

RA60/80/81
UDA50A KDA50Q

UDA50/KDA50-Q FMTR
CZUDKAO

COPYRIGHT (c) 1984
AH-T939A-MC FICHE
FICHE 01 OF 01

FEB 1985
digital
Made In USA

Microfiche grid containing multiple frames of data, including text and tables. The data is too small to transcribe accurately but appears to be organized in a structured format.

IDENTIFICATION

PRODUCT CODE: AC T938A MC
PRODUCT NAME: CZUDKAO UDA50A/KDASO-Q FORMATTER
PRODUCT DATE: 2-OCT-1984
MAINTAINER: ROGER OAKY
AUTHOR: BRIAN SCHOW

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

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

COPYRIGHT (C) 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DEC	DIBOL	RSX
DEC/CMS	EduSystem	UNIBUS
DECnet	IAS	VAX
DECsystem-10	MASSBUS	VMS
DECSYSTEM-20	PDP	VT
DECUS	PDT	Digital Logo
DECwriter	RSTS	

1

.REM

.TITLE CZUDKO UDASOA/KDASO-Q FORMATTER

TABLE OF CONTENTS

	Page	
1.0	GENERAL INFORMATION	3
1.1	PROGRAM ABSTRACT	3
1.2	SYSTEM REQUIREMENTS	4
2.0	OPERATING INSTRUCTIONS	4
2.1	COMMANDS	4
2.2	SWITCHES	5
2.3	FLAGS	6
2.4	HARDWARE QUESTIONS	7
2.5	SOFTWARE QUESTIONS	8
2.6	MANUAL INTERVENTION QUESTIONS	9
2.7	EXTENDED P-TABLE DIALOGUE	10
2.8	QUICK STARTUP PROCEDURE	12
3.0	ERROR INFORMATION	15
3.1	TYPES OF ERROR MESSAGES	15
3.2	SPECIFIC ERROR MESSAGES	16
3.2.1	HOST PROGRAM ERROR MESSAGES	16
3.2.2	DUP PROGRAM ERROR MESSAGES	24
4.0	PERFORMANCE AND PROGRESS REPORTS	28
5.0	TEST SUMMARIES	29

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

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

1. Reformat - Format the disk with the bad sector information that was written onto the disk at the factory. This is the normal way to format a disk.
2. Reconstruct - Format the disk without using any bad sector information. This should be used only when the bad sector information has been destroyed or for some reason can no longer be read from the disk. This method may also be specified in the disk drive's maintenance manual for special cases (eg. changing an RM/RA80 spare HDA from RM80 format to RA80 format).
3. Restore - Format the disk using bad sector information obtained from a disk file on the XXDP+ system load device. This method is provided for use by manufacturing. No files are provided, nor any method of obtaining the files, at this time.

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

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

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

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

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

1.2 SYSTEM REQUIREMENTS

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

- PDP-11 Unibus or Q-bus processor
- 28K words of memory (minimum)
- Console terminal
- XXDP load media containing this program
- One or more UDA50A or KDA50-Q subsystems.

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

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

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

2.3 FLAGS

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

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

*Error messages are described in section 3.1

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

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

2.4 HARDWARE QUESTIONS

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

CSR ADDRESS (0) 172150 ?

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

VECTOR (0) 154 ?

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

DRIVE NUMBER (0) 0 ?

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

2.5 SOFTWARE QUESTIONS

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

REFORMAT USING EXISTING BAD SECTOR INFORMATION (L) Y ?

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

NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR INFORMATION ON THE DISK.

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

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

RECONSTRUCT BAD SECTOR INFORMATION (L) Y ?

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

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

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

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

2.6 MANUAL INTERVENTION QUESTIONS

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

WARNING:

THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC TOOL. RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK DRIVE'S SERVICE MANUAL.

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

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

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

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

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

1. If the number typed is 70 or greater, a 19 is prefixed.
Eg., 1-JAN-70 translates to year 1970 and 25-DEC-99 translates to year 1999.
2. If the number typed is less than 70, a 20 is prefixed. Eg., 1-APR-21 is translated to year 2021.

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

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

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

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

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

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

2.7 EXTENDED P-TABLE DIALOGUE

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

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

UNITS (D) ? 8<CR>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

CZUDKO UD450A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 10 1
USER DOCUMENTATION

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

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

UNIT 1
CSR ADDRESS (D) ? 160000<CR>
SUB-DEVICE # (D) ? 0 7<CR>
Q-FACTOR (D) 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 ZUDKAO"
4. Type "START"
5. Answer the "CHANGE HW" question with "Y"
6. Answer all the hardware questions
7. Answer the "CHANGE SW" question with "N"
8. Answer "Y" to the "ARE YOU SURE ..." question following the warning. Please read the disk drive's service manual before answering this question.
9. Type today's date.

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 11
USER DOCUMENTATION

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

DR>STA

CHANGE HW (L) ? Y

UNITS (D) ? 2

UNIT 0

CSR ADDRESS (O) 172150 ?

VECTOR (O) 154 ?

DRIVE NUMBER (D) 0 ? 0,1

CHANGE SW (L) ? N

WARNING:

THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC
TOOL. RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK
DRIVE'S SERVICE MANUAL.

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

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

UNIT 0 CONTROLLER AT 172150 DRIVE 0 RUNTIME 0:00:20

Format begun Version 11

STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK
UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN
BROUGHT ONLINE.

UNIT 1 CONTROLLER AT 172150 DRIVE 1 RUNTIME 0:00:23

Format begun Version 11

STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK
UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN
BROUGHT ONLINE.

UNIT 0 CONTROLLER AT 172150 DRIVE 0 RUNTIME 0:42:20

Format completed

2 Rectorated LBNS

2 Primary revectorated LBNS

0 Secondary/tertiary revectorated LBNS

0 Bad RBNS

0 Bad blocks in the RCT area due to data errors

0 Bad blocks in the DBN area due to data errors

0 Bad blocks in the XBN area due to data errors

2 Blocks retried on the check pass

FCT used successfully

CZUDKO UD450A/KD450-Q FORMATTER MACRO V05.01b Monday 01 Oct 84 10:07 Page 12
 USER DOCUMENTATION

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

CZUDK EOP 1
 0 CUMULATIVE ERRORS
 DR>

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

DR>STA

CHANGE HW (L) ? N

CHANGE SW (L) ? Y

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

WARNING:

THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC
 TOOL. RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK
 DRIVE'S SERVICE MANUAL.

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

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

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

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

CZUDK EOP 1
 0 CUMULATIVE ERRORS
 DR>

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

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

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

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

HOST PROGRAM CONTROLLER AT xxxxxx RUNTIME hhh:mm:ss

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

Sample error message:

```

CZUDK DVC FTL ERR 00021 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME 0:00:12
CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE
  SA CONTAINS 104041
REPLACE CONTROLLER PROCESSOR MODULE

```

general message

basic message

extended message

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

3.2 SPECIFIC ERROR MESSAGES

3.2.1 HOST PROGRAM ERROR MESSAGES

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

```

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

```

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

```

00002 CZUDK SYS FTL ERR 00002 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS
MULTIPLE UNITS SELECT THE SAME DRIVE

```

The hardware questions for two units were exactly the same. The program is aborted and returns to the Runtime Services prompt so that the hardware questions may be changed.

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

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

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

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

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

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

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

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

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

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 15
USER DOCUMENTATION

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

All UDA50-0's (modules M7161-2) are not supported by this
formatter. The module sets M7485-6 and M7777-7 are the only
ones that can be used by this formatter. If the controller
is a UDA50-0 (M7161-2) it will not be tested. If the
controller consists of the M7161-2 modules, install one with
M7485-6 modules. Replace both modules, mixing the module
sets will not work.

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

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 16
USER DOCUMENTATION

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

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

- 104000 - Fatal sequencer error
- 104040 - D processor ALU error
- 104041 - D proc ROM parity error
- 105102 - D proc with no Board #2 or RAM parity error
- 105105 - D proc RAM buffer error
- 105152 - D proc SDI error
- 105153 - D proc write mode wrap SERDES error
- 105154 - D proc read mode SERDES, RSGEN, and ECC error
- 106040 - U proc ALU error
- 106041 - U proc Control Register error
- 106042 - U proc DFIL/ROM parity error/Board #1 test count is wrong
- 106047 - U proc Constant ROM error with D proc running SDI test
- 106055 - Unexpected trap found, aborted diagnostic
- 106071 - U proc ROM error
- 106072 - U proc ROM parity error
- 106200 - Step 1 data error (MSB not set)
- 107103 - U proc RAM parity error
- 107107 - U proc RAM buffer error
- 107115 - Board #2 test count was wrong
- 112300 - Step 2 error
- 122240 - NPR error
- 122300 - Step 3 error
- 142300 - Step 4 error

Replace the board specified in the last line of the error message.

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

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

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

Neither the expected step bit nor the error bit set within the expected time.

00023 CZUDK DVC FTL ERR 00023 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx.
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
 CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION
 6 WORDS WERE TO BE CLEARED STARTING AT ADDRESS 040644
 FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):

ADDRESS	CONTENTS
040644	000010
040650	000010
040652	000010

REPLACE CONTROLLER PROCESSOR MODULE

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

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

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

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

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

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

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

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

00031 CZUDK DVC FTL ERR 00031 ON UNIT 00 TST 001 SUB 000 PC: xxxxxx
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME x:xx:xx
 FORMATTER IS HUNG

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

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

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

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

COMMAND PACKET SENT	RESPONSE PACKET RECEIVED
000000 000020	000000 000020
000000 000000	000000 000000
000000 000002	000000 000202
000000 014336	000000 014336
000000 034674	000000 034674
000000 000000	000000 000000
000000 000000	000000 000000
000000 051232	000000 051232
000000 000000	000000 000000
000000 000000	000000 000000
000000 000000	000000 000000
000000 000000	000000 000000

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

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

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

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

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

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

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

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

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

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

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

3.2.2 DUP PROGRAM ERROR MESSAGES

Error messages returned by the formatter are as follows:

GET STATUS failure

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

SDI send error

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

Unsuccessful SDI command

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

SDI receive error

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

BUS read error

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

Formatter initialization error

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

Non-existent unit number

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

DBN/XBN format error (drive FORMAT command failed)

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

FCT does not have enough good copies of each block

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

SEEK error

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

RCT does not have enough good copies of each block

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

LBN format error (drive FORMAT command failed)

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

FCT write error

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

RCT read error

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

RCT write error

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

RCT full

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

FCT read error

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

FCT downline-load error

The formatter was led to believe that a bad block information file was larger than it really was. There may be a BUS or controller processor module problem.

Drive init timeout

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

Illegal response to start-up question

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

FCT corrupted - Format Invalid

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

(ZUDKO UDAS0A/KDAS0-0 FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 23
USER DOCUMENTATION

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

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

MORE THAN 12.5% OF TRACK IS BAD

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

An example of how the errors are presented is below:

RUNTIME 0:00:18
Non-existent unit number

4.0 PERFORMANCE AND PROGRESS REPORTS

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

The DUP program issues the following messages upon normal completion:

Format completed

n Revectored LBNS

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

n Primary revectored LBNS

Where n is the number of LBNS which were primary revector.

n Secondary/tertiary revectored LBNS

Where n is the number of the LBNS which were secondary or tertiary revector.

n Bad RBNS

Where n is the number of RBNS which were bad.

n Bad blocks in the RCT area due to data errors

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

n Bad blocks in the DBN area due to data errors

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

n Bad blocks in the XBN area due to data errors

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 25
USER DOCUMENTATION

n Blocks retried on the check pass

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

FCT used successfully or
FCT was not used

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

An example of how the messages are presented is below.

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

5.0 TEST SUMMARIES

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

```

1
25
26 002000
27
28
29
30
31
32 002000
33
34 002000
002000
002000 103
002001 132
002002 125
002003 104
002004 113
002005 000
002006 000
002007 000
002010
002010 101
002011
002011 060
002012
002012 000001
002014
002014 016040
002016
002016 022754
002020
002020 023032
002022
002022 002130
002024
002024 002136
002026
002026 000124
002030
002030 000000
002032
002032 000000
002034
002034 000001
002036
002036 000000
002040
002040 002124
002042
002042 000340
002044
002044 000000
002046
002046 000000
002050
002050 003
002051 003

```

```

.SBTTL PROGRAM
      BGNMOD
;
; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
;
      POINTER BGNSW, BGNSFT, BGNSETUP
      HEADER CZUDK,A,0,7200.,1,PRI07

```

```

L$NAME::
      .ASCII /C/
      .ASCII /Z/
      .ASCII /U/
      .ASCII /D/
      .ASCII /K/
      .BYTE 0
      .BYTE 0
      .BYTE 0
L$REV::
      .ASCII /A/
L$DEPO::
      .ASCII /O/
L$UNIT::
      .WORD T$PTHV
L$TIML::
      .WORD 7200.
L$HPCP::
      .WORD L$HARD
L$SPCP::
      .WORD L$SOFT
L$HPTP::
      .WORD L$HW
L$SPTP::
      .WORD L$SW
L$LADP::
      .WORD L$LAST
L$STA::
      .WORD 0
L$CO::
      .WORD 0
L$DTYP::
      .WORD 1
L$APT::
      .WORD 0
L$DTP::
      .WORD L$DISPATCH
L$PRIO::
      .WORD PRI07
L$ENVI::
      .WORD 0
L$EXP1::
      .WORD 0
L$MREV::
      .BYTE C$REVISION
      .BYTE C$EDIT

```

```

002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 003454
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 000000
002072
002072 000000
002074
002074 000000
002076
002076 003502
002100
002100 104035
002102
002102 000000
002104
002104 021242
002106
002106 022200
002110
002110 022176
002112
002112 021234
002114
002114 000000
002116
002116 000000
002120
002120 000000
    
```

```

L$EF:: .WORD 0
        .WORD 0
L$SPC:: .WORD 0
L$DEVP:: .WORD L$DVTYP
L$REPP:: .WORD 0
L$EXPA:: .WORD 0
L$EXPS:: .WORD 0
L$AUT:: .WORD 0
L$DUT:: .WORD 0
L$LUN:: .WORD 0
L$DESP:: .WORD L$DESC
L$LOAD:: EMT E$LOAD
L$ETP:: .WORD 0
L$ICP:: .WORD L$INIT
L$CCP:: .WORD L$CLEAN
L$ACP:: .WORD L$AUTO
L$PRT:: .WORD L$PROT
L$TEST:: .WORD 0
L$DLY:: .WORD 0
L$HIME:: .WORD 0
    
```

1
2
3
4
5
6
7
8

.SBTTL DISPATCH TABLE

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

DISPATCH 1

002122
002122 000001
002124
002124 022264

.WORD 1
L#DISPATCH::
.WORD T1

.SBTTL DEFAULT HARDWARE P-TABLE

1
2
3
4
5
6
7
8
9

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

10	002126		BGNHW	DFPTBL			
	002126	000002				.WORD	L10000-L#HW/2
	002130					L#HW::	
	002130					DFPTBL::	
11							
12	002130	172150	.WORD	172150			; UNIBUS ADDRESS
13	002132	000000	.WORD	0.			; LOGICAL DRIVE NUMBER
14	002134		ENDHW				
	002134						L10000:

```

1      .SBTTL  SOFTWARE P-TABLE
2
3
4      ;**
5      ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
6      ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
7      ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
8      ; AT RUN TIME.
9      ;--
10     002134      BGNSW  SFPTBL
11     002134      000001
12     002136
13     002136      .WORD  L10001-L#SW/2
14     002140      USE
15     002140      SFPTBL::
16     002140      ;OFFSET  YES/NO ANSWERS
17     002140      ; 0.
18     002140      L10001:
19     002140      ENDSW  7
20     002140      ENDMOD

```

1
 2
 3 002140
 4
 5
 6
 7
 8
 9
 10 002140

.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

EQUALS

; BIT DIFINITIONS

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

001000	BIT9==	BIT09
000400	BIT8==	BIT08
000200	BIT7==	BIT07
000100	BIT6==	BIT06
000040	BIT5==	BIT05
000020	BIT4==	BIT04
000010	BIT3==	BIT03
000004	BIT2==	BIT02
000002	BIT1==	BIT01
000001	BIT0==	BIT00

; EVENT FLAG DEFINITIONS
 ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

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

; PRIORITY LEVEL DEFINITIONS

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

```

000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
;
;OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000
;
11          000015      CR=      15
12

```

;VALUE TO PASS TO PRINT MACRO TO END LINE

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 31
 GLOBAL EQUATES SECTION

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

;MACRO DEFINITIONS FOR GLOBAL EQUATES

;THESE MACROS ARE USED TO DEFINE INDEXES INTO A TABLE

;CALLING SEQUENCE MUST BE

```

;      TABLE
;      ITEM   NAME   BYTES
;      ITEM   NAME   BYTES
;      ITEM   NAME   BYTES
;      END    SIZE

```

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

```

.MACRO TABLE
      TINDEX=0

```

```

.ENDM

```

```

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

```

```

.ENDM

```

```

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

```

```

.ENDM

```

```

1      ;UDA BIT DEFINITIONS
2
3      ;UDASA REGISTER UNIVERSAL READ BITS
4
5      004000      SA.S1= 004000      ;STEP 1 STATUS BIT
6      010000      SA.S2= 010000      ;STEP 2 STATUS BIT
7      020000      SA.S3= 020000      ;STEP 3 STATUS BIT
8      040000      SA.S4= 040000      ;STEP 4 STATUS BIT
9      100000      SA.ERR= 100000     ;ERROR INDICATOR
10     001000      SA.QB= 1000        ;QB BIT MASK
11     000100      SA.MP= 100         ;MP BIT MASK
12     000040      SA.SM= 40          ;SA BIT MASK
13
14     ;UDASA REGISTER ERROR STATUS BITS
15
16     003777      SA.ERC= 003777     ;ERROR CODE
17
18     ;UDASA REGISTER STEP ONE READ BITS
19
20     002000      SA.NV= 002000     ;NON SETTABLE INTERRUPT VECTOR
21     001000      SA.A2= 001000     ;22 BIT ADDRESS BUS
22     000400      SA.GI= 000400     ;ENHANCED DIAGNOSTICS
23     ;           ;           000377 ;ALL BITS RESERVED
24
25     ;UDASA REGISTER STEP ONE WRITE BITS
26
27     000177      SA.VEC= 000177     ;INTERRUPT VECTOR (DIVIDED BY 4)
28     000200      SA.INT= 000200     ;INTERRUPT ENABLE DURING INITIALIZATION
29     003400      SA.MSG= 003400     ;MESSAGE RING LENGTH
30     034000      SA.CMD= 034000     ;COMMAND RING LENGTH
31     040000      SA.WRP= 040000     ;WRAP BIT
32     100000      SA.STP= 100000     ;STEP - MUST ALWAYS BE WRITTEN A ONE
33
34     000400      SA.MS1= 000400     ;LSB OF MESSAGE RING LENGTH
35     004000      SA.CM1= 004000     ;LSB OF COMMAND RING LENGTH
36
37     ;UDASA REGISTER STEP TWO READ BITS
38
39     000007      SA.MSE= 000007     ;MESSAGE RING LENGTH ECHO
40     000070      SA.CME= 000070     ;COMMAND RING LENGTH ECHO
41     ;           ;           000100 ;RESERVED
42     000200      SA.STE= 000200     ;STEP ECHO
43     003400      SA.CTP= 003400     ;CONTROLLER TYPE
44
45     ;UDASA REGISTER STEP TWO WRITE BITS
46
47     000001      SA.PRG= 000001     ;ENABLE VAX UNIBUS ADAPTER PURGE INTERRUPT
48     ;           ;           177776 ;LOW ORDER MESSAGE RING BYTE ADDRESS

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 33
GLOBAL EQUATES SECTION

```

1      ;UDASA REGISTER STEP THREE READ BITS
2
3      000177      SA.VCE= 000177      ;INTERRUPT VECTOR ECHO
4      000200      SA.INE= 000200      ;INTERRUPT ENABLE ECHO
5      000400      SA.NVE= 000400      ;VECTOR NOT PROGRAMMABLE
6      ;          003000      ;RESERVED
7
8      ;UDASA REGISTER STEP THREE WRITE BITS
9
10     ;          077777      ;HIGH ORDER MESSAGE RING BYTE ADDRESS
11     100000      SA.TST= 100000      ;PURGE POLE TEST ENABLE
12
13     ;UDASA REGISTER STEP FOUR READ BITS
14
15     000017      SA.MCV= 000017      ;UDA MICROCODE VERSION
16     003760      SA.CNT= 003760      ;CONTROLLER MODEL
17
18     ;UDASA REGISTER STEP FOUR WRITE BITS
19
20     000001      SA.GO= 000001      ;GO BIT TO START UDA FIRMWARE
21     000002      SA.LFC= 000002      ;LAST FAILURE CODE REQUEST
22     000374      SA.BST= 000374      ;BURST LEVEL
23
24     ;INIT ROUTINE FLAGS
25
26     000002      ICONT == BIT1      ;CONTINUE EVENT FLAG
27     000004      IREST == BIT2      ;RESTART FLAG
28     000010      ISTRT == BIT3      ;START FLAG
29     000020      ISTRTH == BIT4      ;START FLAG HOLD FOR DMRQA ROUTINE

```

CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 34
 GLOBAL EQUATES SECTION

```

1      ;COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS
2
3      100000      RG.OWN= 100000      ;SET WHEN UDA OWNS RING
4      040000      RG.FLG= 040000      ;FLAG BIT
5
6      ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
7      ;AND TWO PACKET AND BUFFER AREAS.
8
9      000004      HC.ISZ= 4.          ;SIZE OF INTERRUPT INDICATOR WORDS
10     000004      HC.RSZ= 4.          ;SIZE OF RING IN BYTES
11     000004      HC.ESZ= 4.          ;SIZE OF ENVELOPE WORDS BEFORE PACKET
12     000060      HC.PSZ= 48.         ;SIZE OF COMMAND AND MESSAGE PACKETS
13     000244      HC.BSZ= 164.        ;SIZE OF BUFFER
14
15     000000      HC.INT= 0.          ;INTERRUPT INDICATOR WORDS START
16     000004      HC.MSG= HC.INT+HC.ISZ ;MESSAGE RING START
17     000006      HC.MCT= HC.MSG+2.   ;MESSAGE RING CONTROL WORD
18     000010      HC.CMD= HC.MSG+HC.RSZ ;COMMAND RING START
19     000012      HC.CCT= HC.CMD+2.   ;COMMAND RING CONTROL WORDS
20     000014      HC.MEV= HC.CMD+HC.RSZ ;MESSAGE ENVELOPE START
21     000020      HC.MPK= HC.MEV+HC.ESZ ;MESSAGE PACKET START
22     000100      HC.CEV= HC.MPK+HC.PSZ ;COMMAND ENVELOPE START
23     000104      HC.CPK= HC.CEV+HC.ESZ ;COMMAND PACKET START
24     000164      HC.BF1= HC.CPK+HC.PSZ ;FIRST BUFFER
25     000430      HC.BF2= HC.BF1+HC.BSZ ;SECOND BUFFER
26
27     000674      HC.SIZ= HC.BF2+HC.BSZ ;TOTAL SIZE OF HOST COMM AREA
28
29     ;VIRTUAL CIRCUIT IDENTIFIERS
30
31     000000      MSCP= 0              ;MSCP CIRCUIT
32     000001      LOG= 1              ;LOG CIRCUIT
33     177777      DIAG= -1           ;DIAGNOSTIC CIRCUIT
34     001000      DUP= 1000          ;DIAGNOSTIC AND UTILITIES PROTOCOL

```

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

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

CZUDKO UDA50A/KDA50 Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 36
GLOBAL EQUATES SECTION

```

1      ;COMMAND PACKET OPCODES
2
3      000001      OP.ABO= 1      ;ABORT COMMAND
4      000020      OP.ACC= 20     ;ACCESS COMMAND
5      000010      OP.AVL= 10     ;AVAILABLE COMMAND
6      000021      OP.CCD= 21     ;COMPARE CONTROLLER DATA COMMAND
7      000040      OP.CMP= 40     ;COMPARE HOST DATA COMMAND
8      000022      OP.ERS= 22     ;ERASE COMMAND
9      000023      OP.FLU= 23     ;FLUSH COMMAND
10     000002      OP.GCS= 2      ;GET COMMAND STATUS COMMAND
11     000003      OP.GUS= 3      ;GET UNIT STATUS COMMAND
12     000011      OP.ONL= 11     ;ONLINE COMMAND
13     000041      OP.RD= 41      ;READ COMMAND
14     000024      OP.RPL= 24     ;REPLACE COMMAND
15     000004      OP.SCC= 4      ;SET CONTROLLER CHARACTERISTICS COMMAND
16     000012      OP.SUC= 12     ;SET UNIT CHARACTERISTICS COMMAND
17     000042      OP.WR= 42      ;WRITE COMMAND
18     000030      OP.MRD= 30     ;MAINTENANCE READ COMMAND
19     000031      OP.MWR= 31     ;MAINTENANCE WRITE COMMAND
20     000200      OP.END= 200    ;END PACKET FLAG
21     000007      OP.SEX= 7      ;SERIOUS EXCEPTION END PACKET
22     000100      OP.AVA= 100    ;AVAILABLE ATTENTION MESSAGE
23     000101      OP.DUP= 101    ;DUPLICATE UNIT NUMBER ATTENTION MESSAGE
24     000102      OP.SHC= 102    ;SHADOW COPY COMPLETE ATTENTION MESSAGE
25     000103      OP.RLC= 103    ;RESET COMMAND LIMIT ATTENTION MESSAGE
26
27     000001      OP.GDS= 1      ;DUP GET DUST STATUS
28     000001      OP.GSS= 1      ;DUP GET DUST STATUS
29     000002      OP.ESP= 2      ;DUP EXECUTE SUPPLIED PROGRAM
30     000003      OP.ELP= 3      ;DUP EXECUTE LOCAL PROGRAM
31     000004      OP.SSD= 4      ;DUP SEND STUD DATA
32     000005      OP.RSD= 5      ;DUP RECEIVE STUD DATA
33
34     ;NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
35     ;PACKET FLAG TO THE COMMAND OPCODE. FOR EXAMPLE, A READ COMMAND'S END PACKET
36     ;CONTAINS THE VALUE OP.RD+OP.END IN ITS OPCODE FIELD. THE INVALID COMMAND END
37     ;PACKET CONTAINS JUST THE END PACKET FLAG (I.E., OP.END) IN ITS OPCODE FIELD.
38     ;THE SERIOUS EXCEPTION END PACKET CONTAINS THE SUM OF THE END PACKET FLAG
39     ;PLUS THE SERIOUS EXCEPTION OPCODE SHOWN ABOVE (I.E., OP.SEX+OP.END) IN ITS
40     ;OPCODE FIELD.
41
42     ;COMMAND OPCODE BITS 3 THROUGH 5 INDICATE THE COMMAND CLASS, WHICH IS ENCODED
43     ;AS FOLLOWS:
44     ; 000 IMMEDIATE COMMANDS
45     ; 001 SEQUENTIAL COMMANDS
46     ; 010 NON-SEQUENTIAL COMMANDS THAT DO NOT INCLUDE A BUFFER DESCRIPTOR
47     ; 100 NON-SEQUENTIAL COMMANDS THAT DO INCLUDE A BUFFER DESCRIPTOR

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 37
GLOBAL EQUATES SECTION

```

1          ;COMMAND MODIFIERS
2
3          ;      = 020000
4          MD.CMP= 040000      ;CLEAR SERIOUS EXCEPTION
5          MD.EXP= 100000      ;COMPARE
6          MD.ERR= 010000      ;EXPRESS REQUEST
7          MD.SCH= 004000      ;FORCE ERROR
8          MD.SCL= 002000      ;SUPPRESS CACHING (HIGH SPEED)
9          MD.SEC= 000100      ;SUPPRESS CACHING (LOW SPEED)
10         MD.SER= 000400      ;SUPPRESS ERROR CORRECTION
11         MD.SSH= 000200      ;SUPPRESS ERROR RECOVERY
12         MD.WBN= 000100      ;SUPPRESS SHADOWING
13         MD.WBV= 000400      ;WRITE-BACK (NON-VOLATILE)
14         MD.SEQ= 000020      ;WRITE BACK (VOLATILE)
15         MD.SPD= 000001      ;WRITE SHADOW SET ONE UNIT AT A TIME
16         MD.FEU= 000001      ;SPIN-DOWN
17         MD.VOL= 000002      ;FLUSH ENTIRE UNIT
18         MD.NXU= 000001      ;VOLATILE ONLY
19         MD.RIP= 000001      ;NEXT UNIT
20         MD.IMF= 000002      ;ALLOW SELF DESTRUCTION
21         MD.SMP= 000004      ;IGNORE MEDIA FORMAT ERROR
22         MD.CMB= 000010      ;SET WRITE PROTECT
23         MD.PRI= 000001      ;CLEAR WRITE-BACK DATA LOST
24                                     ;PRIMARY REPLACEMENT BLOCK
25         ;END PACKET FLAGS
26
27         EF.BBR= 000200      ;BAD BLOCK REPORTED
28         EF.BBU= 000100      ;BAD BLOCK UNREPORTED
29         EF.LOG= 000040      ;ERROR LOG GENERATED
30         EF.SEX= 000020      ;SERIOUS EXCEPTION
31
32         ;CONTROLLER FLAGS
33
34         CF.ATN= 000200      ;ENABLE ATTENTION MESSAGES
35         CF.MSC= 000100      ;ENABLE MISCELLANEOUS ERROR LOG MESSAGES
36         CF.OTH= 000040      ;ENABLE OTHER HOST'S ERROR LOG MESSAGES
37         CF.THS= 000020      ;ENABLE THIS HOST'S ERROR LOG MESSAGES
38         CF.SHD= 000002      ;SHADOWING
39         CF.576= 000001      ;576 BYTE SECTORS

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 38
GLOBAL EQUATES SECTION

```

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

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 39
GLOBAL EQUATES SECTION

1			;END PACKET OFFSETS	
2				
3				GENERIC END PACKET OFFSETS:
4	000000	P.CRF= 0.		;COMMAND REFERENCE NUMBER
5	000004	P.UNIT= 4.		;UNIT NUMBER
6	000010	P.OPCD= 8.		;OPCODE (ALSO CALLED ENDCODE)
7	000011	P.FLGS= 9.		;END PACKET FLAGS
8	000012	P.STS= 10.		;STATUS
9	000014	P.BCNT= 12.		;BYTE COUNT
10	000034	P.FBBK= 28.		;FIRST BAD BLOCK
11				
12				GET COMMAND STATUS END PACKET OFFSETS:
13	000014	P.OTRF= 12.		;OUTSTANDING REFERENCE NUMBER
14	000020	P.CMST= 16.		;COMMAND STATUS
15				
16				GET UNIT STATUS END PACKET OFFSETS:
17	000014	P.MLUN= 12.		;MULTI-UNIT CODE
18	000016	P.UNFL= 14.		;UNIT FLAGS
19	000020	P.HSTI= 16.		;HOST IDENTIFIER
20	000024	P.UNTI= 20.		;UNIT IDENTIFIER
21	000034	P.MEDI= 28.		;MEDIA TYPE IDENTIFIER
22	000040	P.SHUN= 32.		;SHADOW UNIT
23	000042	P.SHST= 34.		;SHADOW STATUS
24	000044	P.TRCK= 36.		;TRACK SIZE
25	000046	P.GRP= 38.		;GROUP SIZE
26	000050	P.CYL= 40.		;CYLINDER SIZE
27	000054	P.RCTS= 44.		;RCT TABLE SIZE
28	000056	P.RBNS= 46.		;RBNS / TRACK
29	000057	P.RCTC= 47.		;RCT COPIES
30				
31				ONLINE AND SET UNIT CHARACTERISTICS END PACKET AND AVAILABLE
32				ATTENTION MESSAGE OFFSETS:
33	000014	P.MLUN= 12.		;MULTI-UNIT CODE
34	000016	P.UNFL= 14.		;UNIT FLAGS
35	000020	P.HSTI= 16.		;HOST IDENTIFIER
36	000024	P.UNTI= 20.		;UNIT IDENTIFIER
37	000034	P.MEDI= 28.		;MEDIA TYPE IDENTIFIER
38	000040	P.SHUN= 32.		;SHADOW UNIT
39	000042	P.SHST= 34.		;SHADOW STATUS
40	000044	P.UNCL= 36.		;UNIT COMMAND LIMIT
41	000050	P.UNSZ= 40.		;UNIT SIZE
42	000054	P.VSER= 44.		;VOLUME SERIAL NUMBER
43				
44				SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS:
45	000014	P.VRSN= 12.		;MSCP VERSION
46	000016	P.CNTF= 14.		;CONTROLLER FLAGS
47	000020	P.CTMO= 16.		;CONTROLLER TIMEOUT
48	000022	P.CNCL= 18.		;CONTROLLER COMMAND LIMIT
49	000024	P.CNTI= 20.		;CONTROLLER ID
50				
51				GET DUST STATUS END PACKET OFFSETS:
52	000014	P.DEXT= 12.		;DUST PROGRAM EXTENSION
53	000017	P.DFLG= 15.		;STATUS FLAGS
54	000020	P.DPI= 16.		;PROGRESS INDICATOR
55	000024	P.DTO= 20.		;TIMEOUT VALUE

```

1          ;STATUS AND EVENT CODE DEFINITIONS
2
3          000037      ST.MSK= 37          ;STATUS / EVENT CODE MASK
4          000040      ST.SUB= 40          ;SUB-CODE MULTIPLIER
5          000000      ST.SUC= 0           ;SUCCESS
6          000001      ST.CMD= 1           ;INVALID COMMAND
7          000002      ST.ABO= 2           ;COMMAND ABORTED
8          000003      ST.OFL= 3           ;UNIT-OFFLINE
9          000004      ST.AVL= 4           ;UNIT-AVAILABLE
10         000005      ST.MFE= 5           ;MEDIA FORMAT ERROR
11         000006      ST.WPR= 6           ;WRITE PROTECTED
12         000007      ST.CMP= 7           ;COMPARE ERROR
13         000010      ST.DAT= 10          ;DATA ERROR
14         000011      ST.HST= 11          ;HOST BUFFER ACCESS ERROR
15         000012      ST.CNT= 12          ;CONTROLLER ERROR
16         000013      ST.DRV= 13          ;DRIVE ERROR
17         000037      ST.DIA= 37          ;MESSAGE FROM AN INTERNAL DIAGNOSTIC
18
19         ;GET DUST STATUS FLAGS
20
21         000010      DF.ACT= 010         ;SET IF THIS DUST CURRENTLY ACTIVE
22         000004      DF.NES= 004         ;SET IF THIS DUST WILL NOT ACCEPT THE EXECUTE
23                                         ;SUPPLIED PROGRAM COMMAND
24         000002      DF.LCL= 002         ;SET IF THIS DUST HAS A LOCAL LOAD MEDIA FOR LOADING
25                                         ;DIAGNOSTICS AND OTHER UTILITIES
26         000001      DF.GA= 001         ;SET IF ANY PROGRAM EXECUTION UNDER THIS DUST
27                                         ;DISABLES THE OPERATION OF ALL OTHER SERVERS IN THE
28                                         ;SAME SYSTEM AS THE DUST
29
30         ;DUP MESSAGE TYPES
31
32         010000      DU.QUE = 10000      ;QUESTION
33         020000      DU.DFL = 20000      ;DEFAULT QUESTION
34         030000      DU.INF = 30000      ;INFORMATION
35         040000      DU.TER = 40000      ;TERMINATOR
36         050000      DU.FTL = 50000      ;FATAL ERROR
37         060000      DU.SPC = 60000      ;SPECIAL
38
39         170000      DU.TYP= 170000      ;MESSAGE TYPE FIELD
40
41         ;DM PROGRAM HEADER DEFINITIONS
42
43         000000      DMTRLN= 0            ;OFFSET TO SIZE OF PROGRAM NEEDING DOWNLINE LOAD
44         000004      DMOVRL= 4            ;OFFSET TO SIZE OF OVERLAY
45         000021      DMTMO= 21            ;TIMEOUT VALUE IN SECONDS (ONE BYTE)
46         000040      DMMAIN= 40           ;OFFSET TO FIRST WORD OF MAIN PROGRAM
47         001000      DMFRST= 1000         ;ADDRESS IN DM FILE CONTAINING FIRST BYTE OF HEADER

```


CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 42
 GLOBAL EQUATES SECTION

```

1      ;DRIVE TABLE DEFINITIONS
2      ;
3      ;ONE DRIVE TABLE WILL BE SET UP BY THE INITIALIZE SECTION FOR EACH
4      ;DRIVE SELECTED FOR TESTING.  EACH TABLE IS POINTED TO BY A
5      ;WORD IN THE CONTROLLER TABLE ON WHICH THE DRIVE EXISTS.
6
7 002140      TABLE      ;START A TABLE DEFINITION
8
9 002140      ITEM D.DRV      2      ;DRIVE NUMBER
10 002140     ITEM D.UNIT     2
11           DT.UNT= 000077      ; LOGICAL UNIT NUMBER OF DRIVE
12           DT.AVL= BIT15      ; SET WHEN NOT AVAILABLE FOR TESTING
13 002140     ITEM D.SERN     22.   ;DISK SERIAL NUMBER
14
15 002140     END D.SIZE      ;SIZE OF DRIVE TABLE IN BYTES

```

CZUDKO UDA50A/KDA50 Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 43
GLOBAL EQUATES SECTION

```

1      ;USEFUL INSTRUCTION DEFINITIONS
2
3      .MACRO AND ARG,ADR          ;LOGICAL AND INSTRUCTION
4      .LIST
5
6      .NLIST                      BIC #+C<ARG>,ADR
7
8      .ENDM
9
10     .MACRO OR ARG,ADR          ;LOGICAL OR INSTRUCTION
11     .LIST
12
13     .NLIST                      BIS #ARG,ADR
14
15     .ENDM
16
17     .MACRO PUSH ARG           ;PUSH INSTRUCTION
18     .IRP X,<ARG>
19     .LIST
20
21     .NLIST                      MOV X,-(SP)
22     .ENDM
23
24     .MACRO POP ARG            ;POP INSTRUCTION
25     .IRP X,<ARG>
26     .LIST
27
28     .NLIST                      MOV (SP),.X
29     .ENDM
30
31     .MACRO .BR ADR            ;A BRANCH TO THE NEXT LOCATION
32     .IF P2
33         .IF NE .-ADR
34             .ERROR ;ILLEGAL .BR TO ADR
35         .ENDC
36     .ENDC
37
38     .ENDM
39
40     .MACRO ASSUME FIRST CONDITION SECOND
41         .IF CONDITION <FIRST>--<SECOND>
42         .IFF
43             .ERROR ;BAD ASSUME OF <FIRST> CONDITION <SECOND>
44         .ENDC

```


CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 45
GLOBAL EQUATES SECTION

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

```
;PRIN: 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
```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 46
GLOBAL DATA SECTION

```

1          .SBTTL GLOBAL DATA SECTION
2
3          ;**
4          ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
5          ; IN MORE THAN ONE TEST.
6          ;--
7
8 002140    FFREE:: .BLKW 1          ;FIRST FREE WORD IN MEMORY
9 002142    FSIZE:: .BLKW 1         ;SIZE OF FREE MEMORY IN WORDS
10 002144   FMEM:  .BLKW 1         ;COPY OF FFREE AT END OF INIT SECTION
11 002146   FMEMS: .BLKW 1         ;COPY OF FSIZE AT END OF INIT SECTION
12 002150   CTABS:: .BLKW 1        ;START OF CONTROLLER TABLE STORAGE
13 002152   CTRLRS: .BLKW 1        ;COUNT OF UDA CONTROLLERS IN PYABLE
14 002154   TSTTAB: .BLKW 1        ;POINTER TO FIRST CONTROLLER TABLE UNDER TEST
15          .GLOBL RAFMT
16 002156   000000G  DMPROG: .WORD RAFMT ;START ADDRESS OF DM PROGRAM
17 002160   URUN:  .BLKW 1         ;NUMBER OF UNITS TO RUN AT ONE TIME
18 002162   URNING: .BLKW 1        ;NUMBER OF UNITS STILL RUNNING
19 002164   UCNT:  .BLKW 1         ;COUNTER OF UNITS UNDER TEST
20 002166   000000   FILOPN: .WORD 0 ; FILE OPEN
21 002170   UFREEZ: .BLKW 1        ;FREEZE ON UNIT WHEN NOT ZERO
22 002172   NXMAD:  .BLKW 1        ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
23 002174   000000   FDATA: .WORD 0
24 002176   FCTBUF: .BLKB 512.     ;STORAGE FOR FCT BLOCK
25 003176   FCTNUM: .BLKW 1        ;FCT BLOCK NUMBER
26 003200   MODE:  .BLKW 1 ;MODE WORD, SAME BIT DEFS AS SO.BIT
27
28          ;INIT ROUTINE DATA
29
30 003202   DTABS:: .BLKW 1         ;START OF DRIVE TABLE STORAGE
31 003204   IFLAGS::.BLKW 1       ;FLAGS FROM INIT CODE
32
33          ;CLOCK CONTROL
34
35 003206   000000   KW.CSR: .WORD 0 ;CSR OF CLOCK
36 003210   KW.BRL:  .BLKW 1       ;BR LEVEL
37 003212   KW.VEC:  .BLKW 1       ;VECTOR
38 003214   KW.HZ:  .BLKW 1       ;HERTZ (50. OR 60.)
39 003216   KW.EL:  .BLKW 2       ;ELAPSED TIME
40
41 003222   016270   PTYPE: .WORD PF ;PRINT TYPE
42 003224   000     ERRCHR: .BYTE 0,0 ;FIRST BYTE LOADED WITH OUTPUT CHARACTER
43 003226   000000   NULL:  .WORD 0 ;USED TO PRINT A NULL CHARACTER
44 003230   FNAME:  .BLKB 10.

```

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 47
GLOBAL DATA SECTION

1	003242				TEMP:	.BLKB 22.		USED TO GET ANSWER FROM GMANID CALL
2	003270	061	055	112	DATEI:	.ASCIZ\1-JAN-70\		DEFAULT DATE
3	003301					.BLKB 3		
4	003304	000000			DATEO:	.WORD 0 ,DATE STRING IN FORMATTER FORMAT		
5	003306					.BLKB 10.		(FIRST WORD ZERO SAYS NO DATE HERE YET)
6	003320	061	070	064	HIGHEST:	.ASCIZ\18446744073709551615\		HIGHEST DISK SERIAL NUMBER
7	003345	104	105	103	MONTHS:	.ASCII\DEC\		NAME OF MONTHS
8	003350	116	117	126		.ASCII\NOV\		
9	003353	117	103	124		.ASCII\OCT\		
10	003356	123	105	120		.ASCII\SEP\		
11	003361	101	125	107		.ASCII\AUG\		
12	003364	112	125	114		.ASCII\JUL\		
13	003367	112	125	116		.ASCII\JUN\		
14	003372	115	101	131		.ASCII\MAY\		
15	003375	101	120	122		.ASCII\APR\		
16	003400	115	101	122		.ASCII\MAR\		
17	003403	106	105	102		.ASCII\FEB\		
18	003406	112	101	116		.ASCII\JAN\		
19	003411	037			DAYS:	.BYTE 31.		NUMBER OF DAYS IN EACH MONTH
20	003412	035				.BYTE 29.		
21	003413	037				.BYTE 31.		
22	003414	036				.BYTE 30.		
23	003415	037				.BYTE 31.		
24	003416	036				.BYTE 30.		
25	003417	037				.BYTE 31.		
26	003420	037				.BYTE 31.		
27	003421	036				.BYTE 30.		
28	003422	037				.BYTE 31.		
29	003423	036				.BYTE 30.		
30	003424	037				.BYTE 31.		
31	003425	061	071	000	YEAR19:	.ASCIZ\19\		
32	003430	062	060	000	YEAR20:	.ASCIZ\20\		
33						.EVEN		
34	003434	000000			IPADRS:	.WORD 0		
35	003436	000000				.WORD 0		
36	003440	000000				.WORD 0		
37	003442	000000				.WORD 0		
38	003444	000000				.WORD 0		
39	003446	000000				.WORD 0		
40	003450	000000				.WORD 0		
41	003452	000000				.WORD 0		

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

.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

DEV TYP <RA SERIES DISK DRIVE>

003454
003454
003454

122 101 040

L#DVTYP::
.ASCIZ /RA SERIES DISK DRIV
.EVEN

E/

13
14
15
16

; TEST DESCRIPTION

DESCRIPT <CZUDKO UDASOA,KDASO-Q FORMATTER>

003502
003502
003502

103 132 125

L#DESC::
.ASCIZ /CZUDKO UDASOA,KDASO
.EVEN

A Q FORMATTER/

05

UNFORMATTED MESSAGES

1				
2				
3	003544	105	116	124 DATEQ: .ASCIZ\ENTER DATE AS DD-MMM-YY\
4	003574	040	106	117 FILMAQ: .ASCIZ\ FOR DISK TO BE FORMATTED\
5	003626	040	000	SERNQ: .ASCIZ\ \
6	003630	101	122	105 UNQUES: .ASCIZ\ARE YOU SURE YOU WANT TO RUN THIS FORMATTER\

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 50
GLOBAL TEXT SECTION

```

1
2
3 003704 045 124 000 ERRONE: .ASCIZ\#T\
4 003707 045 116 000 ERRNL: .ASCIZ\#N\
5 003712 042 040 040 RNTIM: .ASCIZ\" RUNTIME "D16": "\
6 003735 104 071 042 RNTIM1: .ASCIZ\D9": "\
7 003743 104 071 000 RNTIM2: .ASCIZ\D9\
8 003746 042 040 040 ERRME1: .ASCIZ\" * * * ERROR PROCESSING MESSAGE STRING * * * "N\
9 004035 116 042 125 MESSG: .ASCIZ\N"UNIT "D6" CONTROLLER AT "016" DRIVE "D9S\
10 004110 042 116 117 NOCLOCK: .ASCIZ\N"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\
11 004165 042 110 117 BASNO: .ASCIZ\N"HOST PROGRAM"\
12 004204 042 040 040 BASL2: .ASCIZ\" CONTROLLER AT "016\
13 004232 042 040 040 BASL3: .ASCIZ\" DRIVE "D9\
14 004247 000 BAS: .BYTE 0 ;NULL TO PRINT NOTHING
15
16 004250 122 066 122 BASLN: .ASCIZ\R6R6R6R6\ ;USED TO PRINT BASIC LINE OF ERROR MESSAGE
17 004261 116 042 123 SERNUM: .ASCIZ\N"SERIAL NUMBER FOR UNIT "D6" CONTROLLER AT "016" DRIVE "D9\
18 004355 042 123 124 WNSTOP: .ASCII\N"STOPPING THIS FORMAT AFTER THIS POINT WILL MAKE THE DISK"N\
19 004450 042 125 116 .ASCII\N"UNUSABLE, AND WILL CAUSE THE DISK TO BE SPUN DOWN WHEN"N\
20 4541 042 102 122 .ASCIZ\N"BROUGHT ONLINE."N\
21 004565 116 042 127 WNSTRT: .ASCII\N"WARNING:"N\
22 004601 042 040 040 .ASCII\" THIS FORMATTER PROGRAM SHOULD NOT BE USED AS A DIAGNOSTIC"N\
23 004703 042 040 040 .ASCII\" TOOL. RUN THIS PROGRAM ONLY AS INSTRUCTED IN THE DISK"N\
24 005002 042 040 040 .ASCIZ\" DRIVE'S SERVICE MANUAL."N\

```

1	005043			X1A:	
2	005043			X2A:	
3	005043			X3A:	
4	005043	042	111	116	X8A: .ASCIZ\ "INVALID ANSWERS GIVEN TO HARDWARE QUESTIONS" N\
5	005122	122	065	122	X1: .ASCIZ\RSR6 "CONTROLLER HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE" N\
6	005225	122	065	122	X2: .ASCIZ\RSR6 "MULTIPLE UNITS SELECT THE SAME DRIVE" N\
7	005301	122	065	122	X3: .ASCIZ\RSR6 "MORE THAN EIGHT DRIVES SELECTED ON THIS CONTROLLER" N\
8	005373	122	064	042	X4: .ASCII\R4 "NOT ENOUGH ROOM IN MEMORY TO FORMAT THE UNITS SELECTED" N\
9	005466	042	120	114	.ASCIZ\ "PLEASE START PROGRAM OVER AND FORMAT FEWER UNITS AT A TIME" N\
10	005564	122	065	122	X8: .ASCIZ\RSR6 "TWO CONTROLLERS USE THE SAME VECTOR" N\
11	005637	122	064	042	X9: .ASCII\R4 "ONLY ONE DISK CAN BE SELECTED IN HW QUESTIONS IN RESTORE MODE." N\
12	005742	042	120	114	.ASCIZ\ "PLEASE START PROGRAM OVER AND SELECT ONLY ONE DISK." N\
13	006031	122	064	042	X10: .ASCIZ\R4 "THIS PROGRAM CAN ONLY REFORMAT A DISK IN UNATTENDED MODE." N\
14	006130	122	065	042	X14: .ASCII\RS "CONTROLLER IS NOT SUPPORTED BY THIS FORMATTER PROGRAM. THIS" N\
15	006231	042	120	122	.ASCII\ "PROGRAM REQUIRES A UDA50-A (MODEL 6) OR A KDA50-Q (MODEL 13)" N\
16	006330	042	103	117	.ASCIZ\ "CONTROLLER. CONTROLLER REPORTED MODEL CODE "D4"." N\
17	006415	122	065	042	X20: .ASCII\RS "MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS" N\
18	006502	042	103	110	.ASCII\ "CHECK CSR SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE OR BUS" N\
19	006607	042	117	122	.ASCIZ\ "OR REPLACE CONTROLLER PROCESSOR MODULE" N\
20	006661	122	065	042	X21: .ASCII\RS "CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE" NR8\
21	006750	042	122	105	.ASCIZ\ "REPLACE CONTROLLER SOI MODULE" N\
22	007011	122	065	042	X21A: .ASCIZ\RS "CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE" NR8R7\
23	007103	122	065	042	X22: .ASCII\RS "STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION" N\
24	007201	042	123	124	.ASCIZ\ "STEP BIT EXPECTED "016NR8R7\
25	007236	122	065	042	X23A: .ASCII\RS "CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATI ON"
26	007357	104	071	042	.ASCII\D9 " WORDS WERE TO BE CLEARED STARTING AT ADDRESS "016N\
27	007445	042	106	111	.ASCII\ "FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):" N\
28	007522	123	066	042	.ASCIZ\S6 "ADDRESS" S4 "CONTENTS" N\
29	007553	123	067	117	X23B: .ASCIZ\S7016SS016N\
30	007567	122	065	042	X24: .ASCII\RS "SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION" N\
31	007677	042	120	125	.ASCIZ\ "PURGE/POLE DIAGNOSTICS WERE REQUESTED" NR8R7\
32	007754	122	065	042	X25: .ASCII\RS "CONTROLLER DID NOT RETURN CORRECT DATA IN SA REGISTER DURING" N\
33	010055	042	111	116	.ASCII\ "INITIALIZATION" N\
34	010076	042	040	040	.ASCIZ\ " SA EXPECTED "016NR8R7\
35	010130	122	065	042	X30: .ASCIZ\RS "CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE RUNNING FORMATTER" NR8
36	010246	122	065	042	X31: .ASCIZ\RS "FORMATTER PROGRAM IS HUNG" N\
37	010305	122	065	042	X32: .ASCIZ\RS "MESSAGE BUFFER RECEIVED FROM FORMATTER WITH UNKNOWN REQUEST NUMBER" N\
38	010415	122	065	042	X36: .ASCII\RS "NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS" N\
39	010506	042	127	110	.ASCIZ\ "WHILE LOADING FORMATTER" N\
40	010541	122	065	042	X37: .ASCIZ\RS "CONTROLLER REPORTED FATAL ERROR IN SA REGISTER WHILE LOADING FORMATTER" NR8
41	010661	122	065	042	X100: .ASCIZ\RS "FORMATTER ASKED UNEXPECTED QUESTION ("D12")" N\
42	010742	122	065	042	X101: .ASCIZ\RS "FORMATTER REJECTED ANSWER TO DATE OR SERIAL NUMBER QUESTION" N\

ON"

R7\

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 52
 GLOBAL TEXT SECTION

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 53
GLOBAL ERROR REPORT SECTION

```

1      .SBTTL GLOBAL ERROR REPORT SECTION
2
3      ;**
4      ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
5      ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PRINTB
6      ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
7      ;--
8      SVCINS= -1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
9      SVCTST= -1     ; LIST TEST TAGS, SHIFTED RIGHT
10     SVCSUB= -1     ; LIST SUBTEST TAGS, SHIFTED RIGHT
11     SVCGBL= -1    ; LIST GLOBAL TAGS, SHIFTED RIGHT
12     SVCTAG= -1    ; LIST OTHER TAGS, SHIFTED RIGHT
13
14     011772      BGNMSG ERRO01
15     011772      PNTB X1,#X1A
16     011772      012746      005043      MOV #X1A,-(SP)
17     011776      004137      016422      JSR R1,LPNTB
18     012002      005122      .WORD X1
19     012004      000002      .WORD PNT.CT
20     012006      ENDMSG
21
22     012010      BGNMSG ERRO02
23     012010      PNTB X2,#X2A
24     012010      012746      005043      MOV #X2A,-(SP)
25     012014      004137      016422      JSR R1,LPNTB
26     012020      005225      .WORD X2
27     012022      000002      .WORD PNT.CT
28     012024      ENDMSG
29
30     012026      BGNMSG ERRO03
31     012026      PNTB X3,#X3A
32     012026      012746      005043      MOV #X3A,-(SP)
33     012032      004137      016422      JSR R1,LPNTB
34     012036      005301      .WORD X3
35     012040      000002      .WORD PNT.CT
36     012042      ENDMSG
37
38     012044      BGNMSG ERRO04
39     012044      PNTB X4
40     012044      004137      016422      JSR R1,LPNTB
41     012050      005373      .WORD X4
42     012052      000000      .WORD PNT.CT
43     012054      ENDMSG
44
45     012056      BGNMSG ERRO08
46     012056      PNTB X8,#X8A
47     012056      012746      005043      MOV #X8A,-(SP)
48     012062      004137      016422      JSR R1,LPNTB
49     012066      005564      .WORD X8
50     012070      000002      .WORD PNT.CT
51     012072      ENDMSG
52
53     012074      BGNMSG ERRO09
54     012074      PNTB X9
55     012074      004137      016422      JSR R1,LPNTB
56     012100      005637      .WORD X9
57     012102      000000      .WORD PNT.CT

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 53-1
GLOBAL ERROR REPORT SECTION

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 53-2
GLOBAL ERROR REPORT SECTION

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 53-3
GLOBAL ERROR REPORT SECTION

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

CZUDKO UDAS0A/KDA50-0 FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 53-4
 GLOBAL ERROR REPORT SECTION

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

CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 54
GLOBAL ERROR REPORT SECTION

1 000001
2 000001
3 000001
4 000001
5 000001

SVCINS= 1
SVCTST= 1
SVCSUB= 1
SVCGBL= 1
SVCTAG= 1

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

1
2
3
4
5
6
7
8

.SBTTL GLOBAL SUBROUTINES SECTION
;MEMORY ALLOCATION ERROR
;THIS ROUTINE PRINTS A SYSTEM FATAL ERROR AND EXITS THE TEST
FMERR: ERRSF 4,,ERRO04

012674
012674 104454
012676 000004
012700 000000
012702 012044
012704
012704 104444

DOCLN

;ABORT

TRAP C#ERSF
.WORD 4
.WORD 0
.WORD ERRO04
TRAP C#DCLN

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01 Oct 84 10:07 Page 56
 GLOBAL SUBROUTINES SECTION

```

1      ;ALOCM
2      ;
3      ;ALLOCATE A BLOCK OF FREE MEMORY.  REPORT ERROR IF MEMORY EXHAUSTED
4      ;
5      ;INPUTS:
6      ;      R1 - NUMBER OF WORDS TO ALLOCATE
7      ;      FFREE - FIRST FREE WORD IN MEMORY
8      ;      FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS
9      ;OUTPUTS:
10     ;      R1 - ADDRESS OF FIRST WORD OF ALLOCATED MEMORY
11     ;      FFREE - NEW FIRST FREE WORD IN MEMORY
12     ;      FSIZE - SIZE OF FREE MEMORY LEFT AFTER ALLOCATION
13     ;SYSTEM FATAL ERROR WILL BE REPORTED IF NOT ENOUGH MEMORY AVAILABLE
14     ;AND ENTIRE PROGRAM WILL BE STOPPED.
15
16     012706      013746      002140      ALOCM:  PUSH FFREE                ;SAVE FFREE AT ENTRY
17     012706      013746      002140                ;REDUCE SIZE OF FREE MEMORY      MOV FFREE, -(SP)
18     012712      160137      002142      SUB R1,FSIZE
19     012716      002766                BLT FMERR                ;REPORT ERROR IF NOT ENOUGH MEMORY
20     012720      060101                ADD R1,R1                ;CHANGE WORDS TO BYTES
21     012722      060137      002140      ADD R1,FFREE            ;CALCULATE NEW START OF FREE MEMORY
22     012726      012601                POP R1                   ;GET START OF ALLOCATED MEMORY
22     012730      000207                RETURN                   MOV (SP),.R1

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

012732 012701 000336
012736 004737 012706
012742 010165 000014
012746 000207

```

;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
;
HCOMM:  MOV @MC.SIZ/2,R1      ;GET SIZE OF AREA TO ALLOCATE
        CALL ALOCM          ;ALLOCATE THE MEMORY
        MOV R1,C.RING(R5)   ;GET ADDRESS OF HOST COMM AREA
                                ;PLACE IN CONTROLLER TABLE
        RETURN

```

CZUDKO UDA50A/KDA50-0 FORMATTER MACRO V05.01b Monday 01-Oct 84 10:07 Page 58
GLOBAL SUBROUTINES SECTION

```

1      ; RESET
2      ; RESET ALL UDA-50S IN THE CONTROLLER TABLES
3
4      ; INPUTS:
5      ; IPADRS - CONTAINS ALL IP ADDRESSES
6      ; OUTPUTS:
7      ; NONE
8
9      RESET:  PUSH <R3,R4>
10     MOV R3, -(SP)
11     MOV R4, (SP)
12
13     CLR     NXMAD
14     SETVEC @4, @NXMI, @PRI07
15
16     MOV     @PRI07, (SP)
17     MOV     @NXMI, -(SP)
18     MOV     @4, -(SP)
19     MOV     @3, -(SP)
20     TRAP   C$SVEC
21     ADD     @10, SP
22
23     BREAK
24     TRAP   C$BRK
25
26     MOV     @8, R3      ; R3 = COUNTER OF ENTRIES
27     MOV     @IPADRS, R4 ; R4 -> IP ADDRESS
28
29 1$:  TST     (R4)      ; IS THERE AN ENTRY?
30     BEQ     2$      ; IF NOT, DONE
31     CLR     @R4+     ; INIT UDA
32     TST     NXMAD    ; WAS THERE AN ERROR?
33     BNE     3$      ; IF SO, EXIT
34     DEC     R3      ; MAKE SURE WE DO NOT EXTEND OVER AREA
35     BNE     1$      ; IF NOT DONE, BRANCH
36
37 2$:  CLRVEC @4
38
39     MOV     @4, R0
40     TRAP   C$CVEC
41
42     POP     <R4,R3>
43
44     MOV (SP), R4
45     MOV (SP), R3
46
47     RETURN
48
49 3$:  TST     -(R4)    ; R4 -> UDAIP THAT FAILED
50     MOV     R4, R5  ; SAVE IN R5 FOR REPORT
51     ERROF  20, ERRO20
52
53     TRAP   C$ERDF
54     .WORD  20
55     .WORD  0
56     .WORD  ERRO20
57
58     CLR     (R4)    ; DESTROY ENTRY SO NOT TO FALL INTO RESET ERROR LOOP
59     DOCLN
60     TRAP   C$DCLN

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 59
GLOBAL SUBROUTINES SECTION

```

1      ;RUNDM
2      ;
3      ;LOAD AND RUN A DM PROGRAM IN THE CONTROLLERS. RETURN WHEN ALL
4      ;DM PROGRAMS HAVE TERMINATED.
5      ;
6      ;INPUTS:
7      ;   TSTTAB - POINTER TO FIRST CONTROLLER TABLE
8      ;   R1 - NUMBER OF CONTROLLERS TO TEST
9      ;IMPLICIT INPUTS:
10     ;   DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
11     ;OUTPUTS:
12     ;   Z SET IF NO CONTROLLERS SUCCESSFULLY STARTED
13     ;ALL REGISTERS ARE USED AND PREVIOUS CONTENTS DESTROYED.
14
15 013074 010137 002160  RUNDM:  MOV R1,URUN          ;SAVE NUMBER OF UNITS TO RUN
16 013100 005037 002162          CLR URNING        ;CLEAR NUMBER OF UNITS RUNNING
17
18     ;LOAD DM PROGRAM INTO EACH CONTROLLER
19
20 013104 013737 002160 002164          MOV URUN,UCNT      ;SET COUNTER OF UNITS
21 013112 013705 002154          MOV TSTTAB,R5     ;GET FIRST CONTROLLER TABLE
22 013116          LDDM:
23 013116 005065 000012          CLR C.FLG(R5)    ;CLEAR ALL FLAGS
24 013122 116537 000002 002074        MOVB C.UNIT(R5),L#LUN ;SEE IF UNIT TO BE TESTED
25 013130 005765 000002          TST C.UNIT(R5)
26 013134 100407          BMI LDNEXT        ;IF NOT, DON'T LOAD THIS UNIT
27 013136          ASSUME CT.AVL EQ BIT15
28 013136 004737 012732          CALL HCOMP      ;ALLOCATE SPACE FOR HOST COMM AREA
29 013142 004737 016554          CALL LOADDM    ;LOAD THE DM PROGRAM
30 013146 001402          BEQ LDNEXT        ;IF ERROR, GO TO NEXT CONTROLLER
31 013150 005237 002162          INC URNING     ;IF NO ERROR, COUNT UNIT RUNNING
32 013154 062705 000052          LDNEXT: ADD #C.SIZE,R5 ;MOVE TO NEXT CONTROLLER TABLE
33 013160 005337 002164          DEC UCNT      ;CHECK IF MORE CONTROLLERS
34 013164 001354          BNE LDDM        ;LOAD NEXT
35 013166 005037 002170          CLR UFREEZ    ;CLEAR UNIT FREEZE FLAG
36 013172 012737 177777 003176        MOV #-1,FCTNUM ;INVALIDATE FCT BLOCK NUMBER (BLOCK IN MEMORY)
37
38     ;CHECK IF ANY CONTROLLERS LOADED
39
40 013200 005737 002162          TST URNING     ;ANY UNITS LOADED?
41
42     ;THE DM PROGRAMS ARE NOW IN CONTROL
43     ;RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
44
45 013204 000207          RETURN

```

```

1      ;CLOSEF
2      ;
3      ;CLOSE DATA FILE FOR DM PROGRAMS
4      ;
5      ;INPUTS:
6      ;      FILOPN - ZERO IF FILE NOT OPEN
7      ;OUTPUTS:
8      ;      NONE
9
10     013206 005737 002166      CLOSEF: TST FILOPN      ;SEE IF FILE CURRENTLY OPEN
11     013212 001403              BEQ 14              ; IF SO, CLOSE IT
12     013214 104435              CLOSE              ;AND MARK AS SO      TRAP      C8CLOS
13     013216 005037 002166
14     013222 000207      14:      CLR FILOPN
                                RETURN

```

CZUDKO UDAS0A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 61
GLOBAL SUBROUTINES SECTION

```

1      ;RESPDM
2
3      ;RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
4      ;HAVE TERMINATED.
5
6 013224 013705 002154      RESPDM: MOV TSTTAB,R5      ;GET CONTROLLER TABLE ADDRESS
7 013230 013737 002160 002164      MOV URUN,UCNT      ;SET COUNTER OF UNITS
8 013236      ;RESPCT: BREAK      ;ALLOW DRS TO SEE TERMINAL INPUT
9      013236 104422      ;GET MOST COMM AREA ADDRESS      TRAP      C:BRK
10 013240 016504 000014      MOV C.RING(R5),R4
11 013244 032765 000002 000012      BIT #CT.RN,C.FLG(R5)      ;CHECK IF PROGRAM RUNNING
12 013254 116537 000002 002074      BEQ RSPNXT      ;IF NOT, LOOK AT NEXT
13 013262 032765 000010 000012      MOVB C.UNIT(R5),L#LJN      ;STORE UNIT NUMBER UNDER TEST
14 013270 001150      BIT #CT.MSG,C.FLG(R5)      ;SEE IF INTERRUPT RECEIVED
15 013272 032765 000004 000012      BNE RSPIN      ;IF SO, LOOK AT PACKET
16 013300 001002      BIT #CT.CMD,C.FLG(R5)      ;SEE IF COMMAND HAS BEEN SENT
17 013302 000137 014050      BNE 1#      ;IF NOT, SEND ONE
18      JMP RSPOUT
19
20      ;CHECK IF UDA STILL RUNNING
21 013306 011503      1#: MOV (R5),R3      ;GET ADDRESS OF UDAIP
22 013310 016301 000002      MOV 2(R3),R1      ;LOOK AT UDASA REGISTER
23 013314 001405      BEQ RSPTM      ;IF ZERO, UDA STILL RUNNING
24 013316      ERRODF 30,ERR030      ;REPORT UDA HAS FATAL ERROR
25 013316 104455      ;DROP CONTROLLER FROM TESTING      TRAP      C:ERDF
26 013320 000036      .WORD      30
27 013322 000000      .WORD      0
28 013324 012346      .WORD      ERR030
29 013326 000465      BR RSPDRP
30
31      ;CHECK FOR TIMEOUT OF RESPONSE
32
33 RSPTM: TST C.TOT(R5)      ;SEE IF DUP PROGRAM TO BE TIMED
34 BEQ RSPNTO
35 TST KW.CSR      ;SEE IF A CLOCK ON SYSTEM
36 BEQ RSPNTO      ;DON'T TIME IF NO CLOCK
37 CMP KW.EL+2,C.TOT(R5)      ;COMPARE TO TIMEOUT COUNTER
38 BHI RSPTMO
39 BNE RSPNTO
40 CMP KW.EL,C.TOT(R5)
41 BLO RSPNTO
42 RSPTMO: BIT #CT.STA,C.FLG(R5)      ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
43 BNE RSPTOE      ;SEE IF A GET DUST STATUS COMMAND OUTSTANDING
44 TST HC.CCT(R4)      ;REPORT ERROR IF SO
45 BMI RSPTOE      ;SEE IF UDA TOOK LAST COMMAND PACKET
46 MOV #CT.TM1,R0      ;REPORT ERROR IF NOT
47 BIT #CT.TM1,C.FLG(R5)      ;SEE IF FIRST TIMEOUT ALREADY HAPPENED
48 BEQ 1#
49 ASL R0
50 BIS #CT.STA,R0      ;IF SO,
51 BIS R0,C.FLG(R5)      ;SET SECOND TIME OUT FLAG
52 MOV #OP.GDS,R0      ;SET THE PROPER TIMEOUT BIT
53 CALL BLDCMD      ; AND STATUS REQUESTED BIT
54 MOV #RG.OWN,HC.CCT(R4)      ;BUILD GET DUST STATUS COMMAND
55 TST B(R5)      ;MARK COMMAND TO UDA
56 JMP RSPOU4      ;TELL UDA COMMAND IS THERE

```

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 61 1
GLOBAL SUBROUTINES SECTION

53 013460

RSPNTD:

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 62
GLOBAL SUBROUTINES SECTION

```

1          ;SWITCH TO NEXT CONTROLLER
2
3 013460 005737 002170      RSPNXT: TST UFREEZ          ;FROZEN TO ONE UNIT?
4 013464 001264              BNE RESPCT          ;STAY THERE IF SO
5 013466 062705 000052      ADD #C.SIZE,R5          ;MOVE TO NEXT TABLE
6 013472 005337 002164      DEC UCNT              ;CHECK IF MORE CONTROLLERS
7 013476 001257              BNE RESPCT          ;LOOK AT NEXT CONTROLLER
8 013500 C00651              BR RESPDM           ;LOOK AT FIRST CONTROLLER AGAIN
9
10         ;REMOVE A CONTROLLER FROM TESTING
11
12 013502 005065 000012      RSPDRP: CLR C.FLG(R5)      ;CLEAR PROGRAM RUNNING
13 013506 005037 002170      CLR UFREEZ
14 013512 010504              MOV R5,R4
15 013514 062704 000016      ADD #C.DRO,R4
16 013520 012702 000010      MOV #8.,R2
17 013524 012403              1$: MOV (R4)+,R3
18 013526 001420              BEQ 3$
19 013530 005763 000002      TST D.UNIT(R3)
20 013534                      ASSUME DT.AVL EQ BIT15
21 013534 100003              BPL 2$
22 013536 005302              DEC R2
23 013540 001371              BNE 1$
24 013542 000412              BR 3$
25 013544 052763 100000 000002 2$: BIS #DT.AVL,D.UNIT(R3)
26 013552 005302              DEC R2
27 013554 001405              BEQ 3$
28 013556 005714              TST (R4)
29 013560 001403              BEQ 3$
30 013562 004737 016554      CALL LOADDM          ;START DM PROGRAM AGAIN
31 013566 001223              BNE RESPCT
32 013570 005337 002162      3$: DEC URNING          ;REDUCE RUNNING CONTROLLERS COUNT
33 013574 001331              BNE RSPNXT          ;IF ANY STILL RUNNING, LOOK AT THEM
34 013576 000207              RETURN              ;ELSE RETURN TO TEST SECTION
35
36 013600                      RSPTOE: ERRDF 31,,ERR031      ;REPORT TIMEOUT ERROR
37 013600 104455                      TRAP          C$ERDF
38 013602 000037                      .WORD          31
39 013604 000000                      .WORD          0
40 013606 012362                      .WORD          ERR031
41 013610 000734                      BR RSPDRP          ;DROP CONTROLLER FROM TESTING

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 63
GLOBAL SUBROUTINES SECTION

```

1          ;CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
2
3          ;CHECK FOR PROPER OPCODE IN END PACKET
4
5 013612 012700 000204          RSPIN:  MOV #OP.END+OP.SSD,R0          ;GET SEND DATA END PACKET OPCODE
6 013616 032765 000020 000012  BIT #CT.REQ.C.FLG(R5)          ;LOOK IF SEND DATA OR RECEIVE DATA
7 013624 001402          BEQ RSPMWR
8 013626 012700 000205          MOV #OP.END+OP.RSD,R0          ;CHANGE TO RECEIVE DATA END PACKET OPCODE
9 013632 120064 000030          RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)      ;COMPARE TO OPCODE IN END PACKET
10 013636 001145          BNE RSPERR
11
12          ;LOOK AT STATUS CODE
13
14 013640 032764 000037 000032          BIT #ST.MSK,HC.MPK+P.STS(R4)      ;CHECK FOR STATUS CODE ST.SUC (ZERO)
15 013646 001004          BNE RSPERW
16
17          ;CHECK FOR EXPECTED REFERENCE NUMBER
18
19 013650 026564 000050 000020          CMP C.REF(R5),HC.MPK+P.CRF(R4)  ;CHECK IF CORRECT REF NUMBER
20 013656 001405          BEQ RSPPTW
21 013660          RSPERW: ERRODF 33,ERR033
22          TRAP          C1ERDF
23          .WORD          33
24          .WORD          0
25          .WORD          ERRO33
26          BR RSPDRP          ;DROP UNIT FROM TESTING
27
28          ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
29
30 013672 032765 000020 000012          RSPPTW: BIT #CT.REQ.C.FLG(R5)    ;CHECK IF RESPONSE FROM DM PROGRAM
31 013700 001463          RSPOU:  BEQ RSPOUT          ;LOOK AT REQUEST NUMBER IF SO

```

```

1          ;MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
2
3 013702 016401 000430 RSPPT2: MOV HC.BF2(R4),R1          ;GET REQUEST NUMBER
4 013706 042701 007777 BIC #C<DU.TYP>,R1          ;CHECK TYPE
5 013712 001403 BEQ 1$          ;IF ZERO, ERROR
6 013714 020127 06000$ CMP R1,#DU.SPC          ;CHECK IF IN EXPECTED RANGE
7 013720 101405 BLOS RSPPT3
8 013722 1$: ERRDF 32,,ERR032          ;BAD REQUEST NUMBER
          TRAP C$ERDF
          .WORD 32
          .WORD 0
          .WORD ERR032
9 013732 000663 BR RSPDRP          ;DROP UNIT FROM TESTING
10
11 013734 016403 000034 RSPPT3: MOV HC.MPK+P.BCNT(R4),R3      ;GET BYTE COUNT OF CHARACTERS RECEIVED IN R3
12 013740 162703 000002 SUB #2,R3          ;(FIRST TWO CHARACTERS ARE TYPE WORD)
13 013744 012700 000004 MOV #OP.SSD,R0      ;BUILD A SEND DATA COMMAND PACKET
14 013750 004737 016754 CALL BLDCHD          ; FOR ANSWER TO DM PROGRAM
15 013754 012700 000164 MOV #HC.BF1,R0      ;POINT TO BUFFER IN PACKET
16 013760 004737 017116 CALL CLRBUF          ; AND CLEAR BUFFER
17 013764 010402 MOV R4,R2          ;R2 POINTS TO SEND BUFFER
18 013766 062704 000244 ADD #HC.BSZ,R4      ;R4 POINTS TO CHARACTERS IN RECEIVE BUFFER
19 013772 042724 170000 BIC #DU.TYP,(R4)   ;CLEAR TYPE FIELD IN BUFFER
20 013776 000301 SWAB R1          ;GET TYPE RIGHT JUSTIFIED
21 014000 006201 ASR R1          ;TIMES TWO
22 014002 006201 ASR R1
23 014004 006201 ASR R1
24 014006 010100 MOV R1,R0          ;COPY MESSAGE TYPE TO R0
25 014010 005001 CLR R1          ;R1 CONTAINS ZERO SEND BYTE COUNT
26 014012 004770 014276 CALL BRSPDSP-2(R0) ;CALL REQUESTED ROUTINE
27 014016 001231 BNE RSPDRP          ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
28          ; Z SET IF UNIT TO CONTINUE RUNNING
29 014020 016504 000014 MOV C.RING(R5),R4   ;GET RING ADDRESS
30 014024 032701 000001 BIT #1,R1          ;LOOK AT CHARACTER COUNT TO SEND TO DUP PROGRAM
31 014030 001401 BEQ 1$          ;IF AN ODD COUNT
32 014032 005201 INC R1          ; INCREASE BY ONE
33 014034 010164 000120 1$: MOV R1,HC.CPK+P.BCNT(R4) ;PUT CHARACTER COUNT IN COMMAND PACKET
34 014040 100003 BPL R$POUT          ;IF NEGATIVE BYTE COUNT RETURNED
35 014042 042765 000020 000012 BIC #CT.REQ,C.FLG(R5) ; DON'T SEND ANY DATA TO UDA
36
37          ;SEND COMMAND BACK TO UDA
38
39 014050 042765 000350 000012 R$POUT: BIC #CT.MSG+CT.STA+CT.TM1+CT.TM2,C.FLG(R5) ;CLEAR MESSAGE RECEIVED FLAG
40 014056 032765 000020 000012 BIT #CT.REQ,C.FLG(R5) ;CHECK WHICH COMMAND TO SEND
41 014064 001014 BNE R$POU2          ;BRANCH IF RESPONSE TO REQUEST
42
43 014066 012700 000005 MOV #OP.RSD,R0      ;BUILD RECEIVE DATA COMMAND
44 014072 004737 016754 CALL BLDCHD
45 014076 012700 000430 MOV #HC.BF2,R0      ;POINT TO MESSAGE BUFFER
46 014102 004737 017116 CALL CLRBUF          ; AND CLEAR IT
47 014106 052765 000020 000012 BIS #CT.REQ,C.FLG(R5) ;SET REQUEST BIT
48 014114 000403 BR R$POU3
49
50 014116 042765 000020 000012 R$POU2: BIC #CT.REQ,C.FLG(R5) ;CLEAR REQUEST BIT
51 014124 R$POU3:
52 014124 004737 017040 CALL SNDCMD          ;SEND COMMAND TO UDA
53 014130 016500 000042 R$POU4: MOV C.TOT(R5),R0 ;SET TIMEOUT

```

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 64 1
GLOBAL SUBROUTINES SECTION

54	014134	010501			MOV R5,R1	
55	014136	062701	000036		ADD #C.TO,R1	;PUT TIME IN CONTROLLER TABLE
56	014142	004737	017352		CALL SETTO	
57	014146	000137	013460		JMP RSPNXT	;NOW WAIT FOR END PACKET
58	014152	122764	000201	000030	RSPERR: CMPB #OP.END+OP.GDS,HC.MPK+P.OPCD(R4)	;SEE IF GET DUST STATUS OPCODE
59	014160	001237			BNE RSPERW	
60	014162	132764	000010	000037	BITB #DF.ACT,HC.MPK+P.DFLG(F4)	;IF DUST NO LONGER RUNNING
61	014170	001603			BEQ RSPTOE	; REPORT ERROR
62	014172	042765	000050	000012	BIC #CT.STA+CT.MSG,C.FLG(R5)	;CLEAR CONTROL BITS
63	014200	032765	000200	000012	BIT #CT.TM2,C.FLG(R5)	;IF AT SECOND TIMEOUT
64	014206	001413			BEQ 1:	
65	014210	026465	000040	000044	CMP HC.MPK+P.DPI(R4),C.PRI(R5)	;COMPARE PROGRESS INDICATOR
66	014216	001004			BNE 2:	
67	014220	026465	000042	000046	CMP HC.MPK+P.DPI+2(R4),C.PRI+2(R5)	;COMPARE PROGRESS INDICATOR
68	014226	001422			BEQ 4:	;REPORT ERROR IF NOT CHANGED
69	014230	042765	000200	000012	2: BIC #CT.TM2,C.FLG(R5)	;CLEAR TIMEOUT 2 FLAG
70	014236	032765	000100	000012	1: BIT #CT.TM1,C.FLG(R5)	;IF AT FIRST TIMEOUT
71	014244	001406			BEQ 3:	
72	014246	016465	000040	000044	MOV HC.MPK+P.DPI(R4),C.PRI(R5)	;GET COPY OF PROGRESS INDICATOR
73	014254	016465	000042	000046	MOV HC.MPK+P.DPI+2(R4),C.PRI+2(R5)	;GET COPY OF PROGRESS INDICATOR
74	014262	012764	140000	000006	3: MOV #RG.OW+RG.FLG,HC.MCT(R4)	;GIVE MESSAGE BUFFER BACK TO UDA
75	014270	000137	013460		JMP RSPNXT	
76	014274	000137	013600		4: JMP RSPTOE	

```
1  
2  
3 014300 014314  
4 014302 014366  
5 014304 014540  
6 014306 014666  
7 014310 014676  
8 014312 014706  
9 000006
```

RESPONSE REQUEST DISPATCH TABLE

RSPDSP:	.WORD QUEST	:QUESTION
	.WORD DQUEST	:QUESTION WITH DEFAULT ANSWER
	.WORD INFO	:INFORMATION MESSAGE FOR OPERATOR
	.WORD TERM	:NORMAL TERMINATION
	.WORD ERRTRM	:FATAL ERROR TERMINATION
	.WORD SPECL	:SPECIAL
DSPSIZ=<.-RSPDSP>/2		:LEGAL NUMBERS ARE LOWER THAN THIS

;NORMAL DUP RECEIVE DATA BUFFER DESCRIPTION

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

;BYTE OFFSET FROM
;START OF BUFFER

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

USED TO SELECT ROUTINE
R4 CONTAINS THIS ADDRESS

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

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

!BYTE OFFSET FROM
!START OF BUFFER

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

R2 CONTAINS THIS ADDRESS

CZUDKO UDAS0A/KDAS0-0 FORMATTER MACKO V05.01b Monday 01 Oct-84 10:07 Page 68
GLOBAL SUBROUTINES SECTION

```

1      ;MESSAGE TYPE 1
2      ;
3      ;ANSWER QUESTION FOR DUP PROGRAM
4      ;
5      ;INPUT:
6      ;      R5 - ADDRESS OF CONTROLLER TABLE
7      ;      R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;      R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;      R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;      R1 - ZERO
11     ;OUTPUT:
12     ;      R1 - COUNT OF CHARACTERS IN SEND BUFFER
13     ;      Z SET TO CONTINUE RUNNING DUP PROGRAM
14     ;      Z CLEAR TO STOP THE DUP PROGRAM
15
16 014314 004737 015040      QUEST: CALL GDRVT      ;GET POINTER TO DRIVE TABLE
17 014320 062700 000004      ADD @D.SERN,R0      ;BUMP POINTER TO SERIAL NUMBER
18 014324 014403              MOV -(R4),R3        ;GET QUESTION NUMBER
19 014326 001411              BEQ QUE0            ;BRANCH IF QUESTION NUMBER 0
20 014330 020327 000007      CMP R3,#7           ;IF NOT, SEE IF QUESTION NUMBER 7
21 014334 001410              BEQ QUE7
22 014336              ERROF 100,ERR100      ;ANY OTHER NUMBER IS AN ERROR
23              TRAP      C#ERDF
24              .WORD    100
25              .WORD    0
26              .WORD    ERR100
27
28 014346 000244              CLZ      ;CLEAR Z TO STOP DUP PROGRAM
29 014350 000207              RETURN
30
31 014352 012700 003304      QUE0: MOV @DATE0,R0    ;POINT TO DATE STRING
32 014356              QUE7:
33 014356 005201              QUEL: INC R1      ;COUNT THE CHARACTERS
34 014360 112022              MOV8 (R0),.(R2).    ;AND PUT THEM IN OUTPUT BUFFER
35 014362 001375              BNE QUEL           ;UNTIL A NUL CHARACTER FOUND
36 014364 000207              RETURN      ;RETURN WITH Z SET

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 69
GLOBAL SUBROUTINES SECTION

```

1      ;MESSAGE TYPE 2
2
3      ;ANSWER QUESTION FOR DUP PROGRAM WITH DEFAULT ANSWER
4
5      ;INPUT:
6      ;       R5 - ADDRESS OF CONTROLLER TABLE
7      ;       R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;       R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;       R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;       R1 - ZERO
11     ;OUTPUT:
12     ;       R1 - COUNT OF CHARACTERS IN SEND BUFFER
13     ;       Z SET TO CONTINUE RUNNING DUP PROGRAM
14     ;       Z CLEAR TO STOP THE DUP PROGRAM
15
16 014366 004737 015040 DQUEST: CALL GDRVT      ;GET DRIVE TABLE ADDRESS INTO R0
17 014372 014403      MOV -(R4),R3      ;GET QUESTION NUMBER
18 014374 020327 000006 CMP R3,#DQUESZ
19 014400 101035      BHI DQUEX
20 014402 006303      ASL R3
21 014404 000173 014410 JMP @DQUEJP(R3)
22 014410 014474 DQUEJP: .WORD DQUEX      ; 0 (NOT USED)
23 014412 014426      .WORD DQUNIT      ; 1 ENTER UNIT NUMBER TO FORMAT
24 014414 014474      .WORD DQUEX      ; 2 (NOT USED)
25 014416 014474      .WORD DQUEX      ; 3 (NOT USED)
26 014420 014500      .WORD DQRFMT      ; 4 USE EXISTING BAD SECTOR INFORMATION
27 014422 014520      .WORD DQRSTR      ; 5 DOWN-LINE LOAD BAD SECTOR BLOCK INFORMATION
28 014424 014530      .WORD DQCONT      ; 6 CONTINUE IF BAD BLOCK INFO INACCESSIBLE
29      DQUESZ=<<<.-DQUEJP>/2>-1
30
31     ;ENTER UNIT NUMBER TO FORMAT
32
33 014426      DQUNIT: PUSH R5
34      CLZ R4
35      MOV (R0),R3      ;GET DRIVE NUMBER
36      ASSUME D.DRV EQ 0
37 014434 012700 000012 MOV #10.,R0      ;RADIX 10.
38 014440 004737 016516 DQUNL1: CALL DIVIDE
39 014444      PUSH R5
40      MOV R5,-(SP)
41      INC R1
42      TST R3
43      BNE DQUNL1
44      MOV R1,R0
45      DQUNL2: POP R5
46      MOV (SP),R5
47      ADD #0,R5
48      MOVB R5,(R2).
49      DEC R0
50      BNE DQUNL2
51      POP R5
52      MOV (SP),R5
53 014474 000264 DQUEX: SEZ
54 014476 000207      RETURN
55 014500 032737 000003 003200 DQRFMT: BIT #50.FMT,MODE

```

CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 69 1
GLOBAL SUBROUTINES SECTION

54	014506	001410				BEQ DQNO
55	014510	112712	000131		DQYES:	MOVB #'Y,(R2)
56	014514	005201				INC R1
57	014516	000766				BR DQUEX
58						
59	014520	032737	000010	003200	DQRSTR:	BIT #SO.STR,MODE
60	014526	001370				BNE DQYES
61	014530				DQCONT:	
62	014530	112712	000116		DQNO:	MOVB #'N,(R2)
63	014534	005201				INC R1
64	014536	000756				BR DQUEX

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 70
GLOBAL SUBROUTINES SECTION

```

1      ;MESSAGE TYPE 3
2      ;
3      ;PRINT INFORMATION FROM DUP PROGRAM
4      ;
5      ;INPUT:
6      ;   R5 - POINTER TO CONTROLLER TABLE
7      ;   R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;   R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;   R1 - ZERO
11     ;OUTPUT:
12     ;   R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM
13     ;   Z SET TO CONTINUE RUNNING DUP PROGRAM
14     ;
15 014540 016400 177776 INFO:  MOV -2(R4),R0 ;GET MESSAGE NUMBER
16 014544 001434      BEQ INFOB ;IF ZERO, PRINT BEGUN MESSAGE
17 014546 020027 000100  CMP RO,#100 ;IF OCTAL 100
18 014552 001423      BEQ INFOE ; PRINT ERROR MESSAGE
19 014554 020027 000200  CMP RO,#200 ;SEE IF 200 OR GREATER
20 014560 002005      BGE INFOH ; IF SO, PRINT WITHOUT FREEZING
21 014562 005737 002170  TST W,HEEZ
22 014566 001007      BNE INFOP
23 014570 005237 002170  INC UFREEZ
24 014574 004737 015040  INFOH: CALL GTDRVT
25 014600 010002      MOV RO,R2
26 014602 004737 015064  CALL HEAJER
27 014606 004737 015004  INFOP: CALL MMSG ;PRINT THE MESSAGE
28 014612 012701 100000  INFOX: MOV #BIT15,R1 ;RETURN A NEGATIVE BYTE COUNT
29 014616 000264      SEZ
30 014620 000207      RETURN ;RETURN WITH Z SET
31
32 014622      INFOE: ERROF 101,ERP101 ;ANSWER WAS REJECTED BY DUP PROGRAM
33      014622 104455      TRAP C8ERDF
34      014624 000145      .WORD 101
35      014626 000000      .WORD 0
36      014630 012470      .WORD ERR101
37 014632 000244      CLZ ;RETURN WITH Z CLEAR TO STOP DUP PROGRAM
38 014634 000207      RETURN
39
40 014636 004737 015040  INFOB: CALL GTDRVT ;PRINT FORMAT BEGUN MESSAGE
41 014642 010002      MOV RO,R2
42 014644 004737 015064  CALL HEADER
43 014650 004737 015004  CALL MMSG
44 014654      PNT WNSTOP ;PRINT WARNING NOT TO STOP NOW
45      014654 004137 016450  JSR R1,LPNT
46      014660 004355      .WORD WNSTOP
47      014662 000000      .WORD PNT.CT
48 014664 000752      BR INFOX

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 71
 GLOBAL SUBROUTINES SECTION

```

1      ;MESSAGE TYPE 4
2      ;
3      ;TERMINATION MESSAGE
4      ;
5      ;INPUT:
6      ;   R5 - POINTER TO CONTROLLER TABLE
7      ;   R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;   R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;   R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;   R1 - ZERO
11     ;OUTPUT:
12     ;   Z CLEAR TO TERMINATE DUP PROGRAM
13
14 014666 004737 014540  TERM:  CALL INFO      ;PRINT THE MESSAGE
15 014672 000244          CLZ
16 014674 000207          RETURN          ;RETURN Z CLEAR TO TERMINATE DUP PROGRAM

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 014676 004737 014540
15 014702 000244
16 014704 000207

```

```

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

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 73
GLOBAL SUBROUTINES SECTION

```

1      ;MESSAGE TYPE 6
2
3      ;SPECIAL TYPE - READ FCT BLOCK FROM FILE
4
5      ;INPUT:
6      ;       R5 - POINTER TO CONTROLLER TABLE
7      ;       R4 - POINTER TO DATA IN RECEIVE BUFFER
8      ;       R3 - CHARACTER COUNT IN RECEIVE BUFFER
9      ;       R2 - POINTER TO SEND BUFFER (BUFFER IS CLEARED)
10     ;       R1 - ZERO
11     ;OUTPUT:
12     ;       Z SET TO SEND DATA TO PROGRAM
13
14 014706 023714 003176   SPECL:  CMP FCTNUM,(R4) ;SEE IF DESIRED BLOCK IS IN MEMORY
15 014712 001425         BEQ SPECLX   ; IF SO, SEND TO DUP PROGRAM
16 014714 002407         BLT SPECLR   ; IF LOWER NUMBERED BLOCK IN MEMORY,
17                                     ; GO READ NEXT BLOCK
18 014716               SPECLC:
19 014716               CLOSE      ;OTHERWISE, START READING FROM BEGINNING AGAIN
20 014716 104435         OPEN #FNAME                                TRAP    C#CLOS
21 014720               MOV #512,R3                                MOV    #FNAME,R0
22 014720 012700 003230   MOV #-1,FCTNUM                               TRAP    C#OPEN
23 014724 104434         MOV #512,R3                                ;GET BYTE COUNT IN A BLOCK
24 014726 012737 177777 003176   MOV #FCTBUF,R1             ;POINT TO STORAGE AREA
25 014734 012703 001000   SPECLR:  GETBYTE (R1)+          ;READ THE FILE
26 014740 012701 002176   ;PRINT ERROR IF NO MORE BYTES IN FILE
27 014744 104426         BNCOMPLETE SPECLE                               TRAP    C#GETB
28 014746 110021         ;COUNT THE BYTES                               MOV    RO,(R1)+
29 014750 103005         DEC R3                                     BCC    SPECLE
30 014752 005303         BNE SPECLL
31 014754 001373         INC FCTNUM          ;KEEP COUNT OF BLOCK IN MEMORY
32 014756 005237 003176   BR SPECL
33 014762 000751
34 014764 005212         SPECLE:  INC (R2)          ;TELL DUP PROGRAM DATA NOT AVAILABLE
35 014766 012762 002176 000002   SPECLX:  MOV #FCTBUF,2(R2)      ;PUT ADDRESS OF DATA IN OUTPUT BUFFER
36 014774 012701 000006   MOV #6,R1          ;SEND 3 WORDS TO DUP PROGRAM
37 015000 000264
38 015002 000207         SEZ
39                                     RETURN
40                                     ;RETURN WITH Z SET TO SEND DATA TO DUP PROGRAM

```

```

1      ;PRINT A MESSAGE IN THE RECEIVE BUFFER FROM THE DUP PROGRAM
2      ;
3      ;INPUT:
4      ;      R4 - POINTER TO DATA IN RECEIVE BUFFER
5      ;      R3 - CHARACTER COUNT IN RECEIVE BUFFER
6      ;OUTPUT:
7      ;      R4 - POINTER TO CHARACTER AFTER MESSAGE IN RECEIVE BUFFER
8      ;      R3 - ZERO
9      ;      R1 - BIT 15 SET TO PREVENT SENDING DATA TO DUP PROGRAM
10     ;      R0 - CONTENTS DESTROYED
11     ;      Z SET TO CONTINUE RUNNING DUP PROGRAM
12
13     MSG:
14     1$:      MOVB (R4)+,R0          ;PRINT CHARACTERS FROM DUP PROGRAM
15             BEQ 2$              ; DISCARDING LF AND NULL CHARACTERS
16             CMP R0,#12
17             BEQ 2$
18             PRINT R0
19
20     2$:      DEC R3                ;COUNT THE CHARACTERS
21             BGT 1$
22             PRINT #CR
23
24             MOVB #CR,R0
25             CALL CPNT
26
27     RETURN
    
```

```

13 015004
14 015004 112400
15 015006 001405
16 015010 020027 000012
17 015014 001402
18 015016
   015016 004737 016240
19 015022 005303
20 015024 003367
21 015026
   015026 112700 000015
   015032 004737 016240
22 015036 000207
    
```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 75
 GLOBAL SUBROUTINES SECTION

```

1      ;GDRV
2      ;
3      ;GET DRIVE TABLE ADDRESS FROM CONTROLLER TABLE
4      ;
5      ;INPUTS:
6      ;      R5 - CONTROLLER TABLE ADDRESS
7      ;OUTPUTS:
8      ;      R0 - ADDRESS OF FIRST DRIVE TABLE AVAILABLE FOR TESTING
9      ;              (WITH DT.AVL BIT CLEAR)
10
11 015040      GDRV: PUSH R5
12 015040      010546      MOV R5, (SP)
13 015042      062705      000016      ADD #C.DRO,R5
14 015046      012500      GDRV: MOV (R5)+,R0
15 015050      016037      000002      002074      MOV D.UNIT(R0),L#LUN
16 015056      100773      ASSUME DT.AVL EQ BIT15
17 015060      012605      BMI GDRV
18 015062      000207      POP R5
                                MOV (SP)+,R5
                                RETURN

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 76
GLOBAL SUBROUTINES SECTION

```

1      ;HEADER
2      ;
3      ;PRINT A HEADER IN FRONT OF EACH MESSAGE FROM DUP PROGRAM.
4      ;A UDA ADDRESS IS PRINTED IF MORE THAN ONE UDA IS IN HARDWARE P-TABLE.
5      ;A RUNTIME IS PRINTED IF A CLOCK IS BEING USED TO TIME PROGRAM EXECUTION.
6      ;
7      ;INPUT:
8      ;   R5 - POINTER TO CONTROLLER TABLE
9      ;OUTPUT:
10     ;   R0 - POINTER TO DRIVE TABLE
11     ;   PRINTED MESSAGE
12     ;
13 015064 022737 000001 002012 HEADER: CMP #1,L#UNIT                ;IF MORE THAN ONE UNIT BEING TESTED
14 015072 001411                                BEQ 1$
15 015074                                PNTF MESSG,C.UNIT(R2),(R5),(R2)        ;PRINT UDA ADDRESS
16     015074 011246                                MOV (R2),-(SP)
17     015076 011546                                MOV (R5),-(SP)
18     015100 016246 000002                                MOV D.UNIT(R2),-(SP)
19     015104 004137 016412                                JSR R1,L#PNTF
20     015110 004035                                .WORD MESSG
21     015112 000006                                .WORD PNT.CT
22     16 015114                                ASSUME C.UADR EQ 0
23     17 015114                                ASSUME D.DRV EQ 0
24     18 015114 000407                                BR 2$
25     19 015116 005737 003206 1$: TST KW.CSR                ;IF NO CLOCK BEING USED
26     20 015122 001406                                BEQ 3$                ;BYPASS RUNTIME MESSAGE
27     21 015124                                PRINT #CR
28     22 015134 004737 020402 2$: CALL RNTIME                ;PRINT RUNTIME IF A CLOCK IN USE
29     23 015140 004737 016240 3$: PRINT #CR
30     24 015140 112700 000015                                MOVB #CR,R0
31     25 015130 004737 016240                                CALL CPNT
32     26 015144 004737 016240                                MOVB #CR,R0
33     27 015150 000207                                CALL CPNT
34     28 015150 000207                                RETURN

```

```

1      ;OSTRNG
2      ;
3      ;FORMAT OF THE ASCIZ STRING IS AS FOLLOWS:
4      ;
5      ;CHARACTERS ENCLOSED IN QUOTES ARE TO BE PRINTED AS THEY ARE.
6      ;
7      ;OTHERWISE CODE IS A SINGLE LETTER FOLLOWED BY AN OPTIONAL DECIMAL
8      ;NUMBER:
9      ; ON - PRINT OCTAL NUMBER. N REPRESENTS SIZE OF BINARY NUMBER PASSED
10     ;      IN PARAMETER IN BITS. MAY BE IN RANGE 1 TO 32. IF N>16, TWO PARAMETER
11     ;      WORDS ARE USED, OTHERWISE ONLY ONE WORD. LEADING ZEROS ARE PRINTED.
12     ;      N IS ALWAYS SPECIFIED.
13     ; DN - PRINT UNSIGNED DECIMAL NUMBER FROM N BIT PARAMETER. LEADING ZEROS
14     ;      ARE NOT PRINTED. A 16 BIT NUMBER EQUAL TO ZERO WILL PRINT "0".
15     ; HN - PRINT HEX NUMBER FROM PARAMETER OF N BITS. IF N>16 TWO PARAMETERS
16     ;      ARE USED, OTHERWISE ONLY ONE PARAMETER. LEADING ZEROS ARE PRINTED.
17     ; SN - PRINT N SPACES. N ASSUMED TO BE 1.
18     ; NN - START NEW LINE (CR-LF SEQUENCE). N ASSUMED TO BE 1.
19     ; AN - PRINT N ASCII CHARACTERS FROM PARAMETERS. N ASSUMED TO BE 1.
20     ;      N/2 PARAMETER WORDS USED.
21     ; RN - EXECUTE ROUTINE #N. N MUST BE GIVEN AND DEFINED IN HOST PROGRAM.
22     ;
23     ;A NULL CHARACTER MEANS END OF MESSAGE. A NULL AS FIRST CHARACTER IN STRING
24     ;MUST BE IGNORED.
25     ;
26     ;OUTPUT A MESSAGE ACCORDING TO A FORMAT STRING
27     ;
28     ;INPUTS:
29     ;      R2 - ADDRESS OF START OF FORMAT STRING
30     ;      R4 - ADDRESS OF PARAMETERS
31     ;OUTPUTS:
32     ;      R2 AND R4 UPDATED TO END OF STRING AND PARAMETERS
33     ;
34 015152 112201      OSTRNG: MOVB (R2)+,R1      ;GET CONTROL CHARACTER
35 015154 001421      BEQ OSTRE      ;EXIT IF NULL CHARACTER
36 015156 012700 015452  MOV #ERRC,R0      ;GET POINTER TO CHARACTER TABLE
37 015162 120110      NCONS: CMPB R1,(R0)      ;COMPARE CHARACTER WITH TABLE ENTRY
38 015164 001407      BEQ NCONF      ;BRANCH IF MATCH FOUND
39 015166 105720      TSTB (R0)+      ;INCREMENT POINTER
40 015170 001374      BNE NCONS      ;CONTINUE SEARCH IF NOT END OF TABLE
41 015172      PNTF ERRME1      ;REPORT BAD CONTROL CHARACTER
42      015172 004137 016412      JSR R1,LPNTF
43      015176 003746      .WORD ERRME1
44      015200 000000      .WORD PNT.CT
45 015202 000406      NCONF: BR OSTRE
46 015204 162700 015452  SUB #ERRC,R0      ;GET INCREMENT INTO TABLE
47 015210 006300      ASL R0      ;DOUBLE TO WORD COUNT
48 015212 004770 015464  CALL BERRD(R0)      ;DISPATCH TO PRINT ROUTINE
49 015216 000755      BR OSTRNG      ;GET NEXT
50 015220 000207      OSTRE: RETURN

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 78
GLOBAL SUBROUTINES SECTION

```

1          ;CONTROL CHARACTER WAS A QUOTE. PRINT ALL CHARACTERS TO THE NEXT QUOTE.
2
3 015222 112200          CON.QU: MOVB (R2),R0          ;GET CHARACTER
4 015224 120027 000042  CMPB R0,#'"          ;CHECK IF ENDING QUOTE
5 015230 001403          BEQ CON.QX          ;IF SO, GO GET NEXT CONTROL CHARACTER
6 015232          PRINT R0          ;PRINT THE CHARACTER
7 015232 004737 016240          BR CON.QU          CALL CPNT
8 015236 000771          CON.QX: RETURN          ;CONTINUE PRINTING
9 015240 000207
10         ;CONTROL CHARACTER WAS AN A. PRINT ASCII CHARACTERS FROM PARAMETERS.
11
12 015242 004737 015720  CON.A: CALL GETCNT          ;GET COUNT OF CHARACTERS
13 015246          CON.A1: PRINT (R4)          ;PRINT THE CHARACTER
14 015246 112400          MOVB (R4),R0          ;COUNT THE CHARACTERS
15 015250 004737 016240          CALL CPNT          ;PRINT UNTIL COUNT REACHES ZERO
16 015254 005301          DEC R1          ;CHECK IF R4 NOW ODD
17 015256 001373          BNE CON.A1          ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
18 015260 032704 000001  BIT #1,R4          ;NOW GET NEXT CONTROL CHARACTER
19 015264 001401          BEQ CON.A2
20 015266 005204          INC R4
21 015270 000207          CON.A2: RETURN
22
23         ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.
24
25 015272 012701 000012  CON.D: MOV #10.,R1          ;LOAD RADIX
26 015276 004737 015776  CALL PNTNUM          ;PRINT NUMBER
27 015302 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER
28
29         ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.
30
31 015304 012701 000020  CON.H: MOV #16.,R1          ;LOAD RADIX
32 015310 004737 015776  CALL PNTNUM          ;PRINT NUMBER
33 015314 000207          RETURN          ;NOW GET NEXT CONTROL CHARACTER

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 79
GLOBAL SUBROUTINES SECTION

```

1
2
3 015316 012701 000010
4 015322 004737 015776
5 015326 000207
6
7
8
9 015330 004737 015720
10 015334
    015334 112700 000015
    015340 004737 016240
11 015344 005301
12 015346 001372
13 015350 000207
14
15
16
17 015352 004737 015720
18 015356 020127 000010
19 015362 101004
20 015364 060101
21 015366 004771 015430
22 015372 000207
23 015374
    015374 004137 016412
    015400 003746
    015402 000000
24 015404
    015404 012601
25 015406 000207
26
27
28
29 015410 004737 015720
30 015414
    015414 112700 000040
    015420 004737 016240
31 015424 005301
32 015426 001372
33 015430 000207

;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.
CON.O:  MOV #0.,R1          ;LOAD RADIX
        CALL PNTNUM        ;PRINT NUMBER
        RETURN            ;NOW GET NEXT CONTROL CHARACTER

;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.
CON.N:  CALL GETCNT        ;GET COUNT
CON.N1: PRINT @CR          ;PRINT NEW LINE SEQUENCE
                                MOVB @CR,R0
                                CALL CPNT

        DEC R1             ;COUNT THE SEQUENCES
        BNE CON.N1
        RETURN            ;NOW GET NEXT CONTROL CHARACTER

;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.
CON.R:  CALL GETCNT        ;GET ROUTINE NUMBER
        CMP R1,@ERRRSZ    ;CHECK IF DEFINED ROUTINE NUMBER
        BHI CON.R1
        ADD R1,R1         ;DOUBLE COUNT TO GET WORD INDEX
        CALL BERRRTB-2(R1);CALL ROUTINE
        RETURN            ;NOW GET NEXT CONTROL CHARACTER
CON.R1: PNTF ERRME1        ;REPORT BAD MESSAGE STRING
                                JSR R1,LPNTF
                                .WORD ERRME1
                                .WORD PNT.CT

        POP R1            ;FIX THE STACK
        RETURN            MOV (SP),R1

;CONTROL CHARACTER WAS AN S. PRINT SPACES.
CON.S:  CALL GETCNT        ;GET COUNT
CON.S1: PRINT '<@>'       ;PRINT A SPACE
                                MOVB @',R0
                                CALL CPNT

        DEC R1             ;COUNT THE SPACES
        BNE CON.S1
        RETURN            ;NOW GET NEXT CONTROL CHARACTER

```

```

1
2
3 015432 015504
4 015434 015504
5 015436 015504
6 015440 015516
7 015442 015572
8 015444 015650
9 015446 015664
10 015450 015702
11 000010
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

;ERROR ROUTINE DISPATCH TABLE
ERRRTB: .WORD CALRF ;NOT USED
        .WORD CALRE ;NOT USED
        .WORD CALRE ;NOT USED
        .WORD CALR4 ;PRINT BASIC LINE WITHOUT UDA ADDRESS
        .WORD CALR5 ;PRINT BASIC LINE WITH UDA ADDRESS
        .WORD CALR6 ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
        .WORD CALR7 ;PRINT "REPLACE PROCESSOR MODULE"
        .WORD CALR8 ;PRINT " UDASA CONTAINS XXXXXX"
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

```

```

1
2
3
4
5
6
7
8
9 015452
   015452      042
   015453      101
   015454      104
   015455      110
   015456      117
   015457      116
   015460      122
   015461      123
10 015462      000
11
12
13
14
15
16
17
18
19
20
21 015464
   015464 015222
   015466 015242
   015470 015272
   015472 015304
   015474 015316
   015476 015330
   015500 015352
   015502 015410

```

```

;HERE IS FIRST TABLE
.MACRO ENTRY ARG1,ARG2
  .LIST
  .BYTE ' ' ARG1
  .NLIST
.ENDM

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

```

;FOLLOW WITH A NULL BYTE

```

;HERE IS SECOND TABLE
.MACRO ENTRY ARG1,ARG2
  .LIST
  .WORD ARG2
  .NLIST
.ENDM

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

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 82
GLOBAL SUBROUTINES SECTION

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 83
 GLOBAL SUBROUTINES SECTION

```

1      ;PRE-PROGRAMMED ROUTINE 4
2      ;PRINT BASIC LINE FOR MOST PROGRAM ERROR WITHOUT UDA ADDRESS
3      ;THEN SWITCH TO EXTENDED FORMAT
4
5      CALR4:  PNTB BASLN,#BASNO,#BAS,#BAS,#BAS
6
7      MOV #BAS,-(SP)
8      MOV #BAS,-(SP)
9      MOV #BAS,-(SP)
10     MOV #BASNO,-(SP)
11     JSR R1,LPNTB
12     .WORD BASLN
13     .WORD PNT.CT
14
15     CALL RNTIME
16     PRINT #CR
17
18     MOV #CR,R0
19     CALL CPNT
20
21     MOV #PX,PTYPE
22     RETURN
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 84
 GLOBAL SUBROUTINES SECTION

```

1
2
3
4
5 015572          ;PRE-PROGRAMMED ROUTINE 5
   015572 012746 004247 ;PRINT BASIC LINE FOR HOST PROGRAM ERROR WITH UDA ADDRESS
   015576 012746 004247 ;THEN SWITCH TO EXTENDED FORMAT
   015602 011546
   015604 012746 004204
   015610 012746 004165
   015614 004137 016422
   015620 004250
   015622 000012
6 015624 004737 020402 CALRS: PNTB BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
7 015630          CALL RNTIME
   015630 112700 000015 PRINT #CR
   015634 004737 016240
8 015640 012737 016340 003222 MOV #PX,PTYPE
9 015646 000207          RETURN
                                MOV #BAS,-(SP)
                                MOV #BAS,-(SP)
                                MOV (R5),-(SP)
                                MOV #BASL2,-(SP)
                                MOV #BASNO,-(SP)
                                JSR R1,LPNTB
                                .WORD BASLN
                                .WORD PNT.CT

                                MOVB #CR,R0
                                CALL CPNT
  
```

CZUDKO UASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 85
GLOBAL SUBROUTINES SECTION

```
1
2
3
4 015650          ;PRE-PROGRAMMED ROUTINE 6
   015650 010246  ;CALL ALTERNATE PRINT ROUTINE IN PDP 11 MEMORY
5 015652 012402
6 015654 004737 015152
7 015660          CALR6: PUSH R2          ;SAVE CURRENT STRING POINTER
   015660 012602          MOV R2, (SP)
8 015662 000207          MOV (R4),R2    ;GET NEW STRING POINTER
          CALL OSTRNG    ;OUTPUT USING THIS STRING
          POP R2        ;GET OLD POINTER BACK
          RETURN       ;NOW CONTINUE THE OLD STRING
          MOV (SP),R2
```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 86
GLOBAL SUBROUTINES SECTION

```
1  
2  
3  
4 015664  
   015664 010246  
5 015666 012702 011532  
6 015672 004737 015152  
7 015676  
   015676 012602  
8 015700 000207  
  
;PRE-PROGRAMMED ROUTINE 7  
;PRINT "REPLACE PROCESSOR MODULE"  
  
CALR7: PUSH R2  
  
      MOV #XFRU,R2  
      CALL OSTRNG  
      POP R2  
  
      MOV R2,.(SP)  
  
      MOV (SP),.R2  
  
      RETURN
```

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 87
GLOBAL SUBROUTINES SECTION

```
1
2
3
4 015702          ;PRE-PROGRAMMED ROUTINE 8
   015702 010246  ;PRINT " UDASA CONTAINS XXXXXX"
5 015704 012702 011504 CALR8: PUSH R2
6 015710 004737 015152          MOV #XSA,R2
7 015714          CALL OSTRNG
   015714 012602          POP R2
8 015716 000207          MOV (SP),R2
          RETURN
```

```

1      ;GETCNT
2      ;
3      ;GET COUNT IN NEXT CHARACTERS OF STRING POINTED TO BY R2.
4      ;NUMBER WILL BE IN DECIMAL. IF NO NUMBER, RETURN A
5      ;DEFAULT OF 1.
6      ;
7      ;INPUTS:
8      ;   R2 - POINTER TO ASCII STRING
9      ;
10     ;OUTPUTS:
11     ;   R1 - NUMBER READ OR A ONE
12     ;   R2 - POINTING TO CHARACTER AFTER NUMBER
13     GETCNT: PUSH R0
14     015720 010046
15     015722 005001
16     015724 121227 000060
17     015730 103415
18     015732 121227 000071
19     015736 101012
20     015740 006301
21     015742 010100
22     015744 006301
23     015746 006301
24     015750 060001
25     015752 112200
26     015754 162700 000060
27     015760 060001
28     015762 000760
29     015764 005701
30     015766 001001
31     015770 005201
32     015772 012600
33     015774 000207

```

```

;START WITH ZERO COUNT
;CHECK IF CHARACTER A DIGIT
;BRANCH IF LOWER THAN ZERO
;BRANCH IF HIGHER THAN NINE
;MULTIPLY NUMBER BY 10
;SAVE 2N
;COMPUTE 4N
;COMPUTE 8N
;8N + 2N = 10N
;GET DIGIT FROM STING
;GET RID OF ASCII
;ADD TO NUMBER
;GO TO NEXT CHARACTER
;CHECK IF NUMBER IS ZERO
;IF ZERO, CHANGE
; TO DEFAULT OF ONE

```

```

MOV RO, -(SP)
CLR R1
CMPB (R2), #'0
BLO GETCDN
CMPB (R2), #'9
BHI GETCDN
ASL R1
MOV R1, R0
ASL R1
ASL R1
ADD R0, R1
MOVB (R2), .R0
SUB #'0, R0
ADD R0, R1
BR GETCNX
TST R1
BNE GETCXX
INC R1
POP R0
RETURN

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct 84 10:07 Page 89
GLOBAL SUBROUTINES SECTION

```

1      ;PNTNUM
2      ;
3      ;PRINT A NUMBER
4      ;
5      ;INPUTS:
6      ;   R1 - RADIX OF NUMBER
7      ;   R2 - ASCII STRING TO COUNT OF BITS IN NUMBER
8      ;   R4 - POINTER TO NUMBER (LOW WORD)
9      ;OUTPUTS:
10     ;   NUMBER IS PRINTED. LEADING ZEROS ARE PRINTED EXCEPT FOR
11     ;   DECIMAL NUMBERS.
12     ;   R0 - CONTENTS DESTROYED
13
14 015776 010100      PNTNUM: MOV R1,R0          ;SAVE RADIX
15 016000 004737 015720 CALL GETCNT          ;GET COUNT OF BITS
16 016004      PNTNUS: PUSH <R2,R3,R5>
17 016012 012403      MOV (R4)+,R3          ;GET ONE PARAMETER WORD
18 016014 005005      CLR R5              ;CLEAR STORAGE FOR OTHER
19 016016 020127 000020 CMP R1,#16.          ;MORE THAN 16 BITS IN NUMBER?
20 016022 003401      BLE 1#
21 016024 012405      MOV (R4)+,R5          ;YES, GET SECOND PARAMETER WORD
22 016026      1#:   PUSH R4
23 016030 010446      MOV R5,R4              ;PUT HIGH WORD IN R4
24 016032 012702 000020 MOV #16.,R2          ;COMPUTE BITS NOT WANTED
25 016036 160102      SUB R1,R2              ;BY SUBTRACTING BITS TO USE
26 016040 002002      BGE 2#              ;FROM 16.
27 016042 062702 000020 ADD #16.,R2          ;IF NEGATIVE, ADD 16 FOR FIRST WORD
28 016046 001414      2#:   BEQ 6#              ;IF ZERO, NO BITS NEED BE CLEARED
29 016050 012705 100000 MOV #BIT15,R5       ;START MASK WITH SIGN BIT SET
30 016054 005302      3#:   DEC R2              ;COUNT BITS IN MASK
31 016056 001402      BEQ 4#
32 016060 006205      ASR R5              ;SHIFT MORE BITS TO RIGHT
33 016062 000774      BR 3#
34 016064 020127 000020 4#:   CMP R1,#16.          ;MORE THAN 16 BITS IN NUMBER?
35 016070 003402      BLE 5#
36 016072 040504      BIC R5,R4          ;YES, CLEAR IN HIGH WORD
37 016074 000401      BR 6#
38 016076 040503      5#:   BIC R5,R3          ;NO, CLEAR IN LOW WORD
39 016100 004737 016516 6#:   CALL DIVIDE       ;DIVIDE BY RADIX IN R0
40 016104      PUSH R5          ;PUSH REMAINDER ON STACK
41 016106 010546      MOV R5,(SP)
42 016110 005202      INC R2              ;COUNT DIGITS ON STACK
43 016112 005703      TST R3              ;CHECK IF QUOTIENT IS ZERO
44 016114 001372      BNE 6#
45 016116 005704      TST R4
46 016118 001370      BNE 6#

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct 84 10:07 Page 90
GLOBAL SUBROUTINES SECTION

1	016120	020027	000012						
2	016124	001423							
3	016126	010103							
4	016130	162700	000014						
5	016134	003002							
6	016136	012700	000003						
7	016142	004737	016516	7#:	CALL DIVIDE				
8	016146	005705							
9	016150	001401							
10	016152	005203							
11	016154	160203		8#:	SUB R2,R3				
12	016156	001406							
13	016160			9#:	PRINT #'0				
	016160	112700	000060						
	016164	004737	016240						
14	016170	005303							
15	016172	001372							
16									
17	016174			10#:	POP R5				
	016174	012605							
18	016176	062705	000060						
19	016202	020527	000071						
20	016206	003402							
21	016210	062705	000007						
22	016214			11#:	PRINT R5				
	016214	110500							
	016216	004737	016240						
23	016222	005302							
24	016224	001363							
25	016226								
	016226	012604							
	016230	012605							
	016232	012603							
	016234	012602							
26	016236	000207							

RETURN

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

;IF REMAINDER NOT ZERO
; INCREMENT QUOTIENT

; SUBTRACT DIGITS ON STACK
; NO LEADING ZEROS IF ZERO
; PRINT A ZERO

MOVB #'0,R0
CALL CPNT

; REPEAT UNTIL COUNT REACHES ZERO

; GET CHARACTER FROM STACK
MOV (SP)+,R5

; CONVERT TO ASCII DIGIT
; IF GREATER THAN A 9
; CONVERT TO A OR HIGHER
; FOR HEX DIGIT
; PRINT THE CHARACTER

MOVB R5,R0
CALL CPNT

; REPEAT FOR ALL DIGITS
; ON STACK

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

CZUDKO UDA50A/KDA50 @ FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 91
GLOBAL SUBROUTINES SECTION

```

1      ;PRINT ONE CHARACTER
2
3      ;CALL WITH MACRO PRINT
4
5 016240 110037 003224      CPNT:  MOV B R0,ERRCHR
6 016244                                PUSH R1
7 016246 010146                                MOV R1, (SP)
8 016252 120027 000015      MOV #ERRONE,R1
9 016256 001002                                CMPB R0,#CR
10 016260 012701 003704      BNE 1$
11 016264 000177 164732      1$:  MOV #ERRNL,R1
12 016270                                JMP @TYPE
13 016312 000435                                PF:  PRINTF R1,#ERRCHR
14 016314                                MOV #ERRCHR, (SP)
15 016314 012746 003224      PB:  PRINTB R1,#ERRCHR
16 016320 010146                                MOV R1, -(SP)
17 016322 012746 000002      MOV #2, -(SP)
18 016326 010600                                MOV SP,R0
19 016330 104414                                TRAP C#PNTB
20 016332 062706 000006      ADD #6,SP
21 016336 000423                                BR CPNTX
22 016340 012746 003224      PX:  PRINTX R1,#ERRCHR
23 016344 010146                                MOV #ERRCHR, -(SF)
24 016346 012746 000002      MOV R1, -(SP)
25 016352 010600                                MOV #2, -(SP)
26 016354 104415                                MOV SP,R0
27 016356 062706 000006      TRAP C#PNTX
28 016362 000411                                ADD #6,SP
29 016364 012746 003224      PS:  PRINTS R1,#ERRCHR
30 016370 010146                                MOV #ERRCHR, -(SP)
31 016372 012746 000002      MOV R1, -(SP)
32 016376 010600                                MOV #2, -(SP)
33 016400 104416                                MOV SP,R0
34 016402 062706 000006      TRAP C#PNTS
35 016406 012601                                ADD #6,SP
36 016410 000207      CPNTX: POP R1
37                                MOV (SP)+,R1
38                                RETURN

```

CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 92
GLOBAL SUBROUTINES SECTION

```

1          ;PRINT FORMATTED MESSAGE
2
3          ;CALL WITH MACRO PNT, PNTF, PNTB, PNTX, OR PNTS
4
5 016412 012737 016270 003222 LPNTF:  MOV @PF,PTYPE
6 016420 000413                BR LPNT
7 016422 012737 016314 003222 LPNTB:  MOV @PB,PTYPE
8 016430 000407                BR LPNT
9 016432 012737 016340 003222 LPNTX:  MOV @PX,PTYPE
10 016440 000403                BR LPNT
11 016442 012737 016364 003222 LPNTS:  MOV @PS,PTYPE
12 016450                LPNT:  PUSH <R2,R3,R4,R5>
13 016450 010246                MOV R2,-(SP)
14 016452 010346                MOV R3,-(SP)
15 016454 010446                MOV R4,-(SP)
16 016456 010546                MOV R5,-(SP)
17 016460 012102                MOV (R1),R2
18 016462 010604                MOV SP,R4
19 016464 062704 000012        ADD @10.,R4
20 016470                PUSH R1
21 016472 004737 015152        CALL OSTRING
22 016476 012600                POP <R0,R5,R4,R3,R2,R1>
23 016500 012605
24 016502 012604
25 016504 012603
26 016506 012602
27 016510 012601
28 016512 062006                ADD (R0),SP
29 016514 000110                JMP BRO

```

```

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

```


CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 94
GLOBAL SUBROUTINES SECTION

```

1      ;LOADDM
2      ;
3      ;LOAD AND START A DM PROGRAM INTO A CONTROLLER
4      ;
5      ;INPUTS:
6      ;      R5 - CONTROLLER TABLE ADDRESS
7      ;      DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
8      ;
9      ;OUTPUTS:
10     ;      IF LOAD SUCCEEDS - Z CLEAR
11     ;      CONTROLLER TABLE MARKED LOADED
12     ;      IF ERROR - Z SET
13     016554 013701 002156      LOADDM: MOV DMPROG,R1      ;GET STORAGE ADDRESS OF DM PROGRAM
14     016560 116165 000021 000042  MOVB DMTH0(R1),C.TOT(R5)  ;GET TIMEOUT VALUE
15     016566 105065 000043      CLRB C.TOT+1(R5)
16     016572 016504 000004      MOV C.VEC(R5),R4      ;GET VECTOR OF UDA
17     016576      AND CT.VEC,R4
18     016602 010501      MOV R5,R1      ;GET INTERRUPT SERVICE LINK      BIC #*C<CT.VEC>,R4
19     016604 062701 000006      ADD #C.JSR,R1
20     016610      SETVEC R4,R1,#PRI07      ;SET UP INTERRUPT VECTOR
21     016610 012746 000340      MOV      #PRI07,(SP)
22     016614 010146      MOV      R1,-(SP)
23     016616 010446      MOV      R4,-(SP)
24     016620 012746 000003      MOV      #3,(SP)
25     016624 104437      TRAP    C$VEC
26     016626 062706 000010      ADD     #10,SP
27     ;INITIALIZE UDA WITH SMALLEST
28     ; RING BUFFER AND INTERRUPTS ENABLED
29     ;BRANCH IF AN ERROR
30     016632 004737 017434      CALL UDAINIT
31     016636 001444      BEQ LOADER

```

CZUDKO UDASOA/KDASO-Q FCAMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 95
 GLOBAL SUBROUTINES SECTION

1	016640	012700	000002	
2	016644	004737	016754	
3	016650	013764	002156	000124
4	016656	017764	163274	000120
5	016664	013764	002156	000140
6	016672	067764	163260	000140
7	016700	004737	017040	
8	016704	004737	017160	
9	016710	001417		
10	016712	032764	000037	000032
11	016720	001007		
12	016722	042765	000024	000012
13	016730	052765	000002	000012
14	016736	000207		

```

MOV #OP.ESP,R0          ;BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET
CALL BLDCHD
MOV DMPROG,HC.CPK+P.UADR(R4) ;LOAD MAIN PROGRAM ADDRESS
MOV @DMPROG,HC.CPK+P.BCNT(R4) ; AND SIZE
MOV DMPROG,HC.CPK+P.OVRL(R4) ;LOAD OVERLAY ADDRESS
ADD @DMPROG,HC.CPK+P.OVRL(R4)
CALL SDCMD              ;SEND COMMAND TO UDA
CALL WAITMS             ;WAIT FOR MESSAGE RESPONSE
BEQ LOADER              ;ABORT IF NO RESPONSE
BIT #ST.MSK,HC.MPK+P.STS(R4) ;CHECK FOR ERRORS
BNE LOADE1
BIC #CT.CMD+CT.REQ,C.FLG(R5) ;CLEAR COMMAND OUTSTANDING FLAG
BIS #CT.RN,C.FLG(R5)     ;SET DM PROGRAM RUNNING FLAG
RETURN
  
```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct 84 10:07 Page 96
GLOBAL SUBROUTINES SECTION

1
2
3 016740 104455
016740 000042
016742 000000
016744 012420
4 016746 000264
5 016750 000207
016752

;UDA FAILED TO DOWNLINE LOAD DM PROGRAM

LOADE1: ERRDF 34,,FRRO34

LOADER: SEZ
RETURN

TRAP C1ERDF
.WORD 34
.WORD 0
.WORD ERRO34

;SET Z TO INDICATE ERROR OCCURRED

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 97
GLOBAL SUBROUTINES SECTION

```

1      ;BLDCMD
2
3      ;BUILD A COMMAND IN COMMAND PACKET
4
5      ;INPUTS:
6          R5 - CONTROLLER TABLE ADDRESS
7          R0 - COMMAND CODE
8
9      ;OUTPUTS:
10         R4 - ADDRESS OF HOST COMM AREA
11         COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL OTHER FIELDS CLEARED.
12         CMD REFERENCE NUMBER IN CONTROLLER TABLE INCREMENTED AND RESULT
13         IN COMMAND PACKET.
14         R0 - CONTENTS DESTROYED
15
16 016754      BLDCMD: PUSH <R1,R0>
17 016754      ;
18 016754      ;
19 016754      ;
20 016754      ;
21 016754      ;
22 016754      ;
23 016754      ;
24 016754      ;
25 016754      ;
26 016754      ;
27 016754      ;
28 016754      ;
29 016754      ;
30 016754      ;
31 016754      ;

```

016754	010146				MOV R1,-(SP)
016756	010046				MOV R0,-(SP)
016760	016504	000014		MOV C,RING(R5),R4	;GET ADDRESS OF HOST COMM AREA
016764	010400			MOV R4,R0	;COPY TO R0
016766	062700	000100		ADD #HC.CEV,R0	;COMPUTE ADDRESS OF COMMAND ENVELOPE
016772	012720	000060		MOV #HC.PSZ,(R0)	;LOAD PACKET LENGTH
016776	012701	001000		MOV #DUP,R1	;LOAD DIAG CIRCUIT IDENTIFIER
017002	022716	000031		CMR #OP.MMR,(SP)	;IF CODE IS MAINTENANCE WRITE
017006	001002			BNE BLDC0	; GET OTHER CIRCUIT IDENTIFIER
017010	012701	177777		MOV #DIAG,R1	
017014	010120			BLDC0: MOV R1,(R0)	;PUT IDENTIFIER INTO PACKET
017016	012701	000030		MOV #<HC.PSZ>/2,R1	;GET WORDS TO CLEAR
017022	005020			BLDC1: CLR (R0)	;CLEAR PACKET
017024	005301			DEC R1	
017026	001375			BNE BLDC1	
017030	012664	000114		POP HC.CPK.P.OPCD(R4)	;PUT OPCODE IN PACKET
017034	012601			POP R1	MOV (SP),HC.CPK.P.OPCD(R4)
017036	000207			RETURN	MOV (SP),R1

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 98
 GLOBAL SUBROUTINES SECTION

```

1      ;SNDCMD
2
3      ;SEND A COMMAND TO THE UDA.
4      ;MARK BOTH PACKETS AVAILABLE TO THE
5      ;UDA. SET COMMAND ISSUED BIT IN CONTROLLER TABLE AND INITIALIZE
6      ;TIMEOUT COUNTER.
7
8      ;INPUTS:
9      ;      R5 - CONTROLLER TABLE ADDRESS
10     ;OUTPUTS:
11     ;      R4 - ADDRESS OF HOST COMM AREA
12
13
14     017040      SNDCMD: PUSH <R0,R1>
15     017040      010046
16     017042      010146
17     017044      016504      000014
18     017050      005265      000050
19     017054      016564      000050      000104
20     017062      012764      140000      000006
21     017070      012764      100000      000012
22     017076      005775      000000
23     017102      052765      000004      000012
24     017110
25     017110      012601
26     017112      012600
27     017114      000207
28
29     MOV C.RING(R5),R4
30     INC C.REF(R5)
31     MOV C.REF(R5),HC.CPK+P.CRF(R4)
32     MOV #RG.OMN+RG.FLG,HC.HCT(R4)
33     TST B(R5)
34     BIS #CT.CMD,C.FLG(R5)
35     POP <R1,R0>
36
37     ;LOAD R4 WITH HOST COMM AREA ADDRESS
38     ;INCREMENT CMD REFERENCE NUMBER
39     ;PUT IN PACKET
40     ;MARK MESSAGE PACKET AVAILABLE
41     ;MARK COMMAND TO UDA
42     ;TELL UDA COMMAND IS THERE
43     ;MARK COMMAND ISSUED
44
45     MOV R0,-(SP)
46     MOV R1,-(SP)
47     MOV (SP)+,R1
48     MOV (SP)+,R0
49
50     RETURN

```

```

1      ;CLRBUF
2      ;
3      ;CLEAR THE SPECIFIED DATA BUFFER IN THE HOST COMM AREA
4      ;AND LOAD BUFFER DESCRIPTOR IN COMMAND PACKET TO THE BUFFER
5      ;
6      ;INPUTS:
7      ;   R5 - CONTROLLER TABLE ADDRESS
8      ;   R4 - ADDRESS OF HOST COMM AREA
9      ;   R0 - OFFSET INTO HOST COMM AREA TO DATA BUFFER
10     ;OUTPUTS:
11     ;   DATA BUFFER CLEARED
12     ;   COMMAND PACKET POINTING TO BUFFER
13     ;   BYTE COUNT SET TO SIZE OF BUFFER
14     ;   R4 - ADDRESS OF DATA BUFFER
15
16     CLRBUF: PUSH <R0,R1>
17     017116 010046
18     017116 010146
19     017122 060400
20     017124 010064 000124
21     017130 012764 000244 000120
22     017136 010004
23     017140 012701 000122
24     017144 005020
25     017146 005301
26     017150 001375
27     017152
28     017152 012601
29     017154 012600
30     017156 000207

      ADD R4,R0
      MOV RO,HC.CPK+P.UADR(R4)
      MOV #HC.BSZ,HC.CPK+P.BCNT(R4)
      MOV RO,R4
      MOV #HC.BSZ/2,R1
      CLR (R0)+
      DEC R1
      BNE CLRBFL
      POP <R1,R0>

      ;ADD START OF HOST COMM AREA TO OFFSET
      ;PUT BUFFER ADDRESS IN COMMAND PACKET
      ;PUT SIZE OF BUFFER IN COMMAND PACKET
      ;PUT BUFFER ADDRESS IN R4
      ;GET SIZE OF BUFFER IN WORDS
      ;CLEAR ALL THE WORDS

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

      RETURN

```

```

1      ;WAITMS
2
3      ;WAIT FOR UDA TO RESPOND WITH A MESSAGE PACKET
4
5      ;INPUTS:
6      ;      R5 - ADDRESS OF CONTROLLER TABLE
7      ;OUTPUTS:
8      ;      Z CLEAR IF NO ERROR
9      ;      Z SET IF ERROR, MESSAGE PRINTED
10
11     WAITMS: PUSH <R0,R1>
12     017160 010046                                MOV R0,-(SP)
13     017162 010146                                MOV R1,-(SP)
14     017164 012700 000036                        MOV #30.,R0      ;SET TIME OUT VALUE OF 30 SECONDS
15     017170 010501                                MOV R5,R1        ;POINT TO TIME OUT COUNTER
16     017172 062701 000036                        ADD #C.TO,R1
17     017176 004737 017352                        CALL SETTO
18     017202 011500                                1$: MOV (R5),R0    ;GET ADDRESS OF UDAIP REGISTER
19     017204 032765 000010 000012                BIT #CT.MSG,C.FLG(R5) ;LOOK IF INTERRUPT OCCURRED
20     017212 001030                                BNE 3$           ;BRANCH IF SO
21     017214 016001 000002                        MOV 2(R0),R1     ;LOOK AT UDASA REGISTER
22     017220 001034                                BNE 4$           ;BRANCH IF ERROR CODE PRESENT
23     017222                                BREAK
24     017224 104422                                TRAP            C$BRK
25     017224 005737 003206                        TST KW.CSR      ;SEE IF A CLOCK ON SYSTEM
26     017230 001764                                BEQ 1$
27     017232 023765 003220 000040                CMP KW.EL+2,C.TOM(R5) ;CHECK IF TIMEOUT HAS HAPPENED
28     017240 101005                                BHI 2$
29     017242 001357                                BNE 1$
30     017244 023765 003216 000036                CMP KW.EL,C.TO(R5)
31     017252 103753                                BLO 1$
32     017254 017254 104455                                2$: ERRODF $6.,ERR036
33     017256 000044                                TRAP            C$ERDF
34     017260 000000                                .WORD          36
35     017262 012426                                .WORD          0
36     017264                                .WORD          ERR036
37     017264                                POP <R1,R0>
38     017264 012601                                MOV (SP)-,R1
39     017266 012600                                MOV (SP)+,R0
40     017270 000264                                SEZ
41     017272 000207                                RETURN
    
```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 101
 GLOBAL SUBROUTINES SECTION

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

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 102
GLOBAL SUBROUTINES SECTION

```
1          ;NXMI
2          ;
3          ;NON-EXISTANT MEMORY SERVICE ROUTINE
4          ;
5          ;INPUTS:
6          ;      NXMAD SET TO ZERO
7          ;OUTPUTS:
8          ;      NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
9
10         BGNSRV NXMI
11
12         017332 012737 177777 002172          MOV #-1,NXMAD
13
14         017340          ENDSRV
15         017340          L10031:
16         017340 000002          RTI
```

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 103
 GLOBAL SUBROUTINES SECTION

```

1      ;UDASRV
2      ;
3      ;UDA INTERRUPT SERVICE ROUTINE. MARKS UDA CONTROLLER TABLE THAT AN
4      ;INTERRUPT HAS BEEN RECEIVED.
5      ;
6      ;THIS ROUTINE IS CALLED BY A [JSR RO,UDASRV] INSTRUCTION FROM WITHIN
7      ;THE CONTROLLER TABLE. THE PC STORED IN RO IS THE ADDRESS OF THE C.FLG
8      ;WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
9      ;OF RO FOLLOWED BY THE INTERRUPTED PC AND PS.
10     ;
11     ;INPUTS:
12     ;   RO - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
13     ;   STACK - SAVED CONTENTS OF RO
14     ;OUTPUTS:
15     ;   CT.CMD CLEARED AND CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
16     ;   RO - RESTORED FROM STACK
17
18     BGNSRV UDASRV
19     017342      052710  000010      BIS #CT.MSG,(RO)      ;SET CT.MSG
20     017346      012600                      POP RO              ;RESTORE RO
21     017350
22     017350
23     017350      000002                      MOV (SP)+,RO
24
25     ENDSRV
26
27     L10032:
28     RTI
  
```

CZUDKO UDASOA/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 104
GLOBAL SUBROUTINES SECTION

```

1      ;SETTO
2      ;
3      ;SET TIMEOUT COUNTER TO SOME NUMBER OF SECONDS FROM CURRENT TIME.
4      ;
5      ;INPUTS:
6      ;      R0 - NUMBER OF SECONDS FOR TIMEOUT
7      ;      R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT
8      ;OUTPUTS:
9      ;      R0 - CONTENTS DESTROYED
10     ;      R1 - INCREMENTED BY 2
11
12     ;COMPUTE CLOCK TICKS TIL TIMEOUT
13
14     017352      ;SETTO:  PUSH <R2,R3>
15     017352      010246
16     017354      010346
17     017356      005002
18     017360      013703      003214
19     017364      006200
20     017366      103001
21     017370      060302
22     017372      006303
23     017374      005700
24     017376      001372
25
26     017400      013700      003216
27     017404      013703      003220
28     017410      020037      003216
29     017414      001371
30
31     ;GET CURRENT TIME
32
33     017416      060200
34     017420      005503
35
36     ;ADD TIME TIL TIMEOUT
37
38     017422      010021
39     017424      010311
40
41     017426
42     017426      012603
43     017430      012602
44     017432      000207

```

```

;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
SETTO:  PUSH <R2,R3>
;
;CLEAR PRODUCT
;GET MULTIPLICAND
;SHIFT MULTIPLIER TO RIGHT
;IF A ONE BIT SHIFTED OUT
;ADD MULTIPLICAND TO PRODUCT
;DOUBLE THE MULTIPLICAND
;CONTINUE UNTIL MULTIPLIER IS ZERO
;
;GET TIME
;IF CHANGED DURING RETRIEVAL
;GET IT AGAIN
;
;ADD
;
;PUT RESULT IN STORAGE
MOV R2,-(SP)
MOV R3,-(SP)
CLR R2
MOV KW.HZ,R3
SET00: ASR R0
;CLEAR PRODUCT
;GET MULTIPLICAND
;SHIFT MULTIPLIER TO RIGHT
;IF A ONE BIT SHIFTED OUT
;ADD MULTIPLICAND TO PRODUCT
;DOUBLE THE MULTIPLICAND
;CONTINUE UNTIL MULTIPLIER IS ZERO
BCC SET01
ADD R3,R2
SET01: ASL R3
TST R0
BNE SET00
;
;GET CURRENT TIME
SET02: MOV KW.EL,R0
MOV KW.EL+2,R3
;GET TIME
;IF CHANGED DURING RETRIEVAL
;GET IT AGAIN
CMP R0,KW.EL
BNE SET02
;
;ADD TIME TIL TIMEOUT
ADD R2,R0
ADC R3
;ADD
;
;PUT RESULT IN STORAGE
MOV R0,(R1)+
MOV R3,(R1)
POP <R3,R2>
MOV (SP)+,R3
MOV (SP)+,R2
RETURN

```

```

1      ;UDAINI
2
3      ;FUNCTIONAL DESCRIPTION:
4      ;   SUBROUTINE TO INITIALIZE A UDA AND BRING IT ON-LINE.
5      ;   ALL STEPS ARE CHECKED. AN ERROR MESSAGE IS REPORTED IF ANY ERROR
6      ;   DETECTED.
7
8      ;INPUTS:
9      ;   R5 - ADDRESS OF CONTROLLER TABLE.
10     ;IMPLICIT INPUTS:
11     ;   C.RING(R5) - ADDRESS GIVEN TO UDA AS START OF RING BUFFER.
12     ;   LENGTH OF RING STRUCTURE IS ONE ENTRY EACH.
13     ;OUTPUTS:
14     ;   CONDITION Z - SET IF ANY ERROR REPORTED. CLEAR IF NO ERROR.
15     ;   R4 - ADDRESS OF UDAIP REGISTER IN UDA
16     ;   R5 - UNCHANGED.
17
18     ;FILL MOST COMMUNICATION AREA WITH ALL ONES
19
20     UDAINI: MOV C.RING(R5),R2                ;GET FIRST ADDRESS OF RING BUFFER
21             MOV @<MC.RSZ*2+MC.ISZ>/2,R3    ;GET SIZE OF RING BUFFER
22     UDAI1L: MOV @-1,(R2)*                ;WRITE ONES TO BUFFER
23             DEC R3                        ;COUNT THE WORDS IN BUFFER
24             BGT UDAI1L                    ;LOOP UNTIL ENTIRE BUFFER WRITTEN
25
26     ;DO THE INITIALIZATION
27
28     UDAINI: CALL UDAIST                    ;DO FIRST THREE STEPS
29             BCS UDAIEX                    ;GET OUT IF UDA MICROCODE REPORTED FAILURE
30             MOV (R3)*,2(R4)              ;WRITE NEXT WORD TO UDASA REGISTER
31             MOV @200.,R3                 ;GET TRY COUNTER
32     UDAI1A: MOV 2(R4),R2                  ;LOOK AT UDASA
33             BEQ UDAI1C
34             DEC R3
35             BNE UDAI1A
36             ERROF 24,,ERR024
37
38             TRAP C$ERDF
39             .WORD 24
40             .WORD 0
41             .WORD ERR024
42
43     UDAINI: BR UDAIEX
44     UDAI1C: MOV R2,2(R4)                  ;WRITE 0 TO UDASA (PURGE)
45             MOV (R4),R2                  ;READ FROM UDAIP (POLL)
46             CALL UDARSP                  ;WAIT FOR STEP OR ERROR BIT
47             BCS UDAIEX                    ;GET OUT IF UDA MICROCODE REPORTED FAILURE
48             BIC @+C<SA.CNT>,R2          ;CLEAR OTHER BITS
49             ASR R2                        ;MOVE TO RIGHT OF REGISTER
50             ASR R2
51             ASR R2
52             ASR R2
53             CMP R2,#6                      ;CONTROLLER MODEL MUST BE 6
54             BEQ UDAI2
55             CMP R2,#13                    ; OR 13
56             BEQ UDAI2
57             ERROF 14,,ERR014            ;REPORT CONTROLLER NEEDS NEW REVISION
58
59             TRAP C$ERDF
60             .WORD 14

```

017566 000000
017570 012120
52 017572 000441

BR UDAIEX

.WORD 0
.WORD ERRO14

CZUDKO UDASA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 106
GLOBAL SUBROUTINES SECTION

```

1          ;CHECK MOST COMMUNICATION AREA FOR ALL ZEROS
2
3 017574 016502 000014      UDAI2:  MOV C.RING(R5),R2      ;GET FIRST ADDRESS OF RING BUFFER
4 017600 010201              MOV R2,R1          ;SAVE FOR ERROR MESSAGE
5 017602 012703 000006      MOV #<HC.RSZ*2+HC.ISZ>/2,R3 ;GET SIZE OF RING BUFFER
6 017606 005722              UDAI2L: TST (R2).        ;CHECK WORD IN BUFFER
7 017610 001003              BNE UDAI2E         ;GO TO ERROR REPORTER IF NOT ZERO
8 017612 005303              DEC R3            ;COUNT THE WORDS IN BUFFER
9 017614 003374              BGT UDAI2L         ;LOOP UNTIL ALL WORDS CHECKED
10 017616 000405              BR UDAI3
11
12 017620              UDAI2E: ERDF 23,,ERR02I      ;REPORT BUFFER NOT CLEARED
13 017620 104455              TRAP C1ERDF
14 017622 000027              .WORD 23
15 017624 000000              .WORD 0
16 017626 012236              .WORD ERR023
17 017630 000422              BR UDAIEX
18
19          ;SEND GO BIT i0 UDASA REGISTER TO END INITIALIZATION
20
21          UDAI3:
22          MOV #SA.GO,R0
23          MOV R0,2(R4)      ;SEND TO UDA
24          MOV C.RING(R5),R1
25          MOV R1,HC.MSG(R1)
26          ADD #HC.MPK,HC.MSG(R1)
27          MOV R1,HC.CMD(R1)
28          ADD #HC.CPK,HC.CMD(R1)
29          CLZ              ;CLEAR Z AS NO ERROR INDICATION
30          RETURN
31
32          ;ERROR RETURN
33
34          UDAIEX: SEZ
35          RETURN          ;SET Z TO INDICATE ERROR OCCURRED

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct 84 10:07 Page 107
GLOBAL SUBROUTINES SECTION

```

1      ;UDAIST
2      ;
3      ;START THE INITIALIZATION PROCESS ON THE SELECTED UDA.
4      ;STOP BEFORE WRITING THE THIRD WORD SO UDA DOES NOT
5      ;ATTEMPT ANY UNIBUS TRANSFERS.
6      ;
7      ;INPUTS:
8      ;      R5 - ADDRESS OF CONTROLLER TABLE
9
10     ;LOAD TABLE OF DATA TO SEND TO UDASA REGISTER
11
12     UDAIST: BREAK
13     017702      017702      104422      TRAP      C$BRK
14     017704      017704      010146      PUSH R1
15     017706      017706      016504      000004      MOV C.VEC(R5),R4
16     017712      017712      042704      177000      AND CT.VEC,R4
17     017716      017716      006204      ASR R4
18     017722      017722      052704      100000      ASR R4
19     017726      017726      010437      020120      BIS #SA.STP,R4
20     017732      017732      016537      000014      020124      MOV R4,UDAID1
21     017740      017740      062737      000004      020124      MOV C.RING(R5),UDAID2
22     ;
23     ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER
24
25     017746      017746      016504      000000      MOV C.UADR(R5),R4
26     017752      017752      005037      002172      CLR NXMAD
27     017756      017756      012746      000340      SETVEC #4,#NXMI,#PRI07
28     017762      017762      012746      017332      MOV #PRI07,-(SP)
29     017766      017766      012746      000004      MOV #NXMI,-(SP)
30     017772      017772      012746      000003      MOV #4,-(SP)
31     017776      017776      104437      TRAP      C$SVEC
32     020000      020000      062706      000010      MOV #3,-(SP)
33     020004      020004      005764      000002      TRAP      C$SVEC
34     020010      020010      005014      ADD      #10,SP
35     020012      020012      012700      000004      TST 2(R4)
36     020016      020016      104436      CLR (R4)
37     020020      020020      005737      002172      ;ACCESS UDASA REGISTER
38     020024      020024      001406      CLRVEC #4
39     020026      020026      104455      ;WRITE TO UDAIP
40     020030      020030      000024      ;GIVE UP THE VECTOR
41     020032      020032      000000      MOV #4,R0
42     020034      020034      012134      TRAP      C$CVEC
43     020036      020036      000261      ;SEE IF A MEMORY ERROR OCCURRED
44     020040      020040      000424      ERRDF 20,,ERR020
45
46     TRAP      C$ERDF
47     .WORD      20
48     .WORD      0
49     .WORD      ERR020
50
51     SEC
52     BR UDAISE

```

CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 108
 GLOBAL SUBROUTINES SECTION

```

1          ;SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
2
3 020042 012737 004000 020356 UDASIG: MOV #SA.S1,UDARSD      ;STORE RESPONSE MASK
4 020050 012703 020116          MOV #UDAIDT,R3          ;AND INDEX TO TABLE
5
6          ;WAIT FOR AND CHECK RESPONSE DATA
7
8 020054 004737 020220          UDASL: CALL UDARSP          ;WAIT FOR STEP OR ERROR BITS
9 020060 103414          BCS UDAISE          ;EXIT IF ERROR
10 020062 004733          CALL @R3          ;CALL RESPONSE CHECKER FOR STEP
11 020064 103412          BCS UDAISE          ;GET OUT IF ERROR
12 020066 006337 020356          ASL UDARSD          ;SHIFT TO NEXT STEP BIT
13 020072 032737 040000 020356  BIT #SA.S4,UDARSD      ;CHECK IF NOW AT STEP 4
14 020100 001003          BNE UDAISX          ;GET OUT IF SO
15 020102 012364 000002          MOV (R3)+,2(R4)      ;WRITE DATA TO UDASA REGISTER
16 020106 000762          BR UDASL          ;STAY IN LOOP
17
18 020110 000241          UDASX: CLC          ;CLEAR CARRY FOR NO ERROR INDICATION
19 020112          UDAISE: POP R1
          ;
20 020114 000207          RETURN          MOV (SP)+,R1
  
```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 109
GLOBAL SUBROUTINES SECTION

```

1      ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION
2
3 020116 020132      UDAIDT: .WORD UDAIR1      ;FIRST WORD RESPONSE CHECK ROUTINE
4 020120 000000      UDAID1: .WORD 0      ;FIRST WORD TO SEND TO UDASA
5 020122 020144      ;.WORD UDAIR2      ;SECOND WORD RESPONSE CHECK ROUTINE
6 020124 000000      UDAID2: .WORD 0      ;SECOND WORD TO SEND TO UDASA
7 020126 020164      ;.WORD UDAIR3      ;THIRD WORD RESPONSE CHECK ROUTINE
8 020130 100000      UDAID3: .WORD SA.TST      ;THIRD WORD TO SEND TO UDASA
9
10     ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
11     ;CHECK FOR PROPER CONTROLLER TYPE
12
13 020132 012701 004400      UDAIR1: MOV #SA.S1+SA.DI,R1      ;SET STEP ONE BIT
14 020136 042702 001140      BIC #<SA.QB+SA.MP+SA.SM>,R2      ;MASK OFF UNWANTED BITS
15 020142 000416      BR UDAIRC      ;NOW COMPARE
16
17     ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
18     ;CHECK FOR ECHO OF INTI AND VECTOR
19
20 020144 013701 020120      UDAIR2: MOV UDAID1,R1      ;GET WORD SENT TO UDASA
21 020150 000301      SWAB R1      ;GET HIGH 8 BITS
22 020152 042701 177400      BIC #177400,R1
23 020156 052701 010000      BIS #SA.S2,R1      ;SET STEP 2 BIT
24 020162 000406      BR UDAIRC      ;NOW COMPARE
25
26     ;RESPONSE CHECK FOR THIRD WORD FROM UDASA
27     ;CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
28
29 020164 013701 020120      UDAIR3: MOV UDAID1,R1      ;GET WORD SENT TO UDASA
30 020170 042701 177400      BIC #177400,R1      ;JUST LOW 8 BITS
31 020174 052701 020000      BIS #SA.S3,R1      ;SET STEP 3 BIT
32
33     ;COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
34
35 020200 020102      UDAIRC: CMP R1,R2      ;COMPARE THE DATA
36 020202 001405      BEQ UDAIRX      ;EXIT IF COMPARED CORRECTLY
37 020204      ERRODF 25,ERR025      ;REPORT ERROR
38      ;
39 020214 000261      ;
39 020216 000207      UDAIRX: RETURN

```

```

TRAP      C1ERDF
.WORD     25
.WORD     0
.WORD     ERR025

```


CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 111
 GLOBAL SUBROUTINES SECTION

```

1          ;CHECK IF ERROR BIT SET
2
3 020330 016402 000002      UDARS2: MOV 2(R4),R2          ;GET REGISTER CONTENTS
4 020334 100006              BPL UDARSX              ;EXIT IF ERROR NOT SET
5 020336                      ERROF 21,,ERRO21        ;REPORT ERROR INFO
   020336 104455              TRAP C#ERDF
   020340 000025              .WORD 21
   020342 000000              .WORD 0
   020344 012146              .WORD ERRO21
6 020346 000261      UDARSE: SEC
7 020350 000207          RETURN
8
9          ;NORMAL EXIT
10
11 020352 000241      UDARSX: CLC          ;CLEAR CARRY AS NO ERROR INDICATION
12 020354 000207          RETURN
13
14          ;LOCATION FOR STEP BIT MASK
15
16 020356 000000      UDARSD: .WORD 0      ;LOAD BY CALLING ROUTINE
  
```

```

1          ;KW11I
2          ;
3          ;CLOCK INTERRUPT SERVICE ROUTINE
4
5 020360    BGNSRV KW11I
6 020360    062737 000001 003216          ADD #1,KW.EL          KW11I::
7 020366    005537 003220          ADC KW.EL+2          ;COUNT THE INTERRUPT
8 020372    012777 000105 162606          MOV #KWOUT.,#KW.CSR  ;RESTART THE CLOCK
9 020400    ENDSRV
          L10033:
          RTI
020400    000002

```

```

1      ;RNTIME
2
3      ;PRINT RUNTIME
4
5      ;INPUTS:
6      ;       KW.EL - CONTAINS ELAPSED TIME
7      ;       KW.HZ - HERTZ OF CLOCK
8      ;OUTPUTS:
9      ;       IF CLOCK ON SYSTEM:
10     ;       "   RUNTIME HH:MM:SS " PRINTED
11     ;       IF NO CLOCK: ONE SPACE IS PRINTED
12
13 020402 005737 003206      RNTIME: TST KW.CSR           ;CHECK IF A CLOCK PRESENT
14 020406 001465           BEQ RNTIMX           ;BRANCH IF NOT
15 020410           PUSH <R0,R3,R4,R5>
16     020410 010046           MOV R0,-(SP)
17     020412 010346           MOV R3,-(SP)
18     020414 010446           MOV R4,-(SP)
19     020416 010546           MOV R5,-(SP)
20 020420 013703 003216      MOV KW.EL,R3           ;GET ELAPSED TIME
21 020424 013704 003220      MOV KW.EL+2,R4
22 020430 013700 003214      MOV KW.HZ,R0           ;GET SPEED OF CLOCK
23 020434 004737 016516      CALL DIVIDE           ;COMPUTE SECONDS OF ELAPSED TIME
24 020440 012700 000074      MOV #60.,R0           ;NOW DIVIDE BY 60
25 020444 004737 016516      CALL DIVIDE           ; TO COMPUTE MINUTES
26 020450           PUSH R5           ;SAVE REMAINDER AS SECONDS
27     020450 010546           MOV R5,-(SP)
28 020452 004737 016516      CALL DIVIDE           ;DIVIDE BY 60 AGAIN
29 020456           PNT RNTIM,R3       ;PRINT HOURS
30     020456 010346           MOV R3,-(SP)
31     020460 004137 016450      JSR R1,LPNT           ;PRINT MINUTES
32     020464 003712           .WORD RNTIM
33     020466 000002           .WORD PNT.CT
34 020470 020527 000011      CMP R5,#9             ;IF MINUTES 9 OR LESS
35 020474 003004           BGT 1$
36 020476           PRINT #'0         ;PRINT A LEADING ZERO
37     020476 112700 000060      MOVB #'0,R0
38     020502 004737 016240      CALL CPNT
39 28 020506           1$: PNT RNTIM1,R5       ;NOW PRINT MINUTES
40     020506 010546           MOV R5,-(SP)
41     020510 004137 016450      JSR R1,LPNT
42     020514 003735           .WORD RNTIM1
43     020516 000002           .WORD PNT.CT
44 29 020520           POP R5           ;GET SECONDS
45     020520 012605           MOV (SP)+,R5
46 30 020522 020527 000011      CMP R5,#9             ;IF 9 OR LESS
47 31 020526 003004           BGT 2$
48 32 020530           PRINT #'0         ;PRINT A LEADING ZERO
49     020530 112700 000060      MOVB #'0,R0
50     020534 004737 016240      CALL CPNT
51 33 020540           2$: PNT RNTIM2,R5       ;NOW PRINT SECONDS
52     020540 010546           MOV R5,-(SP)
53     020542 004137 016150      JSR R1,LPNT
54     020546 003743           .WORD RNTIM2
55     020550 000002           .WORD PNT.CT
56 34 020552           POP <R5,R4,R3,R0>       ;HOURS IN R3
57     020552 012605           MOV (SP)+,R5

```

020554	012604	
020556	012603	
020560	012600	
35 020562		
020562	112700	000040
020566	004737	016240
36 020572	000207	

RNTIMX: PRINT '<' >

,PRINT A SPACE

RETURN

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

MOVB #' ,R0
CALL CPNT

```

1 020574          DATE:  GMANID DATEQ,DATEI,A,-1,1,11.,YES      ;GET DATE
   020574 104443
   020576 000406
   020600 003270
   020602 000152
   020604 003544
   020606 177777
   020610 000001
   020612 000013
   020614
2 020614 012705 003270      MOV #DATEI,R5      ;GET POINTER TO ANSWER
3 020620 121527 000060      CMPB (R5),#0
4 020624 103443          BLO DERR
5 020626 122527 000071      DAY:  CMPB (R5)+,#9
6 020632 101040          BHI DERR
7 020634 121527 000055      CMPB (R5),#'-
8 020640 001406          BEQ DAS1
9 020642 121527 000060      CMPB (R5),#0
10 020646 103432         BLO DERR
11 020650 122527 000071      CMPB (R5)+,#9
12 020654 101027         BHI DERR
13 020656 122527 000055      DAS1:  CMPB (R5)+,#'-
14 020662 001024         BNE DERR
15 020664 012704 000014      MOV #12.,R4      ;GET NUMBER OF MONTH
16 020670 012703 003345      MON1:  MOV #MONTHS,R3  ;GET POINTER TO MONTH NAMES
17 020674 005000          CLR R0
18 020676 121523         CMPB (R5),(R3)+
19 020700 001401         BEQ MON2
20 020702 005200          INC R0
21 020704 126523 000001      MON2:  CMPB 1(R5),(R3)+
22 020710 001401         BEQ MON3
23 020712 005200          INC R0
24 020714 126523 000002      MON3:  CMPB 2(R5),(R3)+
25 020720 001401         BEQ MON4
26 020722 005200          INC R0
27 020724 005700          MON4:  TST R0
28 020726 001407         BEQ MON5
29 020730 005304         DEC R4
30 020732 001360         BNE MON1
31 020734          DERR:  PNTF DATEX
   020734 004137 016412
   020740 011671
   020742 000000
32 020744 000713
33 020746 012701 003304      MON5:  BR DATE
   020752 010403          MOV #DATEQ,R1      ;GET POINTER TO DATE FOR FORMATTER
34 020754 020327 000012      MOV R4,R3          ;GET COPY OF MONTH NUMBER
35 020760 103404          CMP R3,#10.        ; IF 10 OR GREATER
36 020762 112721 000061      BLO MON6
37 020766 162703 000012      MOVB #'1,(R1)+    ;PUT A "1" IN OUTPUT
38 020772 062703 000060      SUB #10.,R3
39 020776 110321 000055      MON6:  ADD #'0,R3          ;CONVERT MONTH NUMBER TO ASCII
40 021000 112721 000055      MOVB R3,(R1)+     ;PUT A NUMBER IN OUTPUT
41 021004 062704 003410      MOVB #'-(R1)+     ;PUT A "-" IN OUTPUT
42 021010 012703 003270      ADD #DAYS-1,R4    ;GET POINTER TO DAYS IN MONTH
43 021014 005000          ;INDEXED BY NUMBER OF MONTH
44 021014 012703 003270      MOV #DATEI,R3     ;GET POINTER TO DATE INPUT
45 021014 005000          CLR R0

```

```

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

```

```

C:GMAN
10000$
DATEI
T:CODE
DATEQ
-1
T:LOLIM
T:HILIM

```

10000\$:

```

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

```

CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 114 1
GLOBAL SUBROUTINES SECTION

```

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

```

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01 Oct 84 10:07 Page 114 2
GLOBAL SUBROUTINES SECTION

103
104 021232 000000
105
106 021234

BRSAY: .WORD 0
ENDMOD

!DEFAULT BR LEVEL AND VECTOR

```

1          .SBTTL  PROTECTION TABLE
2
3 021234          BGNMOD
4
5          ;**
6          ; THIS TABLE IS USED BY THE RUNTIME SERVICES
7          ; TO PROTECT THE LOAD MEDIA.
8          ;
9
10         BGNPROT
11
12         021234 177777          LIPROT::
13         021236 177777          ;OFFSET INTO P-TABLE FOR CSR ADDRESS
14         021240 177777          ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
15
16         021242          ENDPROT
17

```


CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 116 1
INITIALIZE SECTION

```

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

```

CZUDKO UDAS0A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 116 2
INITIALIZE SECTION

```

96 021434          BCOMPLETE      2#
   021434 103407
97 021436 005037 003206      CLR      KW.CSR          ;IF NEITHER, CLEAR CSR STORAGE WORD
   021436 005037 003206      PNTF     NOCLOCK
98 021442          BCS          2#
   021442 004137 016412          JSR R1,LPNTF
   021446 004110          .WORD NOCLOCK
   021450 000000          .WORD PNT.CT
99 021452 000426          BR      3#
100
101 021454 012037 003206      2#:   MOV      (R0)+,KW.CSR      ;STORE DATA RETURNED
102 021460 012037 003210      MOV      (R0)+,KW.BRL
103 021464 012037 003212      MOV      (R0)+,KW.VEC
104 021470 012037 003214      MOV      (R0)+,KW.HZ
105
106 021474          SETVEC     KW.VEC,#KW11I,#PRI07      ;SETUP KW11 VECTOR ADDRESS
   021474 012746 000340          MOV      @PRI07,-(SP)
   021500 012746 020360          MOV      @KW11I,-(SP)
   021504 013746 003212          MOV      KW.VEC,-(SP)
   021510 012746 000003          MOV      @3,-(SP)
   021514 104437          TRAP     C$VEC
   021516 062706 000010          ADD     @10,SP
107 021522 012777 000105 161456      MOV      @KWOUT.,@KW.CSR      ;START THE CLOCK
108 021530 004737 012750      3#:   CALL     RESET          ;RESET ALL CONTROLLERS
109 021534          MEMORY     FFREE          ;RESET START OF FREE MEMORY
   021534 104431          TRAP     C$MEM
   021536 010037 002140          MOV      R0,FFREE
110 021542 017737 160372 002142      MOV      @FFREE,F$SIZE      ;RESET SIZE OF FREE MEMORY
111
112 ;
113 ;   ALLOCATE DRIVE TABLES TO MEMORY
114 ;
115
116 021550 013737 002140 003202      INIT2: MOV      FFREE,DTABS      ;STORE START OF DRIVE TABLES AND
117 021556 005077 161420          CLR      @DTABS          ;MARK ZERO END.
118 021562 013700 002012          MOV      L$UNIT,R0      ;GET NUMBER OF LOGICAL UNITS TO RUN.
119 021566 012701 000001          MOV      @1,R1          ;GET INITIAL SIZE OF DRIVE TABLE AND
120 021572 062701 000015      1#:   ADD     @<D.SIZE>/2,R1      ;ACCUMULATE DRIVE TABLE SIZE.
121 021576 005300          DEC     NO              ;SEE IF ANY MORE LOGICAL UNITS.
122 021600 001374          BNE     1#             ;BRANCH IF NOT, ELSE
123 021602 004737 012706          CALL    ALOCM          ;ALLOCATE ALL DRIVE TABLES TO MEMORY.
124
125
126 ;
127 ;   INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
128 ;
129
130 021606 013737 002140 002150      INIT3: MOV      FFREE,CTABS      ; STORE START OF CONTROLLER TABLES AND
131 021614 005077 160330          CLR      @CTABS          ; MARK ZEROS END.
132 021620 005037 002152          CLR      CTRLNS          ; CLEAR CONTROLLER COUNT
133 021624 012701 003434          MOV      @IPADRS,R1      ; R1 -> IP ADDRESS
134 021630 012702 000010          MOV      @8.,R2          ; GET MAXIMUM # OF CONTROLLERS
135 021634 005021      1#:   CLR      (R1)+          ; CLEAR ENTRY
136 021636 005302          DEC     R2              ; DONE?
137 021640 001375          BNE     1#             ; IF NOT, BRANCH
138
139 ;
140 ;   BUILD CONTROLLER TABLES

```

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 116 3
INITIALIZE SECTION

```

141      ;
142      ;
143 021642 005005      INIT4: CLR      R5      ;CLEAR CUSTOMER DATA FLAG
144 021644 005002      CLR      R2      ;START WITH LOGICAL UNIT 0
145 021646 012757 005160 021232      MOV      #5160,BRSV ; SAVE DEFAULT FOR BR LEVEL & VECTOR
146 021647 010200      18:  GPHARD R2,R0 ; GET POINTER TO IT'S P-TABLE
      021654 010200      MOV      R2,R0
      021656 104442      TRAP    C:GPHRD
147 021660      BNCOMplete 168 ; BRANCH TO 168 IF NOT AVAILABLE
      021660 103104      BCC     168
148 021662 013703 002150      MOV      CTABS,R3 ; GET ADDRESS OF 1ST CONTROLLER TABLE
149 021666 005713      28:  TST      (R3) ; CHECK IF ANY MORE TABLES
150 021670 001405      BEQ      68 ; BUILD NEW TABLE IF FOUND ZERO WORD
151 021672 021013      CMP      (R0),(R3) ; CHECK IF SAME CSR ADDRESS.
152 021674      ASSUME  C.UADR EQ 0
153 021674      ASSUME  MO.UBA EQ 0
154 021674 001444      BEQ      118 ; BRANCH IF SO
155
156
157 021676 062703 000052      58:  ADD      #C.SIZE,R3 ;POINT TO BEGINNING OF NEXT CONTROLLER
158 021702 000771      BR       28 ;TABLE IN MEMORY.
159
160      ;
161      ; BUILD NEW CONTROLLER TABLE
162      ;
163
164 021704 012704 003434      68:  MOV      #IPADRS,R4 ;GET BEGINNING OF IP ADDRESS TABLE
165 021710 020427 003444      78:  CMP      R4,#IPADRS*8. ;SEE IF END OF IP ADDRESS TABLE.
166 021714 101004      BHI      98 ;BRANCH IF SO, ELSE
167 021716 005724      TST      (R4). ;DID WE FIND AN OPEN ENTRY ?
168 021720 001401      BEQ      88 ;BRANCH IF SO, ELSE
169 021722 000772      BR       78 ;LOOK AGAIN.
170
171 021724 011044      88:  MOV      (R0),-(R4) ;TAKE CSR ADDRESS FROM P-TABLE
172      ;AND STORE IT IN THE IP ADDRESS TABLE.
173 021726 012701 000025      98:  MOV      #<C.SIZE>/2,R1 ;GET # OF ENTRIES IN CONTROLLER TABLE
174 021732 004737 012706      CALL    ALOCM ;AND ALLOCATE A TABLE TO MEMORY.
175      ; R0 => 1ST WORD P-TABLE
176      ; R1 => 1ST WORD IN CONTROLLER TABLE
177 021736 011021      MOV      (R0),(R1). ; STORE CSR ADDRESS AND
178 021740 010221      MOV      R2,(R1). ; UNIT NUMBER IN THE CONTROLLER TABLE.
179 021742 013704 021232      MOV      BRSV,R4 ; GET DEFAULT VECTOR & BR LEVEL
180 021746 162704 000004      SUB      #4,R4 ; GET NEXT VECTOR
181 021752 010437 021232      MOV      R4,BRSV ; SAVE NEXT VECTOR
182 021756 010421      MOV      R4,(R1). ; STORE IT IN THE CONTROLLER TABLE.
183 021760 012721 004037      MOV      #4037,(R1). ; THE 'JSR R0' INSTRUCTION AND
184 021764 012721 017342      MOV      #UDASRV,(R1). ; THE ADDRESS OF THE INTERRUPT SERVICE
185      ; ROUTINE IN THE CONTROLLER TABLE.
186 021770 012704 000020      MOV      #<C.SIZE-C.FLG>/2,R4 ;GET # OF ENTRIES TO END OF TABLE.
187 021774 005021      108: CLR      (R1). ;CLEAR REST OF TABLE AND
188 021776 005304      DEC      R4 ;ADD ZERO WORD AT END.
189 022000 002375      BGE     108 ;LOOP TIL ALL CLEARED
190 022002 005237 002152      INC     CTRLRS ;KEEP TRACK OF CONTROLLER COUNT
191
192      ;
193      ; BUILD DRIVE TABLES
194      ;

```

CZUDKO UDA50A/KDA50-G FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 116 4
INITIALIZE SECTION

```

195
196 022006 013701 003202      114:  MOV    DTABS,R1      ;GET ADDRESS OF CURRENT DRIVE TABLE
197 022012 062703 000016      ADD    #C.DRO,R3      ; INDEX TO 1ST DRIVE IN TABLE
198 022016 012704 000010      MOV    #8.,R4         ; GET # OF DRIVES PER CONTROLLER
199 022022 005713              124:  TST    (R3)          ; ANY ENTRY TO DRIVE TABLE,
200 022024 001411              BEQ    144            ; BRANCH IF NOT, ELSE
201 022026 026033 000002      CMP    MO.LDR(R0),B(R3) ; COMPARE DRIVE NUMBER IN DRIVE TABLE.
202 022032 001002              BNE    134            ; BRANCH IF DIFFERENT, ELSE
203 022034 000137 022140      JMP    MLDREX         ; FOUND TWO P-TABLES WITH SAME DRIVE.
204
205 022040 005304              134:  DEC    R4             ; COUNT DRIVES
206 022042 001367              BNE    124            ; IF FOUR DRIVE TABLES ALREADY EXIST,
207 022044 000137 022156      JMP    TOOMER         ; THEN REPORT ERROR
208
209 022050 010113              144:  MOV    R1,(R3)        ; STORE ADDRESS OF DRIVE TABLE IN
210                                ; CONTROLLER TABLE.
211 022052 016021 000002      MOV    MO.LDR(R0),(R1) ; STORE DRIVE NUMBER AND
212 022056 010221              MOV    R2,(R1)        ; LOGICAL UNIT NUMBER IN DRIVE TABLE.
213
214 022060 062737 000032 003202  ADD    #D.SIZE,DTABS  ; NEXT DRIVE TABLE ADDRESS AND
215 022066 005077 161110      CLR    #DTABS         ; MARK ZERO END.
216 022072 005202              164:  INC    R2             ; INCREMENT LOGICAL UNIT NUMBER
217 022074 020237 002012      CMP    R2,L#UNIT      ; CHECK IF GOT ALL TABLES
218 022100 002665              BLT    144            ; IF NOT, GO BACK FOR NEXT, ELSE
219 022102 012701 000001      MOV    #1,R1          ; GET 1 WORD TO TERMINATE ALL CONTROLLER
220 022106 004737 012706      CALL  ALOCM           ; TABLES AND ALLOCATE IT TO MEMORY.
221
222                                ;
223                                ; SAVE CURRENT PARAMETERS TO FREE MEMORY SO EACH TEST CAN USE ALL OF IT
224                                ;
225
226 022112 013737 002140 002144  INIT6: MOV    FFREE,FMEM  ; SAVE START ADDRESS
227 022120 013737 002142 002146  MOV    FSIZE,FMEMS    ; SAVE SIZE
228
229                                ;
230                                ; EXIT INITIALIZE SECTION
231                                ;
232
233 022126              INITXX: SETPRI #PRI00  ; SET RUNNING PRIORITY TO ZERO
234 022126 012700 000000      MOV    #PRI00,R0      ;
235 022132 104441              TRAP   C#SPRI
236
237                                ;
238 022134              EXIT  INIT
239 022134 104432              TRAP   C#EXIT
240 022136 000036              .WORD L10035-.
241
242                                ; TWO P-TABLES FOR SAME DRIVE
243 022140 013705 003242      MLDREX: MOV TEMP,R5   ; GET CONTROLLER ADDRESS
244 022144 104454              ERRSF 2.,ERR002
245 022146 000002              TRAP   C#ERSF
246 022150 000000              .WORD 2
247 022152 012010              .WORD 0
248                                .WORD ERR002
249
250                                DOCLN
251                                TRAP   C#DOCLN
252
253                                ; MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER

```

243							
244	022156	013705	003242	TOOMER: MOV TEMP,R5			
245	022162			ERRSF 3.,ERR003			
	022162	104454				TRAP	C#ERSF
	022164	000003				.WORD	3
	022166	000000				.WORD	0
	022170	012026				.WORD	ERR003
246	022172			DOCLN		TRAP	C#DCLN
	022172	104444					
247							
248							
249	022174			ENDINIT			
	022174					L10035:	
	022174	104411				TRAP	C#INIT

.SBTTL AUTODROP SECTION

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

1
2
3
4
5
6
7
8
9
10 022176
 022176
11
12 022176
 022176
 022176 104461

BGNAUTO

L#AUTO::

ENDAUTO

L10036: TRAP C#AUTO

.SBTTL CLEANUP CODING SECTION

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

```

1
2
3
4
5
6
7
8 022200          BGNCLN
   022200
9
10 022200 004737 013206          CALL CLOSE
11 022204          SETVEC #4,#NXMI,#PRI07          ;CLOSE DATA FILE
   022204 012746 000340          MOV          #PRI07,-(SP)
   022210 012746 017332          MOV          #NXMI,-(SP)
   022214 012746 000004          MOV          #4,-(SP)
   022220 012746 000003          MOV          #3,-(SP)
   022224 104437          TRAP          C$SVEC
   022226 062706 000010          ADD          #10,SP
12 022232 012703 000010          MOV          #8,R3          ; R3 = COUNTER OF ENTRIES
13 022236 012704 003434          MOV          #IPADRS,R4      ; R4 -> IP ADDRESS
14 022242 005714          1$: TST          (R4)          ; IS THERE AN ENTRY?
15 022244 001403          BEQ          2$          ; IF NOT, DONE
16 022246 005034          CLR          8(R4)+        ; INIT UDA
17 022250 005303          DEC          R3          ; MAKE SURE WE DO NOT EXTEND OVER AREA
18 022252 001373          BNE          1$          ; IF NOT DONE, BRANCH
19 022254          2$: CLRVEC #4
   022254 012700 000004          MOV          #4,R0
   022260 104436          TRAP          C$CVEC
20
21 022262          ENDCLN
   022262          L10037:
   022262 104412          TRAP          C$CLEAN
22
23 022264          ENDMOD

```

TEST 1: DUP PROGRAM DRIVER

```

1          .SBTTL TEST 1: DUP PROGRAM DRIVER
2
3 022264      BGNMOD
4
5 022264      BGNTST
6 022264      PNTX WNSTRT          ;PRINT WARNING MESSAGE      T1::
   022264      004137  016432      JSR R1,LPNTX
   022270      004565              .WORD WNSTRT
   022272      000000              .WORD PNT.CT
7 022274      MANUAL              ;SEE IF MANUAL INTERVENTION ALLOWED
   022274      104450              TRAP      C#MANI
8 022276      BNCOMPLETE T1MODE    ;IF NOT, JUST RUN THE PROGRAM
   022276      103020              BCC      T1MODE
9 022300      CLR TEMP              ;CLEAR WORD FOR ANSWER
10 022304      GMANIL WNKUES,TEMP,1,YES ;ASK IF STILL WANT TO RUN
   022304      104443              TRAP      C#GMAN
   022306      000404              BR       10000$
   022310      003242              .WORD    TEMP
   022312      000130              .WORD    T$CODE
   022314      003630              .WORD    WNKUES
   022316      000001              .WORD    1
   022320              10000$:
11 022320      005737  003242      TST TEMP          ;LOOK AT ANSWER
12 022324      001417              BEQ T1QUIT      ;IF NO, QUIT NOW
13 022326      005737  003304      TST DATEO        ;SEE IF ALREADY ASKED FOR DATE
14 022332      001002              BNE T1MODE
15 022334      004737  020574      CALL DATE        ;IF NOT, GET IT NOW
16
17 022340      032737  000003  003200  T1MODE: BIT #50,FMT,MODE
18 022346      001164              BNE T1FMT
19 022350      022350      104450              MANUAL
   022350      104450              TRAP      C#MANI
20 022352      BCOMPLETE T1GO
   022352      103406              BCS      T1GO
21 022354      ERRSF 10,,ERR010
   022354      104454              TRAP      C#ERSF
   022356      000012              .WORD    10
   022360      000000              .WORD    0
   022362      012106              .WORD    ERR010
22 022364      T1QUIT: EXIT TST
   022364      104432              TRAP      C#EXIT
   022366      000362              .WORD    L10040-.
23 022370      032737  000010  003200  T1GO: BIT #50,STR,MODE
24 022376      001435              BEQ T1CNS
25 022400      023727  002012  000001      CMP L$UNIT,#1
26 022406      001406              BEQ T1RST
27 022410      ERRSF 9,,ERR009
   022410      104454              TRAP      C#ERSF
   022412      000011              .WORD    9
   022414      000000              .WORD    0
   022416      012074              .WORD    ERR009
28 022420      EXIT TST
   022420      104432              TRAP      C#EXIT
   022422      000326              .WORD    L10040 .
29
30 022424      T1RST: PNTF FILNAM

```


TEST 1: DUP PROGRAM DRIVER

```

59 022624 105710          T1SER6: TSTB (R0)
60 022626 001416          BEQ T1SER8
61 022630 122120          CMPB (R1),.(R0).
62 022632 001774          BEQ T1SER6
63 022634 103413          BLO T1SER8
64 022636 102746 003320    T1SER7: PRINTF #SERNX,#HIGHEST
    022636 012746 003320    MOV #HIGHEST,-(SP)
    022642 012746 011601    MOV #SERNX,-(SP)
    022646 012746 000002    MOV #2,-(SP)
    022652 010600          MOV SP,R0
    022654 104417          TRAP C#PNTF
    022656 062706 000006    ADD #6,SP
65 022662 000724          BR T1SER3
66 022664 062702 000004    T1SER8: ADD #D.SERN,R2 ;PUT ANSWER INTO DRIVE TABLE
67 022670 012701 003242    MOV #TEMP,R1
68 022674 112122          T1SER9: MOVB (R1),.(R2).
69 022676 001376          BNE T1SER9
70 022700 005303          DEC R3
71 022702 001402          BEQ T1SERN
72 022704 005724          TST (R4).
73 022706 000700          BR T1SER2
74 022710 062705 000052    T1SERN: ADD #C.SIZE,R5
75 022714 005715          TST (R5)
76 022716 001267          BNE T1SER1
77 022720 013737 002150 002154 T1FMT: MOV CTABS,TSTTAB ;GET FIRST TABLE ADDRESS
78 022726 013701 002152    MOV CTRLRS,R1 ;RUN DM PROGRAM ON ALL CONTROLLERS
79 022732 004737 013074    CALL RUNDM ; RUN ALL CONTROLLERS OF ONE TYPE AT ONCE
80 022736 001402          BEQ 6$
81 022740 004737 013224    CALL RESPDM
82 022744 104432          6$: EXIT TST
    022744 000002          TRAP C$EXIT
    022746 000002          .WORD L10040
83 022750 104401          ENDTST
    022750 104401          L10040: TRAP C$ETST
84 022752          ENDMOD
    
```

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

.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

000011

.WORD L10041-L#HARD/2
L#HARD::

;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:

TABLE

;START A TEBLE DEFINITION

ITEM NO.UBA 2
ITEM NO.LDR 2
END

; UNIBUS ADDRESS
; DRIVE NUMBER

022752
022752
022754
022754
022754
022754
022754
022754

```

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

```

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01 Oct 84 10:07 Page 122
SOFTWARE PARAMETER CODING SECTION

```

1      .SBTTL  SOFTWARE PARAMETER CODING SECTION
2
3
4      ;
5      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
6      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
9      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10     ; WITH THE OPERATOR.
11     ;--
12     BGNSFT
13
14     ;FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:
15
16     TABLE                                ;START A TABLE DEFINITION
17
18     ITEM SO.BIT      2                    ;YES/NO ANSWERS
19     SO.FM1 = BIT0    ; REFORMAT MODE
20     SO.FM2 = BIT1    ; (AGAIN)
21     SO.FMT = SO.FM1+SO.FM2
22     SO.CNS = BIT2   ; RECONSTRUCT MODE
23     SO.STR = BIT3   ; RESTORE MODE
24
25     END

```

023030
023030 000022
023032

.WORD L10042 L#SOFT/2
L#SOFT::

13
14
15
16 023032
17
18 023032
19
20
21
22
23
24
25 023032

END

1	023032				GPRML S.FMT,SO.BIT,SO.FM1,YES	;REFORMAT?			
	023032	000130					.WORD	T#CODE	
	023034	023247					.WORD	S.FMT	
	023036	000001					.WORD	SO.FM1	
2	023040				XFERT SWEND				
	023040	017024					.WORD	T#CODE	
3	023042				GPRML S.NRF,SO.BIT,SO.FM2,YES	;AGAIN - REFORMAT?			
	023042	000130					.WORD	T#CODE	
	023044	023076					.WORD	S.NRF	
	023046	000002					.WORD	SO.FM2	
4	023050				XFERT SWEND				
	023050	013024					.WORD	T#CODE	
5	023052				GPRML S.CNS,SO.BIT,SO.CNS,YES	;RECONSTRUCT			
	023052	000130					.WORD	T#CODE	
	023054	023326					.WORD	S.CNS	
	023056	000004					.WORD	SO.CNS	
6	023060				XFERT SWEND				
	023060	007024					.WORD	T#CODE	
7	023062				GPRML S.RST,SO.BIT,SO.STR,YES	;RESTORE?			
	023062	000130					.WORD	T#CODE	
	023064	023371					.WORD	S.RST	
	023066	000010					.WORD	SO.STR	
8	023070				XFERT SWEND				
	023070	003024					.WORD	T#CODE	
9	023072				DISPLAY S.NOF	;WARNING			
	023072	000003					.WORD	T#CODE	
	023074	023512					.WORD	S.NOF	
10	023076				SWEND: ENDSFT				
	023076						.EVEN		
								L10042:	
11									
12	023076	015	012		S.NRF:	.BYTE 15,12			
13	023100	116	117	124		.ASCII\NOT USING EXISTING INFORMATION WILL DESTROY THE FACTORY BAD SECTOR\			
14	023202	015	012			.BYTE 15,12			
15	023204	111	116	106		.ASCII\INFORMATION ON THE DISKS.\			
16	023235	015	012			.BYTE 15,12			
17	023237	101	107	101		.ASCII\AGAIN - \			
18	023247	122	105	106	S.FMT:	.ASCII\REFORMAT USING EXISTING BAD SECTOR INFORMATION\			
19	023326	122	105	103	S.CNS:	.ASCII\RECONSTRUCT BAD SECTOR INFORMATION\			
20	023371	104	117	040	S.RST:	.ASCII\DO YOU HAVE A FILE ON THE SYSTEM LOAD DEVICE\			
21	023445	015	012			.BYTE 15,12			
22	023447	040	103	117		.ASCII\ CONTAINING BAD SECTOR INFORMATION\			
23	023512	131	117	125	S.NOF:	.ASCII\YOU CANNOT PROCEED WITHOUT SUCH A FILE.\			
24	023562	122	105	123		.ASCII\RESTART PROGRAM AND SELECT 10 REFORMAT OR RECONSTRUCT DISK.\			
25	023656	000				.BYTE 0			
26						.EVEN			
27									
28						.DSABL AMA			
29	000000					.PSECT END			

1
2
3 000000
4
5
6
7
8 000120

000120 000134'
000122 000004
000124
9
10 000124

.SBTTL PATCH AREA

\$PATCH::
.REPT 40.
.WORD 0
.ENDR

LASTAD

L\$LAST::

ENDMOD

.EVEN
.WORD T\$FREE
.WORD T\$SIZE

```

1 000124          BGNSETUP          1
2
3 000124          BGNPTAB
  000124 000000
  000126 000002
  000130
4
5 000130 172150   .WORD 172150
6 000132 000000   .WORD 0.
7
8 000134          ENDPTAB
  000134
9
10 000134         ENDSETUP
11
12
13
14
15
16
17
18          000001   .END

```

```

          .WORD 0
          .WORD L10045-./2 1
L10043:
; UNIBUS ADDRESS
; LOGICAL DRIVE NUMBER
L10045:

```

Errors detected: 0

*** Assembler statistics

```

Work file reads: 589
Work file writes: 519
Size of work file: 29208 Words ( 115 Pages)
Size of core pool: 14336 Words ( 56 Pages)
Operating system: RT-11 (Under RTEH-11)

```

```

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

```

CZUDKO UDAS0A/KDAS0-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 5-1
 Cross reference table (CREF V05.01)

#PATCH	124-30								
ADR	30-100								
ALOCM	56-160	57-14	116-123	116-174	116-220				
ASSEMB	26-8	26-8							
BAS	50-140	83-5	83-5	83-5	84-5	84-5			
BASL2	50-120	84-5							
BASL3	50-130								
BASLN	50-160	83-5	84-5						
BASNO	50-110	83-5	84-5						
BIT0	30-100	122-19							
BIT00	30-10	30-100							
BIT01	30-10	30-100							
BIT02	30-10	30-100							
BIT03	30-10	30-100							
BIT04	30-10	30-100							
BIT05	30-10	30-100							
BIT06	30-10	30-100							
BIT07	30-10	30-100							
BIT08	30-10	30-100							
BIT09	30-10	30-100							
BIT1	30-100	33-26	41-22	122-20					
BIT10	30-100								
BIT11	30-100								
BIT12	30-100								
BIT13	30-100								
BIT14	30-100								
BIT15	30-100	41-15	42-12	59-27	62-20	70-28	75-15	89-29	
BIT2	30-100	33-27	41-23	122-22					
BIT3	30-100	33-28	41-24	122-23					
BIT4	30-100	33-29	41-26						
BIT5	30-100	41-29							
BIT6	30-100	41-30							
BIT7	30-100	41-32							
BIT8	30-100								
BIT9	30-100								
BLDC0	97-22	97-240							
BLDC1	97-260	97-28							
BLDCHD	61-49	64-14	64-44	95-2	97-150				
BOE	30-100								
BRSAY	114-1040	116-145*	116-179	116-181*					
C#AU	26-80								
C#AUTO	26-80	117-12							
C#BRK	26-80	58-12	61-8	100-21	107-12	110-27			
C#BSEG	26-80								
C#BSUB	26-80								
C#CEFG	26-80								
C#CLCK	26-80	116-93	116-95						
C#CLEA	26-80	118-21							
C#CLOS	26-80	60-12	73-19						
C#CLP1	26-80								
C#CVEC	26-80	58-22	107-30	118-19					
C#DCLN	26-80	55-8	58-30	116-69	116-240	116-246			
C#DODU	26-80								
C#DRPT	26-80								
C#DU	26-80								
C#EDIT	26-80	26-34							

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01 Oct-84 10:07 Page 5-2
Cross reference table (CREF V05.01)

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

CZUDKO UDASOA/KDASO-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 5-5
 Cross reference table (CREF V05.01)

EF.RES	30-100	116-53												
EF.SEX	37-300													
EF.STA	30-100	116-47												
EOFMSG	53-56	53-590												
ERR001	53-140													
ERR002	53-180	116-239												
ERR003	53-220	116-245												
ERR004	53-260	55-7												
ERR008	53-300													
ERR009	53-340	119-27												
ERR010	53-380	119-21												
ERR014	53-420	105-51												
ERR020	53-460	58-28	107-33											
ERR021	53-500	111-5												
ERR022	53-620	110-36												
ERR023	53-670	106-12												
ERR024	53-790	105-36												
ERR025	53-830	109-37												
ERR030	53-870	61-24												
ERR031	53-910	62-36												
ERR032	53-950	64-8												
ERR033	53-1000	63-21												
ERR034	53-1040	96-3												
ERR036	53-1080	100-29												
ERR037	53-1120	101-5												
ERR100	53-1160	68-22												
ERR101	53-1200	70-32												
ERR21A	53-54	53-570												
ERR23A	53-700	53-75												
ERR23B	53-71	53-730												
ERR23C	53-760													
ERRC	77-36	77-43	81-90											
ERRCHR	46-420	91-50	91-12	91-14	91-16	91-18								
ERRD	77-45	81-210												
ERRME1	50-80	77-41	79-23	82-4										
ERRNL	50-40	91-10												
ERRONE	50-30	91-7												
ERRNSZ	79-18	80-110												
ERRRTB	79-21	80-30	80-11											
ERRTRM	65-7	72-140												
EVL	30-100													
F1AU	26-80													
F1AUTO	26-80	117-10	117-12											
F1BGN	26-80	26-26	29-16	30-3	53-14	53-18	53-22	53-26	53-30	53-34	53-38	53-42	53-46	53-50
	53-62	53-67	53-79	53-83	53-87	53-91	53-95	53-100	53-104	53-108	53-112	53-116	53-120	102-10
	103-18	112-5	114-106	115-3	115-10	116-45	116-235	117-10	118-8	118-23	119-3	119-5	119-22	119-28
	119-82	119-83	119-84	120-3	120-14	122-12	123-9	123-9	124-10	125-1	125-3	125-3	125-8	125-10
F1CLEA	26-80	118-8	118-21											
F1DU	26-80													
F1END	26-8	26-8	26-8	26-8	26-8	26-8	26-8	26-8	26-8	26-8	26-8	26-8	26-8	26-8
	26-8	26-8	26-80	26-26	29-16	30-3	53-16	53-20	53-24	53-28	53-32	53-36	53-40	53-44
	53-48	53-60	53-65	53-77	53-81	53-85	53-89	53-93	53-98	53-102	53-106	53-110	53-114	53-118
	53-122	102-14	103-21	112-9	114-106	115-3	116-235	116-249	117-12	118-21	118-23	119-3	119-5	119-5
	119-5	119-22	119-28	119-82	119-83	119-83	119-84	120-3	121-3	123-10	124-10	125-1	125-3	125-8
	125-10													
F1HARD	26-80	120-14	121-3	123-2	123-4	123-6	123-8							

CZUDKO LDA50A/KDA50-Q FORMATTER MACRO V05 J1b Monday 01 Oct-84 10:07 Page 5 9
 Cross reference table (CREF V05.01)

L\$LOAD	26-340				
L\$LUN	26-340	59-240	61-120	75-140	
L\$PREV	26-340				
L\$NAME	26-340				
L\$PRIO	26-340				
L\$PROT	26-34	115 100			
L\$PRT	26-340				
L\$PEPP	26-340				
L\$REV	26-340				
L\$SOFT	26-34	122-12	122-120		
L\$SPC	26-340				
L\$SOP	26-340				
L\$SPTP	26-340				
L\$STA	26-340				
L\$SM	26-34	29-10	29-100		
L\$TEST	26-340				
L\$TIML	26-340				
L\$UNIT	26-340	76-13	116-118	116-217	119-25
L10000	28-10	28-140			
L10001	29-10	29-140			
L10002	53-160				
L10003	53-200				
L10004	53-240				
L10005	53-280				
L10006	53-320				
L10007	53-360				
L10010	53-400				
L10011	53-440				
L10012	53-480				
L10013	53-600				
L10014	53-650				
L10015	53-770				
L10016	53-810				
L10017	53-850				
L10020	53-890				
L10021	53-930				
L10022	53-980				
L10023	53-1020				
L10024	53-1060				
L10025	53-1100				
L10026	53-1140				
L10027	53-1180				
L10030	53-1220				
L10031	102-140				
L10032	103-210				
L10033	112-90				
L10035	116-235	116-2490			
L10036	117-120				
L10037	118-210				
L10040	119-22	119-28	119-82	119-830	
L10041	120-14	121-30			
L10042	122-12	123-100			
L10043	125-30				
L10045	125-3	125-80			
LDDM	59-220	59-34			
LDNEXT	59-26	59-30	59-320		

CZUDKO UDA50A/KDA50-Q FORMATTER MACRO V05.01b Monday 01-Oct-84 10:07 Page 5-11
 Cross reference table (CREF V05.01)

O:BGNS	26-80	26-320	26-34			
O:DU	26-80	26-34				
O:ERRT	26-80	26-34				
O:GNSW	26-80	26-320	26-34			
O:POIN	26-80	26-32	26-320	26-320	26-320	26-34
O:SETU	26-80	26-320	26-34	124-8		
OP.ABO	36-30					
OP.ACC	36-40					
OP.AVA	36-220					
OP.AVL	36-50					
OP.CCD	36-60					
OP.CMP	36-70					
OP.DUP	36-230					
OP.ELP	36-300					
OP.END	36-200	63-5	63-8	64-58		
OP.ERS	36-80					
OP.ESP	36-290	95-1				
OP.FLU	36-90					
OP.GCS	36-100					
OP.GDS	36-270	61-48	64-58			
OP.GSS	36-280					
OP.GUS	36-110					
OP.HRD	36-180					
OP.HMR	36-190	97-21				
OP.ONL	36-120					
OP.RD	36-130					
OP.RLC	36-250					
OP.RPL	36-140					
OP.RSD	36-320	63-8	64-43			
OP.SCC	36-150					
OP.SEX	36-210					
OP.SHC	36-240					
OP.SSD	36-310	63-5	64-13			
OP.SUC	36-160					
OP.MR	36-170					
OSTRE	77-35	77-42	77-470			
OSTRING	77-340	77-46	85-6	86-6	87-6	92-17
P.BCNT	38-210	39-90	64-11	64-33*	95-4*	99-19*
P.BUFF	38-220					
P.CMST	39-140					
P.CNCL	39-480					
P.CNTF	38-400	39-460				
P.CNTI	39-490					
P.CPSP	38-340					
P.CRF	38-170	39-40	63-19	98-17*		
P.CTMD	39-470					
P.CYL	39-260					
P.DEXT	39-520					
P.DFLG	39-530	64-60				
P.DMDT	38-500					
P.OPI	39-540	64-65	64-67	64-72	64-73	
P.DTO	39-550					
P.ELGF	38-320					
P.FBBK	39-100					
P.FLGS	39-70					
P.GRP	39-250					

T#PC	125-10	125-10												
T#PRO	115-100													
T#PTA	125-10	125-3	125-30											
T#SOF	122-12	122-120	123-10											
T#SRV	102-100	102-14	103-180	103-21	112-50	112-9								
T#SM	29-10	29-100	29-14											
T#TES	119-50	119-22	119-28	119-82	119-83									
T#ARGC	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-340	26-340	26-340
	26-340	26-340	26-340	91-12	91-12	91-12	91-120	91-120	91-14	91-14	91-14	91-140	91-140	91-16
	91-16	91-16	91-160	91-160	91-18	91-18	91-18	91-180	91-180	119-64	119-64	119-64	119-640	119-640
T#CODE	114-1	114-1	114-1	114-10	114-10	114-10	119-10	119-10	119-10	119-100	119-100	119-100	119-31	119-31
	119-31	119-310	119-310	119-310	119-44	119-44	119-44	119-440	119-440	119-440	121-1	121-1	121-1	121-10
	121-10	121-10	121-2	121-2	121-2	121-2	121-20	121-20	121-20	123-1	123-1	123-1	123-10	123-10
	123-2	123-2	123-2	123-2	123-2	123-2	123-20	123-20	123-20	123-20	123-20	123-3	123-3	123-30
	123-30	123-30	123-4	123-4	123-4	123-4	123-4	123-4	123-40	123-40	123-40	123-40	123-5	123-5
	123-5	123-50	123-50	123-50	123-6	123-6	123-6	123-6	123-6	123-6	123-60	123-60	123-60	123-60
	123-7	123-7	123-7	123-70	123-70	123-70	123-8	123-8	123-8	123-8	123-8	123-8	123-80	123-80
	123-80	123-80	123-9	123-90										
T#ERRN	26-80	55-7	55-70	58-28	58-280	61-24	61-240	62-36	62-360	63-21	63-210	64-8	64-80	68-22
	68-220	70-32	70-320	96-3	96-30	100-29	100-290	101-5	101-50	105-36	105-360	105-51	105-510	106-12
	106-120	107-33	107-330	109-37	109-370	110-36	110-360	111-5	111-50	116-239	116-2390	116-245	116-2450	119-21
	119-210	119-27	119-270											
T#EXCP	114-1	114-10	119-31	119-310	119-44	119-440	121-1	121-10	121-2	121-20				
T#FLAG	116-235	116-235	116-2350	116-2350	119-22	119-22	119-220	119-220	119-28	119-28	119-280	119-280	119-82	119-82
	119-820	119-820												
T#FREE	124-8	125-100												
T#GMAN	26-80	114-1	114-10	114-10	119-31	119-310	119-310	119-44	119-440	119-440				
T#HILI	114-1	114-10	119-31	119-310	119-44	119-440	121-1	121-10	121-2	121-20				
T#LAST	26-80	124-80	125-1											
T#LCLI	114-1	114-10	119-31	119-310	119-44	119-440	121-1	121-10	121-2	121-20				
T#LSYM	26-8	26-80	28-14	29-14	53-16	53-20	53-24	53-28	53-32	53-36	53-40	53-44	53-48	53-60
	53-65	53-77	53-81	53-85	53-89	53-93	53-98	53-102	53-106	53-110	53-114	53-118	53-122	102-14
	103-21	112-9	116-249	117-12	118-21	119-83	121-3	123-10						
T#LTNO	124-80													
T#NEST	26-80	26-26	26-26	26-260	28-10	28-10	28-100	28-14	28-14	28-14	28-140	29-10	29-10	29-100
	29-14	29-14	29-14	29-140	29-16	29-16	29-16	29-160	30-3	30-3	30-30	53-14	53-14	53-140
	53-16	53-16	53-16	53-160	53-18	53-18	53-180	53-20	53-20	53-20	53-200	53-22	53-22	53-220
	53-24	53-24	53-24	53-240	53-26	53-26	53-260	53-28	53-28	53-28	53-280	53-30	53-30	53-300
	53-32	53-32	53-32	53-320	53-34	53-34	53-340	53-36	53-36	53-36	53-360	53-38	53-38	53-380
	53-40	53-40	53-40	53-400	53-42	53-42	53-420	53-44	53-44	53-44	53-440	53-46	53-46	53-460
	53-48	53-48	53-48	53-480	53-50	53-50	53-500	53-60	53-60	53-60	53-600	53-62	53-62	53-620
	53-65	53-65	53-65	53-650	53-67	53-67	53-670	53-77	53-77	53-77	53-770	53-79	53-79	53-790
	53-81	53-81	53-81	53-810	53-83	53-83	53-830	53-85	53-85	53-85	53-850	53-87	53-87	53-870
	53-89	53-89	53-89	53-890	53-91	53-91	53-910	53-93	53-93	53-93	53-930	53-95	53-95	53-950
	53-98	53-98	53-98	53-980	53-100	53-100	53-1000	53-102	53-102	53-102	53-1020	53-104	53-104	53-1040
	53-106	53-106	53-106	53-1060	53-108	53-108	53-1080	53-110	53-110	53-110	53-1100	53-112	53-112	53-1120
	53-114	53-114	53-114	53-1140	53-116	53-116	53-1160	53-118	53-118	53-118	53-1180	53-120	53-120	53-1200
	53-122	53-122	53-122	53-1220	102-10	102-10	102-100	102-14	102-14	102-14	102-140	103-18	103-18	103-180
	103-21	103-21	103-21	103-210	112-5	112-5	112-50	112-9	112-9	112-9	112-90	114-106	114-106	114-1060
	114-1060	115-3	115-3	115-30	115-10	115-10	115-100	115-16	115-16	115-16	115-160	116-45	116-45	116-450
	116-249	116-249	116-249	116-2490	117-10	117-10	117-100	117-12	117-12	117-12	117-120	118-8	118-8	118-80
	118-21	118-21	118-21	118-210	118-23	118-23	118-23	118-230	119-3	119-3	119-30	119-5	119-5	119-50
	119-83	119-83	119-83	119-830	119-84	119-84	119-84	119-840	120-3	120-3	120-30	120-14	120-14	120-140
	121-3	121-3	121-3	121-30	122-12	122-12	122-120	123-2	123-4	123-6	123-8	123-10	123-10	123-10
	123-100	124-10	124-10	124-10	124-100									
T#NSO	26-260	29-16	30-30	114-106	115-30	118-23	119-30	119-84	120-30	124-10				

UF.WPS	38-110					
UFREEZ	46-210	59-35*	62-3	62-13*	70-21	70-23*
URNING	46-180	59-16*	59-31*	59-40	62 32*	
URUN	46-170	59-15*	59-20	61-7		
WATMS	95-8	100-110				
WNGUES	49-60	119-10				
WNSTOP	50-180	70-40				
WNSTRY	50-210	119-6				
X1ALWA	26-80					
X1FALS	26-80					
X1OFFS	26-80	123-2	123-4	123-6	123-8	
X1TRUE	26-80	123-2	123-4	123-6	123-8	
X1	51-50	53-15				
X10	51-130	53-39				
X100	51-410	53-117				
X101	51-420	53-121				
X14	51-140	53-43				
X1A	51-10	53-15				
X2	51-60	53-19				
X20	51-170	53-47				
X21	51-200	53-55				
X21A	51-220	53-58				
X22	51-230	53-64				
X23A	51-250	53-68				
X23B	51-290	53-72				
X24	51-300	53-80				
X25	51-320	53-84				
X2A	51-20	53-19				
X3	51-70	53-23				
X30	51-350	53-88				
X31	51-360	53-92				
X32	51-370	53-96				
X36	51-380	53-109				
X37	51-400	53-113				
X3A	51-30	53-23				
X4	51-80	53-27				
X8	51-100	53-31				
X8A	51-40	53-31				
X9	51-110	53-35				
XFRU	52-90	53-76	86-5			
XMSG1	52 10	53-137				
XMSG2	52-20	53-141				
XPKT1	52-30	53-124				
XPKT2	52-70	53-130				
XSA	52-80	87-5				
YEAR19	47-310	114-92				
YEAR20	47-320	114-95				
YFR1	114-690	114-82				
YFR2	114-70	114-72	114-830			
YFR3	114-86	114-920				
YFR4	114-94	114-960	114-99			
YFR5	114-91	114-97	114-1000	114-101		

GPHARD	116-146													
GPRMA	121-1													
GPRND	114-1	114-10	119-31	119-310	119-44	119-440	121-2							
GPRML	119-10	119-100	123-1	123-3	123-5	123-7								
HEADER	26-34													
ITEM	31-240	41-12	41-13	41-16	41-19	41-20	41-21	41-33	41-34	41-35	41-36	41-37	41-38	41-39
	41-40	41-41	41-42	41-43	41-44	41-45	41-46	42-9	42-10	42-13	120-20	120-21	122-18	
LASTAD	124-8													
MIBYTE	26-34	26-34	26-34	26-340										
MICHEC	116-235	116-2350	119-22	119-220	119-28	119-280	119-82	119-820						
MICNT0	114-1	114-10	119-10	119-100	119-31	119-310	119-44	119-440	121-1	121-10	121-2	121-20	123-1	123-10
	123-3	123-30	123-5	123-50	123-7	123-70								
MICOUN	91-12	91-120	91-14	91-140	91-16	91-160	91-18	91-180	119-64	119-640				
MIDATA	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-340	26-340	48-12	48-120	48-16
	48-160													
MIDECR	28-14	28-140	29-14	29-140	29-16	29-160	53-16	53-160	53-20	53-200	53-24	53-240	53-28	53-280
	53-32	53-320	53-36	53-360	53-40	53-400	53-44	53-440	53-48	53-480	53-60	53-600	53-65	53-650
	53-77	53-770	53-81	53-810	53-85	53-850	53-89	53-890	53-93	53-930	53-98	53-980	53-102	53-1020
	53-106	53-1060	53-110	53-1100	53-114	53-1140	53-118	53-1180	53-122	53-1220	102-14	102-140	103-21	103-210
	112-9	112-90	114-106	114-1060	115-16	115-160	116-249	116-2490	117-12	117-120	118-21	118-210	118-23	118-230
	119-83	119-830	119-84	119-840	121-3	121-30	123-10	123-100	124-10	124-100	125-3	125-30		
MIDEFA	114-1	114-10	119-10	119-100	119-31	119-310	119-44	119-440	121-1	121-10	121-2	121-20	123-1	123-10
	123-3	123-30	123-5	123-50	123-7	123-70								
MIENDE	28-140	29-140	29-160	53-160	53-200	53-240	53-280	53-320	53-360	53-400	53-440	53-480	53-600	53-650
	53-770	53-810	53-850	53-890	53-930	53-980	53-1020	53-1060	53-1100	53-1140	53-1180	53-1220	102-140	103-210
	112-90	114-1060	116-2490	117-120	118-210	118-230	119-830	119-840	121-30	123-100	124-100			
MISERRI	55-7	55-70	58-28	58-280	61-24	61-240	62-36	62-360	63-21	63-210	64-8	64-80	68-22	68-220
	0-32	70-320	96-3	96-30	100-29	100-290	101-5	101-50	105-36	105-360	105-51	105-510	106-12	106-120
	107-33	107-330	109-37	109-370	110-36	110-360	111-5	111-50	116-239	116-2390	116-245	116-2450	119-21	119-210
	119-27	119-270												
MSEXCP	114-1	114-1	114-10	119-31	119-31	119-310	119-44	119-44	119-440	121-1	121-1	121-10	121-2	121-2
	121-20													
MSEXIT	116-235	116-2350	119-22	119-220	119-28	119-280	119-82	119-820						
MSEXSE	116-2350	119-220	119-280	119-820										
MSEXJ	116-2350	119-220	119-280	119-820										
MIGEN	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34
	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-34	26-340	26-340	26-340	26-340
	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340
	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340	26-340
	26-340	26-340	26-340	26-340	26-340	26-340	27-9	27-90	28-10	28-10	28-100	28-100	28-14	28-140
	29-10	29-10	29-100	29-100	29-14	29-140	48-12	48-120	48-16	48-160	53-14	53-140	53-16	53-160
	53-18	53-180	53-20	53-200	53-22	53-220	53-24	53-240	53-26	53-260	53-28	53-280	53-30	53-300
	53-32	53-320	53-34	53-340	53-36	53-360	53-38	53-380	53-40	53-400	53-42	53-420	53-44	53-440
	53-46	53-460	53-48	53-480	53-50	53-500	53-60	53-600	53-62	53-620	53-65	53-650	53-67	53-670
	53-77	53-770	53-79	53-790	53-81	53-810	53-83	53-830	53-85	53-850	53-87	53-870	53-89	53-890
	53-91	53-910	53-93	53-930	53-95	53-950	53-98	53-980	53-100	53-1000	53-102	53-1020	53-104	53-1040
	53-106	53-1060	53-108	53-1080	53-110	53-1100	53-112	53-1120	53-114	53-1140	53-116	53-1160	53-118	53-1180
	53-120	53-1200	53-122	53-1220	102-10	102-100	102-14	102-140	103-18	103-180	103-21	103-210	112-5	112-50
	112-9	112-90	114-1	114-10	115-10	115-100	116-45	116-450	116-249	116-2490	117-10	117-100	117-12	117-120
	118-8	118-80	118-21	118-210	119-5	119-50	119-10	119-100	119-31	119-310	119-44	119-440	119-83	119-830
	120-14	120-140	121-3	121-30	122-12	122-120	123-10	123-100	124-8	124-80	125-3	125-30	125-8	125-80
MIGENB	114-1	114-10	119-10	119-100	119-31	119-310	119-44	119-440						
MIGETS	28-14	28-140	29-14	29-140	29-16	29-160	53-16	53-160	53-20	53-200	53-24	53-240	53-28	53-280

	116-235	116-239	116-239	116-239	116-239	116-239	116-239	116-239	116-239	116-240	116-240	116-245	116-245	
	116-245	116-245	116-245	116-245	116-245	116-245	116-245	116-246	116-246	116-249	116-249	117-12	117-12	118-11
	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-19	118-19	118-19
	118-19	118-21	118-21	119-7	119-7	119-7	119-8	119-8	119-10	119-10	119-10	119-10	119-10	119-10
	119-10	119-10	119-10	119-19	119-19	119-19	119-20	119-20	119-21	119-21	119-21	119-21	119-21	119-21
	119-21	119-21	119-22	119-22	119-22	119-22	119-27	119-27	119-27	119-27	119-27	119-27	119-27	119-27
	119-27	119-28	119-28	119-28	119-28	119-31	119-31	119-31	119-31	119-31	119-31	119-31	119-31	119-31
	119-31	119-31	119-31	119-32	119-32	119-32	119-32	119-44	119-44	119-44	119-44	119-44	119-44	119-44
	119-44	119-44	119-44	119-44	119-44	119-64	119-64	119-64	119-64	119-64	119-64	119-64	119-64	119-64
	119-64	119-64	119-82	119-82	119-82	119-82	119-83	119-83	120-14	120-14	121-1	121-1	121-1	121-1
	121-1	121-2	121-2	121-2	121-2	121-2	121-2	121-3	121-3	122-12	122-12	123-1	123-1	123-1
	123-1	123-2	123-2	123-3	123-3	123-3	123-3	123-3	123-4	123-4	123-5	123-5	123-5	123-5
	123-6	123-7	123-7	123-7	123-7	123-7	123-8	123-8	123-9	123-9	123-9	123-9	123-10	123-10
	124-8	124-8	124-8	125-3	125-3	125-3	125-3							124-8
M#GNLS	114-1	114-1	119-10	119-10	119-31	119-31	119-44	119-44						
M#GNTA	28-14	28-14	29-14	29-14	53-16	53-16	53-20	53-20	53-24	53-24	53-28	53-28	53-32	53-32
	53-36	53-36	53-40	53-40	53-44	53-44	53-48	53-48	53-60	53-60	53-65	53-65	53-77	53-77
	53-81	53-81	53-85	53-85	53-89	53-89	53-93	53-93	53-98	53-98	53-102	53-102	53-106	53-106
	53-110	53-110	53-114	53-114	53-118	53-118	53-122	53-122	102-14	102-14	103-21	103-21	112-9	112-9
	116-249	116-249	117-12	117-12	118-21	118-21	119-83	119-83	121-3	121-3	123-10	123-10	125-3	125-3
	125-8	125-8												
M#GNTE	119-5	119-5												
M#HAPT	26-34	26-34												
M#HMAP	26-34	26-34												
M#INCR	26-26	26-26	28-10	28-10	28-10	28-10	29-10	29-10	29-10	29-10	30-3	30-3	53-14	53-14
	53-14	53-14	53-16	53-16	53-18	53-18	53-18	53-20	53-22	53-22	53-22	53-22	53-24	53-26
	53-26	53-26	53-26	53-26	53-30	53-30	53-30	53-30	53-32	53-32	53-34	53-34	53-34	53-36
	53-38	53-38	53-38	53-38	53-40	53-40	53-42	53-42	53-42	53-44	53-46	53-46	53-46	53-46
	53-48	53-50	53-50	53-50	53-50	53-60	53-62	53-62	53-62	53-62	53-65	53-67	53-67	53-67
	53-67	53-77	53-79	53-79	53-79	53-79	53-81	53-81	53-83	53-83	53-83	53-85	53-87	53-87
	53-87	53-87	53-89	53-91	53-91	53-91	53-91	53-93	53-95	53-95	53-95	53-95	53-98	53-100
	53-100	53-100	53-100	53-102	53-104	53-104	53-104	53-104	53-106	53-106	53-108	53-108	53-108	53-110
	53-112	53-112	53-112	53-112	53-114	53-116	53-116	53-116	53-116	53-116	53-116	53-120	53-120	53-120
	53-122	55-7	55-8	58-11	58-12	58-22	58-28	58-30	60-12	61-8	61-24	62-36	63-21	64-8
	68-22	70-32	73-19	73-20	73-24	91-12	91-14	91-16	91-18	94-20	96-3	100-21	100-29	101-5
	102-10	102-10	102-10	102-10	103-18	103-18	103-18	103-18	105-36	105-51	106-12	107-12	107-27	107-30
	107-33	109-37	110-27	110-36	111-5	112-5	112-5	112-5	112-5	114-1	114-1	114-1	115-3	115-3
	115-10	115-10	115-10	115-10	116-45	116-45	116-45	116-45	116-47	116-53	116-59	116-66	116-69	116-93
	116-95	116-106	116-109	116-146	116-233	116-235	116-239	116-240	116-245	116-246	116-249	117-10	117-10	117-10
	117-10	117-12	118-8	118-8	118-8	118-8	118-11	118-19	118-21	119-3	119-3	119-5	119-5	119-5
	119-5	119-5	119-5	119-7	119-10	119-10	119-10	119-19	119-21	119-22	119-27	119-28	119-31	119-31
	119-31	119-32	119-44	119-44	119-44	119-64	119-82	119-83	120-3	120-3	120-14	120-14	120-14	120-14
	122-12	122-12	122-12	122-12	125-1	125-1	125-3	125-3	125-3	125-3	125-3			
M#LDRO	58-22	58-22	73-20	73-20	107-30	107-30	116-47	116-47	116-53	116-53	116-59	116-59	116-66	116-66
	116-93	116-93	116-95	116-95	116-146	116-146	116-233	116-233	118-19	118-19	119-32	119-32		
M#MCHI	26-8	26-8												
M#MCLO	26-8	26-8												
M#POP	28-14	28-14	29-14	29-14	29-16	29-16	53-16	53-16	53-20	53-20	53-24	53-24	53-28	53-28
	53-32	53-32	53-36	53-36	53-40	53-40	53-44	53-44	53-48	53-48	53-60	53-60	53-65	53-65
	53-77	53-77	53-81	53-81	53-85	53-85	53-89	53-89	53-93	53-93	53-98	53-98	53-102	53-102
	53-106	53-106	53-110	53-110	53-114	53-114	53-118	53-118	53-122	53-122	102-14	102-14	103-21	103-21
	112-9	112-9	114-106	114-106	115-16	115-16	116-249	116-249	117-12	117-12	118-21	118-21	118-23	118-23
	119-83	119-83	119-84	119-84	121-3	121-3	123-10	123-10	124-10	124-10				
M#PRIN	91-12	91-12	91-14	91-14	91-16	91-16	91-18	91-18	119-64	119-64				
M#PUSH	26-26	26-26	28-10	28-10	29-10	29-10	30-3	30-3	53-14	53-14	53-18	53-18	53-22	53-22
	53-26	53-26	53-30	53-30	53-34	53-34	53-38	53-38	53-42	53-42	53-46	53-46	53-50	53-50

	53-62	53-62	53-67	53-67	53-79	53-79	53-83	53-83	53-87	53-87	53-91	53-91	53-95	53-95
	53-100	53-100	53-104	53-104	53-108	53-108	53-112	53-112	53-116	53-116	53-120	53-120	102-10	102-10
	103-18	103-18	112-5	112-5	115-3	115-3	115-10	115-10	116-45	116-45	117-10	117-10	118-8	118-8
	119-3	119-3	119-5	119-5	120-3	120-3	120-14	120-14	122-12	122-12				
MISPUT	58-11	58-11	58-11	58-11	58-11	58-11	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	116-106	116-106	116-106	116-106	118-11	118-11	118-11	118-11	118-11	119-64
	119-64	119-64	119-64	119-64										
MISPUT1	58-11	58-11	58-11	58-11	58-11	58-11	58-11	58-11	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	116-106	116-106	116-106	116-106	116-106	116-106	116-106	116-106
	118-11	118-11	118-11	118-11	118-11	118-11	118-11	118-11	119-64	119-64	119-64	119-64	119-64	119-64
MIRADI	114-1	114-1	119-10	119-10	119-31	119-31	119-44	119-44	121-1	121-1	121-2	121-2	123-1	123-1
	123-3	123-3	123-5	123-5	123-7	123-7	123-7	123-7						
MIRBRO	73-24	73-24												
MIRBRO	116-93	116-93	116-95	116-95	116-109	116-109	116-146	116-146						
MISETS	26-26	26-26	28-10	28-10	29-10	29-10	30-3	30-3	53-14	53-14	53-18	53-18	53-22	53-22
	53-26	53-26	53-30	53-30	53-34	53-34	53-38	53-38	53-42	53-42	53-46	53-46	53-50	53-50
	53-62	53-62	53-67	53-67	53-79	53-79	53-83	53-83	53-87	53-87	53-91	53-91	53-95	53-95
	53-100	53-100	53-104	53-104	53-108	53-108	53-112	53-112	53-116	53-116	53-120	53-120	102-10	102-10
	103-18	103-18	112-5	112-5	115-3	115-3	115-10	115-10	116-45	116-45	117-10	117-10	118-8	118-8
	119-3	119-3	119-5	119-5	120-3	120-3	120-14	120-14	122-12	122-12				
MISVC	53-16	53-16	53-20	53-20	53-24	53-24	53-28	53-28	53-32	53-32	53-36	53-36	53-40	53-40
	53-44	53-44	53-48	53-48	53-60	53-60	53-65	53-65	53-77	53-77	53-81	53-81	53-85	53-85
	53-89	53-89	53-93	53-93	53-98	53-98	53-102	53-102	53-106	53-106	53-110	53-110	53-114	53-114
	53-118	53-118	53-122	53-122	55-7	55-7	55-8	55-8	58-11	58-11	58-12	58-12	58-22	58-22
	58-30	58-30	60-12	60-12	61-8	61-8	61-8	61-8	62-36	62-36	63-21	64-8	68-22	70-32
	73-20	73-20	73-24	73-24	73-24	73-24	91-12	91-12	91-14	91-14	91-16	91-16	91-18	91-18
	96-3	100-21	100-21	100-21	101-5	101-5	105-36	105-36	106-12	106-12	107-12	107-12	107-27	107-30
	107-33	109-37	110-27	110-27	110-36	110-36	111-5	111-5	114-1	114-1	114-1	114-1	116-47	116-47
	116-66	116-66	116-69	116-69	116-93	116-93	116-95	116-95	116-106	116-106	116-106	116-106	116-109	116-109
	116-233	116-233	116-235	116-235	116-239	116-239	116-240	116-240	116-245	116-245	116-246	116-246	116-249	116-249
	118-11	118-11	118-19	118-19	118-21	118-21	118-21	118-21	119-7	119-7	119-10	119-10	119-19	119-19
	119-22	119-27	119-28	119-28	119-31	119-31	119-31	119-31	119-32	119-32	119-44	119-44	119-64	119-64
	119-83	119-83												
MITLAB	53-16	53-20	53-24	53-24	53-32	53-32	53-40	53-40	53-44	53-44	53-48	53-48	53-60	53-65
	53-89	53-93	53-98	53-98	53-102	53-102	53-110	53-110	53-114	53-114	53-118	53-118	55-7	55-8
	58-28	58-30	60-12	61-8	61-24	62-36	63-21	64-8	64-8	68-22	70-32	73-19	73-20	73-24
	91-14	91-16	91-18	94-20	96-3	100-21	100-29	101-5	105-36	105-51	106-12	107-12	107-27	107-30
	107-33	109-37	110-27	110-36	111-5	114-1	116-47	116-53	116-59	116-66	116-69	116-93	116-95	116-106
	116-109	116-146	116-233	116-235	116-239	116-240	116-245	116-246	116-249	117-12	118-11	118-19	118-21	119-7
	119-10	119-19	119-21	119-22	119-27	119-28	119-31	119-32	119-44	119-64	119-82	119-83		
MITSTL	53-16	53-16	53-20	53-20	53-24	53-24	53-28	53-28	53-32	53-32	53-36	53-36	53-40	53-40
	53-44	53-44	53-48	53-48	53-60	53-60	53-65	53-65	53-77	53-77	53-81	53-81	53-85	53-85
	53-89	53-89	53-93	53-93	53-98	53-98	53-102	53-102	53-106	53-106	53-110	53-110	53-114	53-114
	53-118	53-118	53-122	53-122	55-7	55-7	55-7	55-8	55-8	58-11	58-11	58-12	58-12	58-22
	58-22	58-28	58-28	58-28	58-30	58-30	60-12	60-12	61-8	61-8	61-24	61-24	61-24	62-36
	62-36	62-36	63-21	63-21	63-21	64-8	64-8	64-8	64-8	68-22	68-22	68-22	70-32	70-32
	73-19	73-19	73-20	73-20	73-24	73-24	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	105-36
	105-36	105-36	105-51	105-51	105-51	106-12	106-12	106-12	107-12	107-12	107-27	107-27	107-30	107-30
	107-33	107-33	107-33	109-37	109-37	109-37	110-27	110-27	110-36	110-36	110-36	110-36	111-5	111-5
	114-1	114-1	116-47	116-47	116-53	116-53	116-59	116-59	116-66	116-66	116-69	116-69	116-93	116-93
	116-95	116-95	116-106	116-106	116-109	116-109	116-146	116-146	116-233	116-233	116-235	116-235	116-239	116-239
	116-239	116-240	116-240	116-245	116-245	116-245	116-246	116-246	116-249	116-249	117-12	117-12	118-11	118-11

