31-10
ERRDF    60-24    61-36    62-21    63-8     67-22    69-30    96-3     100-29   101-5    105-35   106-11   107-33   109-44   110-36
         111-5
ERRSF    56-7     123-3    123-8    123-14   123-20   127-10   127-16
EXIT     122-14   127-11   127-17   127-68
GETBYT   72-23
GMANID   124-1    127-20   127-32
```

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

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

CROSS REFERENCE TABLE (CREF V04.00)

	123-8#	123-9	123-9#	123-14	123-14	123-14	123-14	123-14#	123-14#	123-14#	123-14#	123-14#	123-15	123-15#
	123-20	123-20	123-20	123-20	123-20#	123-20#	123-20#	123-20#	123-20#	123-21	123-21#	124-1	124-1	124-1
	124-1	124-1	124-1	124-1	124-1	124-1#	124-1#	124-1#	124-1#	124-104	124-104#	125-12	125-12#	126-10
	126-10#	126-11	126-11#	126-13	126-12#	127-8	127-8#	127-9	127-9#	127-10	127-10	127-10	127-10	127-10#
	127-10#	127-10#	127-10#	127-10#	127-11	127-11	127-11#	127-11#	127-16	127-16	127-16	127-16	127-16#	127-16#
	127-16#	127-16#	127-16#	127-17	127-17	127-17#	127-17#	127-20	127-20	127-20	127-20	127-20	127-20	127-20
	127-20	127-20#	127-20#	127-20#	127-20#	127-21	127-21	127-21#	127-21#	127-32	127-32	127-32	127-32	127-32
	127-32	127-32	127-32	127-32#	127-32#	127-32#	127-32#	127-50	127-50	127-50	127-50	127-50	127-50	127-50#
	127-50#	127-50#	127-50#	127-50#	127-68	127-68	127-68#	127-68#	127-69	127-69#	128-14	128-14#	129-1	129-1
	129-1	129-1	129-1#	129-2	129-2	129-2	129-2	129-2#	129-3	122-3	129-3	129-3	129-3	129-3#
	129-4	129-4	129-4	129-4	129-4	129-4#	129-5	129-5	129-5	129-5	129-5	129-5#	129-6	129-6#
	130-12	130-12#	131-1	131-1	131-1	131-1#	131-2	131-2#	131-3	131-3	131-3	131-3#	131-4	131-4#
	131-5	131-5	131-5	131-5#	131-6	131-6#	131-7	131-7	131-7	131-7#	131-8	131-8#	131-9	131-9
	131-9#	131-9#	131-10	131-10#	132-8	132-8	132-8	132-8#	133-3	133-3	133-3#	133-3#		
M$GNLS	124-1	124-1#	127-20	127-20#	127-32	127-32#								
M$GNTA	29-17	29-17#	30-14	30-14#	54-16	54-16#	54-20	54-20#	54-24	54-24#	54-28	54-28#	54-32	54-32#
	54-36	54-36#	54-40	54-40#	54-44	54-44#	54-53	54-53#	54-59	54-59#	54-73	54-73#	54-77	54-77#
	54-81	54-81#	54-85	54-85#	54-89	54-89#	54-94	54-94#	54-98	54-98#	54-102	54-102#	54-106	54-106#
	54-110	54-110#	54-114	54-114#	54-118	54-118#	102-14	102-14#	103-21	103-21#	113-9	113-9#	124-104	124-104#
	125-12	125-12#	126-13	126-13#	127-69	127-69#	129-6	129-6#	131-10	131-10#	133-3	133-3#	133-11	133-11#
M$GNTE	127-5	127-5#												
M$HAPT	27-34	27-34#												
M$HNAP	27-34	27-34#												
M$INCR	27-26	27-26#	29-10	29-10	29-10#	29-10#	30-10	30-10	30-10#	30-10#	31-3	31-3#	54-14	54-14
	54-14#	54-14#	54-16#	54-18	54-18	54-18#	54-18#	54-20#	54-22	54-22	54-22#	54-22#	54-24#	54-26
	54-26	54-26#	54-26#	54-28#	54-30	54-30	54-30#	54-30#	54-32#	54-34	54-34	54-34#	54-34#	54-36#
	54-38	54-38	54-38#	54-38#	54-40#	54-42	54-42	54-42#	54-42#	54-44#	54-46	54-46	54-46#	54-46#
	54-53#	54-55	54-55	54-55#	54-55#	54-58#	54-59#	54-61	54-61	54-61#	54-61#	54-73#	54-75	54-75
	54-75#	54-75#	54-77#	54-79	54-79	54-79#	54-79#	54-81#	54-83	54-83	54-83#	54-83#	54-85#	54-87
	54-87	54-87#	54-87#	54-89#	54-91	54-91	54-91#	54-91#	54-94#	54-96	54-96	54-96#	54-96#	54-98#
	54-100	54-100	54-100#	54-100#	54-102#	54-104	54-104	54-104#	54-104#	54-106#	54-108	54-108	54-108#	54-108#
	54-110#	54-112	54-112	54-112#	54-112#	54-114#	54-116	54-116	54-116#	54-116#	54-118#	56-7#	56-8#	60-8#
	60-24#	61-36#	62-21#	63-8#	67-22#	69-30#	72-18#	72-19#	72-23#	91-12#	91-14#	91-16#	91-18#	94-20#
	96-3#	100-21#	100-29#	101-5#	102-10	102-10	102-10#	102-10#	103-18	103-18	103-18#	103-18#	105-35#	106-11#
	107-12#	107-27#	107-30#	107-33#	109-44#	110-27#	110-36#	111-5#	113-5	113-5	113-5#	113-5#	115-3	115-3#
	115-10	115-10	115-10#	115-10#	116-8	116-8	116-8#	116-8#	116-10#	116-12#	116-14#	116-16#	116-18#	116-33#
	116-34#	116-43#	116-45#	116-54#	117-10#	120-4#	122-12#	122-13#	122-14#	123-3#	123-4#	123-8#	123-9#	123-14#
	123-15#	123-20#	123-21#	124-1	124-1#	124-1#	124-104#	125-10	125-10	125-10#	125-10#	125-12#	126-8	126-8
	126-8#	126-8#	126-10#	126-11#	126-13#	127-3	127-3#	127-5	127-5	127-5	127-5#	127-5#	127-5#	127-8#
	127-10#	127-11#	127-16#	127-17#	127-20	127-20#	127-20#	127-21#	127-32	127-32#	127-32#	127-50#	127-68#	127-69#
	128-3	128-3#	128-14	128-14	128-14#	128-14#	130-12	130-12	130-12#	130-12#	133-1	133-1#	133-3	133-3
	133-3	133-3#												
M$LDRO	72-19	72-19#	107-30	107-30#	116-10	116-10#	116-12	116-12#	116-14	116-14#	116-16	116-16#	116-43	116-43#
	116-45	116-45#	117-10	117-10#	120-4	120-4#	122-12	122-12#	127-21	127-21#				
M$MCHI	27-8	27-8#												
M$MCLO	27-8	27-8#												
M$POP	29-17	29-17#	30-14	30-14#	30-16	30-16#	54-16	54-16#	54-20	54-20#	54-24	54-24#	54-28	54-28#
	54-32	54-32#	54-36	54-36#	54-40	54-40#	54-44	54-44#	54-53	54-53#	54-59	54-59#	54-73	54-73#
	54-77	54-77#	54-81	54-81#	54-85	54-85#	54-89	54-89#	54-94	54-94#	54-98	54-98#	54-102	54-102#
	54-106	54-106#	54-110	54-110#	54-114	54-114#	54-118	54-118#	102-14	102-14#	103-21	103-21#	113-9	113-9#
	114-38	114-38#	115-16	115-16#	124-104	124-104#	125-12	125-12#	126-13	126-13#	126-15	126-15#	127-69	127-69#
	127-70	127-70#	129-6	129-6#	131-10	131-10#	132-10	132-10#						
M$PRIN	54-58	54-58#	91-12	91-12#	91-14	91-14#	91-16	91-16#	91-18	91-18#	127-50	127-50#		
M$PUSH	27-26	27-26#	29-10	29-10#	30-10	30-10#	31-3	31-3#	54-14	54-14#	54-18	54-18#	54-22	54-22#
	54-26	54-26#	54-30	54-30#	54-34	54-34#	54-38	54-38#	54-42	54-42#	54-46	54-46#	54-55	54-55#
	54-61	54-61#	54-75	54-75#	54-79	54-79#	54-83	54-83#	54-87	54-87#	54-91	54-91#	54-96	54-96#

CROSS REFERENCE TABLE (CREF V04.00)

	54-100	54-100#	54-104	54-104#	54-108	54-108#	54-112	54-112#	54-116	54-116#	102-10	102-10#	103-18	103-18#
	113-5	113-5#	115-3	115-3#	115-10	115-10#	116-8	116-8#	125-10	125-10#	126-8	126-8#	127-3	127-3#
	127-5	127-5#	128-3	128-3#	128-14	128-14#	130-12	130-12#						
M$PUT	54-58	54-58	54-58#	91-12	91-12	91-12	91-12#	91-14	91-14	91-14	91-14#	91-16	91-16	91-16
	91-16#	91-18	91-18	91-18	91-18#	94-20	94-20	94-20	94-20	94-20#	107-27	107-27	107-27	107-27
	107-27#	116-54	116-54	116-54	116-54	116-54#	127-50	127-50	127-50	127-50#				
M$PUT1	54-58	54-58	54-58#	54-58#	91-12	91-12	91-12	91-12#	91-12#	91-12#	91-14	91-14	91-14	91-14#
	91-14#	91-14#	91-16	91-16	91-16	91-16#	91-16#	91-16#	91-18	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#	107-27	107-27	107-27	107-27	107-27#	107-27#
	107-27#	107-27#	116-54	116-54	116-54	116-54	116-54#	116-54#	116-54#	116-54#	127-50	127-50	127-50	127-50#
	127-50#	127-50#												
M$RADI	124-1	124-1#	127-20	127-20#	127-32	127-32#	129-1	129-1#	129-2	129-2#	129-3	129-3#	129-4	129-4#
	129-5	129-5#	131-1	131-1#	131-3	131-3#	131-5	131-5#	131-7	131-7#				
M$RBRO	72-23	72-23#												
M$RNRO	116-34	116-34#	116-43	116-43#	116-45	116-45#	117-10	117-10#	120-4	120-4#				
M$SETS	27-26	27-26#	29-10	29-10#	30-10	30-10#	31-3	31-3#	54-14	54-14#	54-18	54-18#	54-22	54-22#
	54-26	54-26#	54-30	54-30#	54-34	54-34#	54-38	54-38#	54-42	54-42#	54-46	54-46#	54-55	54-55#
	54-61	54-61#	54-75	54-75#	54-79	54-79#	54-83	54-83#	54-87	54-87#	54-91	54-91#	54-96	54-96#
	54-100	54-100#	54-104	54-104#	54-108	54-108#	54-112	54-112#	54-116	54-116#	102-10	102-10#	103-18	103-18#
	113-5	113-5#	115-3	115-3#	115-10	115-10#	116-8	116-8#	125-10	125-10#	126-8	126-8#	127-3	127-3#
	127-5	127-5#	128-3	128-3#	128-14	128-14#	130-12	130-12#						
M$SVC	54-16	54-16#	54-20	54-20#	54-24	54-24#	54-28	54-28#	54-32	54-32#	54-36	54-36#	54-40	54-40#
	54-44	54-44#	54-53	54-53#	54-58	54-58#	54-59	54-59#	54-73	54-73#	54-77	54-77#	54-81	54-81#
	54-85	54-85#	54-89	54-89#	54-94	54-94#	54-98	54-98#	54-102	54-102#	54-106	54-106#	54-110	54-110#
	54-114	54-114#	54-118	54-118#	56-7	56-8	56-8#	60-8	60-8#	60-24	61-36	62-21	63-8	67-22
	69-30	72-18	72-18#	72-19	72-19#	72-23	72-23#	91-12	91-12#	91-14	91-14#	91-16	91-16#	91-18
	91-18#	94-20	94-20#	96-3	100-21	100-21#	100-29	101-5	105-35	106-11	107-12	107-12#	107-27	107-27#
	107-30	107-30#	107-33	109-44	110-27	110-27#	110-36	111-5	116-10	116-10#	116-12	116-12#	116-14	116-14#
	116-16	116-16#	116-18	116-18#	116-33	116-33#	116-34	116-34#	116-43	116-43#	116-45	116-45#	116-54	116-54#
	117-10	117-10#	120-4	120-4#	122-12	122-12#	122-13	122-13#	122-14	122-14#	123-3	123-4	123-4#	123-8
	123-9	123-9#	123-14	123-15	123-15#	123-20	123-21	123-21#	124-1	124-1#	124-104	124-104#	125-12	125-12#
	126-10	126-10#	126-11	126-11#	126-13	126-13#	127-8	127-8#	127-10	127-11	127-11#	127-16	127-17	127-17#
	127-20	127-20#	127-21	127-21#	127-32	127-32#	127-50	127-50#	127-68	127-68#	127-69	127-69#		
M$TLAB	54-16#	54-20#	54-24#	54-28#	54-32#	54-36#	54-40#	54-44#	54-53#	54-58#	54-59#	54-73#	54-77#	54-81#
	54-85#	54-89#	54-94#	54-98#	54-102#	54-106#	54-110#	54-114#	54-118#	56-7#	56-8#	60-8#	60-24#	61-36#
	62-21#	63-8#	67-22#	69-30#	72-18#	72-19#	72-23#	91-12#	91-14#	91-16#	91-18#	94-20#	96-3#	100-21#
	100-29#	101-5#	105-35#	106-11#	107-12#	107-27#	107-30#	107-33#	109-44#	110-27#	110-36#	111-5#	116-10#	116-12#
	116-14#	116-16#	116-18#	116-33#	116-34#	116-43#	116-45#	116-54#	117-10#	120-4#	122-12#	122-13#	122-14#	123-3#
	123-4#	123-8#	123-9#	123-14#	123-15#	123-20#	123-21#	124-1#	124-104#	125-12#	126-10#	126-11#	126-13#	127-8#
	127-10#	127-11#	127-16#	127-17#	127-20#	127-21#	127-32#	127-50#	127-68#	127-69#				
M$TSTL	54-16	54-16#	54-20	54-20#	54-24	54-24#	54-28	54-28#	54-32	54-32#	54-36	54-36#	54-40	54-40#
	54-44	54-44#	54-53	54-53#	54-58	54-58#	54-59	54-59#	54-73	54-73#	54-77	54-77#	54-81	54-81#
	54-85	54-85#	54-89	54-89#	54-94	54-94#	54-98	54-98#	54-102	54-102#	54-106	54-106#	54-110	54-110#
	54-114	54-114#	54-118	54-118#	56-7	56-7#	56-7#	56-8	56-8#	60-8	60-8#	60-24	60-24#	60-24#
	61-36	61-36#	61-36#	62-21	62-21#	62-21#	63-8	63-8#	63-8#	67-22	67-22#	67-22#	69-30	69-30#
	69-30#	72-18	72-18#	72-19	72-19#	72-23	72-23#	91-12	91-12#	91-14	91-14#	91-16	91-16#	91-18
	91-18#	94-20	94-20#	96-3	96-3#	96-3#	100-21	100-21#	100-29	100-29#	100-29#	101-5	101-5#	101-5#
	105-35	105-35#	105-35#	106-11	106-11#	106-11#	107-12	107-12#	107-27	107-27#	107-30	107-30#	107-33	107-33#
	107-33#	109-44	109-44#	109-44#	110-27	110-27#	110-36	110-36#	110-36#	111-5	111-5#	111-5#	116-10	116-10#
	116-12	116-12#	116-14	116-14#	116-16	116-16#	116-18	116-18#	116-33	116-33#	116-34	116-34#	116-43	116-43#
	116-45	116-45#	116-54	116-54#	117-10	117-10#	120-4	120-4#	122-12	122-12#	122-13	122-13#	122-14	122-14#
	123-3	123-3#	123-3#	123-4	123-4#	123-8	123-8#	123-8#	123-9	123-9#	123-14	123-14#	123-14#	123-15
	123-15#	123-20	123-20#	123-20#	123-21	123-21#	124-1	124-1#	124-104	124-104#	125-12	125-12#	126-10	126-10#
	126-11	126-11#	126-13	126-13#	127-8	127-8#	127-10	127-10#	127-10#	127-11	127-11#	127-16	127-16#	127-16#
	127-17	127-17#	127-20	127-20#	127-21	127-21#	127-32	127-32#	127-50	127-50#	127-68	127-68#	127-69	127-69#
M$WORD	27-34	27-34#	28-9	28-9	28-9#	56-7	56-7	56-7	56-7#	60-24	60-24	60-24	60-24#	61-36

CROSS REFERENCE TABLE (CREF V04.00)