

RA60/80/81  
UDA50A KDA50Q

UDA50/KDA50-Q BSC SUB  
CZUDHA0

COPYRIGHT (c) 1984  
AH-T929A-MC FICHE  
FICHE 01 OF 02

FEB 1985  
digital  
Made In USA

This microfiche card contains a grid of frames, each containing technical data. The data is organized into columns and rows, with some frames containing diagrams or tables. The text is small and difficult to read, but it appears to be technical specifications or test results. The card is labeled 'FICHE 01 OF 02' and 'FEB 1985'.

RA60/80/81  
UDA50A KDA50Q

UDA50/KDA50-Q BSC SUB  
CZUDHA0

COPYRIGHT (c) 1984  
AH-T929A-MC FICHE  
FICHE 02 OF 02

FEB 1985  
digital  
Made In USA

Vertical text columns on the left side of the page, likely representing a list of items or a table. The text is faint and difficult to read, but appears to be organized in columns.



IDENTIFICATION  
-----

PRODUCT CODE: AC-T928A-MC  
PRODUCT NAME: CZUDHA0 UDA50A/KDA50Q BASIC SUBSYSTEM DIAGNOSTIC  
PRODUCT DATE: 7-OCT-1984  
MAINTAINER: ROGER OAKY  
AUTHOR: JOHN MERTZ

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  
2  
3

.REM @

TABLE OF CONTENTS

1.0	GENERAL INFORMATION . . . . .	6
1.1	PROGRAM ABSTRACT . . . . .	6
1.2	SYSTEM REQUIREMENTS . . . . .	7
2.0	OPERATING INSTRUCTIONS . . . . .	8
2.1	COMMANDS . . . . .	8
2.2	SWITCHES . . . . .	9
2.3	FLAGS . . . . .	10
2.4	HARDWARE QUESTIONS . . . . .	11
2.5	SOFTWARE QUESTIONS . . . . .	12
2.6	EXTENDED P-TABLE DIALOGUE . . . . .	13
2.7	QUICK START-UP PROCEDURE . . . . .	15
3.0	ERROR INFORMATION . . . . .	16
3.1	TYPES OF ERROR MESSAGES . . . . .	16
3.2	SPECIFIC ERROR MESSAGES . . . . .	18
3.2.1	HOST PROGRAM ERROR MESSAGES (00001 To 00999) . . . . .	18

3.2.1.1	00001 - CONTROLLER HAS MORE THAN ONE VECTOR OR BR LEVEL	18
3.2.1.2	00002 - TWO UNITS SELECT SAME DRIVE	18
3.2.1.3	00003 MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER	19
3.2.1.4	00004 NOT ENOUGH MEMORY TO TEST THE UNITS SELECTED	19
3.2.1.5	00006 - TABLE CONSISTANCY ERROR	19
3.2.1.6	00008 TWO CONTROLLERS USE THE SAME VECTOR	19
3.2.1.7	00014 - CONTROLLER IS NOT SUPPORTED BY DIAGNOSTIC PROGRAM	20
3.2.1.8	00021 - FATAL ERROR REPORTED BY CONTROLLER	20
3.2.1.9	00022 - STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION	22
3.2.1.10	00023 - CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY	22
3.2.1.11	00024 - SA REGISTER DID NOT GO TO ZERO AFTER STEP 3	23
3.2.1.12	00025 - INCORRECT DATA RETURNED IN SA REGISTER	23
3.2.1.13	00026 - DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST	23
3.2.1.14	00027 - SA REGISTER DID NOT CHANGE AFTER WRITING TO IT	24
3.2.1.15	00028 - CONTROLLER DID NOT INTERRUPT THE HOST	24
3.2.1.16	00029 - CONTROLLER INTERRUPTED AT A DIFFERENT BR LEVEL THAN EXPECTED	25
3.2.1.17	00030 - FATAL ERROR REPORTED BY CONTROLLER	25
3.2.1.18	00031 - NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES	26
3.2.1.19	00032 - UNKNOWN REQUEST RECEIVED FROM DM PROGRAM	26
3.2.1.20	00033 - RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA	27
3.2.1.21	00035 - DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE	27
3.2.1.22	00036 - NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS	28
3.2.1.23	00038 - MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS	28
3.2.1.24	00039 - DATA COMPARE ERROR DURING CONTROLLER MEMORY TEST	28
3.2.2	TEST 1 ERROR MESSAGES (01000 TO 01999)	29
3.2.2.1	1000 - DIAGNOSTIC INTERNAL ERROR DETECTED	29
3.2.2.2	1001 - NON-EXISTENT MEMORY ERROR DURING ONE-WORD READ	29
3.2.2.3	1002 - PARITY ERROR DURING ONE-WORD READ	30
3.2.2.4	1003 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD READ	30
3.2.2.5	1004 - NON-EXISTENT MEMORY ERROR DURING ONE-WORD WRITE	31
3.2.2.6	1005 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD WRITE	31
3.2.2.7	1006 - HOST MEMORY ADDRESSING ERROR. TWO ADDRESSES ACCESS THE SAME LOCATION	32

3.2.2.8	1007 - HOST MEMORY ADDRESSING ERROR. LOCATION READ SHOULD CONTAIN . . . . .	33
3.2.2.9	1008 - NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER WRITE . . . . .	33
3.2.2.10	1009 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER WRITE . . . . .	34
3.2.2.11	1010 - NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER READ . . . . .	34
3.2.2.12	1011 - PARITY ERROR DURING LARGE-BUFFER READ . . . . .	35
3.2.2.13	1012 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER READ . . . . .	35
3.2.2.14	1013 - DATA COMPARE FAILED AFTER LARGE-BUFFER WRITE THEN READ . . . . .	36
3.2.3	TEST 2 ERROR MESSAGES (02000 TO 02999) . . . . .	38
3.2.3.1	02000 - HOST SPECIFIED UNIT THAT CAN'T BE FOUND . . . . .	38
3.2.3.2	02001 - CANNOT RECEIVE VALID DRIVE STATE FROM DRIVE . . . . .	38
3.2.3.3	02002 - DRIVE STATE RECEIVED HAS BAD PARITY . . . . .	38
3.2.3.4	02003 - DRIVE IS NOT ASSERTING RECEIVER READY . . . . .	39
3.2.3.5	02004 - TIMEOUT ON SEND OF ECHO COMMAND TO DRIVE . . . . .	39
3.2.3.6	02005 - ERROR DURING RECEIVE OF ECHO RESPONSE FROM DRIVE . . . . .	39
3.2.3.7	02006 - ECHO COMMAND RESPONDED WITH DIFFERENT DATA . . . . .	40
3.2.3.8	02007 - ERROR BIT SET IN GET STATUS RESPONSE AFTER DRIVE CLEAR COMMAND . . . . .	40
3.2.3.9	02008 - TIMEOUT ON SEND OF ONLINE COMMAND TO DRIVE . . . . .	41
3.2.3.10	02009 - ERROR DURING RECEIVE OF ONLINE RESPONSE FROM DRIVE . . . . .	41
3.2.3.11	02010 - ONLINE COMMAND WAS UNSUCCESSFUL . . . . .	43
3.2.3.12	02011 - ONLINE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE . . . . .	43
3.2.3.13	02012 - TIME-OUT ON SEND OF GET UNIT CHARACTERISTICS COMMAND . . . . .	43
3.2.3.14	02013 - ERROR DURING RECEIVE OF GET UNIT CHARACTERISTICS COMMAND . . . . .	44
3.2.3.15	02014 - GET UNIT CHARACTERISTICS COMMAND WAS UNSUCCESSFUL . . . . .	44
3.2.3.16	02015 - GET UNIT CHARACTERISTICS COMMAND DID NOT RETURN EXPECTED RESPONSE . . . . .	44
3.2.3.17	02016 - HOST PROGRAM GAVE DM CODE IMPROPER DATA . . . . .	45
3.2.3.18	02017 - TIME-OUT ON SEND OF DIAGNOSE COMMAND TO DRIVE . . . . .	45
3.2.3.19	02018 - ERROR DURING RECEIVE OF DIAGNOSE RESPONSE FROM DRIVE . . . . .	45
3.2.3.20	02019 - DIAGNOSE COMMAND WAS UNSUCCESSFUL . . . . .	46
3.2.3.21	02020 - DIAGNOSE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE . . . . .	46
3.2.3.22	02021 - DRIVE DIAGNOSTIC REPORTS A HARD ERROR . . . . .	46
3.2.3.23	02024 - TIME-OUT ON SEND OF MEMORY READ COMMAND . . . . .	47

3.2.3.24	02025 - ERROR DURING RECEIVE OF MEMORY READ RESPONSE FROM DRIVE	47
3.2.3.25	02026 - MEMORY READ COMMAND WAS UNSUCCESSFUL	47
3.2.3.26	02027 - MEMORY READ COMMAND DID NOT RETURN EXPECTED RESPONSE CODE	47
3.2.3.27	02028 - TIME-OUT ON SEND OF MEMORY WRITE COMMAND TO DRIVE	48
3.2.3.28	02029 - ERROR DURING RECEIVE OF MEMORY WRITE RESPONSE FROM DRIVE	48
3.2.3.29	02030 - MEMORY WRITE COMMAND WAS UNSUCCESSFUL	48
3.2.3.30	02031 - MEMORY WRITE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE	49
3.2.3.31	02032 - TIME-OUT ON SEND OF RUN COMMAND TO DRIVE	49
3.2.3.32	02033 - ERROR DURING RECEIVE OF RUN RESPONSE FROM DRIVE	49
3.2.3.33	02034 - RUN COMMAND WAS UNSUCCESSFUL	49
3.2.3.34	02035 - RUN COMMAND DID NOT RETURN EXPECTED RESPONSE CODE	50
3.2.3.35	02036 - TIME-OUT ON SEND OF RECALIBRATE COMMAND TO DRIVE	50
3.2.3.36	02037 - ERROR DURING RECEIVE OF RECALIBRATE RESPONSE FROM DRIVE	50
3.2.3.37	02038 - RECALIBRATE COMMAND WAS UNSUCCESSFUL	50
3.2.3.38	02039 - RECALIBRATE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE	51
3.2.3.39	02040 - TIME-OUT ON SEND OF GET STATUS COMMAND TO DRIVE	51
3.2.3.40	02041 - ERROR DURING RECEIVE OF GET STATUS RESPONSE FROM DRIVE	51
3.2.3.41	02042 - GET STATUS COMMAND WAS UNSUCCESSFUL	51
3.2.3.42	02043 - GET STATUS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE	52
3.2.3.43	02044 - TIME-OUT ON SEND OF DRIVE CLEAR COMMAND TO DRIVE	52
3.2.3.44	02045 - ERROR DURING RECEIVE OF DRIVE CLEAR RESPONSE FROM DRIVE	52
3.2.3.45	02046 - DRIVE CLEAR COMMAND WAS UNSUCCESSFUL	53
3.2.3.46	02047 - DRIVE CLEAR COMMAND DID NOT RETURN EXPECTED RESPONSE CODE	53
3.2.4	TEST 3 INFORMATIONAL MESSAGES	54
3.2.4.1	LOGGABLE INFORMATION AFTER RECAL	54
3.2.5	TEST 3 ERROR MESSAGES (03000 TO 03999)	55
3.2.5.1	03001 - TIME-OUT ON SEND	55
3.2.5.2	03002 - TIME-OUT OF RECEIVE	56
3.2.5.3	03003 - FIRST WORD RECEIVED WAS NOT A START FRAME	56
3.2.5.4	03004 - FRAMING ERROR ON LEVEL 0 RESPONSE	56
3.2.5.5	03005 - CHECKSUM ERROR ON LEVEL 0 RESPONSE	57
3.2.5.6	03006 - RESPONSE LONGER THAN EXPECTED	57
3.2.5.7	03007 - CODE FROM RECEIVE WAS UNINTELLIGIBLE FROM SUBSYSTEM = 0000	57
3.2.5.8	03008 - COMMAND DID NOT RETURN EXPECTED RESPONSE CODE	58

3.2.5.9	03009 - DRIVE NOT ASSERTING RECEIVER READY IN DRIVE STATE . . . . .	58
3.2.5.10	03011 - NO VALID STATE FROM DRIVE . . . . .	58
3.2.5.11	03012 - NO VALID STATE FROM DRIVE . . . . .	59
3.2.5.12	03014 - SUBUNIT CHARACTERISTICS SAY THERE ARE ZERO READ ONLY GROUPS . . . . .	59
3.2.5.13	03015 - LESS THAN 1 READ/WRITE GROUP IN DIAGNOSTIC AREA . . . . .	59
3.2.5.14	03016 - NEITHER R/W READY NOR ATTENTION SET AFTER RECALIBRATE COMMAND . . . . .	59
3.2.5.15	03017 - LESS THAN 1 DIAGNOSTIC CYLINDER . . . . .	60
3.2.5.16	03018 - READ/WRITE READY DROPPED BEFORE FORMAT OPERATION . . . . .	60
3.2.5.17	03019 - FORMAT OPERATION REPORTED TIME-OUT FAILURE . . . . .	60
3.2.5.18	03020 - AFTER RECAL, ERROR BITS WERE SET . . . . .	61
3.2.5.19	03022 - READ/WRITE READY DROPPED BEFORE WRITE OPERATION . . . . .	61
3.2.5.20	03023 - COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK . . . . .	61
3.2.5.21	03024 - READ/WRITE READY DROPPED BEFORE READ OPERATION . . . . .	62
3.2.5.22	03025 - READ OPERATION REPORTED FAILURE . . . . .	63
3.2.5.23	03026 - COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK . . . . .	63
3.2.5.24	03027 - SEEK COMPLETE TIME-OUT . . . . .	64
3.2.5.25	03028 - NO BLOCK ON THIS TRACK CAN BE READ . . . . .	64
3.2.5.26	03029 - AVAILABLE WAS NOT ASSERTED AFTER DISCONNECT . . . . .	65
3.2.5.27	03030 - INVALID LEVEL 2 COMMAND OPCODE Aaaa WAS SUCCESSFUL . . . . .	65
3.2.5.28	03031 - COMMAND WITH Type LENGTH = A WAS SUCCESSFUL . . . . .	65
3.2.5.29	03032 - UNIT DID NOT REPORT TRANSMISSION ERROR . . . . .	66
3.2.5.30	03033 - UNIT ACCEPTED AN INVALID GROUP NUMBER . . . . .	66
3.2.5.31	03035 - SUCCESSFULLY WROTE ON DBN AREA WHEN DRIVE WAS WRITE PROTECTED . . . . .	66
3.2.5.32	03036 - DRIVE NOT PROPERLY FORMATTED . . . . .	67
3.2.5.33	03037 - DRIVE IS FORMATTED IN 576 BYTE MODE . . . . .	67
3.2.5.34	03038 - NO COPY OF THE FCT COULD BE READ . . . . .	67
3.2.6	SPECIAL DEVICE FATAL (05000) . . . . .	68
3.3	DEC STANDARD 166 EXCERPTS . . . . .	70
3.3.1	THE REPLACEMENT AND CACHING TABLES . . . . .	70
3.3.2	FCT Structure . . . . .	72
4.0	PERFORMANCE AND PROGRESS REPORTS . . . . .	75
5.0	TEST SUMMARIES . . . . .	76
5.1	TEST 1 - BUS ADDRESSING TEST . . . . .	76
5.2	TEST 2 - DISK RESIDENT DIAGNOSTIC TEST . . . . .	79
5.3	TEST 3 - DISK FUNCTION TEST . . . . .	81



## 1.0 GENERAL INFORMATION

### 1.1 PROGRAM ABSTRACT

This diagnostic program is provided for testing the UDA50A or KDA50-Q Disk Controller and the associated disk drives. There are three tests within this diagnostic:

Test 1 - Bus Addressing Test (BAT). Runs the UDA50A or KDA50-Q ROM resident diagnostics, then further tests the bus address interface and controller memory.

Test 2 - Disk Resident Diagnostic Test. Executes the diagnostics in each disk drive.

Test 3 - Disk Function Test. Functionally tests each disk drive to ensure the drive can seek, read, write and format.

This program is designed to handle all future disk drives that are attached to the UDA50A or KDA50-Q without modifying or rereleasing the program. This is possible because the disk drives are programmed to tell this diagnostic about all their characteristics that make them different from other drives, such as number of cylinders, sectors per cylinder, etc.

The following PDP-11 diagnostic programs are also provided for the UDA50A or KDA50-Q disk subsystem:

CZUDIA0 - UDA50A/KDA50-Q Disk Drive Exerciser.

CZUDKA0 - UDA50A/KDA50-Q Disk Drive Formatter.

This diagnostic has been written for use with the Diagnostic Runtime Services Software (Supervisor). These services provide the interface to the operator and to the software environment. For a complete description of the Runtime Services, refer to the XXDP+ User's Manual. There is a brief description of the Runtime Services in section 2 of this document.

This diagnostic can isolate many controller faults to either the processor module or the SDI module of the controller. Whenever a controller fault is detected and the fault can be isolated to one of the two controller modules, that module should be replaced.

## 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. This program requires the following:

- PDP-11 Unibus processor
- 28K words of memory (minimum)
- Console terminal
- XXDP+ load media containing this program
- One or more UDA50A or KDA50-Q subsystems. The subsystem controller must be at the latest hardware and microcode revision level.
- Line clock - either Type L or P

The line clock is used for all timed loops in the program. If a clock exists on the system it should be enabled so it can interrupt the processor. The diagnostic will run on a system with no clock or with the clock disabled, but will hang whenever an event for which the program is waiting does not happen (i.e., a time-out error message will not result).

The XXDP+ system device does not need to remain on-line during the execution of this diagnostic.

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

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

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

## 2.3 FLAGS

Flags are used to set up certain operational parameters such as looping on error. All flags are cleared at startup and remain cleared until explicitly set using the flags switch. Flags are also cleared after a START or RESTART command unless set using the flag switch. The ZFLAGS command may also be used to clear all flags. With the exception of the START, the 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)
ISR	Inhibit statistical reports
IDU	Inhibit program dropping of units
LOT	Loop on test

\*Error messages are described in section 3.1

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

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

## 2.4 HARDWARE QUESTIONS

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

CSR ADDRESS OF CONTROLLER (0) 172150 ?

Answer with the CSR address of one 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.

BR LEVEL (D) 5 ?

Answer with the interrupt priority used by the controller. Levels 4 to 7 are accepted. If the controller is a UDA50A, this level must match the level "hard wired" in the controller by the priority plug. If the controller is a KDA50-Q, the answer to this question is ignored and level 4 is used because all Q-bus devices interrupt at level 4.

DRIVE # (D) 0 ?

Answer with the drive number of the drive you wish to test. This is the number which appears on the "unit plug" on the front of the disk drive. A maximum of four disk drives may be tested on one controller at a time (controller configuration limit).

## 2.5 SOFTWARE QUESTIONS

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

### ENTER MANUAL INTERVENTION MODE IN TEST 2 (L) N ?

Test 2 has a manual intervention mode which allows additional parameters to be input to alter the normal testing of a disk drive. This question should normally be answered "N" when this diagnostic is first run. Then, depending on the errors detected, it may be desirable to change this answer to "Y" and alter the testing to further isolate the problem. If this question is answered "Y", and the UAM (unattended mode operation) flag is set, test 2 will print a warning message that the mode cannot be entered and will proceed as if answered "N". See the description of the individual tests in section 5 for more information.

## 2.6 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 are testing multiple drives on the same controller, this becomes tedious since most of the answers are repetitious.

To illustrate a more efficient method, suppose you are testing a UDA50A with 4 disk drives attached to it. These units are numbered 0 through 3. There is one hardware parameter that can vary among units, the drive number. This drive number may be 0 through 252. Below is a simple way to build a table for one UDA50A with four units.

```
0 UNITS (D) ? 4<CR>
```

```
UNIT 1
CSR ADDRESS (0) 172150 ? <CR>
VECTOR (0) 154 ? <CR>
BR LEVEL (D) 5 ? <CR>
DRIVE # (D) 0 ? <CR>
```

```
UNIT 2
CSR ADDRESS (0) 172150 ? <CR>
VECTOR (0) 154 ? <CR>
BR LEVEL (D) 5 ? <CR>
DRIVE # (D) 0 ? 1<CR>
```

```
UNIT 3
CSR ADDRESS (0) 172150 ? <CR>
VECTOR (0) 154 ? <CR>
BR LEVEL (D) 5 ? <CR>
DRIVE # (D) 1 ? 2<CR>
```

```
UNIT 4
CSR ADDRESS (0) 172150 ? <CR>
VECTOR (0) 154 ? <CR>
BR LEVEL (D) 5 ? <CR>
DRIVE # (D) 2 ? 3<CR>
```

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. Also, notice that the default value for the drive number changes when a non-default response is given.

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

Example 1:

```
0 UNITS (D) ? 4<CR>
```

```
UNIT 1
```



```
CSR ADDRESS (0) 172150 ? <CR>
VECTOR (0) 154 ? <CR>
BR LEVEL (0) 5 ? <CR>
DRIVE # (0) 0 ? 0-3<CR>
```

## Example 2:

```
# UNITS (0) ? 4<CR>
```

```
UNIT 1
CSR ADDRESS (0) 172150 ? <CR>
VECTOR (0) 154 ? <CR>
BR LEVEL (0) 5 ? <CR>
DRIVE # (0) 0 ? 0.1.2.3<CR>
```

As you can see in the above dialogue, the supervisor will build as many entries as it can with the information given in any one pass through the questions. In each example, four entries are built since four drive numbers were specified. The supervisor assumes that the CSR address is 172150, the vector is 154, and the BR level is 5 for each entry since they were specified only once. In the first example, the "-" construct tells the supervisor to increment the data from the first number to the second. In this case, drive numbers 0, 1, 2, and 3 were specified.

## 2.7 QUICK START-UP PROCEDURE

To start-up this program:

1. Boot XXDP.
2. Enter the date.
3. On certain systems you will be asked if the system has a Unibus.  
If this question is asked, answer it either "Y" or "N".
4. Type "R ZUDHAO"
5. Type "START"
6. Answer the "CHANGE HW" question with "Y"
7. Answer all the hardware questions
8. Answer the "CHANGE SW" question with "N"

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

Sample of terminal dialogue to test two disks on one UDA50A or KDA50-Q:

```
DR>STA/FLA:PNT/PAS:1
```

```
CHANGE HW (L) ? Y
```

```
* UNITS (D) ? 2
```

```
UNIT 0
```

```
CSR ADDRESS OF CONTROLLER (O) 172150 ?
```

```
VECTOR (O) 154 ?
```

```
BR LEVEL (D) 5 ?
```

```
DRIVE # (D) 0? 0,1
```

```
CHANGE SW (L) ? Y
```

```
ENTER MANUAL INTERVENTION MODE IN TEST 2 (L) N ?
```

```
TST: 001
```

```
TESTING INTERRUPT ABILITY OF CONTROLLER AT ADR 172150 VEC 154...COMPLETED
```

```
TST: 002
```

```
TST: 003
```

```
DR>
```

### 3.0 ERROR INFORMATION

#### 3.1 TYPES OF ERROR MESSAGES

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

```
diag severity errnum ON UNIT unit TST tst SUB sub PC: hostpc
error message
```

where:

```
diag = diagnostic name
severity = error type (SYS FTL ERR, DEV FTL ERR, HRD ERR or SFT ERR)
errnum = error number
unit = Arbitrary number assigned by the supervisor to each P-table
tst = test where error occurred
sub = substst where error occurred
hostpc = address of error message call in the host program
```

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

Device fatal errors (DVC FTL ERR) are used to report errors that are fatal to the device (may be either a UDA50A or KDA50-Q or disk drive). Testing stops on that device for the remainder of the current test.

Hard errors (HRD ERR) reports most of the errors detected. Testing will normally continue after the printing of the error.

Soft errors (SFT ERR) are not used by this diagnostic program.

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

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

The general and basic error messages from this diagnostic are always one line each. The basic message defines what program detected the error, the drive being tested and the time of the error.

The PDP-11 program that is loaded into memory when you give the "R ZUDH\*0" command to the XXDP+ monitor contains two parts, the host level code and three programs which are loaded into the UDA50A or KDA50-Q for execution. These three programs are called "diagnostic machine" or DM programs. The "diagnostic machine" is the facility in the controller which executes a PDP-11 like program. The large majority of the testing is done by these three "diagnostic machine" programs. Once the host level program has loaded and started the "diagnostic machine" program, all it does is respond to requests from that program. These requests include such things as telling the "diagnostic machine" which disks on that UDA50A or KDA50-Q are to be tested and printing error messages.

The basic message (the second line of every error message) will be one of the following:

**HOST PROGRAM CONTROLLER AT csr RUNTIME hh:mm:ss**

The host program (PDP-11) detected the error. CONTROLLER AT csr identifies the address of the UDA50A or KDA50-Q being tested. It may be omitted if the error is not specific to one controller.

**BUS ADDRESSING DM PC: dmpc CONTROLLER AT csr RUNTIME hh:mm:ss**

The "diagnostic machine" program loaded in test 1 detected the error. DM PC: dmpc identifies the address in the "diagnostic machine" program where the error message is reported.

**DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hh:mm:ss**

The "diagnostic machine" program loaded in test 2 detected the error. DM PC: dmpc identifies the address in the "diagnostic machine" program where the error message is reported. DRIVE plug identifies the drive number.

**DISK FUNCTIONAL DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hh:mm:ss**

The "diagnostic machine" program loaded in test 3 detected the error.

**Sample error message:**

```

CZUDH DVC FTL ERR 00021 ON UNIT 00 TST 001 SUB 003 PC: 021062 - general message
HOST PROGRAM CONTROLLER AT 172150 RUNTIME 0:00:12           - basic message
CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE           \
  SA REGISTER CONTAINS 104041                               }- extended
REPLACE CONTROLLER PROCESSOR MODULE                         / message

```

Some informational messages are also printed by this program. They are usually one or two lines in length. They are printed as extended messages and are printed unless the "IER", "IBE" or "IXE" flags are set. These messages are for informational purposes only and their contents are self explanatory.

### 3.2 SPECIFIC ERROR MESSAGES

Following is a list of the error messages that may be printed by the diagnostic program. In the list, some of the numbers that may vary with execution or program version are shown as lower case words. These include program counters and runtime. Other numbers, such as unit number, drive number, UDA50A or KDA50-Q address and data in registers are filled with sample numbers. Additional information about the error may follow the error message.

#### 3.2.1 HOST PROGRAM ERROR MESSAGES (00001 To 00999) -

##### 3.2.1.1 00001 - CONTROLLER HAS MORE THAN ONE VECTOR OR BR LEVEL -

```
CZUDH SYS FTL ERR 00001 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss
INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS
CONTROLLER HAS MORE THAN ONE VECTOR OR BR LEVEL
```

When the hardware questions were answered, two units were selected with the same CSR address but with a different vector or BR level. A single UDA50A or KDA50-Q can have only one vector or BR level. The program is aborted and returns to the supervisor prompt so that you can change the hardware questions.

##### 3.2.1.2 00002 - TWO UNITS SELECT SAME DRIVE -

```
CZUDH SYS FTL ERR 00002 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss
INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS
TWO UNITS SELECT THE SAME DRIVE
```

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

### 3.2.1.3 00003 MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER -

CZUDH SYS FTL ERR 00003 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS  
 MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER

Up to four physical disk drives can be attached to a UDA50A or KDA50-Q at one time. The program is aborted and returns to the supervisor prompt so that you can change the hardware questions.

### 3.2.1.4 00004 NOT ENOUGH MEMORY TO TEST THE UNITS SELECTED -

CZUDH SYS FTL ERR 00004 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM RUNTIME hh:mm:ss  
 NOT ENOUGH MEMORY TO TEST THE UNITS SELECTED  
 RESTART PROGRAM AND TEST FEWER UNITS AT A TIME

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

### 3.2.1.5 00006 - TABLE CONSISTANCY ERROR -

CZUDH SYS FTL ERR 00006 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM RUNTIME hh:mm:ss  
 TABLE CONSISTANCY ERROR. PLEASE RE-LOAD PROGRAM

When the host program is started, controller tables are set according to the P-tables. Error 00006 will occur if the tables were corrupted after restarting the diag- nostic. Load and start your program again.

### 3.2.1.6 00008 TWO CONTROLLERS USE THE SAME VECTOR -

CZUDH SYS FTL ERR 00008 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS  
 TWO CONTROLLERS USE THE SAME VECTOR

The hardware questions for two units specified different controller CSR addresses but identical vector addresses. The program is aborted and returns to the supervisor prompt so that you can change the hardware questions.

## 3.2.1.7 00014 - CONTROLLER IS NOT SUPPORTED BY DIAGNOSTIC PROGRAM -

CZUDH DVC FTL ERR 00014 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 CONTROLLER IS NOT SUPPORTED BY THIS DIAGNOSTIC PROGRAM.  
 THIS PROGRAM REQUIRES A UDA50A (MODEL 6, MICROCODE VERSION AT LEAST 3),  
 OR A KDA50-Q (MODEL 13, MICROCODE VERSION AT LEAST 0)

CONTROLLER REPORTED MODEL CODE xx AND MICROCODE VERSION xx

This diagnostic program will only test UDA50A (modules M7485-6) or KDA50-Q (modules M7164-5) controllers. UDA50 (modules M7161-2) controllers will not be tested by this diagnostic controller and should be replaced. If the program detects that the controller being tested is not a UDA50A or a KDA50-Q, it will not be tested. If the microcode version of the controller is not at the current revision level, the test will proceed, but accurate results are not guaranteed.

## 3.2.1.8 00021 - FATAL ERROR REPORTED BY CONTROLLER -

CZUDH DVC FTL ERR 00021 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 FATAL ERROR REPORTED BY CONTROLLER  
 SA REGISTER CONTAINS 105154  
 REPLACE CONTROLLER SDI MODULE

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

- 104000 - Fatal sequencer error
- 104040 - D PROC ALU test error
- 104041 - D PROC ROM parity test error / Timeout test error
- 105102 - D PROC no board 2 error / D PROC control reg test error /  
D PROC RAM parity error
- 105105 - D PROC RAM buffer error
- 105152 - D PROC SDI error
- 105153 - D PROC write mode, wrap SERDES 16 error
- 105154 - D PROC read mode, SERDES 16, 10 RSGEN and ECC error
- 106040 - U PROC ALU error / DFAIL test error / Unexpected trap
- 106041 - U PROC control reg test error
- 106042 - U PROC parity error set erroneously / CROM parity error
- 106047 - U PROC Constant ROM error with D proc running SDI test
- 106055 - Unexpected trap - abnormal termination of diagnostics
- 106071 - U PROC ROM error
- 106072 - U PROC ROM parity test error
- 106200 - STEP 1 data error (MSB wasn't set)
- 107103 - U PROC RAM parity error
- 107107 - U PROC RAM buffer error
- 107115 - Board #2 test count was wrong
- 112300 - STEP 2 INIT error

122240 - DMA test error  
122300 - STEP 3 INIT error  
142300 - STEP 4 INIT error

Replace either the processor module or the SDI interface module as specified in the error message.



3.2.1.9 00022 - STEP BIT DID NOT SET IN SA REGISTER DURING  
INITIALIZATION -

CZUDH DVC FTL ERR 00022 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION  
 STEP BIT EXPECTED 004000  
 SA REGISTER 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

The expected step bit nor the error bit set within the expected time.

3.2.1.10 00023 - CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST  
MEMORY -

CZUDH DVC FTL ERR 00023 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION  
 6 WORDS WERE TO BE CLEARED STARTING AT ADDRESS 040644  
 THE FOLLOWING WORDS NOT CLEARED:

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 bus.

## 3.2.1.11 00024 - SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 -

CZUDH DVC FTL ERR 00024 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION  
 PURGE/POLE DIAGNOSTICS WERE REQUESTED  
 SA REGISTER CONTENTS 004400

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 goes to zero, then error message 00024 is displayed. The controller may have a bad processor module or the bus maybe broken.

## 3.2.1.12 00025 - INCORRECT DATA RETURNED IN SA REGISTER -

CZUDH DVC FTL ERR 00025 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 INCORRECT DATA RETURNED IN SA REGISTER DURING INITIALIZATION  
 SA EXPECTED 004400  
 SA REGISTER CONTAINS 004000  
 REPLACE CONTROLLER PROCESSOR MODULE

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

## 3.2.1.13 00026 - DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST -

CZUDH DVC FTL ERR 00026 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST  
 DATA SENT TO SA REGISTER 000001  
 RECEIVED FROM SA REGISTER 000000  
 REPLACE CONTROLLER PROCESSOR MODULE

The controller can be put into a mode where the SA register acts as a wrap port. While the controller is in this mode, any data being sent to the SA register will be displayed in the SA register within a small period of time. If the data in the SA register does not match the data that was sent to the SA register, then error message 00026 is displayed. Replace controller processor module.

#### 3.2.1.14 00027 - SA REGISTER DID NOT CHANGE AFTER WRITING TO IT -

CZUDH DVC FTL ERR 00027 ON UNIT 00 TST tst SUB sub PC: hostpc  
HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
SA REGISTER DID NOT CHANGE AFTER WRITING TO IT  
IN PORT LOOP DIAGNOSTIC  
SA REGISTER CONTAINS 004400  
REPLACE CONTROLLER PROCESSOR MODULE

The controller can be put into a mode where the SA register acts as a wrap port. While the controller is in this mode, any data being sent to the SA register will be displayed in the SA register within a small period of time. If, after the host program sends data to it while it was in diagnostic wrap mode, the controller does not change the contents of the SA register, error message 00027 is displayed. Replace controller processor module.

#### 3.2.1.15 00028 - CONTROLLER DID NOT INTERRUPT THE HOST -

CZUDH DVC FTL ERR 00028 ON UNIT 00 TST 001 SUB 005 PC: hostpc  
HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
CONTROLLER DID NOT INTERRUPT THE HOST  
REPLACE CONTROLLER PROCESSOR MODULE

The host program timed out while waiting for an interrupt that had to occur. The controller was told to use interrupts during the initialization process. The host program then waited for the interrupt but it did not occur. Replace the controller processor module.

### 3.2.1.16 00029 - CONTROLLER INTERRUPTED AT A DIFFERENT BR LEVEL THAN EXPECTED -

CZUDH DVC FTL ERR 00029 ON UNIT 00 TST 001 SUB 005 PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 CONTROLLER INTERRUPTED AT DIFFERENT BR LEVEL THAN EXPECTED  
 INTERRUPT WAS AT BR LEVEL 5

The controller interrupted at a different BR level than expected. If the controller under test is a UDA50A, be sure the priority plug on the controller and the BR level specified during the hardware questions match. If these are the same, this error indicates a problem with the controller processor module. If the controller is a KDA50-Q, the BR level will always be 4 and this error will indicate a problem with the controller processor module.

### 3.2.1.17 00030 - FATAL ERROR REPORTED BY CONTROLLER -

CZUDH DVC FTL ERR 00030 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 FATAL ERROR REPORTED BY CONTROLLER  
 SA REGISTER CONTAINS 100004

A message from the controller firmware reports an unexpected failure. An error code is presented in the SA register. 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 register.
- 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

### 3.2.1.18 00031 - NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES

CZUDH DVC FTL ERR 00031 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES  
 ASSUME PROGRAM IS HUNG

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

### 3.2.1.19 00032 - UNKNOWN REQUEST RECEIVED FROM DM PROGRAM -

CZUDH DVC FTL ERR 00032 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER  
 MESSAGE BUFFER CONTAINS:

```
000001 000002 000003 000004 000005 000006 000007
000008 000009 000010 000011 000012 000013 000014
000015 000016 000017 000018 000019 000020 000021
000022 000023 000024 000025 000026 000027 000028
000029 000030 000031 000032 000033 000034 000035
```

The DM program and the host program communicate with each other using packets. Each packet must have a request number set up by the DM program and interpreted by the host program. This request number is not a known request number. The problem may be the 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.

## 3.2.1.20 00033 - RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA -

CZUDH DVC FTL ERR 00033 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 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 to 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 words per line, low order word and byte to the right (corresponding to the MSCP long-word entity).

## 3.2.1.21 00035 - DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE -

CZUDH DVC FTL ERR 00035 ON UNIT 00 TST tst SUB sub PC: hostpc  
 HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss  
 DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE  
 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 currently running DM program sent a message to the controller referencing a drive which doesn't have an entry in the host program's drive data tables. The message sent to the host program is also included in the error message. This error indicates either a software

or hardware error. Re-load the diagnostic program and run TEST 1 to diagnose the failure.

### 3.2.1.22 00036 - NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS

```
CZUDH DVC FTL ERR 00036 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss
NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS
WHILE LOADING DM PROGRAM
```

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.

### 3.2.1.23 00038 - MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS -

```
CZUDH DVC FTL ERR 00038 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss
MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS
CHECK BUS 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 controller IP and SA registers while in subtest 2 of test 1. The controller is at another CSR address (check the bus selection switches) or the processor module is broken or the bus is broken.

### 3.2.1.24 00039 - DATA COMPARE ERROR DURING CONTROLLER MEMORY TEST -

```
CZUDH DVC FTL ERR 00039 ON UNIT 00 TST 001 SUB 008 PC: hostpc
HOST PROGRAM CONTROLLER AT 172150 RUNTIME hh:mm:ss
DATA COMPARE ERROR DURING CONTROLLER MEMORY TEST
CONTROLLER MEMORY ADDRESS xxxxxx
DATA WRITTEN xxxxxx DATA READ xxxxxx
```

Subtest 8 of the Bus Addressing Test performs a controller memory test by writing known data patterns to the controller memory and then reading the data back. If the data read is not the same as the data written this error will occur. This usually indicates a bad memory location in the controller or a controller memory addressing problem.

## 3.2.2 TEST 1 ERROR MESSAGES (01000 TO 01999) -

## 3.2.2.1 1000 - DIAGNOSTIC INTERNAL ERROR DETECTED -

CZUDH DVC FTL ERR 01000 ON UNIT unit TST 001 SUB 000 PC: hostpc  
 BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
 DIAGNOSTIC INTERNAL ERROR DETECTED  
 PARAMETERS PASSED TO DIAGNOSTIC FROM HOST ARE INCORRECT

EITHER AT LEAST ONE WORD IS NOT WRITEABLE OR READABLE OR  
 THE HOST WRITEABLE OR READABLE AREA DOES NOT BEGIN OR END  
 ON A WORD BOUNDARY

## WRITEABLE AREA:

START write\_start (HEX)  
 END write\_end (HEX)

## READABLE AREA:

START read\_start (HEX)  
 END read\_end (HEX)

## write\_start:

Starting address of the writeable area in hex.

## write\_end:

Ending address of the writeable area in hex.

## read\_start:

Starting address of the readable area in hex.

## read\_end:

Ending address of the readable area in hex.

Before starting the test, BAT checks the parameters passed to it from the host to make sure that all the parameters are valid. If a problem is found with a parameter (odd address, no readable or writeable area) the above error is printed.

This usually indicates a host processor or host memory with major problems. Run the processor and memory diagnostics before continuing.

NOTE: BAT requires AT LEAST one word of writeable and one word of readable memory. To specify one word, the starting and ending addresses would be the same.

## 3.2.2.2 1001 - NON-EXISTENT MEMORY ERROR DURING ONE-WORD READ -

CZUDH DVC FTL ERR 01001 ON UNIT unit TST 001 SUB 000 PC: hostpc  
 BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
 NON-EXISTENT MEMORY ERROR DURING ONE-WORD READ FROM HOST MEMORY  
 OCTAL HEX



ADDRESS	octal_address	hex_address
---------	---------------	-------------

octal\_address, hex\_address:

The 32 bit octal and hex address, respectively, of the location of the non-existent memory.

During the reading of the address that contain their own addresses (the cooperating host portion of the test), BAT got a NXM error. NXMs are expected in the readable area, but not in the writeable area. When a NXM is detected during the reading of the addresses, the above error is printed.

This usually indicates the controller has a high addressing line that is stuck at one. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

### 3.2.2.3 1002 - PARITY ERROR DURING ONE-WORD READ -

CZUDH DVC FTL ERR 01002 ON UNIT unit TST 001 SUB 000 PC: hostpc  
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
PARITY ERROR DURING ONE-WORD READ FROM HOST MEMORY

	OCTAL	HEX
ADDRESS	octal_address	hex_address

octal\_address, hex\_address:

The 32 bit octal and hex address, respectively, of the location of the parity error.

If a parity error is detected during the reading of a address containing its own address (the cooperating host portion of the test) OR the unique addressing test, the above error error is printed.

This can be caused by bad a bad transmitter on the host memory, a bad receiver on the controller, stuck-at-one-or-zero data lines in the controller or a bad backplane. Run the processor and memory diagnostics before continuing.

### 3.2.2.4 1003 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD READ -

CZUDH DVC FTL ERR 01003 ON UNIT unit TST 001 SUB 000 PC: hostpc  
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
DIAGNOSTIC INTERNAL ERROR DETECTED  
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD READ  
FROM HOST MEMORY

	code (DECIMAL)	
	OCTAL	HEX
ADDRESS	octal_address	hex_address

**code:**

The error code returned in DM R1 that BAT does not recognize.

**octal\_address, hex\_address:**

The 32 bit octal and hex address, respectively, of the location which caused the unknown error

After every one-word read, BAT checks the contents of DM R1. The only errors that are possible are NXM and parity. If BAT sees any other error code in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

**3.2.2.5 1004 - NON-EXISTENT MEMORY ERROR DURING ONE-WORD WRITE -**

```
CZUDH DVC FTL ERR 01004 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
NON-EXISTENT MEMORY ERROR DURING ONE-WORD WRITE TO HOST MEMORY
                                OCTAL      HEX
ADDRESS                octal_address  hex_address
```

**octal\_address, hex\_address:**

The 32 bit octal and hex address, respectively, of the location of the non-existent memory.

If, when writing during the unique addressing test, a NXM is detected within the writeable area, the above error is printed.

This usually indicates the controller has a high addressing line that is stuck at one. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

**3.2.2.6 1005 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD WRITE -**

```
CZUDH DVC FTL ERR 01005 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
DIAGNOSTIC INTERNAL ERROR DETECTED
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD WRITE
TO HOST MEMORY
ERROR CODE RETURNED FROM CONTROLLER: code (DECIMAL)
                                OCTAL      HEX
ADDRESS                octal_address  hex_address
```

**code:**

The error code returned in DM R1 that BAT does not recognize.

**octal\_address, hex\_address:**

The 32 bit octal and hex address, respectively, of the location which caused the unknown error

After every one-word write, BAT checks the contents of DM R1. The only error possible is a NXM. If BAT sees a error code other than a NXM in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

### 3.2.2.7 1006 - HOST MEMORY ADDRESSING ERROR. TWO ADDRESSES ACCESS THE SAME LOCATION -

CZUDH DVC FTL ERR 01006 ON UNIT unit TST 001 SUB 000 PC: hostpc  
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
HOST MEMORY ADDRESSING ERROR. TWO ADDRESSES ACCESS THE SAME LOCATION

	OCTAL	HEX
KNOWN GOOD ADDRESS	octal_good	hex_good
TEST ADDRESS	octal_test	hex_test
ADDRESS BIT IN ERROR	octal_xor	hex_xor

**octal\_good, hex\_good:**

The 32 bit octal and hex address, respectively, of the location which is 'base' of the test.

**octal\_test, hex\_test:**

The 32 bit octal and hex address, respectively, of the address that was under test.

**octal\_xor, hex\_xor:**

The 32 bit octal and hex values, respectively, showing the addressing bit that is stuck-at-one or stuck-at-zero.

If, during the unique addressing test, the same location is found to respond to two different addresses, the above error is printed.

This indicates that the addressing bit shown on the 'ADDRESS BIT IN ERROR' line is either stuck-at-zero or stuck-at-one.

## 3.2.2.8 1007 - HOST MEMORY ADDRESSING ERROR. LOCATION READ SHOULD CONTAIN -

CZUDH DVC FTL ERP 01007 ON UNIT unit TST 001 SUB 000 PC: hostpc  
 BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
 HOST MEMORY ADDRESSING ERROR. LOCATION READ SHOULD CONTAIN  
 THE LOW 16 BITS OF ITS ADDRESS

	OCTAL	HEX
ADDRESS	octal_address	hex_address
DATA READ	octal_data	hex_data
DATA EXPECTED	octal_expected	hex_expected

octal\_address, hex\_address:

The 32 bit octal and hex address, respectively, of the location which has been read.

octal\_data, hex\_data:

The 16 bit octal and hex contents, respectively, of the address above.

octal\_expected, hex\_expected:

The 16 bit octal and hex values, respectively, which was expected to be found in the address above.

If, while reading the addresses that contain their own addresses, BAT detects that an address DOES NOT contain the low 16 bits of it's own address, the above error is printed.

This usually indicates an addressing bit that is stuck-at-one or stuck-at-zero. Look for error 1006 later on in the error reports. This should give a better indication of the problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

## 3.2.2.9 1008 - NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER WRITE -

CZUDH DVC FTL ERR 01008 ON UNIT unit TST 001 SUB 000 PC: hostpc  
 BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
 NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER WRITE TO HOST MEMORY

	OCTAL	HEX
BUFFER STARTING ADDRESS	octal_address	hex_address
BUFFER SIZE IN BYTES	octal_size	hex_size

octal\_address, hex\_address:

The 32 bit octal and hex address, respectively, of the start of the large buffer write.

octal\_size, hex\_size:

The 16 bit octal and hex size, respectively, of the transfer.

If BAT detects a NXM during a large buffer write, the above error is

printed (the entire writeable area should be writeable and addressable)

This usually indicates the controller has a high addressing line that is stuck at one. Another possibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

### 3.2.2.10 1009 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER WRITE -

```
CZUDH DVC FTL ERR 01009 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
DIAGNOSTIC INTERNAL ERROR DETECTED
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER WRITE
ERROR CODE RETURNED FROM CONTROLLER: code (DECIMAL)
                                OCTAL      HEX
BUFFER STARTING ADDRESS  octal_address  hex_address
BUFFER SIZE IN BYTES    octal_size    hex_size
```

code:

The error code returned in DM R1 that BAT does not recognize.

octal\_address, hex\_address:

The 32 bit octal and hex address, respectively, of the start of the large buffer write.

octal\_size, hex\_size:

The 16 bit octal and hex size, respectively, of the transfer.

After every large buffer write, BAT checks the contents of DM R1. The only error possible is a NXM. If BAT sees a error code other than a NXM (or a success) in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another possibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

### 3.2.2.11 1010 - NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER READ -

```
CZUDH DVC FTL ERR 01010 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER READ FROM HOST MEMORY
                                OCTAL      HEX
BUFFER STARTING ADDRESS  octal_address  hex_address
BUFFER SIZE IN BYTES    octal_size    hex_size
```

octal\_address, hex\_address:

The 32 bit octal and hex address, respectively, of the start of the large buffer read.

octal\_size, hex\_size:

The 16 bit octal and hex size, respectively, of the transfer.

If BAT detects a NXM during a large buffer read, the above error is printed (the entire writeable area should readable)

This usually indicates the controller has a high addressing line that is stuck at one. Another possibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

#### 3.2.2.12 1011 - PARITY ERROR DURING LARGE-BUFFER READ -

CZUDH DVC FTL ERR 01011 ON UNIT unit TST 001 SUB 000 PC: hostpc  
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
PARITY ERROR DURING LARGE-BUFFER READ FROM HOST MEMORY

	OCTAL	HEX
BUFFER STARTING ADDRESS	octal_address	hex_address
BUFFER SIZE IN BYTES	octal_size	hex_size

octal\_address, hex\_address:

The 32 bit octal and hex address, respectively, of the start of the large buffer read.

octal\_size, hex\_size:

The 16 bit octal and hex size, respectively, of the transfer.

While doing a large buffer read, BAT detected a parity error in the data read from the host.

This can be caused by bad a bad transmitter on the host memory, a bad receiver on the controller, stuck-at-one-or-zero data lines in the controller or a bad backplane. Another possibility is that the controller has some kind of stress or heat related problem. Run the processor and memory diagnostics before continuing.

#### 3.2.2.13 1012 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER READ -

CZUDH DVC FTL ERR 01012 ON UNIT unit TST 001 SUB 000 PC: hostpc  
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
DIAGNOSTIC INTERNAL ERROR DETECTED  
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER READ  
FROM HOST MEMORY

ERROR CODE RETURNED FROM CONTROLLER: code (DECIMAL)		
	OCTAL	HEX
BUFFER STARTING ADDRESS	octal_address	hex_address
BUFFER SIZE IN BYTES	octal_size	hex_size

**code:**

The error code returned in DM R1 that BAT does not recognize.

**octal\_address, hex\_address:**

The 32 bit octal and hex address, respectively, of the start of the large buffer read.

**octal\_size, hex\_size:**

The 16 bit octal and hex size, respectively, of the transfer.

After every large buffer read, BAT checks the contents of DM R1. The only errors possible are a NXM and parity error. If BAT sees a error code other than a NXM, parity error (or a success) in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another possibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

### 3.2.2.14 1013 - DATA COMPARE FAILED AFTER LARGE-BUFFER WRITE THEN READ -

CZUDH DVC FTL ERR 01013 ON UNIT unit TST 001 SUB 000 PC: hostpc  
BUS ADDRESSING DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss  
DATA COMPARE FAILED AFTER LARGE-BUFFER WRITE THEN READ FROM HOST MEMORY  
CURRENT DATA PATTERN: pattern

	OCTAL	HEX
BUFFER STARTING ADDRESS	octal_address	hex_address
BUFFER SIZE IN BYTES	octal_size	hex_size
ADDRESS OF ERROR	octal_error	hex_error
DATA READ	octal_data	hex_data
DATA EXPECTED	octal_expected	hex_expected

**octal\_address, hex\_address:**

The 32 bit octal and hex address, respectively, of the start of the large buffer read.

**octal\_size, hex\_size:**

The 16 bit octal and hex size, respectively, of the transfer.

**octal\_error, hex\_error:**

The 32 bit octal and hex address, respectively, of where the data comparison failed.

octal\_data, hex\_data:

The 16 bit octal and hex contents, respectively, of the address above.

octal\_expected, hex\_expected:

The 16 bit octal and hex values, respectively, which was expected to be found in the address above.

After a large buffer read, BAT checks the data read against a known pattern. If any difference is found between the pattern read and the known pattern, the above error is printed.

This usually indicates the controller has a data line that is stuck at one or zero. Another possibility is that the controller has some kind of stress or heat related memory problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.



### 3.2.3 TEST 2 ERROR MESSAGES (02000 TO 02999) -

#### 3.2.3.1 02000 - HOST SPECIFIED UNIT THAT CAN'T BE FOUND -

CZUDH HRD ERR 02000 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 HOST SPECIFIED UNIT #0 THAT CAN'T BE FOUND.  
 TEST2 RESTARTING

When test 2 starts executing out of the DM, it doesn't know if it had been started to execute drive diagnostics or restarted to down line load a diagnostic into the drive. If it had been restarted for the latter reason, the host must tell Test 2 which drive was to receive the diagnostic. If the drive specified by the host is not attached to the controller or could not be located by Test 2, this error message will be printed.

#### 3.2.3.2 02001 - CANNOT RECEIVE VALID DRIVE STATE FROM DRIVE -

CZUDH HRD ERR 02001 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 CANNOT RECEIVE VALID DRIVE STATE FROM DRIVE AFTER DRIVE WAS INITED  
 CHECK IF DRIVE IS POWERED ON.

This error message is presented if valid drive state was not received from the drive after the drive was inited. There are two types of invalid states: no clocks or 'hard' errors. If after getting state and no clocks occur, error 2001 is reported. There may be a bad transmitter on the drive side or a bad receiver on the controller side or the SDI cable may have taken a hit.

#### 3.2.3.3 02002 - DRIVE STATE RECEIVED HAS BAD PARITY -

CZUDH HRD ERR 02002 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 DRIVE STATE RECEIVED HAS BAD PARITY AFTER DRIVE WAS INITED

This error message is presented if bad parity was received from the drive after the drive was inited. There may be a bad transmitter on the drive side or a bad receiver on the controller side or the SDI cable may have taken a hit.

## 3.2.3.4 02003 - DRIVE IS NOT ASSERTING RECEIVER READY -

CZUDH HRD ERR 02003 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
DRIVE IS NOT ASSERTING RECEIVER READY IN DRIVE STATE AFTER DRIVE WAS INITED

This error message is presented if receiver ready was not received from the drive after the drive was initied. There may be a bad transmitter on the drive side or a bad receiver on the controller side or the SDI cable may have taken a hit.

## 3.2.3.5 02004 - TIMEOUT ON SEND OF ECHO COMMAND TO DRIVE -

CZUDH HRD ERR 02004 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
TIME-OUT ON SEND OF ECHO COMMAND TO DRIVE  
ECHO DATA FF

This error message is presented if a send of the ECHO command timed out. This may be caused by receiver ready being deasserted. The echo data is presented in hex.

## 3.2.3.6 02005 - ERROR DURING RECEIVE OF ECHO RESPONSE FROM DRIVE -

CZUDH HRD ERR 02005 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
ERROR DURING RECEIVE OF ECHO RESPONSE FROM DRIVE  
ECHO DATA FF

This error message is presented if a receive of an ECHO command was in error. The echo data is presented in hex. There may be a bad transmitter on the drive side or a bad receiver on the controller side or the SDI cable may have taken a hit.

## 3.2.3.7 02006 - ECHO COMMAND RESPONDED WITH DIFFERENT DATA -

```

CZUDM HRD ERR 02006 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss
ECHO COMMAND RESPONDED WITH DIFFERENT DATA
  ECHO DATA SENT      00FE
  ECHO DATA RECEIVED  00FF

```

This error message is presented if the data returning from an ECHO command did not match the data it was suppose to. The data presented is in hex.

## 3.2.3.8 02007 - ERROR BIT SET IN GET STATUS RESPONSE AFTER DRIVE CLEAR COMMAND -

```

CZUDM HRD ERR 02007 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss
ERROR BIT SET IN GET STATUS RESPONSE AFTER DRIVE CLEAR COMMAND
  GET STATUS RESPONSE
  REAL TIME STATE state
  STATUS (FROM R TO L): word6 word5 word4 word3 word2 word1 word0:

```

This error message is presented when an error bit is set in the status of a drive after the drive was cleared of all errors. The data displayed is the response from a GET STATUS command. The error bits in the response are in bit position 3, 5 and 6 of word2. For further description of the GET STATUS response, refer to the SDI Functional Spec v3.6 and the drive's functional spec.

## REAL TIME STATE state:

The real time state is the real time drive state AFTER Test 2 detected the error. THIS VALUE IS DISPLAYED IN HEX. In this example, receiver ready and attention are both asserted.

The bit positions are defined as follows:

- 0001 - Receiver ready (Test 2 able to transmit to drive)
- 0002 - Attention (error occurred or online timeout expired)
- 0040 - Available (drive offline and usable)
- 1000 - Read/Write ready

The complete meaning of these bits is beyond the scope of this text. please refer to the operator documentation for the drive you are working on.

STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0:

The status is the response to the SDI GET STATUS command. These words are printed in HEX. <<NOTE THAT THE STATUS IS PRINTED OUT FROM RIGHT TO LEFT!!>>. The status' meaning is beyond the scope of this text. please refer to the operator documentation for the drive you are working on.

### 3.2.3.9 02008 - TIMEOUT ON SEND OF ONLINE COMMAND TO DRIVE -

CZUDH HRD ERR 02008 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF ONLINE COMMAND TO DRIVE

The ONLINE command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

### 3.2.3.10 02009 - ERROR DURING RECEIVE OF ONLINE RESPONSE FROM DRIVE -

CZUDH HRD ERR 02009 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF ONLINE RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of an ONLINE command was in error. An explanation of what the error was is also presented. These explanations are:

#### TIMEOUT ERROR OCCURED DURING RECEIVE XFC

This error is a failure of the drive to respond to an SDI level 2 command (see the SDI specification) before the drive-supplied command timeout expires.

#### 1ST WORD NOT START FRAME DURING RECEIVE XFC

The first word received by the controller from the drive was not a valid message start frame.

#### FRAMING ERROR OCCURED ON SDI LEVEL 0 READ DURING RECEIVE XFC

This is caused by one of the following conditions: 1) Illegal frame code -- the frame is not a message start, continue, or end frame. 2) Illegal sequence of frames -- such as a message start frame without ever receiving a message end frame. This can be caused by the drive sending a response before the controller asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or controller receiver.

#### CHECKSUM ERROR OCCURED ON SDI LEVEL 0 READ DURING RECEIVE XFC

The checksum attached to a message end frame did not match the checksum computed over the level 2 command. This could be caused by a bad drive transmitter, bad controller receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable.

#### BUFFER SIZE SMALLER THEN RESPONSE DURING RECEIVE XFC

A buffer size size set aside for the response was not large enough for the response received. This is caused by the drive sending a response that is incorrect for the request sent to the drive, or the drive sending some garbage with the response.

CODE FROM RECEIVE XFC WAS UNINTELLIGIBLE FROM SUBSYSTEM 0000

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 43  
USER DOCUMENTATION

UDA50A/KDA50-Q Basic Subsystem Diag. User Guide

Page 42

The response from the drive was not anything that was expected.  
Possible controller microcode change without test 2 update.

## 3.2.3.11 02010 - ONLINE COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02010 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 ONLINE COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The ONLINE command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status. The drive did not assert the RECEIVER READY signal over the SDI.

## 3.2.3.12 02011 - ONLINE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE

CZUDH HRD ERR 02011 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 ONLINE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE 7E  
 ACTUAL RESPONSE 00

The ONLINE command did not return an expected response code. If there were at least an UNSUCCESSFUL response, test 2 will report the drive state and status. The expected response and actual response are in hex.

## 3.2.3.13 02012 - TIME-OUT ON SEND OF GET UNIT CHARACTERISTICS COMMAND

CZUDH HRD ERR 02012 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF GET UNIT CHARACTERISTICS COMMAND TO DRIVE

The GET UNIT CHARACTERISTICS command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

3.2.3.14 02013 - ERROR DURING RECEIVE OF GET UNIT CHARACTERISTICS  
COMMAND -

CZUDH HRD ERR 02013 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF GET UNIT CHARACTERISTICS COMMAND FROM DRIVE  
 explanation

This error message is presented if a receive of a GET UNIT CHARACTERISTICS command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

## 3.2.3.15 02014 - GET UNIT CHARACTERISTICS COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02014 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 GET UNIT CHARACTERISTICS COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The GET UNIT CHARACTERISTICS command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

3.2.3.16 02015 - GET UNIT CHARACTERISTICS COMMAND DID NOT RETURN  
EXPECTED RESPONSE -

CZUDH HRD ERR 02015 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 GET UNIT CHARACTERISTICS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE 78  
 ACTUAL RESPONSE 00

The GET UNIT CHARACTERISTICS command did not return an expected response code. The expected response and actual response are in hex.

### 3.2.3.17 02016 - HOST PROGRAM GAVE DM CODE IMPROPER DATA -

CZUDH HRD ERR 02016 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 HOST PROGRAM GAVE DM CODE IMPROPER DATA  
 EXPECTED VALUE SHOULD BE BETWEEN 0 AND 3  
 ACTUAL VALUE WAS xx

The host tells the DM program what to do after the DM program is done testing the drive's diagnostic. If the value is not within the expected range, this error message is printed. There is no drive problem. The problem is between the host and the controller.

### 3.2.3.18 02017 - TIME-OUT ON SEND OF DIAGNOSE COMMAND TO DRIVE -

CZUDH HRD ERR 02017 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF DIAGNOSE COMMAND TO DRIVE

The DIAGNOSE command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

### 3.2.3.19 02018 - ERROR DURING RECEIVE OF DIAGNOSE RESPONSE FROM DRIVE

CZUDH HRD ERR 02018 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF DIAGNOSE RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of a DIAGNOSE command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.



### 3.2.3.20 02019 - DIAGNOSE COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02019 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 DIAGNOSE COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE state  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The DIAGNOSE command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

### 3.2.3.21 02020 - DIAGNOSE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

CZUDH HRD ERR 02020 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 DIAGNOSE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE FC  
 ACTUAL RESPONSE 00

The DIAGNOSE command did not return an expected response code. The expected response and actual response are in hex.

### 3.2.3.22 02021 - DRIVE DIAGNOSTIC REPORTS A HARD ERROR -

CZUDH HRD ERR 02021 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 DRIVE DIAGNOSTIC REPORTS A HARD ERROR  
 TEST NUMBER 0000  
 DRIVE TYPE 00  
 ERROR NUMBER 0000  
 data

The drive diagnostic found an error and is reporting the error back to the host. All values are in hex. TEST NUMBER shows what test was run. DRIVE TYPE shows what type of drive was being tested. ERROR NUMBER shows the result of the test. The drive may pass back data to the host. This data will be presented in a 32 bit hex format following the error message. More data may follow the 32 bit hex values. This data is printed in ascii format. For definitions of what these values mean, refer to the drive functional spec.

### 3.2.3.23 02024 - TIME-OUT ON SEND OF MEMORY READ COMMAND -

CZUDH HRD ERR 02024 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF MEMORY READ COMMAND TO DRIVE

The MEMORY READ command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

### 3.2.3.24 02025 - ERROR DURING RECEIVE OF MEMORY READ RESPONSE FROM DRIVE -

CZUDH HRD ERR 02025 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF MEMORY READ RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of a MEMORY READ command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

### 3.2.3.25 02026 - MEMORY READ COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02026 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 MEMORY READ COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The MEMORY READ command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

### 3.2.3.26 02027 - MEMORY READ COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

CZUDH HRD ERR 02027 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 MEMORY READ COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE 72  
 ACTUAL RESPONSE 00

The MEMORY READ command did not return an expected response code. The expected response and actual response are in hex.

### 3.2.3.27 02028 - TIME-OUT ON SEND OF MEMORY WRITE COMMAND TO DRIVE -

CZUDH HRD ERR 02028 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF MEMORY WRITE COMMAND TO DRIVE

The MEMORY WRITE command timed out while it was sent to the drive.  
 The drive did not assert the RECEIVER READY signal over the SDI.

### 3.2.3.28 02029 - ERROR DURING RECEIVE OF MEMORY WRITE RESPONSE FROM DRIVE -

CZUDH HRD ERR 02029 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF MEMORY WRITE RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of a MEMORY WRITE command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

|

### 3.2.3.29 02030 - MEMORY WRITE COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02030 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 MEMORY WRITE COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The MEMORY WRITE command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

3.2.3.30 02031 - MEMORY WRITE COMMAND DID NOT RETURN EXPECTED  
 RESPONSE CODE -

CZUDH HRD ERR 02031 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 MEMORY WRITE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE 7E  
 ACTUAL RESPONSE 00

The MEMORY WRITE command did not return an expected response code.  
 The expected response and actual response are in hex.

3.2.3.31 02032 - TIME-OUT ON SEND OF RUN COMMAND TO DRIVE -

CZUDH HRD ERR 02032 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF RUN COMMAND TO DRIVE

The RUN command timed out while it was sent to the drive. The drive  
 did not assert the RECEIVER READY signal over the SDI.

3.2.3.32 02033 - ERROR DURING RECEIVE OF RUN RESPONSE FROM DRIVE -

CZUDH HRD ERR 02033 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF RUN RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of a RUN command was in  
 error. An explanation of what the error was is also presented. These  
 explanations are described in hard error 2009.

3.2.3.33 02034 - RUN COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02034 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 RUN COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The RUN command was not successful. The drive's status is displayed.  
 See hard error 2007 for further information on the format of the  
 status.

## 3.2.3.34 02035 - RUN COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

CZUDH HRD ERR 02035 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 RUN COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE 7E  
 ACTUAL RESPONSE 00

The RUN command did not return an expected response code. The expected response and actual response are in hex.

## 3.2.3.35 02036 - TIME-OUT ON SEND OF RECALIBRATE COMMAND TO DRIVE -

CZUDH HRD ERR 02036 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF RECALIBRATE COMMAND TO DRIVE

The RECALIBRATE command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

## 3.2.3.36 02037 - ERROR DURING RECEIVE OF RECALIBRATE RESPONSE FROM DRIVE -

CZUDH HRD ERR 02037 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF RECALIBRATE RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of a RECALIBRATE command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

## 3.2.3.37 02038 - RECALIBRATE COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02038 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 RECALIBRATE COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The RECALIBRATE command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

3.2.3.38 02039 - RECALIBRATE COMMAND DID NOT RETURN EXPECTED RESPONSE  
CODE -

CZUDH HRD ERR 02039 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 RECALIBRATE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE 7E  
 ACTUAL RESPONSE 00

The RECALIBRATE command did not return an expected response code. The expected response and actual response are in hex.

## 3.2.3.39 02040 - TIME-OUT ON SEND OF GET STATUS COMMAND TO DRIVE -

CZUDH HRD ERR 02040 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF GET STATUS COMMAND TO DRIVE

The GET STATUS command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

3.2.3.40 02041 - ERROR DURING RECEIVE OF GET STATUS RESPONSE FROM  
DRIVE -

CZUDH HRD ERR 02041 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF GET STATUS RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of a GET STATUS command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

## 3.2.3.41 02042 - GET STATUS COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02042 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 GET STATUS COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The GET STATUS command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

3.2.3.42 02043 - GET STATUS COMMAND DID NOT RETURN EXPECTED RESPONSE  
 CODE -

CZUDH HRD ERR 02043 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 GET STATUS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE F6  
 ACTUAL RESPONSE 00

The GET STATUS command did not return an expected response code. The expected response and actual response are in hex.

3.2.3.43 02044 - TIME-OUT ON SEND OF DRIVE CLEAR COMMAND TO DRIVE -

CZUDH HRD ERR 02044 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND OF DRIVE CLEAR COMMAND TO DRIVE

The DRIVE CLEAR command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

3.2.3.44 02045 - ERROR DURING RECEIVE OF DRIVE CLEAR RESPONSE FROM  
 DRIVE -

CZUDH HRD ERR 02045 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 ERROR DURING RECEIVE OF DRIVE CLEAR RESPONSE FROM DRIVE  
 explanation

This error message is presented if a receive of a DRIVE CLEAR command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

3.2.3.45 02046 - DRIVE CLEAR COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02046 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 DRIVE CLEAR COMMAND WAS UNSUCCESSFUL  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The DRIVE CLEAR command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

3.2.3.46 02047 - DRIVE CLEAR COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

CZUDH HRD ERR 02047 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 DRIVE CLEAR COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 EXPECTED RESPONSE 7E  
 ACTUAL RESPONSE 00

The DRIVE CLEAR command did not return an expected response code. The expected response and actual response are in hex.



### 3.2.4 TEST 3 INFORMATIONAL MESSAGES -

#### 3.2.4.1 LOGGABLE INFORMATION AFTER RECAL -

```
UNIT xx CONTROLLER AT car DRIVE plug  RUNTIME hhh:mm:ss  
LOGGABLE INFORMATION AFTER RECAL  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100
```

After sending a RECALIBRATE command, the ATTENTION bit was set. Test 3 then sent a GET STATUS command and found the LOGGABLE INFORMATION bit was set. This is not an error, it is only some information being sent from the drive. Normal operation continues. Check error 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5 TEST 3 ERROR MESSAGES (03000 TO 03999) -

## 3.2.5.1 03001 - TIME-OUT ON SEND -

CZUDH HRD ERR 03001 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT ON SEND  
 COMMAND WAS command  
 REAL TIME STATE state  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

If test 3 tries to send a level 2 command to the drive, and receiver ready is deasserted, error 3001 occurs. Where command is one of the following:

GET COMMON CHARACTERISTICS  
 ONLINE  
 DRIVE CLEAR  
 DISCONNECT  
 GET SUBUNIT CHARACTERISTICS  
 GET STATUS  
 CHANGE MODE  
 INITIATE RECLIBRATE  
 SPIN UP

## REAL TIME STATE state:

The real time state is the real time drive state <<AFTER>> Test 3 detected the error. <<THIS VALUE IS DISPLAYED IN HEX>>. In this example, receiver ready and attention are both asserted.

## The bit positions are defined as follows:

0001 - Receiver ready (Test 3 able to transmit to drive)  
 0002 - Attention (error occurred or online timeout expired)  
 0040 - Available (drive offline and usable)  
 1000 - Read/Write ready

The complete meaning of these bits is beyond the scope of this text, please refer to the operator documentation for the drive you are working on.

## STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0:

The status is the response to the SDI GET STATUS command. These words are printed in HEX. <<NOTE THAT THE STATUS IS PRINTED OUT FROM RIGHT TO LEFT!!>>. The status' meaning is beyond the scope of this text, please refer to the operator documentation for the drive you are working on.

## 3.2.5.2 03002 - TIME-OUT OF RECEIVE -

CZUDH HRD ERR 03002 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 TIME-OUT OF RECEIVE  
 COMMAND WAS GET COMMON CHARACTERISTICS  
 REAL TIME STATE state  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

This error is a failure of the drive to respond to an SDI level 2 command (see the SDI specification) before the drive-supplied command timeout expires. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.3 03003 - FIRST WORD RECEIVED WAS NOT A START FRAME -

CZUDH HRD ERR 03003 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 FIRST WORD RECEIVED WAS NOT A START FRAME  
 COMMAND WAS GET COMMON CHARACTERISTICS  
 REAL TIME STATE state  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The first word received by the controller from the drive was not a valid message start frame. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.4 03004 - FRAMING ERROR ON LEVEL 0 RESPONSE -

CZUDH HRD ERR 03004 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 FRAMING ERROR ON LEVEL 0 RESPONSE  
 COMMAND WAS GET COMMON CHARACTERISTICS  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

Error 3004 is caused by one or more of the following conditions: 1) Illegal frame code -- the frame is not a message start, continue, or end frame. 2) Illegal sequence of frames -- such as a message start frame without ever receiving a message end frame. This can be caused by the drive sending a response before the controller asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or controller receiver. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.5 03005 - CHECKSUM ERROR ON LEVEL 0 RESPONSE -

CZUDH HRD ERR 03005 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 CHECKSUM ERROR ON LEVEL 0 RESPONSE  
 COMMAND WAS GET COMMON CHARACTERISTICS  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The checksum attached to a message end frame did not match the checksum computed over the level 2 command. This could be caused by a bad drive transmitter, bad controller receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.6 03006 - RESPONSE LONGER THAN EXPECTED -

CZUDH HRD ERR 03006 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 RESPONSE LONGER THAN EXPECTED  
 COMMAND WAS GET COMMON CHARACTERISTICS  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The buffer size set aside for the response was not large enough for the response received. This is caused by the drive sending a response that is incorrect for the request sent to the drive, or the drive sending some garbage with the response. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.7 03007 - CODE FROM RECEIVE WAS UNINTELLIGIBLE FROM SUBSYSTEM = 0000 -

CZUDH HRD ERR 03007 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 CODE FROM RECEIVE WAS UNINTELLIGIBLE FROM SUBSYSTEM = 0000  
 COMMAND WAS GET COMMON CHARACTERISTICS  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The unknown error code occurs when the controller returns an error code from an operation that test 3 does not recognize. Possible controller microcode change without test 3 update. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.8 03008 - COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

CZUDH HRD ERR 03008 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 COMMAND DID NOT RETURN EXPECTED RESPONSE CODE  
 COMMAND WAS GET COMMON CHARACTERISTICS  
 EXPECTED RESPONSE 7E  
 ACTUAL RESPONSE 7D  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

This is caused by receiving an UNSUCCESSFUL response from the drive, or the drive sending some response other than the correct response for the request sent to the drive. See the contents of status for the unexpected response error (or reason). Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.9 03009 - DRIVE NOT ASSERTING RECEIVER READY IN DRIVE STATE -

CZUDH HRD ERR 03009 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 DRIVE NOT ASSERTING RECEIVER READY IN DRIVE STATE  
 REAL TIME STATE 0002  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

Test 3 inits the drive and checks the drive's real time state. If RECEIVER READY was not asserted after a period of time this error message is printed. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.10 03011 - NO VALID STATE FROM DRIVE -

CZUDH HRD ERR 03011 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
 NO VALID STATE FROM DRIVE  
 NO DRIVE CLOCKS  
 CHECK THAT DRIVE IS POWERED ON.

If test 3 attempts to get the drive state, and finds that there are no drive clocks on the port, the above message is occurs. This error usually means that the SDI cable is not connected, the drive is not powered on or the drive's port button that connects it to this controller is not depressed.

## 3.2.5.11 03012 - NO VALID STATE FROM DRIVE -

CZUDH HRD ERR 03012 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
NO VALID STATE FROM DRIVE  
HARD PARITY OR PULSE ERROR FOR 1/2 A SECOND

If test 3 attempts to get the drive state, and gets pulse or parity errors for a full 1/2 second, the above message is printed. This error usually indicates a poor connection or grounding of the SDI cables, a bad drive transmitter, a bad controller receiver or a broken SDI cable.

## 3.2.5.12 03014 - SUBUNIT CHARACTERISTICS SAY THERE ARE ZERO READ ONLY GROUPS -

CZUDH HRD ERR 03014 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
SUBUNIT CHARACTERISTICS SAY THERE ARE ZERO READ ONLY GROUPS  
IN THE DIAGNOSTIC AREA

After interrogating the subunit characteristics, test 3 finds out that the drive claims there are zero read only groups in the diagnostic area. There must be at least one for the test to run.

## 3.2.5.13 03015 - LESS THAN 1 READ/WRITE GROUP IN DIAGNOSTIC AREA -

CZUDH HRD ERR 03015 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
SUBUNIT CHARACTERISTICS SAY THERE ARE LESS THAN 1 READ/WRITE  
GROUPS IN THE DIAGNOSTIC AREA

After interrogating the subunit characteristics, test 3 finds out that the drive claims there are zero read/write groups in the diagnostic area. There must be at least one for the test to run.

## 3.2.5.14 03016 - NEITHER R/W READY NOR ATTENTION SET AFTER RECALIBRATE COMMAND -

CZUDH HRD ERR 03016 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
NEITHER R/W READY NOR ATTENTION SET AFTER RECALIBRATE COMMAND  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

After a RECALIBRATE command, R/W READY or ATTENTION did not set. Check the state for further information. This could be caused by a bad

transmitter or receiver or by a hit on the SDI cable. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.15 03017 - LESS THAN 1 DIAGNOSTIC CYLINDER -

CZUDH HRD ERR 03017 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
SUBUNIT CHARACTERISTICS SAY LESS THAN 1 DIAGNOSTIC CYLINDER

After interrogating the subunit characteristics, test 3 finds out that the drive claims there are zero diagnostic cylinders. There must be at least one for the test to run.

### 3.2.5.16 03018 - READ/WRITE READY DROPPED BEFORE FORMAT OPERATION -

CZUDH HRD ERR 03018 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
READ/WRITE READY DROPPED BEFORE FORMAT OPERATION  
CYLINDER aaa. GROUP bb. TRACK cc.  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The R/W READY signal was deasserted by the drive before a format operation was going to be sent by the controller. The drive may have gone off line or is not transmitting properly or the controller may not be receiving properly or the SDI cable took a hit.

Where:

aaa is the cylinder value in decimal.  
bb is the group value in decimal.  
cc is the track value in decimal.

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.17 03019 - FORMAT OPERATION REPORTED TIME-OUT FAILURE -

CZUDH HRD ERR 03019 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
FORMAT OPERATION REPORTED TIME-OUT FAILURE  
CYLINDER aaa. GROUP bb. TRACK cc.  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The format operation sent by the controller failed. The command timed out possibly due to receiver ready being dropped or communication problem (bad transmitter or receiver or hit on the SDI cable)

## Where:

aaa is the cylinder value in decimal.  
bb is the group value in decimal.  
cc is the track value in decimal.

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.18 03020 - AFTER RECAL, ERROR BITS WERE SET -

CZUDH HRD ERR 03020 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
AFTER RECAL, ERROR BITS WERE SET  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

After sending a RECALIBRATE command, the ATTENTION bit was set. Test 3 then sent a GET STATUS command and found the error bits were set. For further information, check the state and the status. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.19 03022 - READ/WRITE READY DROPPED BEFORE WRITE OPERATION -

CZUDH HRD ERR 03022 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
READ/WRITE READY DROPPED BEFORE WRITE OPERATION  
CYLINDER aaa. GROUP bb. TRACK cc.  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 u100

The R/W READY signal was deasserted by the drive before a write operation was going to be sent by the controller. The drive may have gone off line or is not transmitting properly or the controller may not be receiving properly or the SDI cable took a hit.

## Where:

aaa is the cylinder value in decimal.  
bb is the group value in decimal.  
cc is the track value in decimal.

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

## 3.2.5.20 03023 - COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK -

CZUDH HRD ERR 03023 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK. ON LAST BLOCK:  
WRITE OPERATION REPORTED FAILURE -- ERROR CODE aaa OCTAL.



DBN bbb. CYLINDER ccc. GROUP dd. TRACK ee.  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

After each track in the diagnostic space is formatted, at least one block must be able to have data written to it and read from it and the data must be correct. Not one block (DBN bbb.) from track (ee) was able to pass. The error code (aaa) gives the reason for the write operation failure.

Where:

aaa is the error code in octal.

It may have one of the following values:

- 2 = drive failure
- 3 = requested LBN is a secondary revector.

<<< NOTE >>> We are working with DBN's

- 4 = header compare failure  
(desired header not found)

133 = suspected positioner error

213 = read/write ready failure

253 = drive data or state clock timeout  
(indicates cable/transmitter/  
receiver broken)

313 = receiver ready timeout

413 = drive state receive error during write

bbb is the DBN in decimal.

ccc is the cylinder value in decimal.

dd is the group value in decimal.

ee is the track value in decimal.

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.21 03024 - READ/WRITE READY DROPPED BEFORE READ OPERATION -

CZUDH HRD ERR 03024 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
 DISK FUNCTION DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
 READ/WRITE READY DROPPED BEFORE READ OPERATION  
 CYLINDER aaa. GROUP bb. TRACK cc.  
 REAL TIME STATE 0003  
 STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The R/W READY signal was deasserted by the drive before a read operation was going to be sent by the controller. The drive may have gone off line or is not transmitting properly or the controller may not be receiving properly or the SDI cable took a hit.

Where:

aaa is the cylinder value in decimal.

bb is the group value in decimal.

cc is the track value in decimal.

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.22 03025 - READ OPERATION REPORTED FAILURE -

CZUDH HRD ERR 03025 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK. ON LAST BLOCK:  
READ OPERATION REPORTED FAILURE -- ERROR CODE aaa OCTAL.  
CYLINDER ccc. GROUP dd. TRACK ee.  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

After each track in the diagnostic space is formatted, at least one block must be able to have data written to it and read from it and the data must be correct. No block from track (ee) was able to pass. The error code (aaa) gives the reason for the read operation failure.

Where:

aaa is the error code in octal.

It may have one of the following values:

- 2 = drive failure
- 3 = requested LBN is a secondary revector.

<<< NOTE >>> We are working with DBN's

- 4 = header compare failure  
(desired header not found)

- 52 = SERDES overrun error
- 150 = data sync timeout on read
- 153 = suspected positioner error
- 213 = read/write ready failure
- 253 = drive data or state clock timeout  
(indicates cable/transmitter/receiver broken)

- 313 = receiver ready timeout

- 413 = drive state receive error during write

ccc is the cylinder value in decimal.

dd is the group value in decimal.

ee is the track value in decimal.

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.23 03026 - COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK -

CZUDH HRD ERR 03026 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT cer DRIVE plug RUNTIME hhh:mm:ss  
COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK. ON LAST BLOCK:  
DATA COMPARE FAILURE ON WORD aa.  
EXPECTED DATA bbbb  
ACTUAL DATA cccc

CYLINDER ddd. GROUP ee. TRACK ff.

After each track in the diagnostic space is formatted, at least one block must be able to have data written to it and read from it and the data must be correct. Not one block (DBN bbb.) from track (ee) was able to pass. The data read did not match the data written.

Where:

aa is the offset in decimal into the buffer where  
the error occurred.  
bbbb is the expected data in hex.  
cccc is the actual data in hex.  
ddd is the cylinder value in decimal.  
ee is the group value in decimal.  
ff is the track value in decimal.

#### 3.2.5.24 03027 - SEEK COMPLETE TIME-OUT -

CZUDH HRD ERR 03027 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
SEEK COMPLETE TIME-OUT -- READ/WRITE READY DID NOT SET  
SEEK WAS TO CYLINDER aaa. GROUP bb.  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

After a SEEK command has been successfully sent from the controller to the drive, the signal READ/WRITE READY must be set to indicate that the seek completed. If READ/WRITE READY never is asserted by the drive after the seek, the seek times out and error 3027 is presented.

Where:

aaa is the cylinder in decimal.  
bb is the group in decimal.

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

#### 3.2.5.25 03028 - NO BLOCK ON THIS TRACK CAN BE READ -

CZUDH HRD ERR 03028 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
NO BLOCK ON THIS TRACK CAN BE READ. LAST BLOCK TRIED:  
aBN bbbb. CYLINDER ccc. GROUP dd. TRACK ee.

After a seek to a track, at least one block must be able to be read to assure that test 3 can read the header. If not one block was successful, error message 3028 appears.

Where:

a is 'L' for LBN, 'D' for DBN, or 'X' for XBN.

bbbb is the block number in decimal.  
ccc is the cylinder in decimal.  
dd is the group number in decimal.  
ee is the track number in decimal.

### 3.2.5.26 03029 - AVAILABLE WAS NOT ASSERTED AFTER DISCONNECT -

CZUDH HRD ERR 03029 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
AVAILABLE WAS NOT ASSERTED AFTER DISCONNECT  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

After the DISCONNECT command was sent, the AVAILABLE flag should be asserted after a period of time. If it never was, then error 3029 appears. There maybe a problem with a transmitter or a receiver or the SDI cable at this point. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.27 03030 - INVALID LEVEL 2 COMMAND OPCODE Aaaa WAS SUCCESSFUL -

CZUDH HRD ERR 03030 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
INVALID LEVEL 2 COMMAND OPCODE aaaa WAS SUCCESSFUL  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

Some invalid level 2 commands are sent over the SDI. The drive should find these illegal commands and flag them as such. If the drive doesn't, then error 3030 will appear. "aaaa" is the invalid command in hex. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.28 03031 - COMMAND WITH Type LENGTH = A WAS SUCCESSFUL -

CZUDH HRD ERR 03031 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
COMMAND WITH type LENGTH = a WAS SUCCESSFUL  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

SDI level 2 commands with invalid lengths are sent to the drive to check if the drive can find them.

Where:

type could be 'COMMAND' or 'RESPONSE' for which  
field was affected  
a is the invalid length

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.29 03032 - UNIT DID NOT REPORT TRANSMISSION ERROR -

CZUDM HRD ERR 03032 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
UNIT DID NOT REPORT TRANSMISSION ERROR  
WHEN reason  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

Invalid level 1 sequences were sent to the drive. Several sequences are tried and the drive should find fault with everyone of them.

Where reason could be one of the following:

- AN END FRAME WAS SENT AFTER A START FRAME TIMED OUT
- A CONTINUE OR END FRAME DID NOT FOLLOW A START FRAME
- AN END FRAME WAS SENT WITH NO START FRAME
- AN END FRAME WITH A BAD CHECKSUM WAS SENT
- A CONTINUE FRAME WAS SENT WITH NO START FRAME

Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.30 03033 - UNIT ACCEPTED AN INVALID GROUP NUMBER -

CZUDM HRD ERR 03033 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
UNIT ACCEPTED AN INVALID GROUP NUMBER FROM GROUP SELECT LEVEL 1  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

A level 1 select group command with an illegal group number is sent to the drive. If the drive accepted it, then error 3033 will be displayed. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.31 03035 - SUCCESSFULLY WROTE ON DBN AREA WHEN DRIVE WAS WRITE PROTECTED -

CZUDM DVC FTL ERR 03035 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
SUCCESSFULLY WROTE ON DBN AREA WHEN DRIVE WAS WRITE PROTECTED  
REAL TIME STATE 0003  
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

An attempt was made to write on a write protected drive. It should

have resulted in an error response from the disk drive, but it didn't.  
Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

### 3.2.5.32 03036 - DRIVE NOT PROPERLY FORMATTED -

CZUDH DVC FTL ERR 03036 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
DRIVE IS NOT PROPERLY FORMATTED.  
CONTROLLER WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION  
THIS DRIVE NEEDS TO BE FORMATTED.

Test 3 reads a copy of the FCT in the XBN area and determined that the  
FCT was corrupted. Any normal operating system (which uses the UDA50A  
or KDA50-Q as a controller) will spin down the drive, so the drive  
will need to be reformatted.

### 3.2.5.33 03037 - DRIVE IS FORMATTED IN 576 BYTE MODE -

CZUDH DVC FTL ERR 03037 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
DRIVE IS FORMATTED IN 576 BYTE MODE.  
TO RUN WITH A controller, THIS DRIVE NEEDS TO BE FORMATTED IN 512 BYTE MODE.  
CONTROLLER WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION  
THIS DRIVE NEEDS TO BE FORMATTED.

Test 3 reads a copy of the FCT from the XBN area and determined that  
the drive was formatted in 576 byte mode. Any normal operating system  
(which uses the UDA50A or KDA50-Q as a controller) will spin down the  
drive, so the drive will need to be reformatted.

### 3.2.5.34 03038 - NO COPY OF THE FCT COULD BE READ -

CZUDH DVC FTL ERR 03038 ON UNIT 00 TST 003 SUB 000 PC: hostpc  
DISK FUNCTION DM PC: dmpc CONTROLLER AT car DRIVE plug RUNTIME hhh:mm:ss  
NO COPY OF THE FCT COULD BE READ.  
CONTROLLER WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION  
THIS DRIVE NEEDS TO BE FORMATTED.

Test 3 attempted to read every copy of the FCT without success. Any  
normal operating system (which uses the UDA50A or KDA50-Q as a  
controller) will spin down the drive, so the drive will need to be  
reformatted

## 3.2.6 SPECIAL DEVICE FATAL (05000) -

CZUDH DVC FTL ERR 05000 ON UNIT 00 TST 002 SUB 000 PC: hostpc  
 DISK zzzzzzzz DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss  
 UNABLE TO FIND REQUESTED DRIVE FOR TESTING  
 THE FOLLOWING IS VISIBLE ON THE PORTS  
 PORT 0 -- description  
 PORT 1 -- description  
 PORT 2 -- description  
 PORT 3 -- description

Where zzzzzzzz is either 'RESIDENT', 'FUNCION' or 'EXERCISER'. This message is presented when the specified drive was not found by test 2 or test 3 on any of the ports. A description of what was on each port follows.

## NO DRIVE ATTACHED

There is nothing on the port. If there is suppose to be a drive on this port, make sure there is an odd number of cables between the controller and the drive and make sure the cables are properly attached.

## RCVR RDY NEVER ASSERTED

The device on the port did not assert RCVR RDY while trying to get state.

## TIMEOUT OF SEND

Sending an SDI command timed out. RCVR RDY is not asserted.

## TIMEOUT OF RECEIVE

Receiving an SDI command timed out. The drive failed to respond to an SDI level 2 command before a timeout expired.

## FIRST WORD RECEIVED WAS NOT START FRAME

The first word received by the controller from the drive was not a valid message start frame.

## FRAMING ERROR ON LEVEL 0 RECEIVE

The device and the controller are out of sync or an illegal frame code (the frame is not a message start, continue, or end frame) or illegal sequence of frames. This can be caused by the drive sending a response before the controller asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or controller receiver.

## CHECKSUM ERROR ON LEVEL 0 RECEIVE

The checksum attached to a message end frame did not match the checksum computed over the level 2 command. This could be caused by a bad drive transmitter, bad controller receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable.

## RESPONSE LONGER THAN EXPECTED FOR CMD

The buffer size set aside for the response was not large enough for

the response received. This is caused by the drive sending a response that is incorrect for the request sent to the drive, or the drive sending some garbage with the response.

DRIVE n [further explanation]

A drive was found at the end of the cable. A further explanation may be presented. These further explanations are:

**DRIVE NOT AVAILABLE TO THIS CONTROLLER**

The drive was found but is not available to this controller. It may be dual ported and the drive is online to another controller.

**UNSPINABLE DRIVE**

The drive is unspinnable. The drive may be powered up but the RUN/STOP switch may be popped out.



3.3 DEC STANDARD 166 EXCERPTS

3.3.1 THE REPLACEMENT AND CACHING TABLES - The Replacement and Caching Tables record the locations of all revectored LBN sectors and the status of each RBN on the unit. Each copy of the table is organized in ascending RBN order, with an entry for each RBN sector on the unit. There are "n" copies of the table on the unit, where "n" is a device characteristic. The tables are stored at the high address end of the LBN area of the unit. Table entries (and RBNs) are allocated via a hash algorithm described later.

Replacement And Caching Table Format -

Each entry in the Replacement and Caching Table represents an RBN on the unit. The table is ordered in ascending RBN order. Thus the first entry corresponds to the first RBN on the unit, etc. The size of each copy of the table may exceed that required to contain an entry for each RBN on the unit since additional entries may be required to align the table so that adjacent copies can begin on a track boundary. Entries that do not correspond to RBNs on the unit are called "null entries"; there is always at least one null entry at the end of the RCT. All other entries past this last null entry are undefined.

NOTE

The RCT pad area is controller specific and should never be accessed by the host.

The format of a replacement block descriptor in the Replacement and Caching Tables is:

```

!<-----16 bits----->!
!-----!
!           LBN (low)           !
!-----!
! CODE !           LBN (high)  !
!-----!
! 4 bits!<-----12 bits----->!

```

Where:

LBN is the Logical Block Number of a revectored LBN sector.

CODE is one of the following octal values:

- 00 - Unallocated (empty) replacement block.
- 02 - Allocated replacement block - primary RBN.
- 03 - Allocated replacement block - non-primary RBN.
- 04 - Unusable replacement block.
- \* 05 - Alternate unusable replacement block
- 10 - Null entry - no corresponding RBN sector.

For codes 00, 04, and 10 the LBN field is always zero.

## NOTE

\* This code is reserved. Programs should treat this code as if it were code 04.

Embedded-controllers with no distinction between primary and secondary RBN's must use:

1. Code 02 if the replacement block can be retrieved with little degradation of performance for all blocks.
2. Code 03 if accessing the replacement block has a large impact on performance for all blocks.

3.3.2 FCT Structure - Each copy of the FCT is composed of one volume information block, one 512 byte format table, one 576 byte format table, and one subsystem temporary storage area (distributed amongst the alignment pads). An FCT copy has the following format:

volume information block	SECTOR 0
128 bad block descriptors 512 mode	SECTOR 1
128 bad block descriptors 512 mode	SECTOR 2
:	:
128 bad block descriptors 576 mode	SECTOR m
128 bad block descriptors 576 mode	SECTOR m+1
:	:
128 bad block descriptors 576 mode	SECTOR p
subsystem scratch storage	SECTOR p+1
:	:
subsystem scratch storage	SECTOR Fct-1

The XBN area itself is always formatted to contain 512 byte sectors. The calculations for m and p are:

$$m := (((Lc * g * t * r) + 1) / 2) + 127) / 128$$

$$p := 2 * m$$

Sector 0 contains various volume identification information. The format is:

media mode	WORD 0
formatting instance number	WORD 1
volume serial number ! least significant word!	WORD 2
volume serial number	WORD 3
volume serial number	WORD 4
volume serial number ! most significant word!	WORD 5
date that volume was ! first formatted (low)!	WORD 6
date that volume was ! first formatted	WORD 7
date that volume was ! first formatted	WORD 8
date that volume was ! first formatted (high)!	WORD 9
date of most recent ! volume formatting (low)!	WORD 10
date of most recent ! volume formatting	WORD 11
date of most recent ! volume formatting	WORD 12
date of most recent ! volum formatting (high)!	WORD 13
number of used entries! ! in 512 table (low) !	WORD 14

! number of used entries! ! in 512 table (high) !	WORD 15
! number of used entries! ! in 576 table (low) !	WORD 16
! number of used entries! ! in 576 table (high) !	WORD 17
! XBN of scratch area ! in this copy (low) !	WORD 18
! XBN of scratch area ! in this copy (high) !	WORD 19
! size of scratch area ! in this copy !	WORD 20
! zeros !	
! zeros !	WORD 255

**Where:**

**WORD 0:** "Media Mode" - is "126736" for a 512 byte format and "074161" for a 576 byte format. During formatting the media mode word is set to zero.

#### 4.0 PERFORMANCE AND PROGRESS REPORTS

At the end of each pass, the pass count is given along with the total number of errors reported since the diagnostic was started. The "EOP" switch can be used to control how often the end of pass message is printed. Section 2.2 describes switches.

A statistical report will automatically be printed periodically (approximately every fifteen minutes). It can be suppressed by setting the Inhibit Statistical Report flag (e.g. START/FLAGS:ISR). This is the same report that can be printed on demand with the PRINT command.

The report will look like the following example:

```
TEST 1 IN PROGRESS   RUN TIME 2:24:10
```

## 5.0 TEST SUMMARIES

The controller Host Resident Diagnostic consists of one PDP-11 diagnostic supervisor program that runs in the PDP-11 processor and three programs that run in the controller's buffer memory through an interpreter called the "diagnostic machine" which resides in the controller. The PDP-11 program mainly is responsible for downline loading the "diagnostic machine" programs into the controller and starting their execution. The "diagnostic machine" programs control the testing from that point by requesting the PDP-11 processor to supply information, print error messages and update statistics. The "diagnostic machine" programs inform the PDP-11 processor when a test is complete.

### 5.1 TEST 1 - BUS ADDRESSING TEST

The purpose of test 1 is to complete the testing of the Unibus interface in the controller. The controller resident diagnostic is not able to completely test the Unibus interface because communication with the PDP-11 processor is necessary. Specifically, this test will:

1. Check that every address line on the Unibus can be driven to both one and zero states.
2. Check that the controller can interrupt the PDP-11 processor at the proper priority level and vector.
3. Exercise the Unibus interface by transferring blocks of data to and from Unibus memory.

This test assumes that the following are being tested by the controller Resident Diagnostic:

1. All data bits can be written and read correctly.
2. NPR cycles can be executed correctly.

Test 1 is divided into eight subtests. One at a time, each controller selected for testing will run each subtest.

Subtest 1 makes sure that the controller IP and SA registers exist.

Subtest 2 initializes the controller through steps 1 and 2, and runs the first part of the controller's resident diagnostics.

Subtest 3 initializes the controller into diagnostic wrap mode. In this mode any value written into the SA register is echoed in the SA register.

Subtest 4 initializes the controller using the smallest size ring buffer allowed and interrupts disabled. This allows the controller

Resident Diagnostics to perform bus address testing. Any controller Resident Diagnostic errors will be reported.

In subtest 5, the controller is initialized with interrupts enabled. The vector address and priority level will be determined solely from the answers to the hardware questions. If the hardware vectors to the wrong address, it is impossible to determine the result. A descriptive error message of the problem will not occur (the program or processor may hang or an unrelated message may occur). Therefore, the message "TESTING INTERRUPT ABILITY OF CONTROLLER AT ADR csr VEC vec ..." is printed just before the controller is requested to cause an interrupt and the word "COMPLETED" is printed (on the same line) when the interrupt test is completed. If the word "COMPLETED" does not follow the first message, it should be apparent that the interrupt caused the diagnostic or processor to go astray. The priority level of the interrupt request is also verified.

Subtest 6 initializes the controller with a large ring buffer area and interrupts disabled. The large ring buffer area is supplied to allow the controller Resident Diagnostic to do the most bus address testing possible. Any controller Resident Diagnostic errors will be reported.

Subtest 7 downline loads the DM ADDRESSING TEST program into the controller and executes it. The DM addressing test tests the controller's ability to correctly address host memory. In order to test this ability, the DM addressing test consists of three unique parts. Each part is designed to catch a specific set of problems; once the test is complete, the user will have confidence that the controller will be able to correctly address all of host memory.

Part 1 - Read host memory with host cooperation  
Before the DM addressing test is started, the host identifies two unique regions of memory: 1 - Readable memory (usually the host identifies all addresses  $0 - 2^{*18} - 2$  as readable -- the DM test will tolerate NXMs) 2 - Readable AND writeable memory (usually this is a small area that the host sets aside for the unique use of the DM addressing test). Both areas MUST begin and END on word boundaries (NOTE: the word defined by the ending address IS used).

The host, just prior to starting the -DM addressing test, writes into each word of the WRITEABLE area the address of each location; for example, location 12564 will contain the number 12564, location 12566 will contain 12566, etc. If the writeable area is located in addressing space greater than  $2^{*16} - 2$ , only the low order 16 bits are stored in the word.

Once the writeable area is filled with the addresses of each location, the DM addressing test is downline-loaded into the controller. The DM test will first ask the host for the bounding addresses of readable and writeable memory. The DM will then read, one word at a time, each word in the writeable area, and compare its contents against the address that the DM attempted to read. If any mismatch is found between an address read and its contents, an error is printed.



Once the entire writeable area is read, Part 1 is complete.

#### Part 2 - Unique addressing test

The unique addressing test will drive as many addressing lines as possible to both one and zero, checking to make sure that as each one changes, a different and unique location is addressed. This will catch any addressing lines that may be stuck-at-one or stuck-at-zero.

The method used is as follows:

The first word in the writeable area is written with all ones. Now bit 1 of the writeable location's address is toggled. If the resulting address falls within the readable area, the resulting address is read. If a NXH results, or the contents of that address is NOT all ones, the test knows that it has accessed a unique and different location. If the contents of the read location IS all ones, the test then writes all zeroes to the first word of the writeable area. Now it reads the 'toggled' address again. If the contents are NOT zero, the test knows it has accessed a unique location. If the contents of the read location ARE zero, addressing bit 1 is known to be stuck, and an error is reported.

The above process is repeated for bits 2, 3, etc. up to bit 31. However, the 'toggled' address is NOT read from unless it falls within the readable area. For this reason, the readable area should be made as large as possible to facilitate testing.

#### Part 3 - Large buffer reads and writes

In the third part, the DM addressing test tries to do large buffer writes and reads to and from host memory. This is to stress the controller by causing it to do large transfers at a high speed.

The test determines if the internal buffer memory is either larger or smaller than the host writeable area. If the internal buffer is smaller, it will do several writes in order to fill up the host writeable area. Otherwise, it will scale the writes down to the size of the writeable area.

Three patterns will be written and read to and from host memory. The first pattern will be written until the entire writeable area is filled. Then the writeable area will be read back, and a data compare will be run against the data read. This will be repeated for the next two patterns. Once this test is complete, the entire DM addressing test is finished.

Subtest 8 performs extensive maintenance write and read operations between the host and the controller. Each location in the controller memory is written and read 32 times with a "shifting 1" and "shifting 0" data pattern to check for "stuck at 1" and "stuck at 0" errors. After each location in controller memory is read, it is overwritten with a different pattern to verify the memory is being addressed properly. The data read from the controller is compared to the pattern written to the controller. Any differences found indicate

errors.

The next controller selected for testing is then be tested in the same manner. When all controllers have been tested, test 1 ends.

## 5.2 TEST 2 - DISK RESIDENT DIAGNOSTIC TEST

The purpose of test 2 is to execute the diagnostics that run in each disk drive. These diagnostic programs are resident in the disk drive. This controller diagnostic program only knows the procedure to execute the disk resident diagnostics and how to determine whether a test passed or failed.

One at a time, each controller selected for testing is initialized and a "diagnostic machine" program downline loaded. The "diagnostic machine" program asks what drives are to be tested, then issues several commands to the disk drive and check for the correct response from the drive. This should serve as a good indicator that the controller and disk drive can communicate.

A DIAGNOSE command is then issued to the drive to request the drive run all of its diagnostics. If the disk drive requests a downline load of a drive diagnostic, the diagnostic program is read from the XXDP. load device, downline loaded into the disk drive and started. There is no limit to the number of downline loads that can be requested by a drive.

If the "Manual Intervention Mode" software question was answered "N" (default) testing proceeds to the next drive. When all drives on the controller have been tested, the next controller selected for testing is tested in the same manner. When all controller's have been tested, test 2 ends.

If the "Manual Intervention Mode" software question was answered "Y", an interactive mode is entered to allow the operator to perform diagnostic activities on the disk drive as desired. The Service Manual for the disk drive must be used to determine what diagnostic capabilities are available.

First, a brief description of available commands is printed as follows:

TEST #2 MANUAL INTERVENTION ON UNIT unit CONTROLLER AT csr DRIVE plug  
TO WRITE AND READ MEMORY:

W DATA REGION OFFSET  
R REGION OFFSET

TO RUN A DIAGNOSTIC:

D REGION

TO EXIT QUESTIONING:

E

DATA, REGION AND OFFSET ARE HEX VALUES.

?

Commands may be typed after the question mark prompt. Each command is processed as entered and results displayed immediately. The exit command will allow the diagnostic to proceed.

Read and write commands remember the region and offset values. Successive read and successive write commands automatically increment to the next offset if the region and offset values are not typed. If a region is typed but not an offset, offset zero is used.

Examples:

1. W FF FFFC 4
2. W 02
3. R FFFC 4  
FFFC 0004/ FF
4. R  
FFFC 0005/ 02
5. W 21 FFFC
6. R  
FFFC 0000/ 21

Command 1 writes one byte (FF) into region FFFC, offset 4. Command 2 writes one byte (02) into the next byte - region FFFC, offset 0005. Commands 3 and 4 read the bytes back. Command 5 writes one byte (21) into the first byte of region FFFC. Command 6 reads back that byte.

The diagnose command remembers the region from previous diagnose commands only, because the region containing the diagnostic is generally not the same region used to write parameters or read results. If the diagnostic returns any data, the data is printed immediately.

### 5.3 TEST 3 - DISK FUNCTION TEST

The purpose of test 3 is to functionally test the disk drive. On a drive that is well diagnosed by its disk resident diagnostics (executed by test 2) these functional tests will have little value. On a drive that has no or minimal resident diagnostics, these functional tests will have more value.

Test 3 starts by initializing each controller selected for testing and then downline loading a "diagnostic machine" program into each controller. Once all controllers have been started, the PDP-11 program responds to requests from all controllers. When all the controllers have indicated the end of testing, test 3 ends.

The "diagnostic machine" program performs the following functions on each drive:

1. Issue a DRIVE CLEAR command.
2. Issue RECALIBRATE command.
3. Issue a CHANGE MODE command to enable diagnostic cylinder access, set the drive to 512 byte sector size, and write protect.
4. Issue INITIATE SEEK command to last diagnostic cylinder.
5. Read all factory formatted sector headers. If no headers on a track can be read, report the error, otherwise continue.
6. Starting with cylinder 0, group 0 and incrementing through every cylinder on the disk, seek to a group, read a header on track 0 and then seek to the factory formatted diagnostic cylinder. Read from the diagnostic cylinder to verify disk positioned correctly.
7. Attempt to write on the first diagnostic cylinder while write protected.
8. Issue a CHANGE MODE command to enable formatting operations and disable write protect.
9. Format all writable DBNs in 512 byte format.
10. Write and read several data patterns to each writable DBN. Report an error if all DBNs on one track have an error.
11. Send invalid SDI level 2 and level 1 commands and check the results.
12. Go to the XBN area and read a copy of the FCT. Check to see if the drive has been properly formatted in 512 byte mode.

13. Issue a DISCONNECT command.

```

1      .NLIST BEX,CND
2      ;**
3      ;      REVISION HISTORY:
4      ;      REV. A - JFM - 12-SEP-1984
5      ;      THIS PROGRAM HAS BEEN ADAPTED FROM CZUDCO, REVISION E.
6      ;--
7      ;**
8      ;
9      ;      M A C R O   D E F I N I T I O N   S E C T I O N
10     ;--
11     ;**
12     ;      PUSH - PUT DATA ON THE STACK
13     ;
14     ;      ARGUMENTS:
15     ;      A - DATA TO BE PUT ON THE STACK
16     ;--
17     ;
18     .MACRO PUSH      A
19     .IRP      B,<A>
20     MOV      B,-(SP)      ; PUSH B ON STACK
21     .ENDM
22     .ENDM      PUSH
23
24     ;**
25     ;      POP - REMOVE DATA FROM THE STACK
26     ;
27     ;      ARGUMENTS:
28     ;      A - LOCATION TO PUT THE DATA REMOVED FROM THE STACK
29     ;--
30     ;
31     .MACRO POP      A
32     .IRP      B,<A>
33     MOV      (SP)+,B      ; POP STACK INTO B
34     .ENDM
35     .ENDM      POP
36
37     ;**
38     ;      ASSUME - CHECK VALIDITY OF PROGRAM ASSUMPTIONS
39     ;--
40     .MACRO ASSUME FIRST,CONDITION,SECOND
41     .IF CONDITION <FIRST>-<SECOND>
42     .IFF
43     .ERROR ;BAD ASSUME OF <FIRST> CONDITION <SECOND>
44     .ENDC
45     .ENDM      ASSUME
46
47     ;**
48     ;      MACRO DEFINITIONS FOR GLOBAL EQUATES
49     ;
50     ;      THESE MACROS ARE USED TO DEFINE INDEXES INTO A TABLE
51     ;
52     ;      CALLING SEQUENCE MUST BE
53     ;
54     ;      TABLE
55     ;      ITEM      NAME      BYTES      COMMENT
56     ;      ITEM      NAME      BYTES      COMMENT
57     ;      ITEM      NAME      BYTES      COMMENT

```

```

58      :           END      SIZE
59      :
60      :           TABLE - DESIGNATES THAT A TABLE IS ABOUT TO BE DEFINED.
61      :           END - TERMINATES THE DEFINITION.
62      :           ITEM - ENTRY IN THE TABLE. ANY NUMBER OF ITEM LINES CAN APPEAR.
63      :           NAME - THE NAME OF THE SYMBOL BEING EQUATED TO THE INDEX. THE INDEX
64      :                   ALWAYS STARTS AT ZERO.
65      :           BYTES - THE SIZE OF THE VALUE TO BE STORED AT THAT INDEX IN BYTES.
66      :           SIZE - (OPTIONAL) THE SIZE OF THE TABLE IN BYTES.
67      :           TINDEX - KEEPS TRACK OF THE INDEX VALUE AND WILL BE EQUAL TO THE SIZE
68      :                   OF THE TABLE AFTER THE END STATEMENT.
69      :--
70
71      .MACRO TABLE
72      TINDEX = 0
73      .ENDM TABLE
74
75      .MACRO ITEM NAME BYTES COMMENT
76      NAME=TINDEX ;COMMENT
77      TINDEX=TINDEX+BYTES
78      .ENDM ITEM
79
80      .MACRO END SIZE COMMENT
81      .IF NB SIZE
82      SIZE=TINDEX ;COMMENT
83      .ENDC
84      .ENDM END
85
86      ;**
87      :           PRINT - PRINT CHARACTER
88      :
89      :           ARGUMENT MUST BE SOURCE STATEMENT TO MOVE CHARACTER TO PRINT (MOV ARG,RO)
90      :                   EX: "PRINT R1" WILL PRINT THE CHARACTER IN R1
91      :           SPECIAL CASE: "PRINT #CR" WILL PRINT END OF LINE SEQUENCE
92      :           THE PRINTING IS DONE AT THE MODE OF THE LAST PRINT LINE CALL
93      :                   IE., PNTF, PNTB, PNTX, PNTS
94      :--
95
96      .MACRO PRINT ARG1
97      .IF DIF <ARG1>,RO
98      MOVB ARG1,RO ; STORE ARG1 IN RO AND
99      .ENDC
100     CALL PRINTC ; PRINT THE CHARACTER.
101     .ENDM PRINT
102
103     ;**
104     :           PNT... - PROCESSING MACRO FOR NEXT SET OF FORMATTED MESSAGE MACROS
105     :--
106
107     .MACRO PNT... RTN,ADR,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
108     ARG.CT = 0
109     .IRP AA,<ARG8,ARG7,ARG6,ARG5,ARG4,ARG3,ARG2,ARG1>
110     .IF NB,<AA>
111     MOV AA,-(SP) ; PUSH AA ON STACK
112     ARG.CT = ARG.CT+2 ; INCREMENT ARGUMENT COUNT
113     .ENDC
114     .ENDM

```

```

115          JSR      R1,RTN          ; CALL RTN PRINT ROUTINE
116          .WORD   ADR             ; ADDRESS OF ASCIZ STRING
117          .WORD   ARG.CT          ; ARGUMENT COUNT * 2
118          .ENDM   PNT...
119
120          ;**
121          ;       PNTF, PNTB, PNTX, PNTS - PRINT FORMATTED MESSAGE MACROS
122          ;
123          ;       USE THESE MACROS TO PRINT A FORMATTED MESSAGE
124          ;       FIRST ARGUMENT MUST BE ADDRESS OF FIRST CHARACTER OF MESSAGE STRING
125          ;       TO BE PUT INTO WORD (.WORD ARG)
126          ;       UP TO 8 SOURCE STATEMENTS MAY FOLLOW TO SPECIFY PARAMETERS TO BE
127          ;       USED BY THE FORMAT
128          ;--
129
130          .MACRO   PNTF      ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
131          PNT... LPNTF ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
132          .ENDM   PNTF
133
134          .MACRO   PNTB      ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
135          PNT... LPNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
136          .ENDM   PNTB
137
138          .MACRO   PNTX      ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
139          PNT... LPNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
140          .ENDM   PNTX
141
142          .MACRO   PNTS      ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
143          PNT... LPNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
144          .ENDM   PNTS
145
146          .MACRO   PNT       ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
147          PNT... LPNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
148          .ENDM   PNT
149
150          .SBTTL   PROGRAM HEADER
151
152          .MCALL   SVC
153 000000      SVC          ; INITIALIZE SUPERVISOR MACROS
154
155          ;**
156          ;       IF STRUCTURED MACROS ARE TO BE USED, ".MCALL STRUCT" AND "STRUCT"
157          ;       MUST BE ADDED TO INITIALIZE THE STRUCTURED MACROS.
158
159          000000      SVCINS= 0          ; LIST INSTRUCTIONS, SHIFTED RIGHT
160          000000      SVCTST= 0         ; LIST TEST TAGS, SHIFTED RIGHT
161          000000      SVCSUB= 0         ; LIST SUBTEST TAGS, SHIFTED RIGHT
162          000000      SVCGBL= 0        ; LIST GLOBAL TAGS, SHIFTED RIGHT
163          000000      SVCTAG= 0        ; LIST OTHER TAGS, SHIFTED RIGHT
164
165          ;       THE VALUES OF THE SVC... SYMBOLS ARE ZERO TO ALIGN THE MACRO CALLS
166          ;       AND THEIR EXPANSIONS.  SETTING THE SYMBOLS TO BE MINUS-ONE WILL CAUSE
167          ;       THE EXPANSIONS TO NOT BE LISTED.  THE SYMBOLS MAY BE CHANGED AT ANY
168          ;       POINT IN THE PROGRAM.
169          ;--
170
171 000000      .ASECT

```



172  
173 002000

.ENABL AMA  
" 2000

174  
175 :  
176 :  
177 :  
178 :  
179 :  
180 :  
181 :  
182 :  
183 :  
184 :  
185 :  
186 :  
187 :  
188 :  
189 :  
190 :  
191 :  
192 :  
193 :  
194 :  
195 :  
196 :  
197 :  
198 :  
199 :  
200 :  
201 :

\*\*\*  
THE PROGRAM HEADER IS THE INTERFACE BETWEEN  
THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.  
THE FOLLOWING ARE THE OPTIONAL 'HEADER' ARGUMENTS:

ARGUMENT	OPTION
-----	-----
BGNRPT	REPORT CODE
BGNSW	SOFTWARE TABLE
BGNSFT	SOFTWARE TABLE QUESTIONS
BGNAU	ADD CODE
BGNDU	DROP CODE
ERRTBL	ERROR TABLE
BGNSETUP	ASSEMBLED P-TABLES

192 002000

POINTER BGNRPT,BGNSW,BGNSFT,ERRTBL,BGNSETUP

\*\*\*  
THE "HEADER" ARGUMENTS ARE: NAME, REV, PATCH, LONGEST TEST  
TIME, TYPE, AND PRIORITY. "TYPE" = 0 FOR SEQUENTIAL DIAGNOSTIC AND = 1  
FOR EXERCISER. "PRIORITY" SPECIFIES THE PROCESSOR PRIORITY TO BE SET  
WHEN STARTING THE DIAGNOSTIC (DEFAULT IS 0).

002000  
002000  
002000 103  
002001 132  
002002 125  
002003 104  
002004 110  
002005 000  
002006 000  
002007 000  
002010  
002010 101  
002011  
002011 060  
002012  
002012 000001  
002014  
002014 000000  
002016  
002016 024662  
002020  
002020 025012  
002022  
002022 002134  
002024  
002024 002146  
002026  
002026 000044'

HEADER CZUDH,A,0,0,0,PRI07 ; TEST 1-3  
L\$NAME:: ;DIAGNOSTIC NAME  
.ASCII /C/  
.ASCII /Z/  
.ASCII /U/  
.ASCII /D/  
.ASCII /H/  
.BYTE 0  
.BYTE 0  
.BYTE 0  
L\$REV:: ;REVISION LEVEL  
.ASCII /A/  
L\$DEPO:: ;0  
.ASCII /0/  
L\$UNIT:: ;NUMBER OF UNITS  
.WORD T\$PTHV  
L\$TIML:: ;LONGEST TEST TIME  
.WORD 0  
L\$HPCP:: ;POINTER TO H.W. QUES.  
.WORD L\$HARD  
L\$SPCP:: ;POINTER TO S.W. QUES.  
.WORD L\$SOFT  
L\$HPTP:: ;PTR. TO DEF. H.W. PTABLE  
.WORD L\$HW  
L\$SPTP:: ;PTR. TO S.W. PTABLE  
.WORD L\$SW  
L\$LADP:: ;DIAG. END ADDRESS  
.WORD L\$LAST

CZUDHAO UDA50A/KDA50-G BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 84-4  
PROGRAM HEADER

002030		L\$STA::		;RESERVED FOR APT STATS
002030	000000		.WORD 0	
002032		L\$CO::		
002032	000000		.WORD 0	
002034		L\$DTYP::		;DIAGNOSTIC TYPE
002034	000000		.WORD 0	
002036		L\$APT::		;APT EXPANSION
002036	000000		.WORD 0	
002040		L\$DTP::		;PTR. TO DISPATCH TABLE
002040	002124		.WORD L\$DISPATCH	
002042		L\$PRIO::		;DIAGNOSTIC RUN PRIORITY
002042	000340		.WORD PRI07	
002044		L\$ENVI::		;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD 0	
002046		L\$EXP1::		;EXPANSION WORD
002046	000000		.WORD 0	
002050		L\$MREV::		;SVC REV AND EDIT #
002050	003		.BYTE C\$REVISION	
002051	003		.BYTE C\$EDIT	
002052		L\$EF::		;DIAG. EVENT FLAGS
002052	000000		.WORD 0	
002054	000000		.WORD 0	
002056		L\$SPC::		
002056	000000		.WORD 0	
002060		L\$DEVP::		; POINTER TO DEVICE TYPE LIST
002060	002342		.WORD L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	021334		.WORD L\$RPT	
002064		L\$EXP4::		
002064	000000		.WORD 0	
002066		L\$EXP5::		
002066	000000		.WORD 0	
002070		L\$AUT::		;PTR. TO ADD UNIT CODE
002070	000000		.WORD 0	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	000000		.WORD 0	
002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD 0	
002076		L\$DESP::		;PTR. TO DIAG. DESCRIPTION
002076	002366		.WORD L\$DESC	
002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT E\$LOAD	
002102		L\$ETP::		;PTR. TO ERRRTBL
002102	002150		.WORD L\$ERRTBL	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	021500		.WORD L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	022676		.WORD L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	022674		.WORD L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	021472		.WORD L\$PROT	
002114		L\$TEST::		;TEST NUMBER
002114	000000		.WORD 0	
002116		L\$DLY::		;DELAY COUNT
002116	000000		.WORD 0	
002120		L\$HIME::		;PTR. TO HIGH MEM

L7

SEQ 0089

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 84-5  
PROGRAM HEADER

203 002120 000000

.WORD 0

1  
2  
3  
4  
5  
6  
7  
8  
9  
10 002122  
002122 000003  
002124  
002124 022724  
002126 024524  
002130 024622  
12

.SBTTL DISPATCH TABLE

```

: **
: THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
: IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST. THE
: ARGUMENT OF 'DISPATCH' INDICATES THE NUMBER OF HARDWARE TESTS
: IN THE DIAGNOSTIC.
: --

```

```

DISPATCH 3
.WORD 3
L$DISPATCH::
.WORD T1
.WORD T2
.WORD T3

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 86  
 DEFAULT HARDWARE P-TABLE

```

1      .SBTTL  DEFAULT HARDWARE P-TABLE
2
3      : **
4      :       THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
5      :       THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
6      :       IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
7      :       AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
8      :       THE ACTUAL P-TABLE BUILT AT RUNTIME IS STORED IN SUPERVISOR
9      :       SPACE.
10     : --
11
12     002132      BGNHW  DFPTBL
13     002132      .WORD  L10000-L$HW/2
14     002134      L$HW::
15     002134      DFPTBL::
16     002134      .WORD  172150      ; CSR ADDRESS
17     002136      .WORD  154         ; VECTOR ADDRESS
18     002140      .WORD  5.          ; BR LEVEL
19     002142      .WORD  0.          ; LOGICAL DRIVE NUMBER
20
21
22     002144      ENDPHW
23     002144      L10000:

```

```

1      .SBTTL  SOFTWARE P-TABLE
2
3      ;**
4      ;      THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
5      ;      PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
6      ;      SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
7      ;      AT RUN TIME.  THIS TABLE, UNLIKE THE HARDWARE TABLE, WILL CONTAIN
8      ;      THE ACTUAL VALUES ENTERED BY THE OPERATOR.
9      ;--
10
11     002144      BGNSW  SFPTBL
12     002144      .WORD  L10001-L$SW/2
13     002146      L$SW::
14     002146      SFPTBL::
15     002146      .WORD  †B0100000100000000      ;SINGLE BIT QUESTIONS
16     002146      040400
17
18
19
20
21     002150      ENDSW
22     002150      L10001:

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 88  
 GLOBAL EQUATES SECTION

1  
 2  
 3  
 4  
 5  
 6  
 7  
 8 002150

```

.SBTTL GLOBAL EQUATES SECTION
; **
; 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



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 89  
CONTROLLER BIT DEFINITIONS

```

1      .SBTTL  CONTROLLER BIT DEFINITIONS
2
3      ;
4      ;      SA REGISTER UNIVERSAL READ BITS
5      ;
6
7      100000  SA.ERR  = 100000      ; ERROR INDICATOR
8      040000  SA.S4   = 040000      ; STEP 4 STATUS BIT
9      020000  SA.S3   = 020000      ; STEP 3 STATUS BIT
10     010000  SA.S2   = 010000      ; STEP 2 STATUS BIT
11     004000  SA.S1   = 004000      ; STEP 1 STATUS BIT
12
13     ;
14     ;      SA REGISTER ERROR STATUS BITS
15     ;
16
17     003777  SA.ERC  = 003777      ; ERROR CODE
18
19     ;
20     ;      SA REGISTER STEP 1 SEND BITS
21     ;
22
23     000177  SA.VEC  = 000177      ; INTERRUPT VECTOR (DIVIDED BY 4)
24     000200  SA.INT  = 000200      ; INTERRUPT? ENABLE DURING INIT
25     003400  SA.MSG  = 003400      ; MESSAGE RING LENGTH
26     034000  SA.CMD  = 034000      ; COMMAND RING LENGTH
27     040000  SA.WRP  = 040000      ; WRAP BIT
28     100000  SA.STP  = 100000      ; STEP - MUST ALWAYS BE WRITTEN A ONE
29
30     000400  SA.MS1  = 000400      ; LSB OF MESSAGE RING LENGTH
31     004000  SA.CM1  = 004000      ; LSB OF COMMAND RING LENGTH
32
33     ;
34     ;      SA REGISTER STEP 1 RESPONSE BITS
35     ;
36
37     002000  SA.NV   = 002000      ; NON SETTABLE INTERRUPT VECTOR
38     001000  SA.QB   = 001000      ; 22 BIT ADDRESS BUS
39     000400  SA.DI   = 000400      ; ENHANCED DIAGNOSTICS
40     000100  SA.MP   = 000100      ; MAPPING BIT
41     000040  SA.SM   = 000040      ; SPECIAL MODE BIT FOR KDA50-Q
42     ;          000377      ; THESE BITS RESERVED
43
44     ;
45     ;      SA REGISTER STEP 2 SEND BITS
46     ;
47
48     000001  SA.PRG  = 000001      ; ENABLE VAX UBA PURGE INTERRUPT
49     ;          177776      ; LOW ORDER MESSAGE RING BYTE ADDRESS
50
51     ;
52     ;      SA REGISTER STEP 2 RESPONSE BITS
53     ;
54
55     000007  SA.MSE  = 000007      ; MESSAGE RING LENGTH ECHO
56     000070  SA.CME  = 000070      ; COMMAND RING LENGTH ECHO
57     ;          000100      ; RESERVED

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 89-1  
 CONTROLLER BIT DEFINITIONS

58	000200	SA.STE = 000200	; STEP ECHO
59	003400	SA.CTP = 003400	; CONTROLLER TYPE
60			
61		:	
62		:	SA REGISTER STEP 3 SEND BITS
63		:	
64		:	
65		SA.TST = 077777	; HIGH ORDER MESSAGE RING BYTE ADDRESS
67	100000	SA.TST = 100000	; PURGE POLL TEST ENABLE
71		:	
72		:	SA REGISTER STEP 3 RESPONSE BITS
73		:	
74			
75	000177	SA.VCE = 000177	; INTERRUPT VECTOR ECHO
76	000200	SA.INE = 000200	; INTERRUPT ENABLE ECHO
77	000400	SA.NVE = 000400	; VECTOR NOT PROGRAMMABLE
78		:	; RESERVED
79			
80		:	
81		:	SA REGISTER STEP 4 SEND BITS
82		:	
83			
84	000001	SA.GO = 000001	; GO BIT TO START CONTROLLER FIRMWARE
85	000002	SA.LFC = 000002	; LAST FAILURE CODE REQUEST
86	000374	SA.BST = 000374	; BURST LEVEL
87			
88		:	
89		:	SA REGISTER STEP 4 RESPONSE BITS
90		:	
91			
92	000017	SA.MCV = 000017	; CONTROLLER MICROCODE VERSION
93	000360	SA.CNT = 000360	; CONTROLLER TYPE
94		:	; RESERVED
95			

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 90  
 HOST COMMUNICATION AREA DEFINIIONS

```

1      .SBTTL  HOST COMMUNICATION AREA DEFINIIONS
2
3      ;
4      ;      COMMAND/MESSAGE RING BIT DEFINITIONS
5      ;
6
7      100000  RG.OWN  = 100000      ; SET WHEN CONTROLLER OWNS RING
8      040000  RG.FLG  = 040000      ; FLAG BIT
9
10     ;
11     ;      VIRTUAL CIRCUIT IDENTIFIERS
12     ;
13
14     000000  MSCP    = 0           ; MSCP CIRCUIT
15     000001  LOG     = 1           ; LOG CIRCUIT
16     177777  DIAG    = -1         ; DIAGNOSTIC CIRCUIT
17     001000  DUP     = 1000        ; DIAGNOSTIC AND UTILITIES PROTOCOL
18
19     ;
20     ;      OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
21     ;      AND TWO PACKETS
22     ;
23     ;      -----
24     ;      HC.INT  }      INTERRUPT INDICATORS      }      4 BYTES
25     ;      }
26     ;      -----
27     ;      HC.MSG  }      MESSAGE (RESPONSE) RING    }      4 BYTES
28     ;      HC.MCT  }
29     ;      -----
30     ;      HC.CMD  }      COMMAND RING                }      4 BYTES
31     ;      HC.CCT  }
32     ;      -----
33     ;      HC.MEV & HC.CEV }      MESSAGE & COMMAND ENVELOPE }      4 BYTES
34     ;      }
35     ;      HC.MPK & HC.CPK }      MESSAGE & COMMAND PACKET }      48 BYTES
36     ;      }
37     ;      -----
38     ;      HC.BF1  }      BUFFER # 1 (RESPONSE TO DM PROGRAM) }      70 BYTES
39     ;      }
40     ;      -----
41     ;      HC.BF2  }      BUFFER # 2 (REQUEST FROM DM PROGRAM) }      70 BYTES
42     ;      }
43     ;      -----
44     ;
45     ;      NOTE: BYTES ARE GIVEN IN DECIMAL
46     ;
47
48     000004  HC.ISZ  = 4.           ; SIZE OF INTERRUPT INDICATOR WORDS
49     0C0004  HC.RSZ  = 4.           ; SIZE OF RING IN BYTES
50     000004  HC.ESZ  = 4.           ; SIZE OF ENVELOPE WORDS BEFORE PACKET
51     000060  HC.PSZ  = 48.          ; SIZE OF COMMAND AND MESSAGE PACKETS
52     000106  HC.BSZ  = 70.          ; SIZE OF BUFFER
53
54     000000  HC.INT  = 0.           ; INTERRUPT INDICATOR WORDS START
55
56     000004  HC.MSG  = HC.INT+HC.ISZ ; MESSAGE RING START
57     000006  HC.MCT  = HC.MSG+2.    ; MESSAGE RING CONTROL WORD

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 90-1  
HOST COMMUNICATION AREA DEFINIIONS

58				
59	000010	HC.CMD = HC.MSG+HC.RSZ		; COMMAND RING START
60	000012	HC.CCT = HC.CMD+2.		; COMMAND RING CONTROL WORDS
61				
62	000014	HC.MEV = HC.CMD+HC.RSZ		; MESSAGE ENVELOPE START
63	000020	HC.MPK = HC.MEV+HC.ESZ		; MESSAGE PACKET START
64				
65	000014	HC.CEV = HC.MEV		; COMMAND ENVELOPE START
66	000020	HC.CPK = HC.MPK		; COMMAND PACKET START
67				
68	000100	HC.BF1 = HC.CPK+HC.PSZ		; FIRST BUFFER
69	000206	HC.BF2 = HC.BF1+HC.BSZ		; SECOND BUFFER
70				
71	000314	HC.SIZ = HC.BF2+HC.BSZ		; TOTAL SIZE OF HOST COMMUNICATION AREA
72				

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 91  
 COMMAND PACKET OPCODES DEFINITIONS

```

1      .SBTTL  COMMAND PACKET OPCODES DEFINITIONS
2
3      ;**
4      ; NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING
5      ; THE END PACKET FLAG TO THE COMMAND OPCODE. FOR EXAMPLE, A READ
6      ; COMMAND'S END PACKET CONTAINS THE VALUE OP.RD+OP.END IN ITS OPCODE
7      ; FIELD. THE INVALID COMMAND END PACKET CONTAINS JUST THE END PACKET
8      ; FLAG (I.E., OP.END) IN ITS OPCODE FIELD. THE SERIOUS EXCEPTION END
9      ; PACKET CONTAINS THE SUM OF THE END PACKET FLAG PLUS THE SERIOUS
10     ; EXCEPTION OPCODE SHOWN ABOVE (I.E., OP.SEX+OP.END) IN ITS OPCODE FIELD.
11     ;
12     ; COMMAND OPCODE BITS 3 THROUGH 5 INDICATE THE COMMAND CLASS, WHICH IS
13     ; ENCODED AS FOLLOWS:
14     ; 000 IMMEDIATE COMMANDS
15     ; 001 SEQUENTIAL COMMANDS
16     ; 010 NON-SEQUENTIAL COMMANDS THAT DO NOT INCLUDE A BUFFER DESCRIPTOR
17     ; 100 NON-SEQUENTIAL COMMANDS THAT DO INCLUDE A BUFFER DESCRIPTOR
18     ;--
19
20     000001  OP.ABO  = 1      ; ABORT COMMAND
21     000020  OP.ACC  = 20     ; ACCESS COMMAND
22     000010  OP.AVL  = 10     ; AVAILABLE COMMAND
23     000021  OP.CCD  = 21     ; COMPARE CONTROLLER DATA COMMAND
24     000040  OP.CMP  = 40     ; COMPARE HOST DATA COMMAND
25     000022  OP.ERS  = 22     ; ERASE COMMAND
26     000023  OP.FLU  = 23     ; FLUSH COMMAND
27     000002  OP.GCS  = 2      ; GET COMMAND STATUS COMMAND
28     000003  OP.GUS  = 3      ; GET UNIT STATUS COMMAND
29     000011  OP.ONL  = 11     ; ONLINE COMMAND
30     000041  OP.RD   = 41     ; READ COMMAND
31     000024  OP.RPL  = 24     ; REPLACE COMMAND
32     000004  OP.SCC  = 4      ; SET CONTROLLER CHARACTERISTICS COMMAND
33     000012  OP.SUC  = 12     ; SET UNIT CHARACTERISTICS COMMAND
34     000042  OP.WR   = 42     ; WRITE COMMAND
35     000030  OP.MRD  = 30     ; MAINTENANCE READ COMMAND
36     000031  OP.MWR  = 31     ; MAINTENANCE WRITE COMMAND
37     000200  OP.END  = 200    ; END PACKET FLAG
38     000007  OP.SEX  = 7      ; SERIOUS EXCEPTION END PACKET
39     000100  OP.AVA  = 100    ; AVAILABLE ATTENTION MESSAGE
40     000101  OP.DUP  = 101    ; DUPLICATE UNIT NUMBER ATTENTION MESSAGE
41     000102  OP.SHC  = 102    ; SHADOW COPY COMPLETE ATTENTION MESSAGE
42     000103  OP.RLC  = 103    ; RESET COMMAND LIMIT ATTENTION MESSAGE
43
44     000001  OP.GSS  = 1      ; DUP GET DUST STATUS
45     000002  OP.ESP  = 2      ; DUP EXECUTE SUPPLIED PROGRAM
46     000003  OP.ELP  = 3      ; DUP EXECUTE LOCAL PROGRAM
47     000004  OP.SSD  = 4      ; DUP SEND DUST DATA
48     000005  OP.RSD  = 5      ; DUP RECEIVE DUST DATA
49
50     ;
51     ; COMMAND MODIFIERS
52     ;
53
54     ;      = 020000      ; CLEAR SERIOUS EXCEPTION
55     040000  MD.CMP  = 040000 ; COMPARE
56     100000  MD.EXP  = 100000 ; EXPRESS REQUEST
57     010000  MD.ERR  = 010000 ; FORCE ERROR

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 91-1  
 COMMAND PACKET OPCODES DEFINITIONS

58	004000	MD.SCH = 004000	; SUPPRESS CACHING (HIGH SPEED)
59	002000	MD.SCL = 002000	; SUPPRESS CACHING (LOW SPEED)
60	000100	MD.SEC = 000100	; SUPPRESS ERROR CORRECTION
61	000400	MD.SER = 000400	; SUPPRESS ERROR RECOVERY
62	000200	MD.SSH = 000200	; SUPPRESS SHADOWING
63	000100	MD.WBN = 000100	; WRITE-BACK (NON-VOLATILE)
64	000400	MD.WBV = 000400	; WRITE BACK (VOLATILE)
65	000020	MD.SEQ = 000020	; WRITE SHADOW SET ONE UNIT AT A TIME
66	000001	MD.SPD = 000001	; SPIN-DOWN
67	000001	MD.FEU = 000001	; FLUSH ENTIRE UNIT
68	000002	MD.VOL = 000002	; VOLATILE ONLY
69	000001	MD.NXU = 000001	; NEXT UNIT
70	000001	MD.RIP = 000001	; ALLOW SELF DESTRUCTION
71	000002	MD.IMF = 000002	; IGNORE MEDIA FORMAT ERROR
72	000004	MD.SWP = 000004	; SET WRITE PROTECT
73	000010	MD.CWB = 000010	; CLEAR WRITE-BACK DATA LOST
74	000001	MD.PRI = 000001	; PRIMARY REPLACEMENT BLOCK
75			
76		:	
77		; END PACKET FLAGS	
78		:	
79			
80	000200	EF.BBR = 000200	; BAD BLOCK REPORTED
81	000100	EF.BBU = 000100	; BAD BLOCK UNREPORTED
82	000040	EF.LOG = 000040	; ERROR LOG GENERATED
83	000020	EF.SEX = 000020	; SERIOUS EXCEPTION
84			
85		:	
86		; CONTROLLER FLAGS	
87		:	
88			
89	000200	CF.ATN = 000200	; ENABLE ATTENTION MESSAGES
90	000100	CF.MSC = 000100	; ENABLE MISCELLANEOUS ERROR LOG MESSAGES
91	000040	CF.OTH = 000040	; ENABLE OTHER HOST'S ERROR LOG MESSAGES
92	000020	CF.THS = 000020	; ENABLE THIS HOST'S ERROR LOG MESSAGES
93	000002	CF.SH0 = 000002	; SHADOWING
94	000001	CF.576 = 000001	; 576 BYTE SECTORS
95			
96		:	
97		; UNIT FLAGS	
98		:	
99			
100	000001	UF.CMR = 000001	; COMPARE READS
101	000002	UF.CMW = 000002	; COMPARE WRITES
102	100000	UF.RPL = 100000	; HOST INITIATED BAD BLOCK REPLACEMENT
103	040000	UF.INA = 040000	; INACTIVE SHADOW SET UNIT
104	004000	UF.SCH = 004000	; SUPPRESS CACHING (HIGH SPEED)
105	002000	UF.SCL = 002000	; SUPPRESS CACHING (LOW SPEED)
106	000100	UF.WBN = 000100	; WRITE-BACK (NON-VOLATILE)
107	020000	UF.WPH = 020000	; WRITE PROTECT (HARDWARE)
108	001000	UF.WPS = 001000	; WRITE PROTECT (SOFTWARE OR VOLUME)
109	000004	UF.576 = 000004	; 576 BYTE SECTORS
110			

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 92  
 COMMAND PACKET OFFSETS

```

1      .SBTTL  COMMAND PACKET OFFSETS
2
3      ;
4      ;      GENERIC COMMAND PACKET OFFSETS
5      ;
6
7      000000  P.CRF      = 0.      ; COMMAND REFERENCE NUMBER
8      000004  P.UNIT     = 4.      ; UNIT NUMBER
9      000010  P.OPCD     = 8.      ; OPCODE
10     000012  P.MOD      = 10.     ; MODIFIERS
11     000014  P.BCNT     = 12.     ; BYTE COUNT
12     000020  P.BUFF     = 16.     ; BUFFER DESCRIPTOR
13     000020  P.UADR     = 16.     ; BUS ADDRESS OF BUFFER DESCRIPTOR
14     000034  P.LBN      = 28.     ; LOGICAL BLOCK NUMBER
15
16     ;
17     ;      ABORT AND GET COMMAND STATUS COMMAND PACKET OFFSETS
18     ;
19
20     000014  P.OTRF     = 12.     ; OUTSTANDING REFERENCE NUMBER
21
22     ;
23     ;      ONLINE AND SET UNIT CHARACTERISTICS COMMAND PACKET OFFSETS
24     ;
25
26     000016  P.UNFL     = 14.     ; UNIT FLAGS
27     000020  P.HSTI     = 16.     ; HOST IDENTIFIER / RESERVED
28     000034  P.ELGF     = 28.     ; ERROR LOG FLAGS
29     000040  P.SHUN     = 32.     ; SHADOW UNIT
30     000042  P.CPSP     = 34.     ; COPY SPEED
31
32     ;
33     ;      REPLACE COMMAND PACKET OFFSETS
34     ;
35
36     000014  P.RBN      = 12.     ; REPLACEMENT BLOCK NUMBER
37
38     ;
39     ;      SET CONTROLLER CHARACTERISTICS COMMAND PACKET OFFSETS
40     ;
41
42     000014  P.VRSN     = 12.     ; MSCP VERSION
43     000016  P.CNTF     = 14.     ; CONTROLLER FLAGS
44     000020  P.HTMO     = 16.     ; HOST TIMEOUT
45     000022  P.USEF     = 18.     ; USE FRACTION
46     000024  P.TIME     = 20.     ; QUAD-WORD TIME AND DATE
47
48     ;
49     ;      MAINTENANCE READ AND MAINTENANCE WRITE COMMAND PACKET OFFSETS
50     ;
51
52     000034  P.RGID     = 28.     ; REGION ID
53     000040  P.RGOF     = 32.     ; REGION OFFSET
54
55     ;
56     ;      EXECUTE SUPPLIED PROGRAM COMMAND PACKET OFFSETS
57     ;

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 92-1  
COMMAND PACKET OFFSETS

58  
59  
60  
61

000024  
000034

P.DMDT = 20.  
P.OVRL = 28.

; DMDT TERMINAL ADDR. (MAINT WRITE ONLY)  
; BUFFER DESCRIPTOR FOR OVERLAYS



CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 93  
 END PACKET OFFSETS

```

1      .SBTTL END PACKET OFFSETS
2
3      ;
4      ;     GENERIC END PACKET OFFSETS
5      ;
6
7      000000      P.CRF      = 0.      ; COMMAND REFERENCE NUMBER
8      000004      P.UNIT     = 4.      ; UNIT NUMBER
9      000010      P.OPCD     = 8.      ; OPCODE (ALSO CALLED ENDCODE)
10     000011      P.FLGS     = 9.      ; END PACKET FLAGS
11     000012      P.STS      = 10.     ; STATUS
12     000014      P.BCNT     = 12.     ; BYTE COUNT
13     000034      P.FBBK     = 28.     ; FIRST BAD BLOCK
14
15     ;
16     ;     GET COMMAND STATUS END PACKET OFFSETS
17     ;
18
19     000014      P.OTRF     = 12.     ; OUTSTANDING REFERENCE NUMBER
20     000020      P.CHST     = 16.     ; COMMAND STATUS
21
22     ;
23     ;     GET UNIT STATUS END PACKET OFFSETS
24     ;
25
26     000014      P.MLUN     = 12.     ; MULTI-UNIT CODE
27     000016      P.UNFL     = 14.     ; UNIT FLAGS
28     000020      P.HSTI     = 16.     ; HOST IDENTIFIER
29     000024      P.UNTI     = 20.     ; UNIT IDENTIFIER
30     000034      P.MEDI     = 28.     ; MEDIA TYPE IDENTIFIER
31     000040      P.SHUN     = 32.     ; SHADOW UNIT
32     000042      P.SHST     = 34.     ; SHADOW STATUS
33     000044      P.TRKS     = 36.     ; TRACK SIZE
34     000046      P.GRPS     = 38.     ; GROUP SIZE
35     000050      P.CYLS     = 40.     ; CYLINDER SIZE
36     000054      P.RCTS     = 44.     ; RCT TABLE SIZE
37     000056      P.RBNS     = 46.     ; RBNS / TRACK
38     000057      P.RCTC     = 47.     ; RCT COPIES
39
40     ;
41     ;     ONLINE AND SET UNIT CHARACTERISTICS END PACKET AND AVAILABLE
42     ;     ATTENTION MESSAGE OFFSETS
43     ;
44
45     000014      P.MLUN     = 12.     ; MULTI-UNIT CODE
46     000016      P.UNFL     = 14.     ; UNIT FLAGS
47     000020      P.HSTI     = 16.     ; HOST IDENTIFIER
48     000024      P.UNTI     = 20.     ; UNIT IDENTIFIER
49     000034      P.MEDI     = 28.     ; MEDIA TYPE IDENTIFIER
50     000040      P.SHUN     = 32.     ; SHADOW UNIT
51     000042      P.SHST     = 34.     ; SHADOW STATUS
52     000044      P.UNSZ     = 36.     ; UNIT SIZE
53     000050      P.VSER     = 40.     ; VOLUME SERIAL NUMBER
54
55     ;
56     ;     SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS
57     ;

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 93-1  
 END PACKET OFFSETS

```

58
59      000014      P.VRSN = 12.      ; MSCP VERSION
60      000016      P.CNTF = 14.      ; CONTROLLER FLAGS
61      000020      P.CTMO = 16.      ; CONTROLLER TIMEOUT
62      000022      P.CSVR = 18.      ; CONTROLLER SOFTWARE VERSION
63      000023      P.CHVR = 19.      ; CONTROLLER HARDWARE VERSION
64      000024      P.CNTI = 20.      ; CONTROLLER ID
65
66      ;
67      ;      GET DUST STATUS END PACKET OFFSETS
68      ;
69
70      000014      P.DEXT = 12.      ; EXTENSION - DOWNLINE LOADABLE PROGRAM
71      000017      P.DFLG = 15.      ; FLAGS
72      000020      P.DPRG = 16.      ; PROGRESS INDICATOR FOR REMOTE PROGRAM
73      000024      P.DTMO = 20.      ; TIMEOUT
74
75      ;
76      ;      STATUS AND EVENT CODE DEFINITIONS
77      ;
78
79      000037      ST.MSK = 37      ; STATUS / EVENT CODE MASK
80      000040      ST.SUB = 40      ; SUB-CODE MULTIPLIER
81      000000      ST.SUC = 0       ; SUCCESS
82      000001      ST.CMD = 1       ; INVALID COMMAND
83      000002      ST.ABO = 2       ; COMMAND ABORTED
84      000003      ST.OFL = 3       ; UNIT-OFFLINE
85      000004      ST.AVL = 4       ; UNIT-AVAILABLE
86      000005      ST.MFE = 5       ; MEDIA FORMAT ERROR
87      000006      ST.WPR = 6       ; WRITE PROTECTED
88      000007      ST.CMP = 7       ; COMPARE ERROR
89      000010      ST.DAT = 10      ; DATA ERROR
90      000011      ST.HST = 11      ; HOST BUFFER ACCESS ERROR
91      000012      ST.CNT = 12      ; CONTROLLER ERROR
92      000013      ST.DRV = 13      ; DRIVE ERROR
93      000037      ST.DIA = 37      ; MESSAGE FROM AN INTERNAL DIAGNOSTIC
94      000400      ST.AOL = 400     ; ALREADY ON-LINE
95
96      ;
97      ;      DUP MESSAGE TYPES
98      ;
99
100     010000      DU.QUE = 10000    ; QUESTION
101     020000      DU.DFL = 20000    ; DEFAULT QUESTION
102     030000      DU.INF = 30000    ; INFORMATION
103     040000      DU.TER = 40000    ; TERMINATOR
104     050000      DU.FTL = 50000    ; FATAL ERROR
105     060000      DU.SPC = 60000    ; SPECIAL
106

```

```

1      .SBTTL CONTROLLER TABLE DEFINITIONS
2
3      ;**
4      ; CONTROLLER TABLE DEFINITIONS
5      ;
6      ; ONE TABLE WILL BE SET UP BY INITIALIZATION SECTION FOR EACH CONTROLLER
7      ; SELECTED FOR TESTING. TABLES ARE CONTIGUOUS. THE END OF THE TABLES IS
8      ; MARKED BY A WORD OF ZEROS.
9      ;
10     ; THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF CTABS.
11     ; THE NUMBER OF TABLES IS CONTAINED IN CTRLRS.
12     ;--
13
14     000077 CT.UNT = 000077 ; LOGICAL UNIT NUMBER MASK
15     000777 CT.VEC = 000777 ; VECTOR ADDRESS MASK
16     007000 CT.BRL = 007000 ; BR LEVEL MASK
17
18     100000 CT.AVL = BIT15 ; SET WHEN NOT AVAILABLE FOR TESTING
19     000100 CT.VER = BIT6 ; CONTROLLER MICRO VERSION NUMBER WAS
20     ; PRINTED IF SET
21     000020 CT.REQ = BIT4 ; BUFFER HAS BEEN GIVEN TO CONTROLLER
22     ; FOR REQUEST. SET WHENEVER READ
23     ; DUST DATA COMMAND ISSUED.
24     000010 CT.MSG = BIT3 ; MESSAGE RESPONSE RECEIVED. WHENEVER
25     ; SET, CT.CMD IS CLEARED.
26     000004 CT.CMD = BIT2 ; COMMAND ISSUED, WAITING FOR RESPONSE
27     000002 CT.RN = BIT1 ; DM PROGRAM RUNNING
28
29     TABLE ; START A TABLE DEFINITION
30     002150 ITEM C.UADR 2 <BUS ADDRESS OF IP REGISTER>
31     002150 ITEM C.UNIT 2 <UNIT NUMBER TO TEST>
32     002150 ITEM C.VEC 2 <VECTOR ADDRESS/BR LEVEL>
33     002150 ITEM C.JSR 2 <INTERRUPT SERVICE ROUTINE FOR CONTROLLER>
34     002150 ITEM C.JAD 2 <THESE TWO WORDS LOADED WITH [JSR RO CNTSRV]>
35     002150 ITEM C.FLG 2 <FLAGS>
36     002150 ITEM C.HCOM 2 <BEGINNING ADRS OF HOST COMM AREA IN MEMORY>
37     002150 ITEM C.DR0 2 <POINTER TO DRIVE TABLES>
38     002150 ITEM C.DR1 2 <IF ZERO, NO DRIVE TABLE EXISTS>
39     002150 ITEM C.DR2 2
40     002150 ITEM C.DR3 2
41     002150 ITEM C.TO 2 <TIMEOUT COUNTER>
42     002150 ITEM C.TOH 2 <(TWO WORDS)>
43     002150 ITEM C.REF 2 <COMMAND REFERENCE NUMBER>
44
45     002150 END C.SIZE <SIZE OF CONTROLLER TABLE IN BYTES>
46
    
```

```

1      .SBTTL  DRIVE TABLE DEFINITIONS
2
3      ;**
4      ;      DRIVE TABLE DEFINITIONS
5      ;
6      ;      ONE DRIVE TABLE WILL BE SET UP BY THE INITIALIZE SECTION FOR EACH
7      ;      DRIVE SELECTED FOR TESTING.  EACH TABLE IS POINTED TO BY A
8      ;      WORD IN THE CONTROLLER TABLE ON WHICH THE DRIVE EXISTS.
9      ;
10     ;      THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF DTABS.
11     ;--
12
13     000077      DT.UNT      = 000077      ; LOGICAL UNIT NUMBER OF DRIVE
14
15     100000      DT.AVL      = BIT15      ; SET WHEN NOT AVAILABLE FOR TESTING
16     040000      D.IM        = BIT14      ; INITIAL WRITE
17     020000      D.DCY       = BIT13      ; DIAGNOSTIC CYLINDERS
18     010000      D.ECC       = BIT12      ; ECC CORRECTION ENABLED
19     004000      D.RO        = BIT11      ; READ ONLY
20     002000      D.WO        = BIT10      ; WRITE ONLY
21     001000      D.RET       = BIT9       ; RETRIES ENABLED
22     000400      D.CYL       = BIT8       ; START/END CYLINDERS SPECIFIED
23     000100      D.SEQ       = BIT6       ; SEQUENTIAL ACCESS
24     000040      D.BE        = BIT5       ; BEGIN/END BLOCKS USED
25     000020      D.TR        = BIT4       ; WHEN D.BE=0: 1 - TRACKS, 0 - GROUPS
26     000010      D.WC        = BIT3       ; WRITE CHECKS ENABLED
27     000004      D.WCA       = BIT2       ; ALWAYS WRITE CHECK
28     000002      D.DC        = BIT1       ; DATA COMPARES ENABLED
29     000001      D.DCA       = BIT0       ; ALWAYS DATA COMPARE
30     011012      DDEF        = D.ECC.D.WC.D.DC.D.RET ; DEFAULT D.PRM
31     140200      D.ZERO      = BIT15.BIT7.D.IM ; BITS TO BE CLEARED
32
33     002150      TABLE      ; START A TABLE DEFINITION
34     002150      ITEM D.DRV   2      <DRIVE NUMBER>
35     002150      ITEM D.UNIT  2      <LOGICAL UNIT NUMBER>
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     002150      END D.SIZE   <SIZE OF DRIVE TABLE IN BYTES>
79
80
81     ;      DM PROGRAM HEADER DEFINITIONS
82     ;
83
84     000000      DMTRLN     = 0           ; OFFSET TO DOWNLINE LOAD PROGRAM SIZE
85     000004      DMOVRL     = 4           ; OFFSET TO SIZE OF OVERLAY
86     000040      DMMAIN     = 40          ; OFFSET TO FIRST WORD OF MAIN PROGRAM
87     001000      DMFRST     = 1000        ; ADDRESS START OF HEADER IN DM FILE
88

```

```

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          ERRTABL
9          LERRTABL::
10         ERRTABL: 000000 .WORD 0
11         ERRTABL: 000000 .WORD 0
12         ERRTABL: 000000 .WORD 0
13         ERRTABL: 000000 .WORD 0
14         ERRTABL: 000000 .WORD 0
15
16         FFREE:: .BLKW 1 ; FIRST FREE WORD IN MEMORY
17         FSIZE:: .BLKW 1 ; SIZE OF FREE MEMORY IN WORDS
18         FMEM: .BLKW 1 ; COPY OF FFREE AT END OF INIT SECTION
19         FMEMS: .BLKW 1 ; COPY OF FSIZE AT END OF INIT SECTION
20         DTABS:: .BLKW 1 ; START OF DRIVE TABLE STORAGE
21         CTABS:: .BLKW 1 ; START OF CONTROLLER TABLE STORAGE
22         CTRLRS: .BLKW 1 ; COUNT OF CONTROLLERS IN PTABLES
23         TSTTAB: .BLKW 1 ; POINTER TO 1ST CONTROLLER TABLE
24         DMPROG: .BLKW 1 ; START ADDRESS OF DM PROGRAM
25
26         IFLAGS:: .BLKW 1 ; FLAGS FROM INIT CODE
27
28         ICONT == BIT1 ; CONTINUE EVENT FLAG
29         IREST == BIT2 ; RESTART FLAG
30         ISTRT == BIT3 ; START FLAG
31         ISTRTH == BIT4 ; START FLAG HOLD FOR DMRQ4 ROUTINE
32
33         TNUM: .WORD 0 ; NUMBER OF TEST EXECUTING
34         URUN: .BLKW 1 ; NUMBER OF UNITS TO RUN AT ONE TIME
35         URNING: .BLKW 1 ; NUMBER OF UNITS STILL RUNNING
36         UCNT: .BLKW 1 ; COUNTER OF UNITS UNDER TEST
37         INTRCV: .BLKW 1 ; INTERRUPT RECEIVED FLAG
38
39         TEMP: .BLKW 12. ; TEMPORARY STORAGE FOR GMANI RESPONSES
40
41         IPADRS: .WORD 0 ; 4 ENTRIES
42         IPADRS: .WORD 0 ; FOR CONTROLLER
43         IPADRS: .WORD 0 ; CSR
44         IPADRS: .WORD 0 ; ADDRESSES
45
46
47         ;
48         ; KW11 CLOCK CONTROL
49         ;
50
51         KW.CSR: .WORD 0 ; CSR OF CLOCK
52         KW.BRL: .BLKW 1 ; BR LEVEL
53         KW.VEC: .BLKW 1 ; VECTOR
54         KW.HZ: .BLKW 1 ; HERTZ (50. OR 60.)
55         KW.EL: .BLKW 2 ; ELAPSED TIME
56         STIME: .BLKW 2 ; STATISTICAL REPORT TIMER
57
58         NXMAD: .BLKW 1 ; SET TO -1 BY NON-EXISTANT ADDRESS

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 96-1  
GLOBAL DATA SECTION

```

73
75 002300          T2WRR:  .BLKW 1          ; WRITE/READ REGION
76 002302          T2WRO:  .BLKW 1          ; WRITE/READ OFFSET
77 002304          T2DR:   .BLKW 1          ; DIAGNOSE REGION
79
80 002306 012252   PTYPE:  .WORD  PF      ; PRINT TYPE
81 002310          TTYOUT: .BYTE  0      ; TTY OUTPUT BUFFER
82 002311          .BYTE  0      ; TERMINATOR FOR ASCIZ STRING
83                .EVEN
84
85                ;
86                ;   DATA TO BE SENT AND RECEIVED BY CONTROLLER INITIALIZATION
87                ;
88
89 002312 020500   INITBL: .WORD  RSP.S1    ; 1ST WORD RESPONSE CHECK ROUTINE
90 002314 000000   SND.S1: .WORD  0        ; 1ST WORD TO SEND TO SA REGISTER
91 002316 020512   .WORD  RSP.S2    ; 2ND WORD RESPONSE CHECK ROUTINE
92 002320 000000   SND.S2: .WORD  0        ; 2ND WORD TO SEND TO SA REGISTER
93 002322 020532   .WORD  RSP.S3    ; 3RD WORD RESPONSE CHECK ROUTINE
94 002324 000000   SND.S3: .WORD  0        ; 3RD WORD TO SEND TO SA REGISTER
95                ;; .WORD  RSP.S4    ; 4TH WORD RESPONSE CHECK ROUTINE
96
97 002326 000000   SSTEP4: .WORD  0        ; LOCATION TO SAVE STEP 4 VALUE
98 002330 000000   CNTRSD: .WORD  0        ; LOCATION FOR STEP BIT MASK
100 002332         WCHNGD: .BLKW  1        ; SAVED LOOPBACK DATA
101 002334         BRLEV:  .BLKW  1        ; WORD FOR BR LEVEL STORAGE
102
103                T18MSZ = 36413      ; CONTROLLER MEMORY SIZE
104 002336         T18PAT: .BLKW  1        ; STORAGE FOR DATA PATTERN
105 002340         DMPTR:  .BLKW  1        ; STORAGE FOR DM PROGRAM POINTER
107

```

```

1      .SBTTL GLOBAL TEXT SECTION
2
3      ;**
4      ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5      ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
6      ; MORE THAN ONE TEST.
7      ;**
8
9      ;
10     ; NAMES OF DEVICES SUPPORTED BY PROGRAM
11     ;
12
13     002342      DEVTYP <LOGICAL DISK DRIVE>
14     002342      L$DVTYP:
15     002342      .ASCIZ /LOGICAL DISK DRIVE/
16              114      117      107      .EVEN
17
18     ;
19     002366      ; TEST DESCRIPTION
20     002366      ;
21     002366      DESCRIPT <CZUDHAO UDA50A,KDA50-Q BASIC SUBSY>
22              103      132      125      L$DESC:
23              .ASCIZ /CZUDHAO UDA50A,KDA50-Q BASIC SUBSY/
24              .EVEN
25
26     ;
27     ; UNFORMATTED MESSAGES
28     ;
29
30     002432      040      040      000      T4OPT7: .ASCIZ \ \
31
32     ;
33     ; FORMAT STATEMENTS USED IN PRINT CALLS
34     ;
35     002435      045      124      000      FRMTT: .ASCIZ \BT\
36     002440      045      116      000      CRLF: .ASCIZ \NN\
37     002443      042      040      040      RNTIM: .ASCIZ \ " RUNTIME "D16": "\
38     002466      104      071      042      RNTIM1: .ASCIZ \D9": "\
39     002474      104      071      000      RNTIM2: .ASCIZ \D9\
40     002477      042      040      040      ERRME1: .ASCIZ \ " * * * ERROR PROCESSING MESSAGE STRING * * * "\
41     002566      116      042      122      MXFERP: .ASCIZ \N"REACHED TRANSFER LIMIT - TESTING STOPPED"N\
42     002643      116      042      125      ERRLIM: .ASCIZ \N"UNIT "D6" REACHED ERROR LIMIT - UNIT DROPPED FROM TEST"N\
43     002736      116      042      124      INTST0: .ASCIZ \N"TESTING INTERRUPT ABILITY OF CONTROLLER AT ADR "016" VEC "09"..."\
44     003042      042      103      117      INTST1: .ASCIZ \ "COMPLETED"N\
45     003057      116      042      115      T2WARN: .ASCIZ \N"MANUAL INTERVENTION NOT ALLOWED. TEST 2 RUNNING UNATTENDED"N\
46     003156      116      042      124      T2CMS1: .ASCII \N"TEST 2 MANUAL INTERVENTION ON UNIT "D8" CONTROLLER AT "016" DRIVE "D9N\
47     003266      042      124      117      .ASCII \ "TO WRITE AND READ MEMORY: "N\
48     003322      042      040      040      .ASCII \ " W DATA REGION OFFSET"N\
49     003353      042      040      040      .ASCII \ " R REGION OFFSET"N\
50     003377      042      124      117      .ASCII \ "TO RUN A DIAGNOSTIC: "N\
51     003426      042      040      040      .ASCII \ " D REGION"N\
52     003443      042      124      117      .ASCII \ "TO EXIT QUESTIONING: "N\
53     003472      042      040      040      .ASCII \ " E"N\
54     003500      042      104      101      .ASCIZ \ "DATA, REGION AND OFFSET ARE HEX VALUES."N\
55     003553      042      077      040      T2CMS5: .ASCIZ \ "? INPUT ERROR"N\
    
```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 97-1  
GLOBAL TEXT SECTION

66	003574	116	042	125	MESSG:	.ASCIZ \N"UNIT "D6" CONTROLLER AT "016" DRIVE "D9S\ 67 003647 042 116 117 NOCLOCK:.ASCIZ \N"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\ 68 69 003724 042 110 117 BASNO: .ASCIZ \N"HOST PROGRAM"\ 71 003743 042 102 125 BASN1: .ASCIZ \N"BUS ADDRESSING"\ 72 003764 042 104 111 BASN2: .ASCIZ \N"DISK RESIDENT"\ 73 004004 042 104 111 BASN3: .ASCIZ \N"DISK FUNCTION"\ 78 004024 042 040 040 BASL1: .ASCIZ \N" DM PC:"012\ 79 004042 042 040 040 BASL2: .ASCIZ \N" CONTROLLER AT "016\ 80 004070 042 040 040 BASL3: .ASCIZ \N" DRIVE "D9\ 81 004105 000 BAS: .BYTE 0 ; NULL TO PRINT NOTHING 82 83 004106 122 066 122 BASLN: .ASCIZ \R6R6R6R6\ ; FORMAT FOR BASIC LINE OF ERROR MESSAGE 84 85 004117 X1A: 86 004117 X2A: 87 004117 X3A: 88 004117 042 111 116 X8A: .ASCIZ \N"INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS"N\ 90 004202 122 065 122 X1: .ASCIZ \R5R6"CONTROLLER HAS MORE THAN ONE VECTOR OR BR LEVEL"N\ 92 004271 122 065 122 X2: .ASCIZ \R5R6"TWO UNITS SELECT THE SAME DRIVE"N\ 93 004340 122 065 122 X3: .ASCIZ \R5R6"MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER"N\ 94 004426 122 064 042 X4: .ASCII \R4"NOT ENOUGH MEMORY TO TEST THE UNITS SELECTED"N\ 95 004507 042 122 105 .ASCIZ \N"RESTART PROGRAM AND TEST FEWER UNITS AT A TIME"N\ 96 004571 122 064 042 X6: .ASCIZ \R4"TABLE CONSISTANCY ERROR. PLEASE RE-LOAD PROGRAM"N\ 98 004656 122 065 122 X8: .ASCIZ \R5R6"TWO CONTROLLERS USE THE SAME VECTOR"N\ 100 004731 122 065 042 X14: .ASCII \R5"CONTROLLER IS NOT SUPPORTED BY THIS DIAGNOSTIC PROGRAM."N\ 101 005025 042 124 110 .ASCII \N"THIS PROGRAM REQUIRES A UDA50A (MODEL 6, MICROCODE VERSION" 102 005121 042 040 101 .ASCII \N" AT LEAST 3), "N\ 103 005141 042 117 122 .ASCII \N"OR A KDA50-Q (MODEL 13, MICROCODE VERSION AT LEAST 0)"N\ 104 005231 116 042 103 .ASCIZ \N"CONTROLLER REPORTED MODEL CODE "D4" AND MICROCODE VERSION "D4N\ 106 005332 122 065 042 X22: .ASCII \R5"STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION"N\ 107 005430 042 123 124 .ASCIZ \N"STEP BIT EXPECTED "016NR8R7\ 108 005465 122 065 042 X23A: .ASCII \R5"CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY " 109 005560 042 104 125 .ASCII \N"DURING INITIALIZATION"N\ 110 005610 104 071 042 .ASCII \D9" WORDS WERE TO BE CLEARED STARTING AT ADDRESS "016N\ 111 005676 042 124 110 .ASCII \N"THE FOLLOWING WORDS WERE NOT CLEARED:"N\ 112 005746 123 066 042 .ASCIZ \S6"ADDRESS"SA"CONTENTS"N\ 113 005777 123 067 117 X23B: .ASCIZ \S7016S5016N\ 114 006013 122 065 042 X24: .ASCII \R5"SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION"N\ 115 006123 042 120 125 .ASCIZ \N"PURGE/POLE DIAGNOSTICS WERE REQUESTED"NR8R7\ 116 006200 122 065 042 X25: .ASCII \R5"INCORRECT DATA RETURNED IN SA REGISTER DURING INITIALIZATION"N\ 117 006301 042 123 101 .ASCIZ \N"SA EXPECTED: "016NR8R7\ 118 006331 122 065 042 X26: .ASCII \R5"DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST"N\ 119 006424 042 104 101 .ASCII \N"DATA SENT TO SA REGISTER: "016N\ 120 006464 042 122 105 .ASCIZ \N"RECEIVED FROM SA REGISTER: "016NR7\ 121 006530 122 065 042 X27: .ASCII \R5"SA REGISTER DID NOT CHANGE AFTER WRITING TO IT"N\ 122 006613 042 111 116 .ASCIZ \N"IN PORT LOOP DIAGNOSTIC"NR8R7\ 123 006652 122 065 042 X28: .ASCIZ \R5"CONTROLLER DID NOT INTERRUPT THE HOST"NR7\ 124 006727 122 065 042 X29: .ASCII \R5"CONTROLLER INTERRUPTED AT DIFFERENT BR LEVEL THAN EXPECTED"N\ 125 007026 042 111 116 .ASCIZ \N"INTERRUPT WAS AT BR LEVEL "03N\ 127 007066 122 065 042 X31: .ASCII \R5"NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES"N\ 128 007156 042 101 123 .ASCIZ \N"ASSUME PROGRAM IS HUNG"N\ 129 007210 122 065 042 X32: .ASCIZ \R5"MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER"N\ 130 007321 122 065 042 X35: .ASCIZ \R5"DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE"N\ 131 007401 122 065 042 X36: .ASCII \R5"NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS"N\ 132 007472 042 127 110 .ASCIZ \N"WHILE LOADING DM PROGRAM"N\ 133 007526 122 065 042 X38: .ASCII \R5"MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS"N\ 
----	--------	-----	-----	-----	--------	--



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 97-2  
 GLOBAL TEXT SECTION

134	007613	042	103	110		.ASCII \ "CHECK BUS SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE" N \
135	007711	042	117	122		.ASCII \ "OR BUS" N \
136	007722	042	117	122		.ASCIZ \ "OR "R7" \
138	007732	122	065	042	X39:	.ASCII \R5"DATA COMPARE ERROR DURING CONTROLLER MEMORY TEST" N \
139	010017	042	103	117		.ASCII \ "CONTROLLER MEMORY ADDRESS: "016" N \
140	010060	042	104	101		.ASCIZ \ "DATA WRITTEN: "016" DATA READ: "016" N \
146	010126	122	065	042	XFCF:	.ASCIZ \R5"FATAL ERROR REPORTED BY CONTROLLER" NR8 \
147	010200	042	115	105	XMSG1:	.ASCIZ \ "MESSAGE BUFFER CONTAINS: " N \
148	010234	123	063	117	XMSG2:	.ASCIZ \S3016S1016S1016S1016S1016S1016S1016S1016N \
149	010301	122	065	042	XPKT1:	.ASCII \R5"RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA" N \
150	010404	042	105	111		.ASCII \ "EITHER CONTROLLER RETURNED ERROR STATUS OR PACKET WAS RECEIVED INCORRECTLY" N \
N \	151	010521	123	063	042	.ASCIZ \S3"COMMAND PACKET SENT" S6"RESPONSE PACKET RECEIVED" N \
	152	010606	123	066	117	XPKT2: .ASCIZ \S6016S1016S14016S1016N \
	153	010635	042	123	101	XSA: .ASCIZ \ "SA REGISTER CONTAINS: "016" N \
	154	010672	042	122	105	XFRU: .ASCIZ \ "REPLACE CONTROLLER PROCESSOR MODULE" N \
	155	010741	042	122	105	XFRUA: .ASCIZ \ "REPLACE CONTROLLER SDI MODULE" N \
	156					.EVEN
	157					

```

1      .SBTTL GLOBAL ERROR REPORT SECTION
2
3      ;**
4      ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS USED
5      ; BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PNTB
6      ; (BASIC) AND PNTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
7      ;--
8
9
10     011002 BGNMSG ERRO01
11     011002 ERRO01::
12     011002 PNTB X1,@X1A
13     011016 ENDMSG
14     011016 L10002: TRAP C#MSG
15     011016 104423
16
17     011020 BGNMSG ERRO02
18     011020 ERRO02::
19     011020 PNTB X2,@X2A
20     011034 ENDMSG
21     011034 L10003: TRAP C#MSG
22     011034 104423
23
24     011036 BGNMSG ERRO03
25     011036 ERRO03::
26     011036 PNTB X3,@X3A
27     011052 ENDMSG
28     011052 L10004: TRAP C#MSG
29     011052 104423
30
31     011054 BGNMSG ERRO04
32     011054 ERRO04::
33     011054 PNTB X4
34     011064 ENDMSG
35     011064 L10005: TRAP C#MSG
36     011064 104423
37
38     011066 BGNMSG ERRO06
39     011066 ERRO06::
40     011066 PNTB X6
41     011076 ENDMSG
42     011076 L10006: TRAP C#MSG
43     011076 104423
44
45     011100 BGNMSG ERRO08
46     011100 ERRO08::
47     011100 PNTB X8,@X8A
48     011114 ENDMSG
49     011114 L10007: TRAP C#MSG
50     011114 104423
51
52     011116 BGNMSG ERRO14
53     011116 ERRO14::
54     011116 PNTB X14,R3,R1
55     011132 ENDMSG
56     011132 L10010: TRAP C#MSG
57     011132 104423
58
59
60

```

42	011134				BGNMSG	ERR021	
	011134				ERR021::		
43	011134					PNTB	XFCE,R2
44	011146	010201				MOV	R2,R1
45	011150	000301				SWAB	R1
46	011152	042701	177775			BIC	#C<2>,R1
47	011156	001405				BEQ	ERR21A
48	011160					PNTB	XFRUA
49	011170	000404				BR	ERR21E
50	011172				ERR21A:		
51	011172					PNTB	XFRU
52	011202				ERR21E:		
53	011202				ENDMSG		
	011202				L10011:		
	011202	104423				TRAP	C#MSG
54							
55	011204				BGNMSG	ERR022	
	011204				ERR022::		
56	011204	042737	100000	002330		BIC	#SA.ERR,CNTRSD
57	011212					PNTB	X22,CNTRSD,R2
58	011230				ENDMSG		
	011230				L10012:		
	011230	104423				TRAP	C#MSG
59							
60	011232				BGNMSG	ERR023	
	011232				ERR023::		
61	011232					PNTB	X23A,R1,FFREE
62	011250	005742				TST	-(R2)
63	011252	005712			ERR23A:	TST	(R2)
64	011254	001406				BEQ	ERR23B
65	011256					PNTB	X23B,R2,(R2)
66					::	DEC	R4
67					::	BEQ	ERR23C
68	011272	005722			ERR23B:	TST	(R2)*
69	011274	005303				DEC	R3
70	011276	001365				BNE	ERR23A
71					::ERR23C:		
72	011300					PNTB	XFRU
73	011310				ENDMSG		
	011310				L10013:		
	011310	104423				TRAP	C#MSG
74							
75	011312				BGNMSG	ERR024	
	011312				ERR024::		
76	011312					PNTB	X24,R2
77	011324				ENDMSG		
	011324				L10014:		
	011324	104423				TRAP	C#MSG
78							
79	011326				BGNMSG	ERR025	
	011326				ERR025::		
80	011326					PNTB	X25,R1,R2
81	011342				ENDMSG		
	011342				L10015:		
	011342	104423				TRAP	C#MSG
82							
83	011344				BGNMSG	ERR026	

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 98-2  
GLOBAL ERROR REPORT SECTION

84	011344			ERR026::		
	011344			PNTB	X26,2(R4),R2	
85	011362			ENDMSG		
	011362			L10016:		
	011362	104423		TRAP	C#MSG	
86						
87	011364			BGNMSG ERR027		
	011364			ERR027::		
88	011364			PNTB	X27,2(R4)	
89	011400			ENDMSG		
	011400			L10017:		
	011400	104423		TRAP	C#MSG	
90						
91	011402			BGNMSG ERR028		
	011402			ERR028::		
92	011402			PNTB	X28	
93	011412			ENDMSG		
	011412			L10020:		
	011412	104423		TRAP	C#MSG	
94						
95	011414			BGNMSG ERR029		
	011414			ERR029::		
96	011414			PNTB	X29,R1	
97	011426			ENDMSG		
	011426			L10021:		
	011426	104423		TRAP	C#MSG	
99						
100	011430			BGNMSG ERR030		
	011430			ERR030::		
101	011430			PNTB	XFCE,R1	
102	011442			ENDMSG		
	011442			L10022:		
	011442	104423		TRAP	C#MSG	
103						
104	011444			BGNMSG ERR031		
	011444			ERR031::		
105	011444			PNTB	X31	
106	011454			ENDMSG		
	011454			L10023:		
	011454	104423		TRAP	C#MSG	
107						
108	011456			BGNMSG ERR032		
	011456			ERR032::		
109	011456			PNTB	X32	
110	011466	004737	011660	CALL	MSGPKT	
111	011472			ENDMSG		
	011472			L10024:		
	011472	104423		TRAP	C#MSG	
112						
113	011474			BGNMSG ERR033		
	011474			ERR033::		
114	011474	004737	011566	CALL	PNTPKT	
115	011500			ENDMSG		
	011500			L10025:		
	011500	104423		TRAP	C#MSG	
116						
117	011502			BGNMSG ERR035		

CZUDMAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 98-3  
GLOBAL ERROR REPORT SECTION

118	011502			ERR035::				
119	011512	004737	011660		PNTB	X35		
120	011516				CALL	MSGPKT		
	011516			ENDMSG				
	011516	104423		L10026:				
121					TRAP	C#MSG		
122	011520			BGNMSG	ERR036			
	011520			ERR036::				
123	011520				PNTB	X36		
124	011530			ENDMSG				
	011530			L10027:				
	011530	104423			TRAP	C#MSG		
125				BGNMSG	ERR038			
126	011532			ERR038::				
	011532				PNTB	X38		
127	011532			ENDMSG				
128	011542			L10030:				
	011542	104423			TRAP	C#MSG		
129				BGNMSG	ERR039			
131	011544			ERR039::				
	011544				PNTB	X39,R1,T18PAT,(R2)		
132	011544			ENDMSG				
133	011564			L10031:				
	011564	104423			TRAP	C#MSG		
140				PNTPKT:				
141	011566				PNTB	XPKT1		
142	011576	010401			MOV	R4,R1		
143	011600	062701	000020		ADD	#HC.CPK,R1		
144	011604	010402			MOV	R4,R2		
145	011606	062702	000020		ADD	#HC.MPK,R2		
146	011612	012703	000014		MOV	#12.,R3		
147				PNTPKL:				
148	011616				PNTB	XPKT2,2(R1),(R1),2(R2),(R2)		
149	011642	062701	000004		ADD	#4,R1		
150	011646	062702	000004		ADD	#4,R2		
151	011652	005303			DEC	R3		
152	011654	001360			BNE	PNTPKL		
153	011656	000207			RETURN			
154				MSGPKT:				
155	011660				PNTB	XMSG1		
156	011670	016504	000014		MOV	C.HCOM(R5),R4		
157	011674	062704	000206		ADD	#HC.BF2,R4		
158	011700	012703	000005		MOV	#5,R3		
159	011704			MSGPKL:				
160	011746	062704	000016		PNTB	XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)		
161	011752	005303			ADD	#14.,R4		
162	011754	001353			DEC	R3		
163	011756	000207			BNE	MSGPKL		
164					RETURN			
165	011760			BGNMSG	ERR.TN			: ERROR REPORT ROUTINE
	011760			ERR.TN::				
166	011760	013702	002204		MOV	TNUM,R2		: GET TEST NUMBER
167	011764	006302			ASL	R2		: DOUBLE
168	011766	012703	004070		MOV	#BASL3,R3		: GET ADDRESS OF DRIVE PRINT LINE



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 99  
GLOBAL SUBROUTINES SECTION

```

1      .SBTTL GLOBAL SUBROUTINES SECTION
2
3      :++
4      : FMERR - MEMORY ALLOCATION ERROR
5      :
6      : THIS ROUTINE PRINTS A SYSTEM FATAL ERROR AND EXITS THE TEST
7      :--
8
9      012120  FMERR: ERRSF 4, ,ERR004
10     012120  104454 TRAP C#ERSF
11     012122  000004 .WORD 4
12     012124  000000 .WORD 0
13     012126  011054 .WORD ERR004
14
15                                     ; DO CLEAN-UP TRAP
16
17     012130  DOCLN TRAP C#DCLN
18     012130  104444
19
20      :++
21      : ALOCM - ALLOCATE A BLOCK OF FREE MEMORY. REPORT ERROR IF MEMORY
22      : EXHAUSTED.
23      :
24      : INPUTS:
25      : R1 - NUMBER OF WORDS TO ALLOCATE
26      : FFREE - FIRST FREE WORD IN MEMORY
27      : FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS
28      :
29      : OUTPUTS:
30      : R1 - ADDRESS OF FIRST WORD OF ALLOCATED MEMORY
31      : FFREE - NEW FIRST FREE WORD IN MEMORY
32      : FSIZE - SIZE OF FREE MEMORY LEFT AFTER ALLOCATION
33      :
34      : SYSTEM FATAL ERROR WILL BE REPORTED IF NOT ENOUGH MEMORY AVAILABLE
35      : AND ENTIRE PROGRAM WILL BE STOPPED.
36      :--
37
38     012132  ALOCM: PUSH <FFREE> ; SAVE FFREE AT ENTRY
39     012136  160137 002162 SUB R1,FSIZE ; REDUCE SIZE OF FREE MEMORY
40     012142  002766 BLT FMERR ; REPORT ERROR IF NOT ENOUGH MEMORY
41     012144  060101 ADD R1,R1 ; CHANGE WORDS TO BYTES
42     012146  060137 002160 ADD R1,FFREE ; CALCULATE NEW START OF FREE MEMORY
43     012152  000207 POP <R1> ; GET START OF ALLOCATED MEMORY
44     012154  000207 RETURN
45
46      :++
47      : HCOMM - ALLOCATE MEMORY FOR HOST COMMUNICATION AREA AND PACKET
48      : BUFFERS WITH ONE DESCRIPTOR IN EACH RING. THIS SUBROUTINE IS CALLED
49      : AFTER INITIALIZING A CONTROLLER WITH SA.MSG=0 AND SA.CMD=0.
50      :
51      : INPUTS:
52      : R5 - ADDRESS OF CONTROLLER TABLE
53      :
54      : OUTPUTS:
55      : CONTROLLER TABLE POINTING TO HOST COMMUNICATION AREA,
56      : RING POINTERS TO PACKETS,
57      : R4 - ADDRESS OF HOST COMMUNICATION AREA
58      :--
59
60     012156  HCOMM: MOV #<HC.SIZ>/2,R1 ; GET SIZE OF AREA TO ALLOCATE
61     012162  012701 000146 CALL ALOCM ; ALLOCATE THE MEMORY
62     012162  004737 012132

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 99-1  
GLOBAL SUBROUTINES SECTION

```

53 012166 010104          MOV      R1,R4          ; GET ADDRESS OF HOST COMM AREA
54 012170 010465 000014  MOV      R4,C.HCOM(R5) ; PLACE BEGINNING ADRS OF HOST COMM
55                                     ; AREA IN THE CONTROLLER TABLE
56 012174 062701 000020  ADD      @HC.MPK,R1     ; COMPUTE START OF MESSAGE PACKET
57 012200 010164 000004  MOV      R1,HC.MSG(R4)  ; PLACE IN RING
58 012204 062701 000000  ADD      @<HC.CPK-HC.MPK>,R1 ; COMPUTE START OF COMMAND PACKET
59 012210 010164 000010  MOV      R1,HC.CMD(R4)  ; PLACE IN RING
60 012214 000207          RETURN

61
62 ;**
63 ; PRINTC - PRINT A CHARACTER
64 ;
65 ; CALL WITH MACRO PRINT
66 ;--
67
68 012216 110037 002310  PRINTC: MOV      RO,TTYOUT          ; SAVE CHARACTER FOR TTY OUTPUT
69 012222          PUSH      <R1>
70 012224 012701 002435  MOV      @FRMTT,R1     ; GET FORMATTED ASCIZ STRING STATEMENT
71 012230 120027 000015  CMPB    RO,@CR         ; IF NOT A CARRIAGE RETURN, THEN
72 012234 001002          BNE     1$             ; PRINT SOME OTHER CHARACTER, ELSE
73 012236 012701 002440  MOV      @CRLF,R1     ; GET FORMATTED ASCIZ STRING STATEMENT
74                                     ; GO PRINT CR-LF.
75 012242 004777 170040  1$:     CALL      @PTYPE          ; PRINT THE ASCIZ STRING.
76 012246          POP      <R1>
77 012250 000207          RETURN
78 012252          PF:     PRINTF   R1,@TTYOUT
79 012252 012746 002310  MOV      @TTYOUT,-(SP)
80 012256 010146          MOV      R1,-(SP)
81 012260 012746 000002  MOV      @2,-(SP)
82 012264 010600          MOV      SP,RO
83 012266 104417          TRAP    C@PNTF
84 012270 062706 000006  ADD      @6,SP
85 012274 000207          RETURN
86 012276          PB:     PRINTB   R1,@TTYOUT
87 012276 012746 002310  MOV      @TTYOUT,-(SP)
88 012302 010146          MOV      R1,-(SP)
89 012304 012746 000002  MOV      @2,-(SP)
90 012310 010600          MOV      SP,RO
91 012312 104414          TRAP    C@PNTB
92 012314 062706 000006  ADD      @6,SP
93 012320 000207          RETURN
94 012322          PX:     PRINTX   R1,@TTYOUT
95 012322 012746 002310  MOV      @TTYOUT,-(SP)
96 012326 010146          MOV      R1,-(SP)
97 012330 012746 000002  MOV      @2,-(SP)
98 012334 010600          MOV      SP,RO
99 012336 104415          TRAP    C@PNTX
100 012340 062706 000006  ADD      @6,SP
101 012344 000207          RETURN
102 012346          PS:     PRINTS   R1,@TTYOUT
103 012346 012746 002310  MOV      @TTYOUT,-(SP)
104 012352 010146          MOV      R1,-(SP)
105 012354 012746 000002  MOV      @2,-(SP)
106 012360 010600          MOV      SP,RO
107 012362 104416          TRAP    C@PNTS
108 012364 062706 000006  ADD      @6,SP
109 012370 000207          RETURN

```



CZUDMAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 99-2  
GLOBAL SUBROUTINES SECTION

```

86
87
88      ;**      PRINT FORMATTED MESSAGE
89      ;
90      ;      CALL WITH MACRO PNT, PNTF, PNTB, PNTX, OR PNTS
91      ;--
92
93 012372 012737 012252 002306 LPNTF: MOV    @PF,PTYPE
94 012400 000413                BR     LPNT
95
96 012402 012737 012276 002306 LPNTB: MOV    @PB,PTYPE
97 012410 000407                BR     LPNT
98
99 012412 012737 012322 002306 LPNTX: MOV    @PX,PTYPE
100 012420 000403                BR     LPNT
101
102 012422 012737 012346 002306 LPNTS: MOV    @PS,PTYPE
103
104 012430                LPNT:  PUSH   <R2,R3,R4,R5>
105 012440 012102                MOV    (R1),R2      ; GET ADDRESS OF ASCIZ STRING
106 012442 010604                MOV    SP,R4      ; COMPUTE ADDRESS OF 1ST ARGUMENT AND
107 012444 062704 000012        ADD    @12,R4     ; SAVE IT IN R4.
108 012450                PUSH   <R1>      ; SAVE R1 ON THE STACK
109 012452 004737 012476        CALL   OSTRNG    ; PRINT THE FORMATTED MESSAGE
110 012456                POP     <R0,R5,R4,R3,R2,R1> ; RESTORE REGISTERS FROM STACK
111 012472 062006                ADD    (R0),SP   ; ADJUST STACK POINTER OVER ARGUMENTS
112 012474 000110                JMP    BRO      ; RETURN
113
114      ;**
115      ;      OSTRNG - OUTPUT A MESSAGE ACCORDING TO A FORMAT STRING
116      ;
117      ;      FORMAT OF THE ASCIZ STRING IS AS FOLLOWS:
118      ;
119      ;      CHARACTERS ENCLOSED IN QUOTES ARE TO BE PRINTED AS THEY ARE.
120      ;
121      ;      OTHERWISE CODE IS A SINGLE LETTER FOLLOWED BY AN OPTIONAL DECIMAL
122      ;      NUMBER:
123      ;      ON - PRINT OCTAL NUMBER. N REPRESENTS SIZE OF BINARY NUMBER PASSED
124      ;      IN PARAMETER IN BITS. MAY BE IN RANGE 1 TO 32. IF N>16, TWO
125      ;      PARAMETER WORDS ARE USED, OTHERWISE ONLY ONE WORD. LEADING ZEROS
126      ;      ARE PRINTED. N IS ALWAYS SPECIFIED.
127      ;      DN - PRINT UNSIGNED DECIMAL NUMBER FROM N BIT PARAMETER. LEADING
128      ;      ZEROS ARE NOT PRINTED. A 16 BIT NUMBER EQUAL TO ZERO WILL
129      ;      PRINT "0".
130      ;      HN - PRINT HEX NUMBER FROM PARAMETER OF N BITS. IF N>16 TWO
131      ;      PARAMETERS ARE USED, OTHERWISE ONLY ONE PARAMETER. LEADING
132      ;      ZEROS ARE PRINTED.
133      ;      SN - PRINT N SPACES. N ASSUMED TO BE 1.
134      ;      NN - START NEW LINE (CR-LF SEQUENCE). N ASSUMED TO BE 1.
135      ;      AN - PRINT N ASCII CHARACTERS FROM PARAMETERS, N ASSUMED TO BE 1.
136      ;      N/2 PARAMETER WORDS USED.
137      ;      RN - EXECUTE ROUTINE @N. N MUST BE GIVEN AND DEFINED IN HOST PROGRAM.
138      ;
139      ;      A NULL CHARACTER MEANS END OF MESSAGE. A NULL AS FIRST CHARACTER IN
140      ;      STRING MUST BE IGNORED.
141      ;
142      ;      INPUTS:

```

CZUDMAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 99-3  
GLOBAL SUBROUTINES SECTION

```

143      ;           R2 - ADDRESS OF START OF FORMAT STRING
144      ;           R4 - ADDRESS OF PARAMETERS
145      ;           OUTPUTS:
146      ;           R2 AND R4 UPDATED TO END OF STRING AND PARAMETERS
147      ;--
148
149 012476 112201      OSTRNG: MOV  B (R2),R1      ; SEE IF TERMINATOR IN ASCIZ STRING.
150 012500 001421      BEQ  B OSTRE      ; EXIT
151 012502 012700 013006  MOV  B #ERRC,R0      ; GET POINTER TO CHARACTER TABLE
152 012506 120110      NCONS: CMP  B R1,(R0)      ; COMPARE CHARACTER WITH TABLE ENTRY
153 012510 001407      BEQ  B NCONF      ; BRANCH IF MATCH FOUND
154 012512 105720      TST  B (R0),      ; INCREMENT POINTER
155 012514 001374      BNE  B NCONS      ; CONTINUE SEARCH IF NOT END OF TABLE
156 012516      PNTF  ERRME1      ; REPORT BAD CONTROL CHARACTER
157 012526 000406      BR   B OSTRE      ;
158 012530 162700 013006  NCONF: SUB  B #ERRC,R0      ; GET INCREMENT INTO TABLE
159 012534 006300      ASL  B R0      ; DOUBLE TO WORD COUNT
160 012536 004770 013020  CALL BERRD(R0)      ; DISPATCH TO PRINT ROUTINE
161 012542 000755      BR   B OSTRNG      ; GET NEXT
162 012544 000207      OSTRE: RETURN
163
164      ;
165      ; CONTROL CHARACTER WAS A QUOTE, SO PRINT ALL CHARACTERS TO
166      ; THE NEXT QUOTE.
167      ;
168
169 012546 112200      CON.QU: MOV  B (R2),R0      ; GET CHARACTER
170 012550 120027 000042  CMP  B R0,#'"      ; CHECK IF ENDING QUOTE
171 012554 001403      BEQ  B CON.QX      ; IF SO, GO GET NEXT CONTROL CHARACTER
172 012556      PRINT B R0      ;
173 012562 000771      BR   B CON.QU      ; CONTINUE PRINTING
174 012564 000207      CON.QX: RETURN
175
176      ;
177      ; CONTROL CHARACTER WAS AN 'A', SO PRINT ASCII CHARACTERS FROM
178      ; PARAMETERS.
179      ;
180
181 012566 004737 015474  CON.A: CALL  GETCNT      ; GET COUNT OF CHARACTERS
182 012572      CON.A1: PRINT (R4),      ;
183 012600 005301      DEC  B R1      ; COUNT THE CHARACTERS
184 012602 001373      BNE  B CON.A1      ; PRINT UNTIL COUNT REACHES ZERO
185 012604 032704 000001  BIT  B #1,R4      ; CHECK IF R4 NOW ODD
186 012610 001401      BEQ  B CON.A2      ;
187 012612 005204      INC  B R4      ; IF SO, INCREMENT TO NEXT EVEN ADDRESS
188 012614 000207      CON.A2: RETURN      ; NOW GET NEXT CONTROL CHARACTER
189
190      ;
191      ; CONTROL CHARACTER WAS A 'D', SO PRINT A DECIMAL NUMBER.
192      ;
193
194 012616 012701 000012  CON.D: MOV  B #10.,R1      ; LOAD RADIX
195 012622 004737 015552  CALL  PNTNUM      ; PRINT NUMBER
196 012626 000207      RETURN      ; NOW GET NEXT CONTROL CHARACTER
197
198      ;
199      ; CONTROL CHARACTER WAS AN 'H', SO PRINT A HEX NUMBER.

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 99-4  
GLOBAL SUBROUTINES SECTION

```

200      ;
201      ;
202 012630 012701 000020      CON.H:  MOV    #16.,R1      ; LOAD RADIX
203 012634 004737 015552      CALL    PNTNUM      ; PRINT NUMBER
204 012640 000207              RETURN      ; NOW GET NEXT CONTROL CHARACTER
205      ;
206      ;
207      ; CONTROL CHARACTER WAS AN 'O', SO PRINT AN OCTAL NUMBER.
208      ;
209      ;
210 012642 012701 000010      CON.O:  MOV    #8.,R1      ; LOAD RADIX
211 012646 004737 015552      CALL    PNTNUM      ; PRINT NUMBER
212 012652 000207              RETURN      ; NOW GET NEXT CONTROL CHARACTER
213      ;
214      ;
215      ; CONTROL CHARACTER WAS AN 'N', SO PRINT A CARRIAGE RETURN-LINE FEED.
216      ;
217      ;
218 012654 004737 015474      CON.N:  CALL    GETCNT      ; GET COUNT
219 012660              CON.N1: PRINT    #CR      ;
220 012670 005301              DEC     R1      ; COUNT THE SEQUENCES
221 012672 001372              BNE    CON.N1      ;
222 012674 000207              RETURN      ; NOW GET NEXT CONTROL CHARACTER
223      ;
224      ;
225      ; CONTROL CHARACTER WAS AN 'R', SO CALL ONE OF THE PRE-PROGRAMMED
226      ; ROUTINE.
227      ;
228      ;
229 012676 004737 015474      CON.R:  CALL    GETCNT      ; GET ROUTINE NUMBER
230 012702 020127 000011      CMP    R1,#ERR.SZ      ; CHECK IF DEFINED ROUTINE NUMBER
231 012706 101004              BHI    CON.R1      ;
232 012710 060101              ADD    R1,R1      ; DOUBLE COUNT TO GET WORD INDEX
233 012712 004771 012754      CALL    BERR.TB-2(R1)      ; CALL ROUTINE
234 012716 000207              RETURN      ; NOW GET NEXT CONTROL CHARACTER
235      ;
236 012720              CON.R1: PNTF    ERRME1      ; REPORT BAD MESSAGE STRING
237 012730              PCP     R1      ; FIX THE STACK
238 012732 000207              RETURN
239      ;
240      ;
241      ; CONTROL CHARACTER WAS AN 'S', SO PRINT SOME NUMBER OF SPACES.
242      ;
243      ;
244 012734 004737 015474      CON.S:  CALL    GETCNT      ; GET COUNT
245 012740              CON.S1: PRINT    '<#>'      ;
246 012750 005301              DEC     R1      ; COUNT THE SPACES
247 012752 001372              BNE    CON.S1      ;
248 012754 000207              RETURN      ; NOW GET NEXT CONTROL CHARACTER
249      ;
250      ;
251      ; PRE-PROGRAMMED ERROR ROUTINE DISPATCH TABLE
252      ;
253      ;
254 012756 015036      ERR.TB: .WORD CALR1      ; CALL ALTERNATE PRINT STRING IN DM PGM
255 012760 015064      .WORD CALR2      ; PRINT AN SDI DIAGNOSE RESPONSE
256 012762 015162      .WORD CALR3      ; DECIDE WHETHER TO PRINT RBN

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 99-5  
GLOBAL SUBROUTINES SECTION

```

257 012764 015176      .WORD CALR4      ; PRINT BASIC LINE W/O CONTROLLER ADDR.
258 012766 015252      .WORD CALR5      ; PRINT BASIC LINE W/ CONTROLLER ADDR.
259 012770 015330      .WORD CALR6      ; CALL ALTERNATE PRINT STRING IN POP-11
260 012772 015344      .WORD CALR7      ; PRINT "REPLACE PROCESSOR MODULE"
261 012774 015362      .WORD CALR8      ; PRINT "SA REGISTER CONTAINS  XXXXXX"
262 012776 015400      .WORD CALR9      ; REPRINT LAST NUMBER
263
264      000011      ERR.SZ  = <.-ERR.TB>/2
265
266 013000      T_NAMES:
268 013000 003743      .WORD BASN1
269 013002 003764      .WORD BASN2
270 013004 004004      .WORD BASN3
275
276      ;
277      ; CONTROL CHARACTER TABLE
278      ;
279
280 013006      042      ERRC:  .BYTE  ' "
281 013007      101      .BYTE  ' A
282 013010      104      .BYTE  ' D
283 013011      110      .BYTE  ' H
284 013012      117      .BYTE  ' O
285 013013      116      .BYTE  ' N
286 013014      122      .BYTE  ' R
287 013015      123      .BYTE  ' S
288
289 013016      000      .BYTE  0      ; FOLLOW WITH A NULL BYTE
290      .EVEN
291
292      ;
293      ; ROUTINE ADDRESS TABLE
294      ;
295
296 013020 012546      ERRD:  .WORD  CON.QU
297 013022 012566      .WORD  CON.A
298 013024 012616      .WORD  CON.D
299 013026 012630      .WORD  CON.H
300 013030 012642      .WORD  CON.O
301 013032 012654      .WORD  CON.N
302 013034 012676      .WORD  CON.R
303 013036 012734      .WORD  CON.S

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 100  
 GLOBAL SUBROUTINES SECTION

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15 013040
16 013040 010137 002204
17 013044 004737 020744
21 013050 013737 002164 002160
22 013056 013737 002166 002162
23 013064 006301
24 013066 016137 013074 002200
25 013074 000207
26
27 013076
29 013076 000000G
30 013100 000000G
31 013102 000000G

: **
:
: TINIT - INITIALIZE VARIABLES FOR TEST
:
: INPUTS:
: R1 - TEST NUMBER
:
: OUTPUTS:
: DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
: LBUFS - CLEARED (DELETES ERROR LOG)
: TNUM - TEST NUMBER FROM R1
: ALL REGISTERS CLOBERED
: --
TINIT:
MOV R1,TNUM ; SAVE TEST NUMBER
CALL RESET ; RESET / L CONTROLLERS
MOV FMEM,FFREE ; INIT FREE
MOV FMEMS,FSIZE ; INIT FSIZE
ASL R1 ; R1 IS WORD INDEX
MOV READDT-2(R1),DMPROG ; STORE ADDRESS OF DM TEST INTO DMPROG
RETURN

READDT:
.WORD TEST1
.WORD TEST2
.WORD TEST3

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 013104 010137 002206
17 013110 005037 002210
18
19
20
21 013114 013705 002176
22 013120 013737 002206 002212
23 013126
24 013126 005065 000012
25 013132 116537 000002 002074
26 013140 005765 000002
27 013144 100405
28 013146
29 013146 004737 016524
30 013152 001402
31 013154 005237 002210
32 013160 062705 000034
33 013164 005337 002212
34 013170 001356
35
36
37
38 013172 005737 002210
39
40
41
42
43 013176 000207
44

```

```

; **
;
; RUNDM - LOAD AND RUN A DM PROGRAM IN THE CONTROLLERS. RETURN WHEN ALL
; DM PROGRAMS HAVE TERMINATED.
;
; INPUTS:
; R1 - NUMBER OF CONTROLLERS TO TEST
; R5 - POINTER TO FIRST CONTROLLER TABLE
; IMPLICIT INPUTS:
; DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
; OUTPUTS:
; Z SET IF NO CONTROLLERS SUCCESSFULLY STARTED
; R1-R5 - CONTENTS DESTROYED
; --
RUNDM: MOV R1,URUN ; SAVE NUMBER OF UNITS TO RUN
CLR URNING ; CLEAR NUMBER OF UNITS RUNNING
;
; LOAD DM PROGRAM INTO EACH CONTROLLER
;
MOV TSTTAB,R5 ; GET CONTROLLER TABLE ADDRESS
MOV URUN,UCNT ; SET COUNTER OF UNITS
LDDH: CLR C.FLG(R5) ; CLEAR ALL FLAGS
MOV C.UNIT(R5),L#LUN ; SEE IF UNIT TO BE TESTED
TST C.UNIT(R5) ;
BHI LDNEXT ; IF NOT, DON'T LOAD THIS UNIT
ASSUME CT.AVL EQ BIT15
CALL LOADDM ; LOAD THE DM PROGRAM
BEQ LDNEXT ; IF ERROR, GO TO NEXT CONTROLLER
INC URNING ; IF NO ERROR, COUNT UNIT RUNNING
LDNEXT: ADD #C.SIZE,R5 ; MOVE TO NEXT CONTROLLER TABLE
DEC UCNT ; CHECK IF MORE CONTROLLERS
BNE LDDM ; LOAD NEXT CONTROLLER
;
; CHECK IF ANY CONTROLLERS LOADED
;
TST URNING ; ANY UNITS LOADED?
;
; THE DM PROGRAMS ARE NOW IN CONTROL
; RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
;
RETURN ;

```

```

1
2
3      ;**
4      ;   RESPDM - RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
5      ;   HAVE TERMINATED.
6      ;
7      ;   INPUTS:
8      ;   R5 - CONTROLLER TABLE ADDRESS
9      ;--
10     013200 013705 002176      RESPDM: MOV     TSTTAB,R5      ; GET CONTROLLER TABLE ADDRESS
11     013204 013737 002206 002212      MOV     URUN,UCNT      ; SET COUNTER OF UNITS
12     013212 016504 000014      RESPCT: MOV     C.HCOM(R5),R4      ; GET HOST COMM AREA ADDRESS
13     013216 032765 000002 000012      BIT     @CT.RN,C.FLG(R5)      ; CHECK IF PROGRAM RUNNING
14     013224 001446      BEQ     RSPNXT      ; IF NOT, LOOK AT NEXT
15     013226 116537 000002 002074      MOVB    C.UNIT(R5),L#LUN      ; STORE UNIT NUMBER UNDER TEST
16     013234 032765 000010 000012      BIT     @CT.MSG,C.FLG(R5)      ; SEE IF INTERRUPT RECEIVED
17     013242 001071      BNE     RSPIN      ; IF SO, LOOK AT PACKET
18     013244 032765 000004 000012      BIT     @CT.CMD,C.FLG(R5)      ; SEE IF COMMAND HAS BEEN SENT
19     013252 001520      BEQ     RSPOU      ; IF NOT, SEND ONE
20
21     ;
22     ;   CHECK IF CONTROLLER STILL RUNNING
23     013254 011503      MOV     (R5),R3      ; GET ADDRESS OF IP REGISTER
24     013256 016301 000002      MOV     2(R3),R1      ; LOOK AT SA REGISTER
25     013262 001405      BEQ     RSPTH      ; IF ZERO, CONTROLLER STILL RUNNING
26     013264      ERDF 30,,ERR030      ; ELSE, REPORT FATAL CONTROLLER ERROR
27     013264 104455      TRAP   C#ERDF
28     013266 000036      .WORD 30
29     013270 000000      .WORD 0
30     013272 011430      .WORD ERR030
31     013274 000445      BR     RSPDRP      ; DROP CONTROLLER FROM TESTING
32
33     ;
34     ;   CHECK FOR TIMEOUT OF RESPONSE
35     013276      RSPTH:
36     013276 005737 002256      TST     KW.CSR      ; SEE IF A CLOCK ON SYSTEM
37     013302 001416      BEQ     RSPNTO      ; DON'T TIME IF NO CLOCK
38     013304 023765 002270 000030      CMP     KW.EL+2,C.TOH(R5)      ; CHECK HIGH WORD OF ELAPSED TIME
39     013312 101005      BHI     RSPTHMO      ; IF GREATER, RESPONSE TIMED OUT
40     013314 001011      BNE     RSPNTO      ; IF NOT SAME, ITS OK
41     013316 023765 002266 000026      CMP     KW.EL,C.TO(R5)      ; CHECK LOW WORD OF ELAPSED TIME
42     013324 103405      BLO     RSPNTO      ; IF LESS, PLENTY OF TIME LEFT
43     013326      RSPTHMO: ERDF 31,,ERR031      ; REPORT TIMEOUT ERROR
44     013326 104455      TRAP   C#ERDF
45     013330 000037      .WORD 31
46     013332 000000      .WORD 0
47     013334 011444      .WORD ERR031
48     013336 000424      BR     RSPDRP      ; DROP CONTROLLER FROM TESTING
49     013340      RSPNTO:
50     013340      BREAK      ; >>>>>>BREAK BACK TO MONITOR<<<<<<<<
51     013340 104422      TRAP   C#BRK
52
53     ;
54     ;   CHECK FOR TIME TO PRINT STATISTICAL REPORT
55     013342 005737 002256      RSPNXT: TST     KW.CSR      ; ANY CLOCK ON SYSTEM?
56     013346 001412      BEQ     RSPNRP      ; BYPASS IF NOT
57     013350 023737 002270 002274      CMP     KW.EL+2,STIME+2      ; CHECK HIGH WORD OF ELAPSED TIME

```

```

54 013356 101005          BHI      RSPRPT          ; IF GREATER PRINT REPORT
55 013360 001005          BNE      RSPNRP          ; IF NOT SAME, ITS NOT TIME YET
56 013362 023737 002266 002272  CMP      KW.EL,STIME      ; CHECK LOW WORD OF ELAPSED TIME
57 013370 103401          BLO      RSPNRP          ; IF LESS, ITS NOT TIME YET
58 013372          RSPRPT:          ;
59 013372          DORPT          ; PRINT A STATISTICAL REPORT
    013372 104424          TRAP      C4DRPT
60          ;
61          ; SWITCH TO NEXT CONTROLLER
62          ;
63 013374 062705 000034  RSPNRP: ADD      #C.SIZE,R5      ; MOVE TO NEXT TABLE
64 013400 005337 002212  DEC      UCNT            ; CHECK IF MORE CONTROLLERS
65 013404 001302          BNE      RESPCT          ; LOOK AT NEXT CONTROLLER
66 013406 000674          BR       RESPDM          ; LOOK AT FIRST CONTROLLER AGAIN
67          ;
68          ; REMOVE A CONTROLLER FROM TESTING
69          ;
70 013410 042765 000012 000012 RSPDRP: BIC      #CT.RN+CT.MSG,C.FLG(R5) ; CLEAR PROGRAM RUNNING
71 013416 005337 002210  DEC      URNING          ; REDUCE RUNNING CONTROLLERS COUNT
72 013422 001347          BNE      RSPNXT          ; IF ANY STILL RUNNING, LOOK AT THEM
73 013424 000207          RETURN          ; ELSE RETURN TO TEST SECTION
74          ;
75          ; CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
76          ; CHECK FOR PROPER OPCODE IN END PACKET
77          ;
78 013426 012700 000204  RSPIN:  MOV      #OP.END+OP.SSD,R0      ; GET SEND DATA END PACKET OPCODE
79 013432 032765 000020 000012  BIT      #CT.REQ,C.FLG(R5)          ; LOOK IF SEND DATA OR RECEIVE DATA
80 013440 001402          BEQ      RSPMWR          ;
81 013442 012700 000205  MOV      #OP.END+OP.RSD,R0          ; CHANGE TO RECEIVE DATA ENDCODE
82 013446 120064 000030  RSPMWR: CMPB     RO,HC.MPK+P.OPCD(R4) ; COMPARE TO OPCODE IN END PACKET
83 013452 001010          BNE      RSPERR          ;
84          ;
85          ; LOOK AT STATUS CODE
86          ;
87 013454 032764 000037 000032  BIT      #ST.MSK,HC.MPK+P.STS(R4) ; CHECK FOR STATUS CODE ST.SUC (ZERO)
88 013462 001004          BNE      RSPERR          ;
89          ;
90          ; CHECK FOR EXPECTED REFERENCE NUMBER
91          ;
92 013464 026564 000032 000020  CMP      C.REF(R5),HC.MPK+P.CRF(R4) ; CHECK IF CORRECT REF NUMBER
93 013472 001405          BEQ      RSPPTW          ;
94 013474          RSPERR: ERPDF 33,ERR033          ;
    013474 104455          TRAP      C4ERDF
    013476 000041          .WORD     33
    013500 000000          .WORD     0
    013502 011474          .WORD     ERR033
95 013504 000741          BR       RSPDRP          ; DROP UNIT FROM TESTING
96          ;
97          ; CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
98          ;
99 013506 032765 000020 000012 RSPPTW: BIT      #CT.REQ,C.FLG(R5)          ; CHECK IF RESPONSE FROM DM PROGRAM
100 013514 001445          RSPDU:  BEQ      RSPDU          ; LOOK AT REQUEST NUMBER IF SO
101          ;
102          ; MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
103          ;
104 013516 016401 000206  RSPPT2: MOV      HC.BF2(R4),R1          ; GET REQUEST NUMBER
105 013522 042701 007777  BIC      #007777,R1          ; CHECK TYPE
    
```



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 102-2  
GLOBAL SUBROUTINES SECTION

```

106 013526 022701 060000          CMP    #DU.SPC,R1          ; IS SPECIAL TYPE SET?
107 013532 001010          BNE    1$                ; IF NOT, ERROR
108 013534 042764 170000 000206    BIC    #+C007777,HC.BF2(R4) ; CLEAR TYPE
109 013542 016401 000206          MOV    HC.BF2(R4),R1      ; GET REQUEST NUMBER
110 013546 020127 000017          CMP    R1,#DSPSIZ        ; CHECK IF IN EXPECTED RANGE
111 013552 103405          BLO    RSPPT3           ;
112 013554          1$: ERRDF 32,ERR032      ; BAD REQUEST NUMBER
    013554 104455          TRAP  C$ERDF
    013556 000040          .WORD 32
    013560 000000          .WORD 0
    013562 011456          .WORD ERR032
113 013564 000711          BR     RSPDRP           ; DROP UNIT FROM TESTING
114
115 013566 012700 000004          RSPPT3: MOV    #OP.SSD,R0   ; BUILD A SEND DATA COMMAND PACKET
116 013572 004737 017176          CALL   BLDCHD           ; FOR ANSWER TO DM PROGRAM
117 013576 012700 000100          MOV    #HC.BF1,R0      ; POINT TO BUFFER IN PACKET
118 013602 004737 017336          CALL   CLRBUF           ; AND CLEAR BUFFER
119 013606 010403          MOV    R4,R3           ; R3 POINTS TO COMMAND BUFFER
120 013610 062704 000106          ADD    #HC.BSZ,R4      ; R4 POINTS TO MESSAGE BUFFER
121 013614 011401          MOV    (R4),R1         ; GET REQUEST NUMBER
122 013616 012423          MOVL  (R4)+,(R3)+      ; PUT REQUEST # INTO COMMAND PACKET
123 013620 060101          ADD    R1,R1           ; DOUBLE REQUEST NUMBER
124 013622 004771 013732          CALL   BRSPDSP(R1)     ; CALL REQUESTED ROUTINE
125 013626 001270          BNE    RSPDRP           ; RETURN Z CLEAR TO DROP UNIT
126
127          ;
128          ; SEND COMMAND BACK TO CONTROLLER
129          ;
130 013630 042765 000010 000012    RSPOUT: BIC    #CT.MSG,C.FLG(R5) ; CLEAR MESSAGE RECEIVED FLAG
131 013636 032765 000020 000012    BIT    #CT.REQ,C.FLG(R5) ; CHECK WHICH COMMAND TO SEND
132 013644 001014          BNE    RSP0U2          ; BRANCH IF RESPONSE TO REQUEST
133 013646 012700 000005          MOV    #OP.RSD,R0      ; BUILD RECEIVE DATA COMMAND
134 013652 004737 017176          CALL   BLDCHD           ;
135 013656 012700 000206          MOV    #HC.BF2,R0      ; POINT TO MESSAGE BUFFER
136 013662 004737 017336          CALL   CLRBUF           ; AND CLEAR IT
137 013666 052765 000020 000012    BIS    #CT.REQ,C.FLG(R5) ; SET REQUEST BIT
138 013674 000403          BR     RSP0U3          ;
139
140 013676 042765 000020 000012    RSP0U2: BIC    #CT.REQ,C.FLG(R5) ; CLEAR REQUEST BIT
141 013704          RSP0U3:
142 013704 004737 017270          CALL   SNDCMD           ; SEND COMMAND TO CONTROLLER
143 013710 012700 000264          MOV    #3,*60.,R0      ; SET TIMEOUT FOR 3 MINUTES
144 013714 010501          MOV    R5,R1           ;
145 013716 062701 000026          ADD    #C.TO,R1        ; PUT TIME IN CONTROLLER TABLE
146 013722 004737 017614          CALL   SETTO           ;
147 013726 000137 013342          JMP    RSPNXT          ; NOW WAIT FOR END PACKET
148
149          ;
150          ; RESPONSE REQUEST DISPATCH TABLE
151          ;
151 013732 013770          RSPDSP: .WORD DMRQ0     ; 0 SET UP MEMORY FOR ADDRESS TESTING
152 013734 000000          .WORD 0                ; 1 *** REQUEST NO LONGER VALID ***
154 013736 014110          .WORD DMRQ2           ; 2 GET MANUAL INTERVENTION COMMAND
155 013740 000000          .WORD 0                ; 3 *** NOT VALID FOR CZUDHO ***
156 013742 000000          .WORD 0                ; 4 *** NOT VALID FOR CZUDHO ***
157 013744 000000          .WORD 0                ; 5 *** NOT VALID FOR CZUDHO ***
158 013746 000000          .WORD 0                ; 6 *** NOT VALID FOR CZUDHO ***
159 013750 014560          .WORD DMRQ7           ; 7 ADD TO SOFT ERROR AND ECC COUNTS

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 102-3  
GLOBAL SUBROUTINES SECTION

160 013752 000000  
161 013754 000000  
162 013756 014564  
163 013760 014626  
164 013762 014626  
165 013764 014720  
166 013766 015032  
182  
183 000017  
184

.WORD 0  
.WORD 0  
.WORD DMRQA  
.WORD DMRQB  
.WORD DMRQB  
.WORD DMRQD  
.WORD DMRQE  
DSPSIZ = <.-RSPDSP>/2

; 8 \*\*\* NOT VALID FOR CZUDHO \*\*\*  
; 9 \*\*\* NOT VALID FOR CZUDHO \*\*\*  
;10 TELL WHICH DRIVES TO TEST  
;11 REPORT ERROR MESSAGE  
;[TEMP] 12 REPORT ERROR (FIX DM BUG)  
;13 PRINT A DESCRIPTIVE MESSAGE  
;14 MARK DM PROGRAM AS NOT RUNNING  
;LEGAL NUMBERS ARE LOWER THAN THIS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39

\*\*\*  
:NORMAL MAINTENANCE READ BUFFER DESCRIPTION

:BYTE OFFSET FROM  
:START OF BUFFER

0	} REQUEST NUMBER }
2	} DATA ARGUMENT #1 }
4	} DATA ARGUMENT #2 }
6	} DATA ARGUMENT #3 }
.	.
.	.
68	} DATA ARGUMENT #34 }

USED TO SELECT ROUTINE  
R4 CONTAINS THIS ADDRESS

:NORMAL PSEUDO-TERMINAL IN PACKET DESCRIPTION GIVEN IN RESPONSE TO ABOVE PACKET

:BYTE OFFSET FROM  
:START OF PACKET

0	} REQUEST NUMBER }
2	} DATA ARGUMENT #1 }
4	} DATA ARGUMENT #2 }
6	} DATA ARGUMENT #3 }
.	.
.	.
68	} DATA ARGUMENT #34 }

ECHOED FROM REQUEST PACKET  
R3 CONTAINS THIS ADDRESS  
ALL DATA ARGUMENTS ARE RETURNED  
CONTAINING ZEROS UNLESS  
SPECIFICALLY INDICATED BY  
RESPONSE ROUTINE.

```

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 013770
30 013770 013701 002160
31 013774 013702 002162
32
33
34
35 014000 010111
36 014002 062701 000002
37 014006 005302
38 014010 001373
39
40
41
42 014012 013723 002160
43 014016 005023
44 014020 013700 002162
45 014024 006300
46 014026 063700 002160
47 014032 162700 000002
48 014036 010023
49 014040 005023
50
51
52
53 014042 005023
54 014044 005023
55 014046 013700 002120
56 014052 005001
57 014054 006300

```

```

: **
:
: DMRQO - DM REQUEST 0
:
: SET UP MEMORY FOR ADDRESS TESTING FROM CONTROLLER.
: PLACE ADDRESS OF EACH LOCATION INTO EACH LOCATION IN FREE
: MEMORY. RETURN FIRST LOCATION OF FREE MEMORY IN CMD.02 (LOW BITS)
: AND CMD.03 (HIGH BITS). RETURN LAST LOCATION OF FREE MEMORY IN
: CMD.04 AND CMD.05. ALSO RETURN FIRST EXISTANT LOCATION IN CMD.06
: AND CMD.07; LAST EXISTANT LOCATION IN CMD.08 AND CMD.09.
:
: INPUTS:
: R5 - CONTROLLER TABLE ADDRESS
: R4 - MESSAGE PACKET DATA ADDRESS (POINTING TO MSG.02)
: R3 - COMMAND PACKET DATA ADDRESS (POINTING TO CMD.02)
:
: OUTPUTS:
: COMMAND PACKET CONTAINING:
: 0.(R3) LOW ADDRESS BITS OF FIRST WRITABLE ADDRESS
: 2.(R3) HIGH ADDRESS BITS OF FIRST WRITABLE ADDRESS
: 4.(R3) LOW ADDRESS BITS OF LAST WRITABLE ADDRESS
: 6.(R3) HIGH ADDRESS BITS OF LAST WRITABLE ADDRESS
: 8.(R3) LOW ADDRESS BITS OF FIRST READABLE ADDRESS
: 10.(R3) HIGH ADDRESS BITS OF FIRST READABLE ADDRESS
: 12.(R3) LOW ADDRESS BITS OF LAST READABLE ADDRESS
: 14.(R3) HIGH ADDRESS BITS OF LAST READABLE ADDRESS
: Z SET
:
: --
:
: DMRQO:
: MOV FFREE,R1 ;GET FIRST ADDRESS OF FREE MEMORY
: MOV FSIZE,R2 ;GET SIZE
:
: FILL MEMORY WITH ADDRESS PATTERN
:
: MEMFIL: MOV R1,(R1) ;WRITE DATA INTO LOCATION
: ADD #2,R1 ;INCREASE ADDRESS TO NEXT LOCATION
: DEC R2 ;COUNT THE WORDS
: BNE MEMFIL ;FILL ALL WORDS
:
: SEND LOCATION OF FREE MEMORY TO CONTROLLER
:
: MOV FFREE,(R3)+ ;LOAD FIRST ADDRESS OF FREE MEMORY
: CLR (R3)+ ;HIGH ORDER BITS ARE ZERO
: MOV FSIZE,R0 ;GET SIZE OF FREE MEMORY
: ASL R0 ;CONVERT TO BYTES
: ADD FFREE,R0 ;COMPUTE LAST LOCATION
: SUB #2,R0
: MOV R0,(R3)+ ;LOAD LAST LOCATION
: CLR (R3)+ ;CLEAR HIGH ORDER BITS
:
: SEND LOCATION OF READABLE MEMORY
:
: CLR (R3)+ ;SEND ZERO AS START OF READABLE MEMORY
: CLR (R3)+
: MOV L#HIMEM,R0 ;GET HIGH MEMORY ADDRESS
: CLR R1 ;CLEAR HIGH BITS
: ASL R0 ;SHIFT LEFT 6 PLACES

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 104-1  
GLOBAL SUBROUTINES SECTION

58	014056	006300		ASL	R0	
59	014060	006300		ASL	R0	
60	014062	006300		ASL	R0	
61	014064	006300		ASL	R0	
62	014066	006101		ROL	R1	
63	014070	006300		ASL	R0	
64	014072	006101		ROL	R1	
65	014074	052700	000076	BIS	#76,R0	;SET LOW ORDER BITS
66	014100	010023		MOV	R0,(R3).	;PUT INTO BUFFER
67	014102	010123		MOV	R1,(R3).	
68	014104	000264		SEZ		
69	014106	000207		RETURN		
70						

```

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 014110
33 014110 032737 000200 002146
34 014116 001002
35 014120 000137 014542
36 014124
   014124 104450
37 014126
   014126 103406
38 014130
39 014140 000137 014542
40 014144 012401
41 014146 012402
42 014150 001022
43 014152 004737 015404
44 014156 001401
45 014160 000207
46
47 014162
48 014202 005037 002300
49 014206 005037 002302
50 014212 005037 002304
51
52 014216 022702 000002
53 014222 001027
54 014224
55 014234 013701 002300
;
```

```

***
:
:   DMRQ2 - DM REQUEST 2
:
:   GET MANUAL INTERVENTION COMMAND
:
:   INPUTS:
:       R5 - CONTROLLER TABLE ADDRESS
:       R4 - MESSAGE DATA ADDRESS
:           (R4) DRIVE NUMBER
:       2.(R4) OPERATION CODE
:           0 ON FIRST REQUEST FOR DRIVE. ECHO OF PREVIOUS RESPONSE
:           ALL OTHER TIMES.
:       IF OPERATION CODE = 2
:       4.(R4) DATA BYTE READ (TO BE PRINTED)
:       R3 - COMMAND DATA ADDRESS
:
:   OUTPUTS:
:       COMMAND DATA FILLED WITH THE FOLLOWING:
:       (R3) OPERATION CODE
:           0 - EXIT
:           1 - WRITE
:           2 - READ
:           3 - DIAGNOSE
:       IF OPERATION CODE = 1, 2 OR 3
:       2.(R3) REGION NUMBER
:       4.(R3) OFFSET INTO REGION
:       IF OPERATION CODE = 1
:       6.(R3) DATA BYTE
:       Z SET IF DATA RETURNED
:       Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
:
:---
DMRQ2:
:   BIT   #SM.MAN,SFPTBL*SO.BIT ;LOOK AT MANUAL INTERVENTION MODE
:   BNE   DMRQ2H ;EXIT IF NOT WANTED
:   JMP   DMRQ2X
DMRQ2H:
:   MANUAL ;MANUAL INTERVENTION ALLOWED?
:   TRAP  C1MANI
:   BCOMPLETE DMRQ20 ;PRINT WARNING IF NOT
:   BCS   DMRQ20
:   PNTF  T2WARN
:   JMP   DMRQ2X
DMRQ20:
:   MOV   (R4),R1 ;GET DRIVE NUMBER
:   MOV   (R4),R2 ;GET OPERATION CODE
:   BNE   DMRQ22 ;BRANCH IF NOT ZERO
:   CALL  GDRVY ;GET DRIVE TABLE ADDRESS
:   BEQ   10 ;CHECK IF DRIVE FOUND
:   RETURN ;RETURN WITH Z CLEAR IF NOT

10:
:   PNTF  T2CMS1,D.UNIT(R4),(R5),(R4) ;PRINT DESCRIPTION
:   CLR   T2WRR ;CLEAR ALL STORAGE WORDS
:   CLR   T2WRO
:   CLR   T2DR
DMRQ22:
:   CMP   #2,R2 ;SEE IF LAST OPERATION WAS READ
:   BNE   DMRQ20 ;BRANCH IF NOT TO QUESTION
:   PRINT '<0' ;PRINT ONE SPACE
:   MOV   T2WRR,R1 ;PRINT REGION
;
```

LZUDHAO JDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 105-1  
GLOBAL SUBROUTINES SECTION

```

56 014240 004737 016120      CALL    T2PNTW      ;
57 014244 013701 002302      MOV     T2WRO,R1   ; PRINT OFFSET
58 014250 004737 016120      CALL    T2PNTW      ;
59 014254                    PRINT   #' /        ; PRINT A SLASH
60 014264 012401                    MOV     (R4),R1    ; PRINT THE DATA
61 014266 004737 016150      CALL    T2PNTB      ;
62 014272                    PRINT   #CR          ; END THE LINE
63                                ;
64                                ; NOW ASK FOR COMMAND INPUT
65                                ;
66 014302 DMRQ2Q: GMANID T4OPT7,TEMP,A,-1,1,20.,NO; ASK FOR COMMAND
    014302 104443      TRAP    C$GMAN
    014304 000406      BR      10000$
    014306 002216      .WORD   TEMP
    014310 000142      .WORD   T$CODE
    014312 002432      .WORD   T4OPT7
    014314 177777      .WORD   -1
    014316 000001      .WORD   T$LOLIM
    014320 000024      .WORD   T$HILIM
    014322                    10000$:
67 014322 012701 002216      MOV     #TEMP,R1   ; GET POINTER TO STRING
68 014326 112100      MOVB   (R1),R0    ; GET COMMAND CHARACTER
69 014330 022700 000105      CMP     #'E,R0
70 014334 001415      BEQ    DMRQ2V
71 014336 022700 000104      CMP     #'D,R0
72 014342 001016      BNE    DMRQ23
73 014344 012713 000003      MOV     #3,(R3)   ; STORE DIAGNOSE OPERATION CODE
74 014350 004737 016232      CALL    T2GNUM     ; GET REGION FROM COMMAND
75 014354 001402      BEQ    1$
76 014356 010437 002304      MOV     R4,T2DR
77 014362 013763 002304 000002 1$: MOV     T2DR,2(R3)
78 014370 004737 016232      DMRQ2V: CALL    T2GNUM     ; MAKE SURE AT END OF LINE
79 014374 001064      BNE    DMRQ2E
80 014376 000461      BR     DMRQ2X
81                                ;
82                                ; COMMAND MUST BE EITHER READ OR WRITE
83                                ;
84 014400 012713 000002      DMRQ23: MOV     #2,(R3)   ; CHECK IF READ
85 014404 022700 000122      CMP     #'R,R0
86 014410 001415      BEQ    DMRQ2R
87 014412 022700 000127      CMP     #'W,R0
88 014416 001053      BNE    DMRQ2E     ; CHECK IF WRITE
89 014420 012713 000001      MOV     #1,(R3)   ; IF NOT - ERROR
90 014424 004737 016232      CALL    T2GNUM     ; GET DATA BYTE
91 014430 001446      BEQ    DMRQ2E     ; ERROR IF NO DATA
92 014432 162700 000002      SUB     #2,R0
93 014436 003043      BGT    DMRQ2E     ; OR GREATER THAN TWO DIGITS
94 014440 010463 000006      MOV     R4,6(R3)  ; STORE DATA BYTES IN BUFFER
95 014444 013763 002300 000002 DMRQ2R: MOV     T2WRO,2(R3)  ; PUT REGION AND OFFSET
96 014452 013763 002302 000004      MOV     T2WRO,4(R3) ; INTO BUFFER
97 014460 021302      CMP     (R3),R2
98 014462 001002      BNE    DMRQ2N
99 014464 005263 000004      INC     4(R3)
100 014470 004737 016232      DMRQ2N: CALL    T2GNUM     ; CONVERT ASCII TO NUMERIC
101 014474 001411      BEQ    DMRQ2W
102 014476 010463 000002      MOV     R4,2(R3)
103 014502 005063 000004      CLR     4(R3)

```





CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 106  
GLOBAL SUBROUTINES SECTION

```
1
2
3      ;**      DMRQ7 - DM REQUEST 7
4      ;
5      ;      THIS REQUEST IS USED ONLY TO KEEP THE HOST FROM TIMING OUT
6      ;      THE CONTROLLER WHILE DM TESTS 2 AND 3 ARE RUNNING.
7      ;
8      ;--
9
10 014560 000264      DMRQ7: SEZ      ; INDICATE SUCCESS
11 014562 000207      RETURN      ; EXIT
```

CZUDHA0 JDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 107  
 GLOBAL SUBROUTINES SECTION

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 014564 010504
19 014566 062704 000016
20 014572 012702 000004
21 014576 012400
22 014600 001406
23 014602 005760 000002
24 014606 100401
25 014610
26 014610 011023
34 014612 005302
35 014614 001370
36 014616 012723 100000
37 014622 000264
38 014624 000207

;DMRQA - DM REQUEST 10
;
; TELL DM PROGRAM WHICH DRIVES ARE SELECTED FOR TESTING
; AND CLEAR STATISTICS IN DRIVE TABLE
;
; INPUTS:
; R5 - CONTROLLER TABLE ADDRESS
; R4 - MESSAGE DATA ADDRESS
; (NO DATA)
; R3 - COMMAND DATA ADDRESS
; OUTPUTS:
; COMMAND PACKET CONTAINING UP TO 4 DRIVE NUMBERS.
; LIST IS ENDED BY A WORD WITH BIT 15 SET,
; D.XFRW, D.XFRR, D.HERR, D.SERR, D.SEEK AND D.ECC CLEARED
; Z SET

DMRQA: MOV R5,R4 ;GET ADDRESS OF CONTROLLER TABLE
ADD #C.DR0,R4 ;BUMP TO DRIVE TABLE POINTERS
MOV #4.,R2 ;GET COUNT OF FORTS
UTOT1: MOV (R4)+,R0 ;SEE IF DRIVE TABLE POINTER EXISTS
BEQ UTOT2 ;BRANCH IF NOT
TST D.UNIT(R0) ;LOOK IF UNIT AVAILABLE FOR TESTING
BMI UTOT1A
ASSUME DT.AVL EQ BIT15
UTOT1A: MOV (R0),(R3)+ ;LOAD DRIVE NUMBER FROM TABLE
DEC R2 ;COUNT THE DRIVE TABLES
BNE UTOT1 ;REPEAT FOR EACH TABLE
UTOT2: MOV #BIT15,(R3)+ ;TERMINATE LIST
SEZ
RETURN ;RETURN WITH Z SET

```

```

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 014626
27
28
29
30
31
32
33
34
35
36
37
38 014626 005764 000002
39 014632 100406
40 014634
   014634 104421
41 014636 032700 000040
42 014642 001014
43 014644 052713 100000
44 014650 016400 000002
45 014654 005100
46 014656 032700 140000
47 014662 001004
48 014664 032737 000400 002146
49 014672 001010
50 014674
51 014674 042765 000010 000012
56 014702 004737 016360
57 014706 103002
58 014710
59 014710 000244
60 014712 000207

;DMRQB - DM REQUEST 11
;
;PRINT AN ERROR MESSAGE
;
;INPUTS:
;   R5 - CONTROLLER TABLE ADDRESS
;   R4 - MESSAGE DATA ADDRESS
;       (R4) ERROR PC IN DM PROGRAM
;       2.(R4) <15:14> ERROR TYPE
;           <13:0 > ERROR NUMBER
;       4.(R4) DRIVE NUMBER (-1 IF NOT GIVEN)
;       6.(R4) MESSAGE POINTER
;       8.(R4) OPTIONAL PARAMETERS FOR ERROR PRINT ROUTINE
;       10.(R4) "
;       :
;       :
;       58.(R4) "
;   R3 - COMMAND DATA ADDRESS
;OUTPUTS:
;   COMMAND PACKET CONTAINING THE FOLLOWING:
;   (R3) - BIT 15 SET IF FATAL ERROR TO INDICATE DRIVE SHOULD
;           NO LONGER BE TESTED
;   Z SET TO INDICATE DATA RETURNED
;   Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
;DMRQB:
;:   PUSH      R4           ; SAVE R4
;:   MOV       4(R4),R1     ; R1 = DRIVE #
;:   BMI      1$           ; IF -1, THEN NO DRIVE # GIVEN
;:   CALL     GTDRVT       ; GET DRIVE TABLE ADDRESS
;:   TST      D.UNIT(R4)   ; IF DRIVE HAS BEEN DROPPED, EXIT
;:   BGE     1$           ; NEGATIVE VALUE MEANS DRIVE DROPPED
;:   POP      R4           ;
;:   BIS     @BIT15,(R3)   ; SET DROP DRIVE BIT
;:   BR      8$           ; EXIT
;:1$:  POP      R4         ;RESTORE R4
;:
;:   TST      2(R4)        ;CHECK IF FATAL ERROR
;:   BMI     5$           ;BRANCH IF NOT
;:   RFLAGS  R0           ;LOOK AT FLAGS
;:   TRAP    C@RFLA
;:   BIT     @IDU,R0      ;SEE IF ALLOWED TO DROP UNITS
;:   BNE     6$           ;BRANCH IF NOT
;:   BIS     @BIT15,(R3)  ;SET DROP DRIVE BIT
;:   MOV     2(R4),R0     ;SEE IF SOFT ERROR
;:   COM     R0
;:   BIT     @140000,R0
;:   BNE     6$           ;BRANCH IF NOT
;:   BIT     @SM.SSF,SO.BIT+SFPTBL ;SEE IF SOFT ERRORS SUPPRESSED
;:   BNE     ERRMSX      ;DON'T PRINT IF SO
;:6$:
;:   BIC     @CT.MSG,C.FLG(R5) ;CLEAR MESSAGE RECEIVED FLAG
;:   CALL    PNTERR      ; PRINT ERROR MESSAGE
;:   BCC     ERRMSX      ; IF DRIVE HASN'T BEEN DROPPED, PRINT
;:8$:
;:   CLZ
;:   RETURN
;           ELSE RETURN

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 108-1  
GLOBAL SUBROUTINES SECTION

61  
83 014714 000264  
84 014716 000207  
85

ERRMSX: SEZ  
RETURN

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 109  
GLOBAL SUBROUTINES SECTION

```

1
2           ;DMRQD - DM REQUEST 13.
3           ;
4           ;PRINT A MESSAGE WITH HEADER AS FOLLOWS:
5           ; "UNIT XX CONTROLLER AT XXXXXX DRIVE XXX RUNTIME HH:MM:SS "
6           ;ENTIRE MESSAGE IS PRINTED WITH PRINTX CALLS.
7           ;
8           ;INPUTS:
9           ;
10          ; R5 - CONTROLLER TABLE ADDRESS
11          ; R4 - MESSAGE DATA ADDRESS
12          ; (R4) DRIVE NUMBER
13          ; 2.(R4) MESSAGE POINTER
14          ; 2.(R4) MESSAGE POINTER
15          ; 4.(R4) OPTIONAL MESSAGE PARAMETERS
16          ;
17          ;
18          ; 58.(R4) COMMAND DATA ADDRESS
19 014720 042765 000010 000012 DMRQD: BIC      #CT.MSG,C.FLG(R5)      ;CLEAR MESSAGE RECEIVED FLAG
20 014726 012401                MOV      (R4)+,R1          ;GET DRIVE NUMBER
21 014730                PUSH     R4              ;SAVE DATA POINTER
22 014732 004737 015404        CALL     GTDRV          ;GET DRIVE TABLE ADDRESS
23 014736 001033                BNE      1$              ;CHECK IF DRIVE FOUND
24 014740 005764 000002        TST      D.UNIT(R4)      ;IF UNIT DROPPED FROM TESTING
25 014744 100430                BMI      1$              ; DON'T PRINT ANYTHING
26 014746                PNTX    MESSG,D.UNIT(R4),(R5),(R4); PRINT HEADER
27 014766 004737 021046        CALL     RNTIME         ; GET RUNTIME PARAMETERS
28 014772                POP      R4              ; RESTORE MESSAGE MOINTER
29 014774 012402                MOV      (R4)+,R2          ;GET MESSAGE POINTER
30 014776 006302                ASL     R2              ;DOUBLE TO MAKE BYTE OFFSET
31 015000 063702 002200        ADD     DMPROG,R2      ;ADD TO START OF MESSAGE STRINGS
32 015004 067702 165170        ADD     SDMPROG,R2    ;ADD SIZE OF MAIN PROGRAM
33 015010 105712                TSTB   (R2)           ;CHECK FIRST BYTE
34 015012 001001                BNE     2$              ;IF ZERO
35 015014 005202                INC     R2              ; INCREMENT TO NEXT BYTE
36 015016 004737 012476        2$:    CALL     OSTRNG    ;OUTPUT ACCORDING TO STRING
37 015022 000264                SEZ
38 015024 000207                RETURN
39 015026                1$:    POP     R4
40 015030 000207                RETURN

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 110  
GLOBAL SUBROUTINES SECTION

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14 015032 000244
15 015034 000207

;DMRQE - DM REQUEST 14
;
;MARK DM PROGRAM AS NO LONGER RUNNING
;
;INPUTS:
; R5 - CONTROLLER TABLE ADDRESS
; R4 - MESSAGE DATA ADDRESS
; (NO DATA)
; R3 - COMMAND DATA ADDRESS
;OUTPUTS:
; Z CLEAR TO DROP UNIT FROM TESTING

DMRQE: CLZ ;DROP UNIT FROM TESTING
RETURN
```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 111  
 PRE-PROGRAMMED SUBROUTINES

```

1      .SBTTL  PRE-PROGRAMMED SUBROUTINES
2
3      ;**
4      ;      CALR1 - PRE-PROGRAMMED PRINT ROUTINE 1
5      ;
6      ;      CALL ALTERNATE PRINT STRING IN DM PROGRAM IMAGE
7      ;--
8
9      015036
10     015040 012402
11     015042 006302
12     015044 063702 002200
13     015050 067702 165124
14     015054 004737 012476
15     015060
16     015062 000207
17
18
19     ;**
20     ;      CALR2 - PRE-PROGRAMMED PRINT ROUTINE 2
21     ;
22     ;      PRINT AN SDI DIAGNOSE RESPONSE
23     ;--
24
25     015064
26     015066 012402
27     015070
28     015072 042702 177400
29     015076 001414
30     015100 012700 000020
31     015104 012701 000040
32     015110 004737 015560
33     015114
34     015124 005302
35     015126 001364
36     015130
37     015132 000301
38     015134 042701 177400
39     015140 001406
40     015142 004737 012572
41     015146
42     015156
43     015160 000207
44
45     ;**
46     ;      CALR3 - PRE-PROGRAMMED PRINT ROUTINE 3
47     ;
48     ;      DECIDE WHETHER TO PRINT RBN
49     ;
50     ;      FOUR PARAMETERS ARE PROVIDED FOR THIS ROUTINE. THE FIRST PARAMETER
51     ;      SHOULD BE CHECKED TO SEE IF BIT 7 IS SET:
52     ;      IF SET - TURN INTO A CALL TO ROUTINE 1 (WHICH WILL USE OTHER 3
53     ;      PARAMETERS).
54     ;      IF CLEAR - SKIP OVER NEXT 3 PARAMETERS AND END ROUTINE
55     ;--
56     015162 032724 000200
57     015166 001323

```

.SBTTL PRE-PROGRAMMED SUBROUTINES

;  
 ;\*\*  
 ; CALR1 - PRE-PROGRAMMED PRINT ROUTINE 1  
 ;  
 ; CALL ALTERNATE PRINT STRING IN DM PROGRAM IMAGE  
 ;  
 ;--

CALR1: PUSH R2  
 MOV (R4)+,R2 ;GET NEW STRING POINTER  
 ASL R2 ;DOUBLE FOR WORD COUNT  
 ADD DMPROG,R2 ;ADD START OF STRING STORAGE  
 ADD @DMPROG,R2 ;ADD SIZE OF MAIN PROGRAM  
 CALL OSTRNG ;OUTPUT USING THIS STRING  
 POP R2 ;GET OLD POINTER BACK  
 RETURN ;NOW CONTINUE THE OLD STRING

;  
 ;\*\*  
 ; CALR2 - PRE-PROGRAMMED PRINT ROUTINE 2  
 ;  
 ; PRINT AN SDI DIAGNOSE RESPONSE  
 ;  
 ;--

CALR2: PUSH R2  
 MOV (R4)+,R2 ;GET COUNTS  
 PUSH R2 ;SAVE COUNTS  
 BIC #177400,R2 ;GET BINARY COUNT  
 BEQ 2\$ ;BYPASS BINARY IF COUNT IS ZERO  
 1\$: MOV #16.,R0 ;RADIX IS HEX  
 MOV #32.,R1 ;32 BIT NUMBERS  
 CALL PNTNUS ;PRINT THE NUMBER  
 PRINT #CR ;GO TO NEW LINE  
 DEC R2  
 BNE 1\$  
 2\$: POP R1 ;GET ASCII COUNT  
 SWAB R1 ;GET ASCII COUNT  
 BIC #177400,R1 ;BYPASS IS COUNT IS ZERO  
 BEQ 3\$ ;PRINT THE ASCII  
 CALL CON,A1 ;GO TO NEW LINE  
 PRINT #CR  
 3\$: POP R2  
 RETURN

;  
 ;\*\*  
 ; CALR3 - PRE-PROGRAMMED PRINT ROUTINE 3  
 ;  
 ; DECIDE WHETHER TO PRINT RBN  
 ;  
 ; FOUR PARAMETERS ARE PROVIDED FOR THIS ROUTINE. THE FIRST PARAMETER  
 ; SHOULD BE CHECKED TO SEE IF BIT 7 IS SET:  
 ; IF SET - TURN INTO A CALL TO ROUTINE 1 (WHICH WILL USE OTHER 3  
 ; PARAMETERS).  
 ; IF CLEAR - SKIP OVER NEXT 3 PARAMETERS AND END ROUTINE  
 ;  
 ;--

CALR3: BIT #BIT7,(R4)+ ;CHECK BIT 7 IN FIRST PARAMETER WORD  
 BNE CALR1 ;IF SET, TURN INTO A CALR1

```

58 015170 062704 000006      ADD    #6,R4          ;ELSE, SKIP OVER NEXT 3 PARAMETERS
59 015174 000207      RETURN
60
61      ;**
62      ;      CALR4 - PRE-PROGRAMMED PRINT ROUTINE 4
63      ;
64      ;      PRINT BASIC LINE FOR HOST PROGRAM ERROR WITHOUT CONTROLLER ADDRESS
65      ;      THEN SWITCH TO EXTENDED FORMAT
66      ;--
67
68 015176      CALR4:  PNTB    BASLN,#BASNO,#BAS,#BAS,#BAS
69 015226 004737 021046      CALL    RNTIME
70 015232      PRINT   #CR
71 015242 012737 012322 002306      MOV     #PX,PType
72 015250 000207      RETURN
73
74      ;**
75      ;      CALR5 - PRE-PROGRAMMED PRINT ROUTINE 5
76      ;
77      ;      PRINT BASIC LINE FOR HOST PROGRAM ERROR WITH CONTROLLER ADDRESS
78      ;      THEN SWITCH TO EXTENDED FORMAT
79      ;--
80
81 015252      CALR5:  PNTB    BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
82 015304 004737 021046      CALL    RNTIME
83 015310      PRINT   #CR
84 015320 012737 012322 002306      MOV     #PX,PType
85 015326 000207      RETURN
86
87      ;**
88      ;      CALR6 - PRE-PROGRAMMED PRINT ROUTINE 6
89      ;
90      ;      CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
91      ;--
92
93 015330      CALR6:  PUSH    R2
94 015332 012402      MOV     (R4)+,R2          ;GET NEW STRING POINTER
95 015334 004737 012476      CALL    OSTRNG          ;OUTPUT USING THIS STRING
96 015340      POP     R2
97 015342 000207      RETURN          ;NOW CONTINUE THE OLD STRING
98
99      ;**
100     ;      CALR7 - PRE-PROGRAMMED PRINT ROUTINE 7
101     ;
102     ;      PRINT "REPLACE CONTROLLER PROCESSOR MODULE"
103     ;--
104
105 015344      CALR7:  PUSH    R2
106 015346 012702 010672      MOV     #XFRU,R2
107 015352 004737 012476      CALL    OSTRNG
108 015356      POP     R2
109 015360 000207      RETURN
110
111     ;**
112     ;      CALR8 - PRE-PROGRAMMED PRINT ROUTINE 8
113     ;
114     ;      PRINT " SA REGISTER CONTAINS XXXXXX"

```



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 111-2  
PRE-PROGRAMMED SUBROUTINES

```
115      ;--
116
117 015362      CALR8:  PUSH   R2
118 015364 012702 010635      MOV   #XSA,R2
119 015370 004737 012476      CALL  OSTRNG
120 015374      POP    R2
121 015376 000207      RETURN
122
123      ;++
124      ;      CALR9 - PRE-PROGRAMMED PRINT ROUTINE 9
125      ;
126      ;      REPRINT LAST NUMBER
127      ;      R4 -> TABLE
128      ;--
129
130 015400 005744      CALR9:  TST   -(R4)
131 015402 000207      RETURN
132
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 015404
15 015406 010504
16 015410 062704 000016
17 015414 012702 000004
18 015420 005714
19 015422 001406
20 015424 027401 000000
21 015430 001412
22 015432 005724
23 015434 005302
24 015436 001370
25 015440
    015440 104455
    015442 000043
    015444 000000
    015446 011502
26 015450
27 015452 000244
28 015454 000207
29
30 015456 011404
31 015460 116437 000002 002074
32 015466
33 015470 000264
34 015472 000207

;GDRVT
;
;GET DRIVE TABLE POINTER
;
;INPUTS:
;   R5 - CONTROLLER TABLE ADDRESS
;   R1 - DRIVE NUMBER
;OUTPUTS:
;   R4 - DRIVE TABLE ADDRESS
;   L#LUN - LOADED WITH UNIT NUMBER OF DRIVE
;   Z CLEAR IF DRIVE TABLE NOT FOUND AFTER ERROR PRINTED

GDRVT: PUSH    R2
        MOV     R5,R4
        ADD     @C.DRO,R4
        MOV     @4.,R2
1$:     TST     (R4)
        BEQ     3$
        CMP     @R4),R1
        BEQ     4$
2$:     TST     (R4)+
        DEC     R2
        BNE     1$
3$:     ERDF   35.,.ERR035
        TRAP   C#ERDF
        .WORD  35
        .WORD  0
        .WORD  ERR035
        POP    R2
        CLZ
        RETURN
;GET CONTROLLER TABLE ADDRESS
;ADD OFFSET TO DRIVE TABLE ADDRESS
;GET COUNT OF DRIVES
;CHECK IF AN ADDRESS HERE
;COMPARE DRIVE NUMBERS
;BRANCH IF A MATCH
;BUMP ADDRESS
;LOOK AT ALL OF THEM
;UNIT NUMBER NOT FOUND
;CLEAR Z AS ERROR FLAG

4$:     MOV     (R4),R4
        MOVB   D.UNIT(R4),L#LUN
        POP    R2
        SEZ
        RETURN
;GET ADDRESS OF TABLE
;GET UNIT NUMBER
;SET Z FLAG

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 113  
 PRE-PROGRAMMED SUBROUTINES

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 015474
15 015476 005001
16 015500 121227 000060
17 015504 103415
18 015506 121227 000071
19 015512 101012
20 015514 006301
21 015516 010100
22 015520 006301
23 015522 006301
24 015524 060001
25 015526 112200
26 015530 162700 000060
27 015534 060001
28 015536 000760
29 015540 005701
30 015542 001001
31 015544 005201
32 015546
33 015550 000207

;GETCNT
;
;GET COUNT IN NEXT CHARACTERS OF STRING POINTED TO BY R2.
;NUMBER WILL BE IN DECIMAL. IF NO NUMBER, RETURN A
;DEFAULT OF 1.
;
;INPUTS:
; R2 - POINTER TO ASCII STRING
;OUTPUTS:
; R1 - NUMBER READ OR A ONE
; R2 - POINTING TO CHARACTER AFTER NUMBER

GETCNT: PUSH R0
        CLR R1
        ;START WITH ZERO COUNT
GETCNX: CMPB (R2),#0
        BLO GETCDN
        ;CHECK IF CHARACTER A DIGIT
        ;BRANCH IF LOWER THAN ZERO
        CMPB (R2),#9
        BHI GETCDN
        ;BRANCH IF HIGHER THAN NINE
        ASL R1
        ;MULTIPLY NUMBER BY 10
        MOV R1,R0
        ;SAVE 2N
        ASL R1
        ;COMPUTE 4N
        ASL R1
        ;COMPUTE 8N
        ADD R0,R1
        ;8N + 2N = 10N
        MOVB (R2),R0
        ;GET DIGIT FROM STING
        SUB #0,R0
        ;GET RID OF ASCII
        ADD R0,R1
        ;ADD TO NUMBER
        BR GETCNX
        ;GO TO NEXT CHARACTER
GETCDN: TST R1
        BNE GETCXX
        ;CHECK IF NUMBER IS ZERO
        ;IF ZERO, CHANGE
        INC R1
        ; TO DEFAULT OF ONE
GETCXX: POP R0
        RETURN

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15 015552 010100
16 015554 004737 015474
17 015560
18 015566 012403
19 015570 005005
20 015572 020127 000020
21 015576 003401
22 015600 012405
23 015602
24 015604 010504
25 015606 012702 000020
26 015612 160102
27 015614 002002
28 015616 062702 000020
29 015622 001414
30 015624 012705 100000
31 015630 005302
32 015632 001402
33 015634 006205
34 015636 000774
35 015640 020127 000020
36 015644 003402
37 015646 040504
38 015650 000401
39 015652 040503
40 015654 004737 016014
41 015660
42 015662 005202
43 015664 005703
44 015666 001372
45 015670 005704
46 015672 001370
47 015674 020027 000012
48 015700 001423
49 015702 010103
50 015704 162700 000014
51 015710 003002
52 015712 012700 000003
53 015716 004737 016014
54 015722 005705
55 015724 001401
56 015726 005203
57 015730 160203

;PNTNUM
;
;PRINT A NUMBER
;
;INPUTS:
; R1 - RADIX OF NUMBER
; R2 - ASCII STRING TO COUNT OF BITS IN NUMBER
; R4 - POINTER TO NUMBER (LOW WORD)
;OUTPUTS:
; NUMBER IS PRINTED. LEADING ZEROS ARE PRINTED EXCEPT FOR
; DECIMAL NUMBERS (LEFT JUSTIFIED).
; R0 - CONTENTS DESTROYED

PNTNUM: MOV R1,R0 ; SAVE RADIX
CALL GETCNT ; GET COUNT OF BITS
PNTNUS: PUSH <R2,R3,R5> ; SAVE REGISTERS ON STACK
MOV (R4),R3 ; GET ONE PARAMETER WORD
CLR R5 ; CLEAR STORAGE FOR OTHER
CMP R1,#16. ; MORE THAN 16 BITS IN NUMBER?
BLE 1# ; NO, SKIP
MOV (R4),R5 ; YES, GET SECOND PARAMETER WORD
1#: PUSH R4 ; SAVE R4 ON STACK
MOV R5,R4 ; PUT LOW WORD IN R4
MOV #16.,R2 ; COMPUTE BITS NOT WANTED
SUB R1,R2 ; BY SUBTRACTING BITS TO USE
BGE 2# ; FROM 16.
ADD #16.,R2 ; IF NEGATIVE, ADD 16 FOR FIRST WORD
2#: BEQ 6# ; IF ZERO, NO BITS NEED BE CLEARED
MOV #BIT15,R5 ; START MASK WITH SIGN BIT SET
3#: DEC R2 ; COUNT BITS IN MASK
BEQ 4# ;
ASR R5 ; SHIFT MORE BITS TO RIGHT
BR 3# ;
4#: CMP R1,#16. ; MORE THAN 16 BITS IN NUMBER?
BLE 5# ;
BIC R5,R4 ; YES, CLEAR IN HIGH WORD
BR 6# ;
5#: BIC R5,R3 ; NO, CLEAR IN LOW WORD
6#: CALL DIVIDE ; DIVIDE BY RADIX IN R0
PUSH R5 ; PUSH REMAINDER ON STACK
INC R2 ; COUNT DIGITS ON STACK
TST R3 ; CHECK IF QUOTIENT IS ZERO
BNE 6# ;
TST R4 ;
BNE 6# ;
CMP R0,#10. ; IF RADIX IS DECIMAL
BEQ 10# ; JUST GO PRINT DIGITS ON STACK
MOV R1,R3 ; OTHERWISE COMPUTE NUMBER OF LEADING 0
SUB #12.,R0 ; DIVIDEND IS BITS IN NUMBER
BGT 7# ; DIVISOR IS BITS PER DIGIT PRINTED
MOV #3,R0 ; (3 OR 4)
7#: CALL DIVIDE ;
TST R5 ; IF REMAINDER NOT ZERO
BEQ 8# ; INCREMENT QUOTIENT
INC R3 ;
8#: SUB R2,R3 ; SUBTRACT DIGITS ON STACK

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 114-1  
PRE-PROGRAMMED SUBROUTINES

58 015732 001406		BEG	10:		; NO LEADING ZEROS IF ZERO
59 015734	9:	PRINT		0'0	;
60 015744 005303		DEC		R3	;
61 015746 001372		BNE		9:	; REPEAT UNTIL COUNT REACHES ZERO
62 015750	10:	POP		R5	; GET CHARACTER FROM STACK
63 015752 062705 000060		ADD		0'0,R5	; CNVERT TO ASCII DIGIT
64 015756 020527 000071		CMP		R5,0'9	; IF GREATER THAN A 9
65 015762 003402		BLE		11:	; CONVERT TO A OR HIGHER
66 015764 062705 000007		ADD		0<'A-'9-1>,R5	; FOR HEX DIGIT
67 015770	11:	PRINT		R5	;
68 015776 005302		DEC		R2	; REPEAT FOR ALL DIGITS
69 016000 001363		BNE		10:	; ON STACK
70 016002		POP		<R4,R5,R3,R2>	;
71 016012 000207		RETURN			;
72					;

CZUDMAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-94 10:38 Page 115  
 PRE-PROGRAMMED SUBROUTINES

```

1
2      ;DIVIDE
3      ;
4      ;DIVIDE A 32 BIT UNSIGNED NUMBER BY A 16 BIT UNSIGNED NUMBER.
5      ;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
6      ;WILL NOT CHECK FOR DIVIDE BY ZERO.
7      ;
8      ;INPUTS:
9      ;   R3 - LOW 16 BITS OF DIVIDEND
10     ;   R4 - HIGH 16 BITS OF DIVIDEND
11     ;   R0 - DIVISOR
12     ;OUTPUTS:
13     ;   R3 - LOW 16 BITS OF QUOTIENT
14     ;   R4 - HIGH 16 BITS OF QUOTIENT
15     ;   R5 - REMAINDER
16
17 016014      DIVIDE: PUSH      R2
18 016016      012702 000040    MOV      #32.,R2      ;SET UP SHIFT COUNT
19 016022      005005          CLR      R5          ;START WITH ZERO REMAINDER
20 016024      006303          14:  ASL      R3          ;SHIFT LEFT INTO R5
21 016026      006104          ROL      R4
22 016030      006105          ROL      R5
23 016032      020005          CMP      R0,R5      ;WILL DIVISOR GO INTO REMAINDER
24 016034      101002          BHI      24          ;ONLY SUBTRACT IF IT WILL
25 016036      160005          SUB      R0,R5      ;SUBTRACT DIVISOR
26 016040      005203          INC      R3          ;PUT A ONE INTO QUOTIENT
27 016042      005302          24:  DEC      R2          ;COUNT THE SHIFTS
28 016044      001367          BNE      14
29 016046          POP      R2
30 016050      000207          RETURN

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 016052
21 016054 012700 000100
22 016060 005005
23 016062 006301
24 016064 006102
25 016066 006103
26 016070 006104
27 016072 006105
28 016074 022705 000012
29 016100 101003
30 016102 162705 000012
31 016106 005201
32 016110 005300
33 016112 001363
34 016114
35 016116 000207
36

```

```

;DIV10
;
;DIVIDE A 64 BIT UNSIGNED NUMBER BY A 10.
;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
;WILL NOT CHECK FOR DIVIDE BY ZERO.
;
;INPUTS:
;   R1 - LOW 16 BITS OF DIVIDEND
;   R2 - NEXT 16 BITS OF DIVIDEND
;   R3 - NEXT 16 BITS OF DIVIDEND
;   R4 - HIGH 16 BITS OF DIVIDEND
;OUTPUTS:
;   R1 - QUOTIENT,
;   R2 - QUOTIENT,
;   R3 - QUOTIENT,
;   R4 - QUOTIENT,
;   R5 - REMAINDER

```

```

DIV10:  PUSH    R0
        MOV     #64.,R0
        CLR    R5
14:     ASL     R1
        ROL    R2
        ROL    R3
        ROL    R4
        ROL    R5
        CMP    #10.,R5
        BHI    24
        SUB    #10.,R5
        INC    R1
24:     DEC    R0
        BNE   14
        POP    R0
        RETURN
;DIVIDEND IS IN <R4,R3,R2,R1>
;SET UP SHIFT COUNT
;START WITH ZERO REMAINDER
;SHIFT LEFT INTO R5
;SILL DIVISOR GC INTO REMAINDER?
;ONLY SUBTRACT IF IT WILL
;SUBTRACT DIVISOR
;PUT A ONE INTO QUOTIENT
;COUNT THE SHIFTS
;RETURN WITH QUOTIENT IN
; <R4,R3,R2,R1> AND REMAINDER IN R5

```

```

1
2
3
4
5
6
7
8 016120
9 016130
10 016132 000301
11 016134 004737 016160
12 016140
13 016142 004737 016160
14 016146 000207
15
16
17
18
19
20
21
22 016150
23
24
25
26
27
28
29
30 016160
31 016162 006001
32 016164 006001
33 016166 006001
34 016170 006001
35 016172 004737 016200
36 016176
37 016200 042701 177760
38 016204 062701 000060
39 016210 020127 000071
40 016214 003402
41 016216 062701 000007
42 016222
43 016230 000207
44

```

```

; **
;
; T2PNTW - PRINT HEX NUMBERS WITH LEADING SPACE
;
; THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTERVENTION QUESTIONS
; --
T2PNTW: PRINT <@' > ; PRINT A SPACE
        PUSH R1 ; SAVE R1 ON STACK
        SWAB R1 ; GET HIGH BYTE
        CALL T2PNT ; PRINT HIGH TWO DIGITS
        POP R1 ; RESTORE R1
        CALL T2PNT ; PRINT LOW TWO DIGITS
        RETURN ;
; **
;
; T2PNTB - PRINT A SPACE
;
; THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTERVENTION QUESTIONS
; --
T2PNTB: PRINT <@' > ; PRINT A SPACE
; **
;
; T2PNT - PRINT TWO HEX DIGITS FROM NUMBER IN R1
;
; THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTERVENTION QUESTIONS
; --
T2PNT: PUSH R1 ; SAVE NUMBER
        ROR R1 ; SHIFT
        ROR R1 ; TO GET
        ROR R1 ; HIGH
        ROR R1 ; DIGIT
        CALL T2PNT0 ; PRINT TWO DIGITS
        POP R1 ; GET LOW DIGIT AGAIN
T2PNT0: BIC @'C17,R1 ; CLEAR OTHER BITS
        ADD @'0,R1 ; CONVERT TO ASCII CHARACTER
        CMP R1,@'9 ; IF GREATER THAN 9
        BLE T2PNTD ; CONVERT
        ADD @<'A-'9-1>,R1 ; TO A - F
T2PNTD: PRINT R1 ; PRINT THE DIGIT
        RETURN

```



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15 C16232 005000
16 016234 105711
17 016236 001442
18 016240 121127 000040
19 016244 001002
20 016246 005201
21 016250 000770
22 016252 005004
23 016254
24 016256 112102
25 016260 162702 000060
26 016264 100431
27 016266 020227 000011
28 016272 003410
29 016274 020227 000021
30 016300 103423
31 016302 020227 000026
32 016306 101020
33 016310 162702 000007
34 016314 006304
35 016316 006304
36 016320 006304
37 016322 006304
38 016324 050204
39 016326 005200
40 016330
41 016332 105711
42 016334 001403
43 016336 121127 000040
44 016342 001344
45 016344 005700
46 016346 000207
47
48 016350
49 016354 000137 014546
51

```

```

: **
:
: T2GNUM - GET A HEX DIGIT FROM AN ASCII INPUT STRING
:
: THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTERVENTION QUESTIONS
:
: INPUTS:
: R1 - STRING POINTER
:
: OUTPUTS:
: R4 - NUMBER
: R1 - UPDATED STRING TO CHARACTER AFTER NUMBER
: R0 - COUNT OF DIGITS (0 IF END OF LINE FOUND)
: --

```

```

T2GNUM: CLR R0 ; CLEAR DIGIT COUNT
TSTB (R1) ; CHECK IF END OF LINE
BEQ T2GNX ; REPORT NULL CHARACTER FOUND
CMPB (R1),#' ; CHECK IF A SPACE
BNE T2GND1 ; IF SO, IGNORE IT
INC R1 ; GET NEXT
BR T2GNUM ; DIGIT
T2GND1: CLR R4 ; CLEAR NUMBER STORAGE
T2GND2: PUSH R2 ; SAVE REGISTER
MOV B (R1),R2 ; GET CHARACTER
SUB #'0,R2 ; CONVERT ASCII TO NUMERIC
BMI T2GNE ; IF LESS THAN 0, INVALID DIGIT
CMP R2,#9 ; IF LESS THAN OR EQUAL TO 9
BLE T2GND3 ; CONVERSION DONE
CMP R2,#'A-'0> ; IF LESS THAN A (10.),
BLO T2GNE ; INVALID DIGIT
CMP R2,#'F-'0> ; IF GREATER THAN F (15.),
BHI T2GNE ; INVALID DIGIT
SUB #'A-'9-1>,R2 ; CONVERT FROM A-F TO 10.-15.
T2GND3: ASL R4 ; MAKE
ASL R4 ; ROOM
ASL R4 ; FOR
ASL R4 ; DIGIT
BIS R2,R4 ; STORE DIGIT
INC R0 ; GET NEXT
POP R2 ; DIGIT
TSTB (R1) ; IF 0,
BEQ T2GNX ; END OF NUMBER
CMPB (R1),#' ; IF NOT A SPACE,
BNE T2GND2 ; CONTINUE
T2GNX: TST R0 ; Z = 1 FOR ERROR, 0 FOR SUCCESS
RETURN ;
T2GNE: POP <R2,R0> ; INVALID DIGIT
JMP DMRQ2E ; ERROR

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 016360
15 016366 005764 000004
16 016372 002004
17 016374 116537 000002 002074
18 016402 000417
19 016404
20 016406 016401 000004
21 016412 004737 015404
22 016416 001037
23 016420 005764 000002
24 016424 100005
25 016426 052713 100000
26 016432
    016432 104424
27 016434
28 016436 000423
29
30 016440
31 016442 012702 002150
32 016446 016412 000002
33 016452 006112
34 016454 006112
35 016456 006112
36 016460 042722 177774
37 016464 016412 000002
38 016470 042722 140000
39 016474 005022
40 016476 012712 011760
41 016502
    016502 104460
42 016504 000241
43 016506
44 016514 000207
45 016516
46 016520 000261
47 016522 000771
48

; **
;
; PNTERR - PRINT ERROR MESSAGE FROM DM PROGRAM REQUEST !1 OR 12.
;
; INPUTS:
; R5 - CONTROLLER TABLE ADDRESS
; R4 - MESSAGE DATA ADDRESS
; R3 - COMMAND DATA ADDRESS
;
; OUTPUTS:
; ERROR MESSAGE PRINTED
; BIT 15 SET IN COMMAND DATA IF DRIVE HAS BEEN DROPPED
; --

PNTERR: PUSH <R0,R1,R2> ; SAVE REGISTERS ON STACK
        TST 4(R4) ; CHECK IF BIT 15 SET
        BGE 1$ ; IF SO, GET UNIT FROM MESSAGE DATA
        MOVB C.UNIT(R5),L$LUN ; ELSE, GET UNIT FROM CONTROLLER TABLE
        BR 2$ ;
        1$: PUSH R4 ; SAVE DATA ADDRESS
            MOV 4(R4),R1 ; GET DRIVE NUMBER
            CALL GTDRVT ; GET DRIVE TABLE ADDRESS
            BNE 5$ ; IF UNIT DROPPED, EXIT
            TST D.UNIT(R4) ; IS TESTING DONE ON UNIT?
            BPL 3$ ; IF NOT DONE, PROCEED
            BIS #BIT15,(R3) ; MARK UNIT AS DONE TESTING
            DRPT ; PRINT A STATISTICAL REPORT
            TRAP C$DRPT
            POP R4 ; RESTORE DATA ADDRESS
            BR 4$ ;
        3$: POP R4 ; RESTORE DATA ADDRESS
        2$: MOV #ERRTYP,R2 ; GET POINTER TO ERROR TABLE
            MOV 2(R4),(R2) ; GET ERROR TYPE
            ROL (R2) ;
            ROL (R2) ;
            ROL (R2) ;
            BIC #C3,(R2) ; CLEAR LOW 2 BITS
            MOV 2(R4),(R2) ;
            BIC #140000,(R2) ; MASK LOW 14 BITS
            CLR (R2) ; CLEAR MESSAGE POINTER
            MOV #ERR.TN,(R2) ; GET ROUTINE NUMBER
            ERROR ; PRINT THE ERROR MESSAGE
            TRAP C$ERROR
            CLC ; DRIVE HAS NOT BEEN DROPPED
        4$: POP <R2,R1,R0> ; RESTORE REGISTERS
            RETURN ;
        5$: POP <R4> ; RESTORE STACK
            SEC ; DRIVE HAS BEEN DROPPED
            BR 4$ ;
    
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15 016524
16 016524 016504 000004
17 016530 042704 177000
18 016534 010501
19 016536 062701 000006
20 016542
    016542 012746 000340
    016546 010146
    016550 010446
    016552 012746 000003
    016556 104437
    016560 062706 000010
21 016564 006204
22 016566 006204
23 016570 004737 017676
24 016574 001526
25 016576 004737 012156
27 016602 023727 002204 000001
28 016610 001053
29
30
31
32
33 016612 017704 163362
34 016616 162704 000040
35 016622 013700 002200
36 016626 062700 000040
37 016632 005001
38
39 016634 012703 000214
40 016640 020403
41 016642 103001
42 016644 010403
43 016646 013702 002160
44 016652 162702 000214
45 016656
46 016662 012022
47 016664 162703 000002
48 016670 001374
49 016672
50 016676 010037 002340
51 016702 012700 000031
52 016706 004737 017056

; **
;
; LOADDM - LOAD AND START A DM PROGRAM IN A CONTROLLER
;
; INPUTS:
; R5 - CONTROLLER TABLE ADDRESS
;
; IMPLICIT INPUTS:
; DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
;
; OUTPUTS:
; IF LOAD SUCCEEDS - Z CLEAR
; CONTROLLER TABLE MARKED LOADED
; IF ERROR - Z SET
; --

LOADDM:
MOV C.VEC(R5),R4 ; GET VECTOR OF CONTROLLER
BIC #C<CT.VEC>,R4 ; CLEAR ALL BUT VECTOR
MOV R5,R1 ; GET INTERRUPT SERVICE LINK
ADD #C.JSR,R1 ;
SETVEC R4,R1,#PRI07 ; SET UP INTERRUPT VECTOR
MOV #PRI07,-(SP)
MOV R1,-(SP)
MOV R4,-(SP)
MOV #3,-(SP)
TRAP C+SVEC
ADD #10,SP
ASR R4 ; INITIALIZE CONTROLLER WITH SMALLEST
ASR R4 ; POSITION VECTOR FOR RING
CALL CNTINT ; BUFFER AND INTERRUPTS ENABLED
BEQ LOADER ; IF ERROR, EXIT
CALL HCOMM ; ALLOCATE SPACE FOR HOST COMM AREA
CMP TNJM,#1 ; IF TEST NUMBER <> 1
BNE LOADTX ; DON'T DO SPECIAL LOAD

;
; LOAD DM PROGRAM INTO MEMORY SPACE TESTED DURING
; INITIALIZATION IN TEST 1
;
MOV #DMPROG,R4 ;GET SIZE OF DM PROGRAM IN BYTES
SUB #DMMAIN,R4 ;
MOV DMPROG,R0 ;GET ADDRESS OF DM PROGRAM
ADD #DMMAIN,R0 ;
CLR R1 ;START WITH OFFSET OF ZERO

LT1L1: MOV #<HC.BSZ*2>,R3 ; GET SIZE OF BUFFER
CMP R4,R3 ; IF FEWER BYTES REMAINING IN PROGRAM
BHIS LT11 ; SKIP
MOV R4,R3 ; ELSE, USE ACTUAL BYTE COUNT
LT11: MOV FFREE,R2 ; GET ADDRESS OF BUFFER
SUB #<HC.BSZ*2>,R2 ;
PUSH <R3,R2> ; SAVE THE BYTE COUNT & BUFFER ADDRESS
LT1L2: MOV (R0)+,(R2)+ ; MOVE DATA TO BUFFER
SUB #2.,R3 ; DECREMENT WORDS REMAINING
BNE LT1L2 ; LOOP IF MORE WORDS TO LOAD
POP <R2,R3> ; RESTORE BUFFER ADDRESS
MOV R0,DMPTR ; SAVE POINTER INTO DM PROGRAM
MOV #OP.MWR,R0 ; GET DOWNLINE LOAD COMMAND
CALL LOAD ; LOAD INTO CONTROLLER
    
```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 120-1  
 PRE-PROGRAMMED SUBROUTINES

```

53 016712 001457          BEQ      LOADER          ; IF ERROR, GET OUT NOW
54 016714 013700 002340   MOV      DMPTR,R0        ; RESTORE POINTER INTO DM PROGRAM
55 016720 006203          ASR      R3              ; CONVERT BYTES TO WORDS
56 016722 060301          ADD      R3,R1          ; INCREASE OFFSET FOR NEXT BUFFER
57 016724 006303          ASL      R3              ; CONVERT WORDS TO BYTES
58 016726 160304          SUB      R3,R4          ; REDUCE REMAINING BYTE COUNT
59 016730 001341          BNE     LT1L1          ; GET NEXT BUFFER
60 016732 012701 000040   MOV      @DMMAIN,R1     ; GET A BYTE COUNT OF HEADER ONLY
61 016736 000402          BR       LOADB         ; NOW START
63
64 016740 017701 163234   LOADTX: MOV     @DMPROG,R1 ; GET SIZE OF PROGRAM
65 016744 012700 000002   LOADB: MOV     @OP.ESP,R0 ; BUILD EXECUTE SUPPLIED
66 016750 004737 017176   CALL    BLD CMD        ; PROGRAM COMMAND PACKET
67 016754 013764 002200 000040   MOV     DMPROG,HC.CPK+P.UADR(R4); LOAD MAIN PROGRAM ADDRESS
68 016762 010164 000034   MOV     R1,HC.CPK+P.BCNT(R4) ; AND SIZE
69 016766 013764 002200 000054   MOV     DMPROG,HC.CPK+P.OVRL(R4); LOAD OVERLAY ADDRESS
70 016774 067764 163200 000054   ADD     @DMPROG,HC.CPK+P.OVRL(R4)
78 017002 004737 017270   CALL    SNDCMD         ; SEND COMMAND TO CONTROLLER
79 017006 004737 017400   CALL    WAITMS         ; WAIT FOR MESSAGE RESPONSE
80 017012 001417          BEQ     LOADER         ; EXIT IF ERROR
81 017014 032764 000037 000032   BIT     @ST.MSK,HC.MPK+P.STS(R4); CHECK FOR ERRORS
82 017022 001007          BNE     LOADE1         ;
83 017024 042765 000024 000012   BIC     @CT.CMD+CT.REQ,C.FLG(R5); CLEAR COMMAND OUTSTANDING FLAG
84 017032 052765 000002 000012   BIS     @CT.RN,C.FLG(R5) ; SET DM PROGRAM RUNNING FLAG
85
86 017040 000207          RETURN                ; SUCCESS RETURN
87
88
89
90
91 017042          LOADE1: ERRDF    33,,ERR033 ;
   017042 104455   TRAP    C#ERRDF ;
   017044 000041   .WORD  33 ;
   017046 000000   .WORD  0 ;
   017050 011474   .WORD  ERR033 ;
92 017052 000264          LOADER: SEZ          ; SET Z TO INDICATE ERROR OCCURRED
93 017054 000207          RETURN                ; ERROR RETURN
94

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 017056          LOAD:  PUSH    <R0,R3,R4>
18 017064 004737 017176  CALL    BLDCMD          ; BUILD COMMAND PACKET
19 017070 010264 000040  MOV     R2,HC.CPK+P.UADR(R4) ; STUFF IN BUFFER ADDRESS
20 017074 010364 000034  MOV     R3,HC.CPK+P.BCNT(R4) ; STUFF IN BYTE COUNT
21 017100 010164 000060  MOV     R1,HC.CPK+P.RGOF(R4) ; STUFF IN OFFSET
22 017104 012764 000001 000054  MOV     #1,HC.CPK+P.RGID(R4) ; STUFF IN REGION ID 1
23 017112 004737 017270  CALL    SNDCMD          ; SEND COMMAND TO CONTROLLER
24 017116 004737 017400  CALL    WAITMS          ; WAIT FOR MESSAGE RESPONSE
25 017122 001420          BEQ     LOADX          ; IF FAILED, EXIT
26 017124 032764 000037 000032  BIT     #ST.MSK,HC.MPK+P.STS(R4); LOOK FOR ANY ERROR
27 017132 001010          BNE     LOADX1         ;
28 017134 042765 000004 000012  BIC     #CT.CMD,C.FLG(R5) ; CLEAR COMMAND ISSUED
29 017142          POP     <R4,R3,R0>   ; RESTORE REGISTERS
30 017150 000244          CLZ          ; CLEAR Z TO INDICATE NO ERROR
31 017152 000207          RETURN          ;
32
33          ;
34          ;
35 017154          LOADX1: ERRDF  33,,ERR033          ; PRINT ERROR INFO FROM CONTROLLER
    017154 104455          TRAP   C#ERDF
    017156 000041          .WORD 33
    017160 000000          .WORD 0
    017162 011474          .WORD ERR033
36 017164          LOADX:  POP     <R4,R3,R0>   ; RESTORE STACK AFTER ERROR
37 017172 000264          SEZ          ; SET Z TO INDICATE ERROR OCCURRED
38 017174 000207          RETURN

```

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

```

: **
:
: BLD CMD - BUILD A COMMAND IN COMMAND PACKET
:
: INPUTS:
: R5 - CONTROLLER TABLE ADDRESS
: R0 - COMMAND CODE
:
: OUTPUTS:
: R4 - ADDRESS OF HOST COMM AREA
: COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL
: OTHER FIELDS CLEARED.
: COMMAND REFERENCE NUMBER IN CONTROLLER TABLE INCREMENTED
: AND RESULT IN COMMAND PACKET.
: R0 - CONTENTS DESTROYED
: --
    
```

```

17 017176
18 017202 016504 000014
19 017206 010400
20 017210 062700 000014
21 017214 012720 000060
22 017220 012701 001000
23 017224 022716 000031
24 017230 001403
25 017232 022716 000030
26 017236 001002
27 017240 012701 177777
28 017244 010120
29 017246 012701 000030
30 017252 005020
31 017254 005301
32 017256 001375
33 017260
34 017264
35 017266 000207
36
    
```

```

BLDCMD: PUSH <R1,R0> ; SAVE REGISTERS ON STACK
        MOV C.HCOM(R5),R4 ; GET ADDRESS OF HOST COMM AREA
        MOV R4,R0 ; COPY TO R0
        ADD #HC.CEV,R0 ; COMPUTE ADDRESS OF COMMAND ENVELOPE
        MOV #HC.PSZ,(R0)+ ; LOAD PACKET LENGTH
        MOV #DUP,R1 ; LOAD DUP CIRCUIT IDENTIFIER
        CMP #OP.MWR,(SP) ; IF CODE IS MAINTENANCE WRITE
        BEQ 1$ ; USE DIAGNOSTIC CIRCUIT ID
        CMP #OP.MRD,(SP) ; IF CODE IS NOT MAINTENANCE READ
        BNE BLDC0 ; SKIP
1$: MOV #DIAG,R1 ; ELSE, USE DIAGNOSTIC CIRCUIT ID
BLDC0: MOV R1,(R0)+ ; PUT IDENTIFIER INTO PACKET
        MOV #<HC.PSZ>/2,R1 ; GET WORDS TO CLEAR
BLDC1: CLR (R0)+ ; CLEAR PACKET
        DEC R1 ; ANY MORE
        BNE BLDC1 ; WORDS TO CLEAR?
        POP HC.CPK+P.OPCD(R4) ; PUT OPCODE IN PACKET
        POP R1 ; RESTORE R1
        RETURN
    
```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRU V05.01b Sunday 07-Oct-84 10:38 Page 123  
 PRE-PROGRAMMED SUBROUTINES

```

1
2
3
4
5
6
7
8
9
10
11
12
13 017270 016504 000014
14 017274 005265 000032
15 017300 016564 000032 000020
16 017306 012764 140000 000006
17 017314 012764 100000 000012
18 017322 005775 000000
19 017326 052765 000004 000012
20 017334 000207
21

```

```

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

```

```

SNDCMD: MOV C.HCOM(R5),R4 ; LOAD R4 WITH HOST COMM AREA ADDRESS
INC C.REF(R5) ; INCREMENT CMD REFERENCE NUMBER
MOV C.REF(R5),HC.CPK.P.CRF(R4); PUT IN PACKET
MOV @RG.OWN+RG.FLG,HC.MCT(R4) ; MARK MESSAGE PACKET AVAILABLE
MOV @RG.OWN,HC.CCT(R4) ; MARK COMMAND TO CONTROLLER
TST @R5 ; TELL CONTROLLER COMMAND IS THERE
BIS @CT.CMD,C.FLG(R5) ; MARK COMMAND ISSUED
RETURN

```

```

1
2
3      ;**
4      ; CLRBUF - CLEAR THE SPECIFIED DATA BUFFER IN THE HOST COMM AREA
5      ; AND LOAD BUFFER DESCRIPTOR IN COMMAND PACKET TO THE BUFFER
6      ;
7      ; INPUTS:
8      ; R5 - CONTROLLER TABLE ADDRESS
9      ; R4 - ADDRESS OF HOST COMM AREA
10     ; R0 - OFFSET INTO HOST COMM AREA TO DATA BUFFER
11     ;
12     ; OUTPUTS:
13     ; DATA BUFFER CLEARED
14     ; COMMAND PACKET POINTING TO BUFFER
15     ; BYTE COUNT SET TO SIZE OF BUFFER
16     ; R4 - ADDRESS OF DATA BUFFER
17     ;--
17 017336 CLRBUF: PUSH <R0,R1> ; SAVE REGISTERS ON STACK
18 017342 060400 ADD R4,R0 ; CREATE BUFFER ADDRESS
19 017344 010064 000040 MOV R0,HC.CPK.P.UADR(R4) ; PUT BUFFER ADDRESS IN COMMAND PACKET
20 017350 012764 000106 000034 MOV #HC.BSZ,HC.CPK.P.BCNT(R4) ; PUT SIZE OF BUFFER IN COMMAND PACKET
21 017356 010004 MOV R0,R4 ; PUT BUFFER ADDRESS IN R4
22 017360 012701 000043 MOV #<HC.BSZ>/2,R1 ; GET SIZE OF BUFFER IN WORDS
23 017364 005020 CLRBFL: CLR (R0) ; CLEAR ALL THE WORDS
24 017366 005301 DEC R1 ; ANY MORE :
25 017370 001375 BNE CLRBFL ; WORDS TO CLEAR?
26 017372 POP <R1,R0> ;
27 017376 000207 RETURN ;
28

```



```

1
2
3      ;**
4      ;      WAITMS - WAIT FOR CONTROLLER TO RESPOND WITH A MESSAGE PACKET
5      ;
6      ;      INPUTS:
7      ;      R5 - ADDRESS OF CONTROLLER TABLE
8      ;
9      ;      OUTPUTS:
10     ;      Z CLEAR IF NO ERROR
11     ;      Z SET IF ERROR, MESSAGE PRINTED
12     ;--
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

12	017400				WAITMS: PUSH	<R0,R1>	:	SAVE REGISTERS ON STACK
13	017404	012700	000036		MOV	#30.,R0	:	SET TIME OUT VALUE OF 30 SECONDS
14	017410	010501			MOV	R5,R1	:	POINT TO TIME OUT COUNTER
15	017412	062701	000026		ADD	#C.TO,R1	:	POINTER TO TIMER FIELD
16	017416	004737	017614		CALL	SETTO	:	START TIMER
17	017422	011500			MOV	(R5),R0	:	GET ADDRESS OF IP REGISTER
18	017424	032765	000010	000012	BIT	#CT.MSG,C.FLG(R5)	:	LOOK IF INTERRUPT OCCURRED
19	017432	001025			BNE	3#	:	BRANCH IF SO
20	017434	016001	000002		MOV	2(R0),R1	:	LOOK AT SA REGISTER
21	017440	001031			BNE	4#	:	BRANCH IF ERROR CODE PRESENT
27	017442				BREAK		:	>>>>>>BREAK BACK TO MONITOR<<<<<<<<
	017442	104422			TRAP	C#BRK		
28	017444	005737	002256		TST	KW.CSR	:	SEE IF A CLOCK ON SYSTEM
29	017450	001764			BEQ	1#	:	IF NOT, DON'T TIMEOUT
30	017452	023765	002270	000030	CMP	KW.EL+2,C.TOH(R5)	:	CHECK HIGH WORD OF INTERVAL
31	017460	101005			BHI	2#	:	IF GREATER, TIMED OUT
32	017462	001357			BNE	1#	:	IF NOT EQUAL, NO TIMEOUT OCCURRED
33	017464	023765	002266	000026	CMP	KW.EL,C.TO(R5)	:	CHECK LOW WORD OF INTERVAL
34	017472	103753			BLO	1#	:	IF LOWER, NO TIMEOUT OCCURRED
35	017474				ERRDF	36,,ERR036	:	PRINT TIMEOUT ERROR
	017474	104455			TRAP	C#ERDF		
	017476	000044			.WORD	36		
	017500	000000			.WORD	0		
	017502	011520			.WORD	ERR036		
36	017504	000413			BR	5#	:	ERROR EXIT
37								
38	017506	042765	000010	000012	BIC	#CT.MSG,C.FLG(R5)	:	CLEAR MESSAGE RECEIVED FLAG
39	017514				POP	<R1,R0>	:	SAVE REGISTERS ON STACK
40	017520	000244			CLZ		:	GIVE NO ERROR RETURN
41	017522	000207			RETURN		:	
42	017524						:	
44	017524				ERRDF	30,,ERR030	:	CONTROLLER DETECTED ERROR
	017524	104455			TRAP	C#ERDF		
	017526	000036			.WORD	30		
	017530	000000			.WORD	0		
	017532	011430			.WORD	ERR030		
51	017534				POP	<R1,R0>	:	RESTORE REGISTERS
52	017540	0C0264			SEZ		:	Z SET OR ERROR RETURN
53	017542	000207			RETURN		:	

```

1
2
3      ;**
4      ;      NXMI - NON-EXISTANT MEMORY SERVICE ROUTINE
5      ;
6      ;      INPUTS:
7      ;      NXMAD SET TO ZERO
8      ;      OUTPUTS:
9      ;      NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
10     ;--
11     017544      BGNSRV  NXMI
12     017544      NXMI::
13     017544      012737  177777  002276      MOV      @-1,NXMAD
14     017552      ENDSRV
15     017552      L10033:
16     017552      000002      RTI
17
18     ;**
19     ;      CNTSRV - CONTROLLER INTERRUPT SERVICE ROUTINE. MARKS CONTROLLER TABLE
20     ;      THAT AN INTERRUPT HAS BEEN RECEIVED.
21     ;
22     ;      THIS ROUTINE IS CALLED BY A [JSR RO,CNTSRV] INSTRUCTION FROM WITHIN
23     ;      THE CONTROLLER TABLE. THE PC STORED IN RO IS THE ADDRESS OF THE C.FLG
24     ;      WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
25     ;      OF RO FOLLOWED BY THE INTERRUPTED PC AND PS.
26     ;
27     ;      INPUTS:
28     ;      RO - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
29     ;      STACK - SAVED CONTENTS OF RO
30     ;      OUTPUTS:
31     ;      CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
32     ;      RO - RESTORED FROM STACK
33     ;--
34     017554      BGNSRV  CNTSRV
35     017554      CNTSRV::
36     017554      052710  000010      BIS      @CT.MSG,(RO)      ; SET CT.MSG
37     017560      POP      RO      ; RESTORE RO
38     017562      ENDSRV
39     017562      L10034:
40     017562      000002      RTI
41
42     ;**
43     ;      INTSRV - CONTROLLER INTERRUPT SERVER
44     ;--
45     017564      BGNSRV  INTSRV
46     017564      INTSRV::
47     017564      005237  002214      INC      INTRCV      ; FLAG INTERRUPT AS RECEIVED
48     017570      ENDSRV
49     017570      L10035:
50     017570      000002      RTI

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 126-1  
PRE-PROGRAMMED SUBROUTINES

```
017572
51 017572 062737 000001 002266 KW11I:: ADD #1,KW.EL ; COUNT THE INTERRUPT
52 017600 005537 002270 ADC KW.EL+2 ; PUT CARRY IN HIGH WORD
53 017604 012777 000105 162444 MOV #KW.OUT,@KW.CSR ; RESTART THE CLOCK
54 017612 ENDSRV
017612 L10036:
017612 000002 RTI
55
```

```

1
2
3      ;**
4      ;   SETTO - SET TIMEOUT COUNTER TO A GIVEN NUMBER OF SECONDS FROM CURRENT
5      ;   TIME.
6      ;
7      ;   INPUTS:
8      ;   R0 - NUMBER OF SECONDS FOR TIMEOUT
9      ;   R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT
10     ;
11     ;   OUTPUTS:
12     ;   R0 - CONTENTS DESTROYED
13     ;   R1 - INCREMENTED BY 2
14     ;--
15     SETTO:  PUSH    <R2,R3>
16             CLR     R2                ; CLEAR PRODUCT
17             MOV     KW.HZ,R3          ; GET MULTIPLICAND
18     SET00:  ASR     R0                ; SHIFT MULTIPLIER TO RIGHT
19             BCC     SET01             ; IF A ONE BIT SHIFTED OUT
20             ADD     R3,R2            ; ADD MULTIPLICAND TO PRODUCT
21     SET01:  ASL     R3                ; DOUBLE THE MULTIPLICAND
22             TST     R0
23             BNE     SET00            ; CONTINUE UNTIL MULTIPLIER IS ZERO
24
25     ;
26     ;   GET CURRENT TIME
27     ;
28     SET02:  MOV     KW.EL,R0          ; GET TIME (LOW WORD)
29             MOV     KW.EL+2,R3       ; GET TIME (HIGH WORD)
30             CMP     R0,KW.EL         ; IF CHANGED DURING RETRIEVAL
31             BNE     SET02            ; GET IT AGAIN
32
33     ;
34     ;   ADD TIME TIL TIMEOUT
35     ;
36     ;
37     ;
38     ADD     R2,R0                    ; ADD TIMEOUT TO CURRENT TIME
39     ADC     R3                        ; INCREMENT HIGH WORD IF CARRY
40
41     ;
42     ;   PUT RESULT IN STORAGE
43     ;
44     ;
45     MOV     R0,(R1)                  ; SAVE LOW WORD OF TIMEOUT
46     MOV     R3,(R1)                  ; SAVE HIGH WORD OF TIMEOUT
47
48     POP     <R3,R2>
49     RETURN
50

```

```

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 017676
32 017700 010400
33 017702 000300
34 017704 042700 177770
35 017710 004737 020724
36 017714 010102
37 017716 010400
38 017720 000300
39 017722 006000
40 017724 006000
41 017726 006000
42 017730 042700 177770
43 017734 004737 020724
44 017740 060201
45 017742 006301
46 017744 062701 000002
50 017750 020137 002162
51 017754 101402
52 017756 000137 012120
53 017762
55
56
57
58 017762 013702 002160
59 017766 010103
60 017770 012722 177777
61 017774 005303
62 017776 003374

; **
;
; CNTINT
;
; FUNCTIONAL DESCRIPTION:
; SUBROUTINE TO INITIALIZE A CONTROLLER AND BRING IT ON-LINE.
; ALL STEPS ARE CHECKED. AN ERROR MESSAGE IS REPORTED IF ANY ERROR
; DETECTED.
;
; INPUTS:
; R5 - ADDRESS OF CONTROLLER TABLE.
; R4 - LENGTH, INTERRUPT AND VECTOR FIELDS TO SEND TO CONTROLLER
;
; IMPLICIT INPUTS:
; FFREE - FIRST FREE ADDRESS OF MEMORY. THIS ADDRESS IS GIVEN TO
; CONTROLLER AS START OF RING BUFFER.
; FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS.
;
; OUTPUTS:
; R1 - SIZE OF RING BUFFER IN WORDS IF NO ERROR
; R4 - ADDRESS OF IP REGISTER IN CONTROLLER
; R5 - UNCHANGED
; Z - CLR IF NO ERROR, SET IF ANY ERROR REPORTED
;
;
; CHECK IF ENOUGH FREE MEMORY FOR RING BUFFER
;
CNTINT: PUSH <R3> ; SAVE R3 ON STACK
MOV R4,R0 ; GET MESSAGE LENGTH
SWAB R0 ; RIGHT JUSTIFY MESSAGE LENGTH
BIC #177770,R0 ; REMOVE EXTRA INFO
CALL CLOG ; COMPUTE POWER OF 2
MOV R1,R2 ; SAVE MESSAGE LENGTH+2 IN R2
MOV R4,R0 ; GET COMMAND LENGTH
SWAB R0 ; RIGHT
ROR R0 ; JUSTIFY
ROR R0 ; COMMAND
ROR R0 ; LENGTH
BIC #177770,R0 ; REMOVE EXTRA INFO
CALL CLOG ; COMPUTE POWER OF 2
ADD R2,R1 ; ADD THE TWO RESULTS
ASL R1 ; MULTIPLY BY 2 WORDS PER RING
ADD #<HC.ISZ>/2,R1 ; ADD SPACE FOR INTERRUPT INDICATORS
CMP R1,FSIZE ; COMPARE WITH SIZE OF FREE MEMORY
BLOS 1$ ; THERE IS PLENTY OF MEMORY
JMP FMERR ; FATAL ERROR IF NOT ENOUGH MEMORY

1$:
;
; FILL HOST COMMUNICATION AREA WITH ALL ONES
;
MOV FFREE,R2 ; GET FIRST ADDRESS OF RING BUFFER
MOV R1,R3 ; GET SIZE OF RING BUFFER
2$: MOV #-1,(R2) ; WRITE ONES TO BUFFER
DEC R3 ; COUNT THE WORDS IN BUFFER
BGT 2$ ; LOOP UNTIL ENTIRE BUFFER WRITTEN
    
```

```

64      ;
65      ; DO THE INITIALIZATION
66      ;
67 020000 004737 020272      CALL CNTIST      ; DO FIRST THREE STEPS
68 020004 103527              BCS 9#             ; EXIT IF MICROCODE REPORTED FAILURE
69 020006 012364 000002      MOV (R3)+,2(R4)   ; WRITE STEP 3 RESPONSE TO SA REGISTER
71 020012 012700 000310      MOV #200.,R0     ; GET TRY COUNTER
72 020016 016402 000002      3#: MOV 2(R4),R2     ; LOOK AT SA REGISTER
73 020022 001407              BEQ 5#           ; EXIT LOOP IF ZERO
74 020024 005300              DEC R0            ;
75 020026 001373              BNE 3#           ; KEEP LOOPING IF NOT ZERO
76 020030              ERRDF 24.,,ERR024 ;
      020030 104455          TRAP C#ERRDF ;
      020032 000030          .WORD 24 ;
      020034 000000          .WORD 0 ;
      020036 011312          .WORD ERR024 ;
77 020040 000511              BR 9#              ;
78 020042 005064 000002      5#: CLR 2(R4)         ; WRITE 0 TO SA REGISTER (PURGE)
79 020046 005714              TST (R4)           ; READ FROM IP REGISTER (POLL)
81 020050 004737 020566      CALL CNTRSP      ; WAIT FOR STEP OR ERROR BIT
82 020054 103503              BCS 9#             ; EXIT IF MICROCODE REPORTED FAILURE
83 020056 010237 002326      MOV R2,SSTEP4    ; SAVE STEP 4 RESPONSE VALUE.
85      ;
86      ; CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
87      ;
88 020062 013702 002160      MOV FFREE,R2     ; GET FIRST ADDRESS OF RING BUFFER
89 020066 010103              MOV R1,R3        ; GET SIZE OF RING BUFFER
90 020070 005722      6#: TST (R2)+           ; CHECK WORD IN BUFFER
91 020072 001003              BNE 7#           ; GO TO ERROR REPORTER IF NOT ZERO
92 020074 005303              DEC R3            ; COUNT THE WORDS IN BUFFER
93 020076 003374              BGT 6#           ; LOOP UNTIL ALL WORDS CHECKED
94 020100 000405              BR 8#            ; START CONTROLLER AND EXIT
95      ;
96 020102      7#: ERRDF 23.,,ERR023    ; REPORT BUFFER NOT CLEARED
      020102 104455          TRAP C#ERRDF ;
      020104 000027          .WORD 23 ;
      020106 000000          .WORD 0 ;
      020110 011232          .WORD ERR023 ;
97 020112 000464              BR 9#              ; ERROR EXIT
99      ;
100     ; SEND GO BIT TO SA REGISTER TO END INITIALIZATION
101     ;
102 020114      8#: MOV #SA.GO,R0         ; SET THE GO BIT
103 020114 012700 000001      MOV R0,2(R4)     ; WRITE TO SA REGISTER
104 020120 010064 000002      ;
105     ;
106     ; NOW CHECK IF THE CONTROLLER TYPE IS VALID AND MICROCODE VERSION
107     ; IS CURRENT
108     ;
109 020124 013703 002326      23#: MOV SSTEP4,R3    ; GET SAVED CONTROLLER STEP 4 RESPONSE
110 020130 010301              MOV R3,R1        ; R3 HAS STEP 4 INFO
111 020132 042701 177760      BIC #C<SA.MCV>,R1 ; R1 = MICRO CODE LEVEL
112 020136 006003              ROR R3            ; RIGHT
113 020140 006003              ROR R3            ; JUSTIFY
114 020142 006003              ROR R3            ; MODEL
115 020144 006003              ROR R3            ; NUMBER
116 020146 042703 177760      BIC #C<SA.CNT/16.>,R3 ; R3 = CONTROLLER MODEL NUMBER

```

```

117 020152 022703 000006          CMP      #6.,R3          ; CHECK IF UDA50A
118 020156 001413                BEQ      24$             ; IF SO, BRANCH
119 020160 022703 000015          CMP      #13.,R3         ; CHECK IF KDA50-Q
120 020164 001422                BEQ      25$             ; IF SO BRANCH
121 020166 052765 100000 000002   BIS      #BIT15,C.UNIT(R5) ; ELSE MARK AS DO NOT EXECUTE
122 020174                ERRDF   14.,ERR014      ; REPORT ERROR
    020174 104455                TRAP    C$ERDF
    020176 000016                .WORD  14
    020200 000000                .WORD  0
    020202 011116                .WORD  ERR014
123 020204 000427                BR      9$              ; EXIT
124 020206                24$:
125 020206 005037 002326          CLR      SSTEP4         ; CLEARED FOR TEST 1, SUBTEST 4
126 020212 020127 000003          CMP      R1,#3.         ; UDA50A MICROCODE VERSION UP TO DATE?
127 020216 002017                BGE     26$             ; IF SO, BRANCH
128 020220                ERRSF   14.,ERR014      ; ELSE, REPORT ERROR
    020220 104454                TRAP    C$ERSF
    020222 000016                .WORD  14
    020224 000000                .WORD  0
    020226 011116                .WORD  ERR014
129 020230 000412                BR      26$             ;
130 020232                25$:
131 020232 012737 000004 002326   MOV      #4.,SSTEP4     ; DEFAULT BR LVL, TST 1,SUBTST 4
132 020240 020127 000000          CMP      R1,#0.         ; KDA50-Q MICROCODE VERSION UP TO DATE?
133 020244 002004                BGE     26$             ; IF SO, BRANCH
134 020246                ERRSF   14.,ERR014      ; ELSE, REPORT ERROR
    020246 104454                TRAP    C$ERSF
    020250 000016                .WORD  14
    020252 000000                .WORD  0
    020254 011116                .WORD  ERR014
135 020256                26$:
136 020260 000244                POP     <R3>            ; RESTORE R3 FROM STACK
137 020262 000207                CLZ                    ; CLEAR Z TO INDICATE NO ERROR
138                                RETURN
139                                ;
140                                ; ERROR RETURN
141                                ;
142                                ;
143 020264                9$:
144 020266 000264                POP     <R3>            ; RESTORE R3 FROM STACK
145 020270 000207                SEZ                    ; SET Z TO INDICATE ERROR OCCURRED
146                                RETURN

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 020272
17 020272 104422
18 020274
19 020276 052704 100000
20 020302 010437 002314
21 020306 013737 002160 002320
22 020314 062737 000004 002320
23 020322 012737 100000 002324
24
25
26
27
28 020330 016504 000000
29 020334 005037 002276
30 020340
   020340 012746 000340
   020344 012746 017544
   020350 012746 000004
   020354 012746 000003
   020360 104437
   020362 062706 000010
31 020366 005764 000002
32 020372 005014
33 020374
   020374 012700 000004
   020400 104436
34 020402 005737 002276
35 020405 001406
36 020410
   020410 104455
   020412 000046
   020414 000000
   020416 011532
37 020420 000261
38 020422 000424
39
40
41
42
43
44 020424 012737 004000 002330 1$:

```

```

: **
:
: CNTIST
:
: START THE INITIALIZATION PROCESS ON THE SELECTED CONTROLLER.
: STOP BEFORE WRITING THE THIRD WORD SO CONTROLLER DOES NOT
: ATTEMPT ANY BUS TRANSFERS.
:
: INPUTS:
:   R5 - ADDRESS OF CONTROLLER TABLE
:   R4 - LEN, INTI AND VECTOR FIELDS TO SEND TO CONTROLLER
:
: LOAD TABLE OF DATA TO SEND TO SA REGISTER
: --
CNTIST:
BREAK                               ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<<
TRAP      C#BRK
PUSH      R1
BIS       #SA.STP,R4                ; SET STEP BIT IN DATA WORD
MOV       R4,SND.S1                 ; LOAD SEND DATA FOR STEP 1 OF INIT
MOV       FFREE,SND.S2              ; GET MEMORY ADDRESS AND
ADD       #HC.MSG,SND.S2            ; LOAD SEND DATA FOR STEP 2 OF INIT
MOV       #SA.TST,SND.S3            ; LOAD SEND DATA FOR STEP 3 OF INIT
:
: VERIFY THE ADDRESS OF THE SA AND IP REGISTERS ARE VALID AND
: START THE INITIALIZATION BY WRITING ZEROS TO IP REGISTER
:
MOV       C.UADR(R5),R4              ; GET ADDRESS OF IP REGISTER
CLR       NXMAD                      ; CLEAR MEMORY ERROR FLAG
SETVEC   #ERRVEC,#NXMI,#PRI07       ; SETUP TIMEOUT ERROR VECTOR
MOV       #PRI07,-(SP)
MOV       #NXMI,-(SP)
MOV       #ERRVEC,-(SP)
MOV       #3,-(SP)
TRAP     C#SVEC
ADD       #10,SP
TST      2(R4)                       ; ACCESS SA REGISTER
CLR      (R4)                        ; WRITE TO IP
CLRVEC   #ERRVEC                    ; RETURN TIMEOUT ERROR VECTOR
MOV      #ERRVEC,R0
TRAP     C#CVEC
TST      NXMAD                       ; SEE IF A MEMORY ERROR OCCURRED
BEQ      1$                          ; IF NO ERROR, CONTINUE
ERRDF    38.,ERR038                  ; PRINT ERROR MESSAGE
TRAP     C#ERDF
        .WORD 38
        .WORD 0
        .WORD ERR038
SEC
BR       4$                          ; CARRY SET TO INDICATE ERROR
:
: SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
:
MOV      #SA.S1,CNTRSD               ; STORE RESPONSE MASK

```



```

45 020432 012703 002312      MOV      #INITBL,R3      ; GET INDEX TO SEND/REPOIND INIT TABLE
46
47      ;
48      ;      WAIT FOR AND CHECK RESPONSE DATA
49      ;
50
51 020436 004737 020566      2$:      CALL      CNTRSP      ; WAIT FOR STEP OR ERROR BITS
52 020442 103414              BCS      4$            ; EXIT IF ERROR
53 020444 004733              CALL     @R3)+         ; CALL RESPONSE CHECKER FOR STEP
54 020446 103412              BCS      4$            ; GET OUT IF ERROR
55 020450 006337 002330      ASL      CNTRSD        ; SHIFT TO NEXT STEP BIT
56 020454 032737 040000 002330  BIT      @SA.S4,CNTRSD  ; CHECK IF NOW AT STEP 4
57 020462 001003              BNE      3$            ; GET OUT IF SO
58 020464 012364 000002      MOV      (R3)+,2(R4)   ; WRITE DATA TO SA REGISTER
59 020470 000762              BR       2$            ; STAY IN LOOP
60
61 020472 000241              3$:      CLC              ; CLEAR CARRY FOR NO ERROR INDICATION
62 020474              4$:      POP       R1        ; RESTORE R1
63 020476 000207              RETURN              ;
64      ;
65      ;      RESPONSE CHECK FOR FIRST WORD (STEP 1) FROM SA REGISTER
66      ;      CHECK FOR PROPER CONTROLLER TYPE
67      ;
68 020500 012701 004400      RSP.S1: MOV      @SA.S1+SA.DI,R1  ; SET STEP ONE BIT
69 020504 042702 001140      BIC      @SA.QB+SA.MP+SA.SM,R2  ; CLEAR Q22 & SM BIT FOR KDA50-Q
70 020510 000416              BR       RSP.CK       ; NOW DO A RESPONSE CHECK
71      ;
72      ;      RESPONSE CHECK FOR SECOND WORD (STEP 2) FROM SA REGISTER
73      ;      CHECK FOR ECHO OF INTERRUPT ENABLE FLAG AND INTERRUPT VECTOR
74      ;
75 020512 013701 002314      RSP.S2: MOV      SND.S1,R1      ; GET WORD SENT TO SA REGISTER
76 020516 000301              SWAB     R1            ; GET HIGH 8 BITS
77 020520 042701 177400      BIC      @177400,R1     ;
78 020524 052701 010000      BIS      @SA.S2,R1     ; SET STEP 2 BIT
79 020530 000406              BR       RSP.CK       ; NOW DO A RESPONSE CHECK
80      ;
81      ;      RESPONSE CHECK FOR THIRD WORD (STEP 3) FROM SA REGISTER
82      ;      CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
83      ;
84 020532 013701 002314      RSP.S3: MOV      SND.S1,R1      ; GET WORD SENT TO SA REGISTER
85 020536 042701 177400      BIC      @177400,R1     ; JUST LOW 8 BITS
86 020542 052701 020000      BIS      @SA.S3,R1     ; SET STEP 3 BIT
87      ;
88      ;      RESPONSE CHECK, COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
89      ;
90 020546 020102              RSP.CK: CMP      R1,R2      ; COMPARE THE DATA
91 020550 001405              BEQ      1$            ; EXIT IF COMPARED CORRECTLY
93 020552              ERRDF   25,,ERR025    ; ERROR - WRONG DATA IN SA REGISTER
    020552 104455              TRAP    C$ERRDF
    020554 000031              .WORD   25
    020556 000000              .WORD   0
    020560 011326              .WORD   ERR025
97 020562 000261              SEC              ; SET CARRY TO INDICATE ERROR
98 020564 000207              1$:      RETURN              ;
99

```

```

1
2
3      : **
4      : CNTRSP
5      :
6      : WAIT FOR CONTROLLER TO RESPOND WITH DATA IN SA REGISTER.
7      : EITHER STEP BIT FROM MASK IN LOCATION CNTRSD OR ERROR BIT
8      : WILL CAUSE A TERMINATION.
9      : AN ERROR MESSAGE WILL BE PRINTED IF THE CONTROLLER DOES NOT RESPOND
10     : IN 10 SECONDS OR IF ERROR SETS.
11     :
12     : INPUTS:
13     : CNTRSD - MASK OF STEP BIT TO LOOK FOR
14     : R5 - ADDRESS OF CONTROLLER TABLE
15     : R4 - ADDRESS OF IP REGISTER
16     :
17     : OUTPUTS:
18     : ERROR MESSAGE IF TIME OUT ON RESPONSE OR ERROR BIT SETS
19     : R2 - DATA FROM SA REGISTER
20     : CARRY SET IF ERROR BIT SETS OR TIME OUT
21     : --
22     :
23     :
24     :
25     :
26     :
27     :
28     :
29     :
30     :
31     :
32     :
33     :
34     :
35     :
36     :
37     :
38     :
39     :
40     :
41     :
42     :
43     :
44     :
45     :
46     :
47     :
48     :
49     :
50     :
51     :
52     :
53     :
54     :
55     :
56     :

```

21	020566				CNTRSP: PUSH	R1		
22	020570	052737	100000	002330	BIS	#SA.ERR,CNTRSD		: SET ERROR BIT IN MASK WORD
23	020576	012700	000012		MOV	#10,,R0		: SET UP FOR 10 SECOND TIMEOUT
24	020602	010501			MOV	R5,R1		: POINT TO COUNTER IN CONTROLLER TABLE
25	020604	062701	000026		ADD	#C.TO,R1		
26	020610	004737	017614		CALL	SETTO		
27	020614				POP	R1		
28	020616	033764	002330	000002	1\$: BIT	CNTRSD,2(R4)		: LOOK AT ERROR AND STEP BIT
29	020624	001024			BNE	3\$		: BRANCH IF EITHER SET
30	020626				BREAK			: >>>>>>>BREAK BACK TO MONITOR<<<<<<<<
	020626	104422			TRAP	C#BRK		
31	020630	005737	002256		TST	KW.CSR		: SEE IF CLOCK ON SYSTEM
32	020634	001770			BEQ	1\$		
33	020636	023765	002270	000030	CMP	KW.EL+2,C.TO(R5)		: CHECK IF TIME OUT OCCURRED
34	020644	101005			BHI	2\$		
35	020646	001353			BNE	1\$		
36	020650	023765	002266	000026	CMP	KW.EL,C.TO(R5)		
37	020656	103757			BLO	1\$		
38	020660	016402	000002		2\$: MOV	2(R4),R2		: GET REGISTER CONTENTS
40	020664				ERRDF	22,,ERR022		: REPORT TIME OUT ERROR
	020664	104455			TRAP	C#ERDF		
	020666	000026			.WORD	22		
	020670	000000			.WORD	0		
	020672	011204			.WORD	ERR022		
44	020674	000407			BR	4\$		
45					:			
46					:			
47					:			
48	020676	016402	000002		3\$: MOV	2(R4),R2		: GET REGISTER CONTENTS
49	020702	100006			BPL	5\$		: EXIT IF ERROR NOT SET
51	020704				ERRDF	21,,ERR021		: REPORT ERROR INFO
	020704	104455			TRAP	C#ERDF		
	020706	000025			.WORD	21		
	020710	000000			.WORD	0		
	020712	011134			.WORD	ERR021		
55	020714	000261			4\$: SEC			
56	020716	000207			RETURN			

```
57      ;  
58      ;      NORMAL EXIT  
59      ;  
60 020720 000241      S# :      CLC      ; CLEAR CARRY AS NO ERROR INDICATION  
61 020722 000207      ;      RETURN      ;
```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 131  
 PRE-PROGRAMMED SUBROUTINES

```

1
2
3      ;**
4      ;   CLOG - COMPUTE SELECTED POWER OF 2.
5      ;
6      ;   INPUTS:
7      ;           R0 - SELECTED POWER OF 2
8      ;   OUTPUTS:
9      ;           R1 - VALUE OF 2 RAISED TO POWER OF INPUT NUMBER
10     CLOG:  PUSH   R0           ; SAVE INPUT VALUE
11           CLR    R1           ; SET UP ZERO START VALUE
12           SEC    ;           ; CARRY READY TO SHIFT IN
13           1$:  ROL    R1           ; SHIFT 1 TO LEFT
14           DEC    R0           ;   UNTIL R0
15           BPL   1$           ;   GOES NEGATIVE
16           POP   R0           ; RESTORE INPUT VALUE
17           RETURN              ;

```

```

1
2
3
4
5
6
7
8
9
10
11
12 020744 005037 002276
13 020750
14 020754
    020754 012746 000340
    020760 012746 017544
    020764 012746 000004
    020770 012746 000003
    020774 104437
    020776 062706 000010
15 021002 012703 000004
16 021006 012704 002246
17 021012 005714
18 021014 001403
19 021016 005034
20 021020 005303
21 021022 001373
22 021024 005737 002256
23 021030 001403
24 021032 012777 000105 161216
25 021040
26 021044 000207
27

;RESET
;
; RESET ALL CONTROLLERS IN THE CONTROLLER TABLES
;
; INPUTS:
; IPADRS - CONTAINS ALL IP ADDRESSES
; OUTPUTS:
; NONE
;
;
RESET: CLR NXMAD ; CLEAR NON-EXISTANT MEMORY ADDRESS
        PUSH <R3,R4> ; SAVE R3 AND R4 ON STACK
        SETVEC @ERRVEC,@NXMI,@PRI07 ; SETUP TIMEOUT ERROR VECTOR
        MOV @PRI07,-(SP)
        MOV @NXMI,-(SP)
        MOV @ERRVEC,-(SP)
        MOV @3,-(SP)
        TRAP C+SVEC
        ADD @10,SP
        MOV @4,R3 ; STORE MAXIMUM # OF CONTROLLERS IN R3
        MOV @IPADRS,R4 ; STORE IP ADDRESS IN R4
16:     TST (R4) ; IS THERE AN ENTRY?
        BEQ 28 ; IF NOT, DONE
        CLR @R4) ; INIT CONTROLLER
        DEC R3 ; MAKE SURE WE DO NOT EXTEND OVER AREA
        BNE 18 ; IF NOT DONE, BRANCH
28:     TST KW.CSR ; SEE IF CLOCK PRESENT,
        BEQ 38 ; BRANCH IF NOT, ELSE
38:     MOV @KW.OUT,@KW.CSR ; START THE CLOCK.
        POP <R4,R3> ; RESTORE R3,R4 FROM STACK
        RETURN
;

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 021046 005737 002256
15 021052 001465
16 021054
17 021064 013703 002266
18 021070 013704 002270
19 021074 013700 002264
20 021100 004737 016014
21 021104 012700 000074
22 021110 004737 016014
23 021114
24 021116 004737 016014
25 021122
26 021134 020527 000011
27 021140 003004
28 021142
29 021152
30 021164
31 021166 020527 000011
32 021172 003004
33 021174
34 021204
35 021216
36 021226
37 021236 000207

;RNTIME
;
;PRINT RUNTIME
;
;INPUTS:
; KW.EL - CONTAINS ELAPSED TIME
; KW.HZ - HERTZ OF CLOCK
;OUTPUTS:
; IF CLOCK ON SYSTEM:
; " RUN TIME HH:MM:SS " PRINTED
; IF NO CLOCK: ONE SPACE IS PRINTED

RNTIME: TST KW.CSR ;CHECK IF A CLOCK PRESENT
        BEQ RNTIMX ;BRANCH IF NOT
        PUSH <R0,R3,R4,R5>
        MOV KW.EL,R3 ;GET ELAPSED TIME
        MOV KW.HZ,R4 ;GET SPEED OF CLOCK
        CALL DIVIDE ;COMPUTE SECONDS OF ELAPSED TIME
        MOV #60.,R0 ;NOW DIVIDE BY 60
        CALL DIVIDE ; TO COMPUTE MINUTES
        PUSH R5 ;SAVE REMAINDER AS SECONDS
        CALL DIVIDE ;DIVIDE BY 60 AGAIN
        PNT RNTIM,R3 ;PRINT HOURS
        CMP R5,#9. ;IF MINUTES 9 OR LESS
        BGT 1$ ;PRINT A LEADING ZERO
        PRINT #'0 ;NOW PRINT MINUTES
        PNT RNTIM1,R5 ;GET SECONDS
        POP R5 ;IF 9 OR LESS
        CMP R5,#9. ;PRINT A LEADING ZERO
        BGT 2$ ;NOW PRINT SECONDS
        PRINT #'0 ;HOURS IN R3
        PNT RNTIM2,R5 ;PRINT A SPACE
        POP <R5,R4,R3,R0>
RNTIMX: PRINT '<'
        RETURN

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 134  
 PRE-PROGRAMMED SUBROUTINES

```

1
2
3      ;:
4      ;      WCHNG
5      ;
6      ;      WAIT UNTIL LOOPED DATA APPEARS IN SA REGISTER
7      ;--
8 021240 012700 000012      WCHNG:  MOV      #10.,R0      ; SET TIMEOUT FOR 10 SECONDS
9 021244 010501              MOV      R5,R1      ; POINT TO CONTROLLER TABLE
10 021246 062701 000026      ADD      #C.TO,R1      ;
11 021252 004737 017614      CALL     SETTO      ;
12 021256 026437 000002 002332 1#:  CMP      2(R4),WCHNGD  ; IS LOOPED DATA IN SA REGISTER?
13 021264 001422              BEQ      2#          ;
14 021266              BREAK     ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<<
    021266 104422              TRAP     C#BRK      ;
15 021270 005737 002256      TST      KW.CSR      ; SEE IF CLOCK ON SYSTEM
16 021274 001770              BEQ      1#          ;
17 021276 023765 002270 000030  CMP      KW.EL+2,C.TOH(R5) ; CHECK IF TIME OUT OCCURRED
18 021304 101005              BHI      3#          ;
19 021306 001363              BNE      1#          ;
20 021310 023765 002266 000026  CMP      KW.EL,C.TO(R5) ;
21 021316 103757              BLO      1#          ;
22 021320              3#:  ERDF     27.,ERR027  ; REPORT ERROR
    021320 104455              TRAP     C#ERDF
    021322 C00033              .WORD   27
    021324 000000              .WORD   0
    021326 011364              .WORD   ERR027
23 021330 000244              CLZ      ; FLAG AS ERROR
24 021332 000207              2#:  RETURN      ; RETURN TO CALLING PROGRAM
26

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 135  
 REPORT CODING SECTION

```

1      .SBTTL  REPORT CODING SECTION
2
3      ;**
4      ;      THE REPORT CODING SECTION CONTAINS THE CODE FOR PRINTING
5      ;      STATISTICAL INFORMATION GATHERED BY THE DIAGNOSTIC.  IT IS
6      ;      EXECUTED BY THE OPERATOR COMMAND "PRINT" OR BY THE MACRO CALL
7      ;      "DORPT".
8      ;--
9
10     021334      BGNRPT
11     021334      L#RPT::
12     021334      PUSH    <R0,R1,R2,R3,R4,R5>
13     021350      PNTS    RPTMSG,TNUM      ; PRINT TEST NUMBER
14     021364      004737  021046      CALL    RNTIME      ; GET RUNTIME PARAMETERS
15     021370      PRINT   #CR              ; END THE LINE
16     021400      012701  002272      MOV     #STIME,R1    ; GET REPORT TIMER
17     021404      012700  001604      MOV     #15.*60.,R0 ; GET REPORT INTERVAL
18     021410      004737  017614      CALL    SETTO       ; SET TIME FOR NEXT REPORT
19
91     021414      RPTXX: POP     <R5,R4,R3,R2,R1,R0>
92
93     021430      EXIT    RPT
94     021430      000167      .WORD   J#JMP
95     021432      000034      .WORD   L10037-2-.
96
97     021434      116      042      124  RPTMSG: .ASCIZ  \N"TEST "D3" IN PROGRESS.  "\
98     .EVEN
99
100    021470      ENDRPT
101    021470      L10037:
102    021470      104425      TRAP   C#RPT

```



```

1
2      .SBTTL  PROTECTION TABLE
3
4      ;**
5      ;      THIS TABLE IS USED BY THE RUNTIME SERVICES
6      ;      TO PROTECT THE LOAD MEDIA.
7      ;--
8
9 021472      BGNPROT
021472      L#PROT::
10
11 021472 177777      -1      ; P-TABLE OFFSET FOR CSR ADDRESS
12 021474 177777      -1      ; P-TABLE OFFSET FOR MASSBUS ADDRESS
13 021476 177777      -1      ; P-TABLE OFFSET FOR DRIVE NUMBER
14
15 021500      ENDPROT
16

```

INITIALIZE 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  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

.SBTTL INITIALIZE SECTION

```

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

```

44 021500
   021500
45
46 021500
   021500 012700 000040
   021504 104447
47
48 021506
   021506 103004
49 021510 012737 000010 002202
50 021516 000525
51 021520
52 021520
   021520 012700 000037
    
```

```

BGNINIT
L$INIT::
;HERE FROM START COMMAND?
READEF #EF.STA
MOV #EF.STA,R0
TRAP C$REFG
;BRANCH TO 1$ IF NOT, ELSE
BNCOMPLETE 1$
BCC 1$
MOV #ISTRT,IFLAGS ;SET START BIT IN FLAG.
BR INIT1
;HERE FROM RESTART? COMMAND?
1$:
READEF #EF.RES
MOV #EF.RES,R0
    
```

## INITIALIZE SECTION

```

53 021524 104447          TRAP    C$REFG                ;BRANCH TO 2$ IF NOT, ELSE
54 021526                BNCOMPLETE    2$
55 021530 052737 000004 002202  BCC      2$
56 021536 000515                BIS      @IREST,IFLAGS    ;SET RESTART BIT IN FLAG.
57 021540                BR        INIT1
58 021540                2$:    READDEF  @EF.CON
59 021540 012700 000036      MOV      @EF.CON,R0
60 021544 104447          TRAP    C$REFG                ;BRANCH TO 3$ IF NOT, ELSE
61 021546                BNCOMPLETE    3$
62 021550 042737 000020 002202  BCC      3$
63 021556 052737 000002 002202  BIC      @ISTRH,IFLAGS    ;CLEAR 1ST TIME THRU CZUDIO FLAG AND
64 021564 000472                BIS      @ICONT,IFLAGS    ;SET CONTINUE BIT IN FLAG.
65 021566                3$:    READDEF  @EF.PWR
66 021566 012700 000034      MOV      @EF.PWR,R0
67 021572 104447          TRAP    C$REFG                ;BRANCH TO 4$ IF NOT, ELSE
68 021574                BNCOMPLETE    4$
69 021574 103001          BCC      4$
70 021576 000465          BR        13$
71                ;
72                ;    MAKE ALL CONTROLLER/DRIVE TABLES NOT AVAILABLE FOR TESTING
73                ;
74 021600 013705 002172 4$:    MOV      CTABS,R5                ;GET ADDRESS OF 1ST CONTROLLER TABLE
75 021604 052765 100000 000002 5$:    BIS      @CT.AVL,C.UNIT(R5)    ;SET CONTROLLER TABLE NOT AVAILABLE
76 021612 010502                MOV      R5,R2                ;GET POINTER TO DRIVE TABLES
77 021614 062702 000016      ADD      @C.DR0,R2
78 021620 012703 000004      MOV      @4.,R3                ; GET NUMBER OF DRIVES PER CONTROLLER
79 021624 012200 6$:    MOV      (R2)+,R0                ;SEE IF THIS DRIVES HAS A TABLE,
80 021626 001403                BEQ      7$                    ;BRANCH IF NOT, ELSE
81 021630 052760 100000 000002 7$:    BIS      @DT.AVL,D.UNIT(R0)    ;SET DRIVE TABLE NOT AVAILABLE.
82 021636 005303                DEC      R3                    ;LOOK AT NEXT DRIVE IN CONTROLLER TABLE.
83 021640 001371                BNE      6$                    ;BRANCH IF NO DRIVES, ELSE
84 021642 062705 000034      ADD      @C.SIZE,R5            ;MOVE TO NEXT CONTROLLER TABLE
85 021646 005715                TST     (R5)                    ;IS THERE A NEXT ONE?
86 021650 001355                BNE      5$                    ;IF SO, CLEAR THE BITS THERE
87                ;
88                ;    NOW GET EACH P-TABLE AND MAKE THE APPROPRIATE CONTROLLER/DRIVE
89                ;    TABLES AVAILABLE FOR TESTING.
90                ;
91 021652 005003                CLR      R3                    ;START WITH LOGICAL UNIT 0
92 021654                8$:    GPHARD  R3,R0                ;GET POINTER TO IT'S P-TABLE
93 021654                MOV      R3,R0
94 021656 010300 104442          TRAP    C$GPHRD
95 021660                ;BRANCH TO 12$ IF NOT AVAILABLE
96 021660 103030                BNCOMPLETE    12$
97 021662 013705 002172 9$:    MOV      CTABS,R5                ;GET ADDRESS OF 1ST CONTROLLER TABLE
98 021666 021015                CMP      (R0),(R5)            ;SEE IF CSR ADDRESSES ARE THE SAME.
99 021670 001411                BEQ      11$                    ;BRANCH IF SO, ELSE

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 137-2  
INITIALIZE SECTION

```

99 021672 062705 000034      ADD      #C.SIZE,R5      ;LOOK AT NEXT CONTROLLER TABLE,
100 021676 005715           TST      (R5)           ;SEE IF THERE IS ANOTHER CONTROLLER TABLE,
101 021700 001372           BNE      9$            ;BRANCH IF SO, ELSE
102 021702           10$:           ;REPORT TABLE CONSISTANCY ERROR.
103 021702           ERRSF   6,,ERR006
    021702 104454          TRAP   C$ERSF
    021704 000006          .WORD  6
    021706 000000          .WORD  0
    021710 011066          .WORD  ERR006
104                               ;DO CLEAN-UP TRAP
105 021712           DOCLN
    021712 104444          TRAP   C$DCLN
106
107 021714 016001 000006      11$:      MOV      H.DRV(R0),R1      ;GET DRIVE NUMBER FROM P-TABLE
108 021720 004737 015404      CALL   GTDRV            ;FIND THE DRIVE TABLE ADDRESS
109 021724 001366           BNE      10$           ;BRANCH IF NOT FOUND, ELSE
110 021726 042765 100000 000002  BIC     #CT.AVL,C.UNIT(R5) ;CLEAR AVAILABLE BIT IN CONTROLLER AND
111 021734 042764 100000 000002  BIC     #DT.AVL,D.UNIT(R4) ;THE DRIVE TABLES.
112 021742 005203           12$:      INC     R3              ;INCREMENT TO NEXT UNIT IN P-TABLE
113 021744 020337 002012      CMP     R3,L$UNIT       ;SEE IF ALL P-TABLES CHECKED,
114 021750 002741           BLT     8$            ;BRANCH IF NOT, ELSE
115 021752 012701 002272      13$:      MOV     #STIME,R1        ; GET REPORT TIMER
116 021756 012700 001604      MOV     #15.*60.,R0     ; GET REPORT INTERVAL
117 021762 004737 017614      CALL   SETTO           ; SET TIME FOR NEXT REPORT
118 021766 000137 022574      JMP     INITXX         ;EXIT THE INITIALIZE SECTION.
119
120
121
122
123
124
125 021772 005037 002266      INIT1:  CLR     KW.EL          ;CLEAR ELAPSED TIME
126 021776 005037 002270      CLR     KW.EL+2
127 022002           CLOCK   L,R0          ;SEE IF L-CLOCK PRESENT
    022002 012700 000114      MOV     #'L,R0
    022006 104462          TRAP   C$CLCK
128 022010           BCOMPLETE 1$
    022010 103413          BCS    1$
129 022012           CLOCK   P,R0          ;SEE IF P-CLOCK PRESENT
    022012 012700 000120      MOV     #'P,R0
    022016 104462          TRAP   C$CLCK
130 022020           BCOMPLETE 1$
    022020 103407          BCS    1$
131 022022 005037 002256      CLR     KW.CSR         ;IF NEITHER, CLEAR CSR STORAGE WORD
132 022026           PNTF   NOCLOCK
133 022036 000434           BR     2$
134
135 022040 012037 002256      1$:      MOV     (R0)+,KW.CSR     ;STORE DATA RETURNED
136 022044 012037 002260      MOV     (R0)+,KW.BRL
137 022050 012037 002262      MOV     (R0)+,KW.VEC
138 022054 012037 002264      MOV     (R0)+,KW.HZ
139
140 022060           SETVEC  KW.VEC,#KW11I,#PRI07 ;SETUP KW11 VECTOR ADDRESS
    022060 012746 000340      MOV     #PRI07,-(SP)
    022064 012746 017572      MOV     #KW11I,-(SP)
    022070 013746 002262      MOV     KW.VEC,-(SP)
    022074 012746 000003      MOV     #3,-(SP)

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 137-3  
INITIALIZE SECTION

```

022100 104437          TRAP  C$SVEC
022102 062706 000010  ADD   #10,SP
141 022106 012777 000105 160142  MOV   #KW.OUT,#KW.CSR      ;START THE CLOCK
142 022114 012701 002272  MOV   #STIME,R1           ; GET REPORT TIMER
143 022120 012700 001604  MOV   #15.*60.,R0        ; GET REPORT INTERVAL
144 022124 004737 017614  CALL  SETTO              ; SET TIME FOR NEXT REPORT
145 022130 004737 020744 2$:  CALL  RESET            ;RESET ALL CONTROLLERS
146 022134          MEMORY FFREE          ;RESET START OF FREE MEMORY
    022134 104431  TRAP  C$MEM
    022136 010037 002160  MOV   R0,FFREE
147 022142 017737 160012 002162  MOV   #FFREE,FSIZE      ;RESET SIZE OF FREE MEMORY
148 022150 005037 002204  CLR   TNUM              ;INITIALIZE TEST NUMBER TO NO TEST RUNNING
149
150          ;
151          ;   ALLOCATE DRIVE TABLES TO MEMORY
152          ;
153
154 022154 013737 002160 002170  INIT2:  MOV   FFREE,DTABS      ;STORE START OF DRIVE TABLES AND
155 022162 005077 160002  CLR   @DTABS           ;MARK ZERO END.
156 022166 013700 002012  MOV   L$UNIT,R0        ;GET NUMBER OF LOGICAL UNITS TO RUN,
157 022172 012701 000001  MOV   #1,R1            ;GET INITIAL SIZE OF DRIVE TABLE AND
158 022176 062701 000002 1$:  ADD   #<D.SIZE>/2,R1  ;ACCUMULATE DRIVE TABLE SIZE.
159 022202 005300  DEC   R0              ;SEE IF ANY MORE LOGICAL UNITS.
160 022204 001374  BNE   1$              ;BRANCH IF NOT, ELSE
161 022206 004737 012132  CALL  ALOCH           ;ALLOCATE ALL DRIVE TABLES TO MEMORY.
162          ;   ; R1 POINTS TO 1ST WORD IN DRIVE TABLE
163
164          ;
165          ;   INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
166          ;
167
168 022212 013737 002160 002172  INIT3:  MOV   FFREE,CTABS      ; STORE START OF CONTROLLER TABLES AND
169 022220 005077 157746  CLR   @CTABS          ; MARK ZEROS END.
170 022224 005037 002174  CLR   CTRLRS         ; CLEAR CONTROLLER COUNT
171 022230 012701 002246  MOV   #IPADRS,R1      ; R1 -> IP ADDRESS
172 022234 012702 000004  MOV   #4.,R2         ; GET MAXIMUM # OF CONTROLLERS
173 022240 005021 1$:  CLR   (R1)+          ; CLEAR ENTRY
174 022242 005302  DEC   R2              ; DONE?
175 022244 001375  BNE   1$              ; IF NOT, BRANCH
176
177          ;
178          ;   BUILD CONTROLLER TABLES
179          ;
180
181 022246 005005  INIT4:  CLR   R5              ;CLEAR CUSTOMER DATA FLAG
182 022250 005002  CLR   R2              ;START WITH LOGICAL UNIT 0
184 022252 1$:  GPHARD R2,R0      ;GET POINTER TO IT'S P-TABLE
    022252 010200  MOV   R2,R0
    022254 104442  TRAP  C$GPHRD
185 022256 185:  BNCOMPLETE 16$      ;BRANCH TO 16$ IF NOT AVAILABLE
    022256 103130  BCC   16$
186 022260 013703 002172  MOV   CTABS,R3        ;GET ADDRESS OF 1ST CONTROLLER TABLE
187 022264 005713 2$:  TST   (R3)         ;CHECK IF ANY MORE TABLES
188 022266 001431  BEQ   6$              ;BUILD NEW TABLE IF FOUND ZERO WORD
189 022270 021013  CMP   (R0),(R3)      ;CHECK IF SAME CSR ADDRESS.
190 022272  ASSUME C.UADR EQ 0
191 022272  ASSUME H.UBA EQ 0

```

CZUDHAO UDA50A/KDA50-G BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 137-4  
INITIALIZE SECTION

```

192 022272 001013          BNE      4$          ;BRANCH IF NOT, ELSE
193                               ;CHECK THAT OTHER PARAMETERS MATCH.
194 022274 016004 000004    MOV      H.BRL(R0),R4    ;GET BR LEVEL FROM P-TABLE
195 022300 000304          SWAB     R4              ;SWAP TO HIGH BYTE
196 022302 006104          ROL      R4              ;SHIFT ONE MORE TO LEFT
197 022304 056004 000002    BIS      H.VEC(R0),R4    ;ADD VECTOR ADDRESS
198 022310 020463 000004    CMP      R4,C.VEC(R3)    ;COMPARE VECTOR AND BR LEVELS,
199 022314 001457          BEQ      11$          ;BRANCH IF SAME, ELSE
200 022316 000137 022606    3$:     JMP      CTABER      ;FOUND SAME CONTROLLER WITH DIFFERENT
201                               ; BR LEVEL OR VECTOR ADDR
202 022322 016304 000004    4$:     MOV      C.VEC(R3),R4    ;GET VECTOR FROM CONTROLLER TABLE
203 022326 042704 177000    BIC      @C<CT.VEC>,R4    ;AND
204 022332 026004 000002    CMP      H.VEC(R0),R4    ;COMPARE VECTOR ADDRESSES,
205 022336 001002          BNE      5$          ;BRANCH IF DIFFERENT, ELSE
206 022340 000137 022656    JMP      SAMVEC          ;FOUND TWO CONTROLLERS WITH SAME VECTOR
219
220 022344 062703 000034    5$:     ADD      @C.SIZE,R3      ;POINT TO BEGINNING OF NEXT CONTROLLER
221 022350 000745          BR      2$          ;TABLE IN MEMORY.
222
223          ;
224          ;     BUILD NEW CONTROLLER TABLE
225          ;
226
227 022352 012704 002246    6$:     MOV      @IPADRS,R4      ;GET BEGINNING OF IP ADDRESS TABLE
228 022356 020427 002256    7$:     CMP      R4,@IPADRS+8.    ;SEE IF END OF IP ADDRESS TABLE,
229 022362 101004          BHI      9$          ;BRANCH IF SO, ELSE
230 022364 005724          TST      (R4)+          ;DID WE FIND AN OPEN ENTRY ?
231 022366 001401          BEQ      8$          ;BRANCH IF SO, ELSE
232 022370 000772          BR      7$          ;LOOK AGAIN.
233
234 022372 011044          8$:     MOV      (R0),-(R4)      ;TAKE CSR ADDRESS FROM P-TABLE
235                               ;AND STORE IT IN THE IP ADDRESS TABLE.
236 022374 012701 000016    9$:     MOV      @<C.SIZE>/2,R1    ;GET # OF ENTRIES IN CONTROLLER TABLE
237 022400 004737 012132    CALL     ALOCH          ;AND ALLOCATE A TABLE TO MEMORY.
238                               ; R0 => 1ST WORD P-TABLE
239                               ; R1 => 1ST WORD IN CONTROLLER TABLE
240 022404 011021          MOV      (R0),(R1)+      ; STORE CSR ADDRESS AND
241 022406 010221          MOV      R2,(R1)+      ; UNIT NUMBER IN THE CONTROLLER TABLE.
242 022410 016004 000004    MOV      H.BRL(R0),R4    ;GET THE BR LEVEL.
243 022414 000304          SWAB     R4              ;SWAP TO HIGH BYTE.
244 022416 006104          ROL      R4              ;SHIFT ONE MORE TO LEFT.
245 022420 056004 000002    BIS      H.VEC(R0),R4    ;ADD VECTOR ADDRESS AND
246 022424 010421          MOV      R4,(R1)+      ;STORE IT IN THE CONTROLLER TABLE.
247 022426 012721 004037    MOV      @4037,(R1)+     ;THE 'JSR R0' INSTRUCTION AND
248 022432 012721 017554    MOV      @CNTSRV,(R1)+   ;THE ADDRESS OF THE INTERRUPT SERVICE
249                               ;ROUTINE IN THE CONTROLLER TABLE.
250 022436 012704 000011    10$:    MOV      @<C.SIZE-C.FLG>/2,R4    ;GET # OF ENTRIES TO END OF TABLE.
251 022442 005021          CLR      (R1)+          ;CLEAR REST OF TABLE AND
252 022444 005304          DEC      R4              ;ADD ZERO WORD AT END.
253 022446 002375          BGE      10$          ;LOOP TIL ALL CLEARED
254 022450 005237 002174    INC      CTRLRS          ;KEEP TRACK OF CONTROLLER COUNT
255
256
257          ;
258          ;     BUILD DRIVE TABLES
259          ;
260
261
262
263
264
265
266 022454 013701 002170    11$:    MOV      DTABS,R1          ;GET ADDRESS OF CURRENT DRIVE TABLE

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 137-5  
INITIALIZE SECTION

```

267 022460 062703 000016          ADD    #C.DRO,R3          ; INDEX TO 1ST DRIVE IN TABLE
268 022464 012704 000004          MOV    #4.,R4            ; GET # OF DRIVES PER CONTROLLER
269 022470 005713          12$:  TST    (R3)          ; ANY ENTRY TO DRIVE TABLE,
270 022472 001411          BEQ    14$              ; BRANCH IF NOT, ELSE
271 022474 026033 000006          CMP    H.DRV(R0),@R3)+  ; COMPARE DRIVE NUMBER IN DRIVE TABLE,
272 022500 001002          BNE    13$              ; BRANCH IF DIFFERENT, ELSE
273 022502 000137 022622          JMP    MLDRER           ; FOUND TWO P-TABLES WITH SAME DRIVE.
274
275 022506 005304          13$:  DEC    R4            ; COUNT DRIVES
276 022510 001367          BNE    12$              ; IF FOUR DRIVE TABLES ALREADY EXIST,
277 022512 000137 022640          JMP    TOOMER           ; THEN REPORT ERROR
278
279 022516 010113          14$:  MOV    R1,(R3)      ; STORE ADDRESS OF DRIVE TABLE IN
280
281 022520 016021 000006          MOV    H.DRV(R0),(R1)+  ; STORE DRIVE NUMBER AND
282 022524 010221          MOV    R2,(R1)+        ; LOGICAL UNIT NUMBER IN DRIVE TABLE.
295
296 022526 062737 000004 002170  ADD    #D.SIZE,DTABS    ; NEXT DRIVE TABLE ADDRESS AND
297 022534 005077 157430          CLR    @DTABS           ; MARK ZERO END.
298 022540 005202          16$:  INC    R2            ; INCREMENT LOGICAL UNIT NUMBER
299 022542 020237 002012          CMP    R2,L#UNIT       ; CHECK IF GOT ALL TABLES
300 022546 002641          BLT    1$              ; IF NOT, GO BACK FOR NEXT, ELSE
301 022550 012701 000001          MOV    #1,R1           ; GET 1 WORD TO TERMINATE ALL CONTROLLER
302 022554 004737 012132          CALL  ALOCM            ; TABLES AND ALLOCATE IT TO MEMORY.
303
340
341          ;
342          ;
343          ;
344 022560 013737 002160 002164  INIT6: MOV    FFREE,FMEM      ; SAVE START ADDRESS
345 022566 013737 002162 002166  MOV    FSIZE,FMEMS     ; SAVE SIZE
346
347          ;
348          ;
349          ;
350
351 022574          INITXX: SETPRI #PRI00          ; SET RUNNING PRIORITY TO ZERO
352 022574 012700 000000          MOV    #PRI00,R0
353 022600 104441          TRAP  C#SPRI
352
353 022602          EXIT  INIT
353 022602 104432          TRAP  C#EXIT
353 022604 000066          .WORD L10041-..

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 138  
INITIALIZE ERRORS

```

1      .SBTTL  INITIALIZE ERRORS
3
4      ;
5      ;      DIFFERENT VECTORS OR BR LEVELS FOR ONE CONTROLLER
6      ;
7
8 022606 010305      CTABER:  MOV    R3,R5                ;GET CONTROLLER ADDRESS
9 022610                ERRSF 1,,ERR001
   022610 104454      TRAP   C#ERSF
   022612 000301      .WORD  1
   022614 000000      .WORD  0
   022616 011002      .WORD  ERR001
10                                ;DO CLEAN-UP TRAP
11 022620                DOCLN
   022620 104444      TRAP   C#DCLN
13
14      ;
15      ;      TWO P-TABLES FOR SAME DRIVE
16      ;
17
18 022622 013705 002216      MLDRER:  MOV    TEMP,R5                ;GET CONTROLLER ADDRESS
19 022626                ERRSF 2,,ERR002
   022626 104454      TRAP   C#ERSF
   022630 000002      .WORD  2
   022632 000000      .WORD  0
   022634 011020      .WORD  ERR002
20                                ;DO CLEAN-UP TRAP
21 022636                DOCLN
   022636 104444      TRAP   C#DCLN
22
23      ;
24      ;      MORE THAN FOUR DRIVES SELECTED ON ONE CONTROLLER
25      ;
26
27 022640 013705 002216      TOOMER:  MOV    TEMP,R5                ;GET CONTROLLER ADDRESS
28 022644                ERRSF 3,,ERR003
   022644 104454      TRAP   C#ERSF
   022646 000003      .WORD  3
   022650 000000      .WORD  0
   022652 011036      .WORD  ERR003
29                                ;DO CLEAN-UP TRAP
30 022654                DOCLN
   022654 104444      TRAP   C#DCLN
32
33      ;
34      ;      TWO CONTROLLERS USE THE SAME VECTOR
35      ;
36 022656 010305      SAMVEC:  MOV    R3,R5                ;GET CONTROLLER ADDRESS
37 022660                ERRSF 8,,ERR008
   022660 104454      TRAP   C#ERSF
   022662 000010      .WORD  8
   022664 000000      .WORD  0
   022666 011100      .WORD  ERR008
38                                ;DO CLEAN-UP TRAP
39 022670                DOCLN
   022670 104444      TRAP   C#DCLN
41

```



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 138-1  
INITIALIZE ERRORS

42 022672  
022672  
022672 104411  
43

ENDINIT  
L10041:  
TRAP C\$INIT

CZUDHAO LDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 139  
AUTODROP SECTION

```
1      .SBTTL  AUTODROP SECTION
2
3      ;**
4      ;      THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
5      ;      THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
6      ;      SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
7      ;      DROPPED FROM TESTING.
8      ;--
9
10     022674      BGNAUTO
11     022674      L$AUTO::
12     022674      ENDAUTO
13     022674      L10042:
14     022674      TRAP      C$AUTO
15     104461
```

```

1      .SBTTL  CLEANUP CODING SECTION
2
3      ;**
4      ;      THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
5      ;      AFTER EACH PASS AND AFTER THE PROGRAM IS INTERRUPTED BY "tC".
6      ;--
7
8 022676      BGNCLN
022676      L$CLEAN::
9
11 022676 004737 020744      CALL  RESET          ;RESET ALL CONTROLLERS
13
14 022702      EXIT  CLN
022702 104432      TRAP  C$EXIT
022704 000002      .WORD L10043-.
15
16 022706      ENDCLN
022706      L10043:
022706 104412      TRAP  C$CLEAN
17

```

```

1
2
3
4
5
6
7
8 022710
   022710
9
10 022710
    022710 000167
    022712 000000
11
12 022714
    022714
    022714 104453
13

```

```

.SBTTL DROP UNIT SECTION

: **
: THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
: TO NO LONGER BE TESTED.
: --

BGNDU
L#DU::

EXIT DU
.WORD J#JMP
.WORD L10044-2-.

ENDDU
L10044:
TRAP C#DU

```

CZUDMAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 142  
 ADD UNIT SECTION

1		.SBTTL	ADD	UNIT SECTION
2				
3		;	**	
4		;		THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
5		;		TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
6		;		TO THE TEST CYCLE.
7		;	--	
8				
9	022716			BGNAU
	022716	L\$AU::		
10				
11				
12	022716		EXIT	AU
	022716		.WORD	J\$JMP
	022720		.WORD	L10045-2-.
13				
14	022722		ENDAU	
	022722	L10045:		
	022722		TRAP	C\$AU
15				

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 143  
 HARDWARE TESTS

```

1      .SBTTL HARDWARE TESTS
2
3
4      .SBTTL TEST 1: BUS ADDRESSING TEST
5
6      ;**
7      ; TEST 1 - BUS ADDRESSING TEST
8      ;--
9
10     022724      BGNTST
11     022724      T1::
12     022724 012701 000001      MOV      #1,R1      ; INITIALIZE TEST PARAMETERS
13     022730 004737 013040      CALL     TINIT      ;
14     022734 013737 002172 002176      MOV      CTABS,TSTTAB      ; GET ADDRESS OF 1ST CONTROLLER TABLE
15     022742 013705 002176      T1NEXT: MOV      TSTTAB,R5      ; GET CONTROLLER TABLE ADDRESS
16     022746 116537 000002 002074      MOVB    C.UNIT(R5),L#LUN      ; CHECK IF UNIT AVAILABLE FOR TESTING
17     022754 005765 000002      TST     C.UNIT(R5)      ;
18     022760 100010      BPL     T1NOW      ; TEST IF AVAILABLE
19     022762      ASSUME CT.AVL EQ BIT15
20     022762 062737 000034 002176      T1SKIP: ADD     #C.SIZE,TSTTAB      ; GET NEXT CONTROLLER TABLE ADDRESS
21     022770 005777 157202      TST     @TSTTAB      ; CHECK IF ANOTHER CONTROLLER TABLE
22     022774 001362      BNE     T1NEXT      ; DO NEXT CONTROLLER
23     022776      EXIT TST
24     022776 104432      TRAP   C#EXIT
25     023000 001522      .WORD  L10046-.
26
27
28     023002 004737 020744      T1NOW: CALL    RESET      ; RESET ALL CONTROLLERS
29

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 144  
 TEST 1: BUS ADDRESSING TEST

```

1
2
3
4
5
6 023006
   023006
   023006 104402
7 023010 005037 002276
8 023014
   023014 012746 000340
   023020 012746 017544
   023024 012746 000004
   023030 012746 000003
   023034 104437
   023036 062706 000010
9 023042 011504
10 023044 005714
11 023046 005764 000002
12 023052
   023052 012700 000004
   023056 104436
13 023060 005737 002276
14 023064 001406
15 023066
   023066 104455
   023070 000046
   023072 000000
   023074 011532
16 023076
   023076 104406
17 023100 000730
18 023102
19 023102
   023102
   023102 104403
20

;
; TEST 1, SUBTEST 1
; HOST MEMORY I/O PAGE ADDRESSING TEST
;
; BGNSUB; 1
; T1.1:
; TRAP C#BSUB
; CLR NXMAD ; CLEAR MEMORY ERROR FLAG
; SETVEC #ERRVEC,#NXMI,#PRI07 ; SETUP TIMEOUT ERROR VECTOR
; MOV #PRI07,-(SP)
; MOV #NXMI,-(SP)
; MOV #ERRVEC,-(SP)
; MOV #3,-(SP)
; TRAP C#SVEC
; ADD #10,SP
; MOV (R5),R4 ; GET ADDRESS OF IP REGISTER
; TST (R4) ; READ IP REGISTER
; TST 2(R4) ; READ SA REGISTER
; CLRVEC #ERRVEC ; RETURN TIMEOUT ERROR VECTOR
; MOV #ERRVEC,R0
; TRAP C#CVEC
; TST NXMAD ; CHECK FLAG
; BEQ 1$ ; IF OK, GO TO NEXT SUBTEST
; ERDF 38,,ERR038 ; ERROR - MEMORY ADDRESS ERROR
; TRAP C#ERDF
; .WORD 38
; .WORD 0
; .WORD ERR038
; CKLOOP ; LOOP ON ERROR IF SELECTED
; TRAP C#CLP1
; BR T1$SKIP ; END TEST NOW

1$:
ENDSUB
L10047:
; TRAP C#ESUB

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 145  
 TEST 1: BUS ADDRESSING TEST

```

1
2
3
4
5
6
7
8 023104          ;
   023104          ;   TEST1, SUBTEST 2
   023104 104402   ;   MAKE SURE CONTROLLER PASSES INTERNAL DIAGNOSTIC
9 023106 005014   ;   MAKE SURE CONTROLLER CAN SENSE STEP 1 AND 2
10 023110 012737 004000 002330 ;
11 023116 004737 020566          ;
12 023122 103717          ;
13 023124 012764 100000 000002 ;
14 023132 012737 010000 002330 ;
15 023140 004737 020566          ;
16 023144 103706          ;
17 023146          ;
   023146          ;
   023146 104403   ;
18

```

```

;
;   TEST1, SUBTEST 2
;   MAKE SURE CONTROLLER PASSES INTERNAL DIAGNOSTIC
;   MAKE SURE CONTROLLER CAN SENSE STEP 1 AND 2
;
BGNSUB; 2
T1.2:
TRAP   C1BSUB
CLR    (R4)
MOV    #SA.S1,CNTRSD
CALL   CNTRSP
BCS    T1SKIP
MOV    #SA.STP,2(R4)
MOV    #SA.S2,CNTRSD
CALL   CNTRSP
BCS    T1SKIP
; INIT CONTROLLER
; STEP 1 ASSERTED?
; WAIT FOR RESPONSE
; IF ERROR, EXIT
; SEND STEP 1
; STEP 2 ASSERTED?
; WAIT FOR RESPONSE
; IF ERROR, EXIT
ENDSUB
L10050:
TRAP   C1ESUB

```



```

1
2
3
4
5
6
7 023150          BGNSUB: 3
   023150          T1.3:
   023150 104402
8 023152 011504   TRAP   C#BSUB
   023154          MOV    (R5),R4           ; R4 POINTS TO IP REGISTER
9 023154          CLR    (R4)               ; INITIALIZE THE CONTROLLER
10 023154 005014   MOV    #SA.S1,CNTRSD          ; LOOK FOR STEP 1
11 023156 012737 004000 002330   CALL  CNTRSP                ; WAIT FOR RESPONSE
12 023164 004737 020566          BCS   T1SKIP                ; IF ERROR, BRANCH
13 023170 103674   MOV    #<SA.STP+SA.WRP>,WCHNGD ; SAVE NEW PORT CONTENTS
14 023172 012737 140000 002332   MOV    #<SA.STP+SA.WRP>,2(R4) ; INITIALIZE FOR PORT WRAP
15 023200 012764 140000 000002   CALL  WCHNG                 ; WAIT FOR THE PORT TO CHANGE
16 023206 004737 021240          BNE   T1SKIP                ; IF ERROR, BRANCH
17 023212 001263   CMP    #<SA.STP+SA.WRP>,2(R4) ; [TEMP] COMPARE WITH DATA WRITTEN
18
19
20 023214 012702 000001          BNE   3#                    ; [TEMP]
21 023220 012703 000020          1#: MOV    #1,R2              ; SET UP FOR SHIFTING '1'
22 023224 010237 002332          MOV    #16,,R3              ; SET UP LOOP COUNT
23 023230 010264 000002          2#: MOV    R2,WCHNGD          ; SAVE NEW PORT CONTENTS
24 023234 004737 021240          MOV    R2,2(R4)              ; WRITE PATTERN TO SA FOR LOOP
25 023240 001250          CALL  WCHNG                 ; WAIT FOR SA TO CHANGE
26
27
28 023242 000405          BNE   T1SKIP                ; IF ERROR, BRANCH
29 023244          CMP    R2,2(R4)          ; [TEMP] COMPARE R0 WITH ECHOED DATA
   023244 104455          BEQ   4#                    ; [TEMP] IF MATCH, BRANCH
   023246 000032          BR    4#                    ; [TEMP]
   023250 000000          3#: ERDF  26,,ERR026        ; REPORT ERROR
   023252 011344          TRAP  C#ERDF
   023254 000642          .WORD 26
   023256 006302          .WORD 0
   023260 005303          .WORD ERR026
   023262 001360          BR    T1SKIP                ; BRANCH
30 023254 000642          4#: ASL    R2                  ; MOVE THE SHIFTING ONE LEFT BY 1
31 023256 006302          DEC    R3                    ; DECREMENT COUNT
32 023260 005303          BNE   2#                    ; IF LOOP INCOMPLETE, BRANCH
33 023262 001360
34 023264          ENDSUB
   023264          L10051:
   023264 104403          TRAP  C#ESUB
35
    
```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 147  
 TEST 1: BUS ADDRESSING TEST

```

1
2
3      ;
4      ;   TEST1, SUBTEST 4
5      ;   INITIALIZE CONTROLLER WITH SMALLEST RING BUFFER AND INTERRUPTS
6      ;   DISABLED
7      ;
8 023266      BGNSUB: 4
   023266      T1.4:
   023266      104402      TRAP      C#BSUB
9 023270      005004      CLR      R4
10 023272      004737      017676      CALL     CNTINT      ; LOAD INIT CHARACTERISTICS IN R4
11 023276      001631      BEQ      T1SKIP    ; INIT CONTROLLER
12 023300      ENDSUB
   023300      L10052:
   023300      104403      TRAP      C#ESUB
13

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 148  
 TEST 1: BUS ADDRESSING TEST

```

1
2
3
4
5
6
7 023302
023302
023302 104402
8 023304 011504
9 023306
10 023306 016503 000004
11 023312 010302
12 023314 042703 177000
13 023320 042702 170777
14 023324 000302
15 023326 006202
16 023330 010237 002334
17 023334
18 023350
19 023350
023350 012746 000000
023354 012746 017564
023360 010346
023362 012746 000003
023366 104437
023370 062706 000010
20 023374
023374 012700 000000
023400 104441
21 023402 006203
22 023404 006203
23 023406 052703 100200
24 023412 005037 002214
25 023416 005014
26 023420 012737 004000 002330
27 023426 004737 020566
28 023432 103514
29 023434 010364 000002
30 023440 012700 000012
31 023444 010501
32 023446 062701 000026
33 023452 004737 017614
34 023456 005737 002214
35 023462 001016
36 023464
023464 104422
37
38 023466 005737 002256
39 023472 001771
40 023474 023765 002270 000030
41 023502 101043
42 023504 001364
43 023506 023765 002266 000026
44 023514 103760
45 023516 000435
46

```

```

:
: TEST 1, SUBTEST 5
: TEST THE INTERRUPTS VECTOR AND BR LEVEL
:
:
BGNSUB: 5
T1.5:
TRAP C#BSUB
MOV (R5),R4 ; R4 POINTS TO IP REGISTER
ASSUME C.UADR EQ 0
MOV C.VEC(R5),R3 ; GET VECTOR AND BR LEVEL
MOV R3,R2 ; COPY TO R2 FOR BR LEVEL
BIC #+CCT.VEC,R3 ; CLEAR UNUSED VECTOR BITS
BIC #+CCT.BRL,R2 ; CLEAR UNUSED BRANCH LEVEL BITS
SWAB R2 ; GET BR LEVEL IN LOW BYTE
ASR R2 ; DIVIDE BY 2
MOV R2,BRLEV ; SAVE THE BUS REQUEST LEVEL
PNTX INTST0,(R5),R3 ; PRINT BEGINNING OF INTERRUPT MESSAGE
ASSUME C.UADR EQ 0
SETVEC R3,#INTSRV,#PRI00 ; SETUP INTERRUPT VECTOR ADDRESS
MOV #PRI00,-(SP)
MOV #INTSRV,-(SP)
MOV R3,-(SP)
MOV #3,-(SP)
TRAP C#SVEC
ADD #10,SP
SETPRI #PRI00 ; SET PRIORITY = 0 TO CHECK INTERRUPTS
MOV #PRI00,R0
TRAP C#SPRI
ASR R3 ; DIVIDE VECTOR BY 4 FOR
; CONTROLLER INITIALIZATION
ASR R3 ; SET OTHER BITS FOR INITIALIZATION
BIS #<SA.STP+SA.INT>,R3 ; FLAG AS NO INTERRUPTS RECEIVED
CLR INTRCV ; INIT CONTROLLER
CLR (R4) ; LOOK FOR STEP 1 COMPLETION
MOV #SA.S1,CNTRSD ; WAIT FOR COMPLETION
CALL CNTRSP ; IF ERROR, EXIT
BCS 9# ; MOVE STEP 1 DATA TO CONTROLLER
MOV R3,2(R4) ; SET UP TIMEOUT OF 10 SECONDS
MOV #10.,R0 ; POINT TO CONTROLLER TABLE
MOV R5,R1 ; MOVE POINTER TO TIMEOUT FIELD
ADD #C.TO,R1 ; INITIALIZE TIMER
CALL SETTO ; SEE IF INTERRUPTED
TST INTRCV ; IF SO, EVERYTHING'S OK, SO BRANCH
BNE 3# ; >>>>>>BREAK BACK TO MONITOR<<<<<<<<
BREAK TRAP C#BRK
TST KW.CSR ; SEE IF CLOCK ON SYSTEM
BEQ 2# ; IF NOT, DON'T TIMEOUT
CMP KW.EL+2,C.TO(R5) ; CHECK HIGH WORD OF TIMER
BHI 6# ; IF GREATER, TIMER EXPIRED
BNE 2# ; IF NOT SAME, KEEP WAITING
CMP KW.EL,C.TO(R5) ; CHECK LOW WORD OF TIMER
BLO 2# ; IF LESS, KEEP WAITING
BR 6# ; ELSE, TIMER EXPIRED

```

## TEST 1: BUS ADDRESSING TEST

```

47 023520 005037 002214      3$: CLR      INTRCV      ; FLAG AS NO INTERRUPTS RECEIVED
48 023524      012700 000340      SETPRI   #PRI07      ; SET PRIORITY AS HIGHEST PRIORITY
      023524 012700 000340      MOV      #PRI07,R0
      023530 104441      TRAP    C$SPRI
49 023532 016402 000002      MOV      2(R4),R2      ; READ SA REGISTER
50 023536 005064 000002      CLR      2(R4)        ; WRITE SECOND STEP TO CONTROLLER
51 023542 012702 000144      MOV      #100.,R2     ; SET UP DELAY TO WAIT FOR INTERRUPT
52 023546 005302      4$: DEC      R2        ; DECREMENT COUNT
53 023550 001376      BNE     4$           ; IF INCOMPLETE, BRANCH
54 023552 012701 000007      MOV      #7.,R1      ; R1 IS PROCESS PRIORITY LEVEL
55 023556      5$: PUSH     R1        ; SAVE PRIORITY
56 023560 006301      ASL     R1          ; SHIFT
57 023562 006301      ASL     R1          ; PRIORITY
58 023564 006301      ASL     R1          ; LEFT
59 023566 006301      ASL     R1          ; 5
60 023570 006301      ASL     R1          ; BITS
61 023572      SETPRI   R1        ; SET RUNNING PRIORITY TO R1
      023572 010100      MOV     R1,R0
      023574 104441      TRAP    C$SPRI
62 023576      POP     R1        ; RESTORE R1
63 023600 005737 002214      TST     INTRCV      ; SEE IF INTERRUPT RECEIVED
64 023604 001007      BNE     7$           ; IF SO, BRANCH
65 023606 005301      DEC     R1          ; DECREMENT PRIORITY LEVEL
66 023610 100362      BPL     5$           ; IF ALL LEVELS UNTESTED, BRANCH
67 023612      6$: ERRDF   28,,ERR028 ; REPORT NO INTERRUPTS ERROR
      023612 104455      TRAP    C$ERDF
      023614 000034      .WORD  28
      023616 000000      .WORD  0
      023620 011402      .WORD  ERR028
68 023622 000420      BR      9$           ; BRANCH
69
70 023624      7$: SETPRI   #PRI00      ; SET RUNNING PRIORITY TO 0
      023624 012700 000000      MOV     #PRI00,R0
      023630 104441      TRAP    C$SPRI
71 023632 005201      INC     R1          ; SO PRIORITY = BR LEVEL
72 023634 013703 002326      MOV     SSTEP4,R3   ; GET BR LEVEL SAVED IN CNTINT
73 023640 001402      BEQ    8$           ; IF ZERO, THIS IS UDA50A - SKIP
74 023642 010337 002334      MOV     R3,BRLEV    ; KDA50-Q MUST USE DEFAULT BR LEVEL
75 023646 023701 002334      8$: CMP     BRLEV,R1   ; SEE IF BR LEVEL MATCHES PRIORITY
76 023652 001414      BEQ    10$          ; IF SO, BRANCH
77 023654      ERRDF   29,,ERR029 ; REPORT ERROR
      023654 104455      TRAP    C$ERDF
      023656 000035      .WORD  29
      023660 000000      .WORD  0
      023662 011414      .WORD  ERR029
78 023664 016503 000004      9$: MOV     C.VEC(R5),R3 ; GET VECTOR ADDRESS
79 023670 042703 177000      BIC    #+CCT.VEC,R3 ; CLEAR UNUSED BITS
80 023674      CLRVEC  R3         ; CLEAR VECTOR
      023674 010300      MOV     R3,R0
      023676 104436      TRAP    C$CVEC
81 023700 000137 022762      JMP     T1SKIP      ; EXIT TEST ON ERROR
82 023704      10$: PNTX    INTST1     ; PRINT TESTING COMPLETED
83 023714 016503 000004      MOV     C.VEC(R5),R3 ; GET VECTOR ADDRESS
84 023720 042703 177000      BIC    #+CCT.VEC,R3 ; CLEAR UNUSED BITS
85 023724      CLRVEC  R3         ; CLEAR VECTOR
      023724 010300      MOV     R3,R0
      023726 104436      TRAP    C$CVEC

```



TEST 1: BUS ADDRESSING TEST

```

1
2
3
4
5
6
7
8
9 023732
   023732
   023732 104402
10 023734 012704 126400
11 023740 004737 017676
12 023744 00:002
13 023746 000137 022762
14 023752
15 023752
   023752
   023752 104403
16

```

```

;
;
; TEST 1, SUBTEST 6
; INITIALIZE CONTROLLER WITH LARGE RING BUFFER LARGE ENOUGH TO COVER
; NORMAL HOST COMMUNICATIONS AREA PACKET AND BUFFER SPACE (A 5 IN
; MESSAGE LENGTH AND A 5 IN COMMAND LENGTH).
;
;
BGNSUB; 6
T1.6:
TRAP C#BSUB
MOV @<SA.STP>+<5*SA.MS1>+<5*SA.CH1>>,R4 ; LOAD CHARACTERISTICS
CALL CNTINT ; INIT CONTROLLER
BNE 1$ ; IF NO ERROR, CONTINUE
JMP T1SKIP ; ELSE, EXIT TEST

1$:
ENDSUB
L10054:
TRAP C#ESUB

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 150  
 TEST 1: BUS ADDRESSING TEST

```

1
2
3      ;**
4      ;   TEST 1, SUBTEST 7
5      ;   RUN DM PROGRAM IN CONTROLLER CURRENTLY UNDER TEST
6      ;--
7 023754      BGNSUB; 7
   023754      T1.7:
   023754      104402      TRAP      C#BSUB
8 023756      012701      000001      PUSH      <FFREE,FSIZE>      ; SAVE ON STACK
9 023766      004737      013104      MOV       #1,R1      ; SAVE COUNT OF CONTROLLERS TO RUN
10 023772     001407      013200      CALL      RUNDM      ; RUN DM PROGRAM
11 024000     001407      013200      BEQ      1$          ; EXIT IF ERROR
12 024004     004737      013200      CALL      RESFDM     ; WAIT FOR RESPONSE
13 024014     000406      013200      POP      <FSIZE,FFREE> ; RESTORE FROM STACK
14 024016     000406      013200      BR       2$          ;
15 024016     000137      022762      1$:      POP      <FSIZE,FFREE> ; RESTORE FROM STACK
16 024026     000137      022762      JMP      T1SKIP     ; IF ERROR, EXIT TEST
17 024032
18 024032      104403      TRAP      C#ESUB
19

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 151  
 TEST 1: BUS ADDRESSING TEST

```

1
2
3
4
5
6
7 024034
   024034
   024034 104402
8 024036
9 024046 013705 002176
10 024052 016504 000004
11 024056 042704 177000
12 024062 010501
13 024064 062701 000006
14 024070
   024070 012746 000340
   024074 010146
   024076 010446
   024100 012746 000003
   024104 104437
   024106 062706 000010
15 024112 006204
16 024114 006204
17 024116 004737 017676
18 024122 001422
19 024124 004737 012156
20 024130 010402
21 024132 062702 000100
22 024136 012704 036413
23
24
25
26 024142 012737 100000 002336
27 024150 004737 024206
28 024154 103405
29 024156 012737 077777 002336
30 024164 004737 024206
31 024170
32 024200
   024200
   024200 104403
33
34 024202 000137 022762
35
36
37
38
39
40
41
42
43
44
45 024206 012701 000020
46 024212 012700 000031
47 024216 013703 002336

```

```

***
:
: TEST 1, SUBTEST 8
: LARGE BUFFER I/O CONTROLLER MEMORY TEST
:
:--
BGNSUB: 8
T1.8:
TRAP C#BSUB
PUSH <FFREE,FSIZE> ; SAVE FREE MEMORY ADDRESS AND SIZE
MOV TSTTAB,R5 ; GET POINTER TO CONTROLLER TABLE
MOV C.VEC(R5),R4 ; GET VECTOR OF CONTROLLER
BIC #C<CT.VEC>,R4 ; MASK OTHER BITS
MOV R5,R1 ; GET INTERRUPT SERVICE LINK
ADD #C.JSR,R1 ;
SETVEC R4,R1,#PRI07 ; SET UP INTERRUPT VECTOR
MOV #PRI07,-(SP)
MOV R1,-(SP)
MOV R4,-(SP)
MOV #3,-(SP)
TRAP C#SVEC
ADD #10,SP
ASR R4 ; INITIALIZE CONTROLLER WITH SMALLEST
ASR R4 ; POSITION VECTOR FOR RING
CALL CNTINT ; BUFFER AND INTERRUPTS ENABLED
BEQ 16 ; IF ERROR, EXIT
CALL HCOMM ; ALLOCATE SPACE FOR HOST COMM AREA
MOV R4,R2 ; GET HOST COMM ADDRESS
ADD #MC.BF1,R2 ; POINT TO BUFFER AREA
MOV #T18MSZ,R4 ; SAVE CONTROLLER MEMORY SIZE
:
: PERFORM I/O AND DATA COMPARISON
:
:
MOV #100000,T18PAT ; INITIALIZE DATA PATTERN
CALL T18BLD ; PERFORM I/O
BCS 16 ; IF ERROR, EXIT TEST
MOV #77777,T18PAT ; INITIALIZE DATA PATTERN
CALL T18BLD ; PERFORM I/O
POP <FSIZE,FFREE> ; RESTORE FREE MEMORY ADDRESS AND SIZE
16:
ENDSUB
L10056:
TRAP C#ESUB
JMP T1SKIP ; GO TEST NEXT CONTROLLER
***
:
: T18BLD - BUILD THE DATA PACKETS FOR MAINTENANCE I/O OPERATIONS.
: WRITE THE PACKETS, READ BACK THE DATA AND PERFORM A DATA COMPARISON.
:
: INPUTS:
: R2 - DATA BUFFER LOCATION
: R4 - CONTROLLER MEMORY SIZE
:--
T18BLD: MOV #16.,R1 ; INITIALIZE PATTERN COUNT
T18BL1: MOV #OP.MWR,R0 ; GET MAINTENANCE WRITE COMMAND
MOV T18PAT,R3 ; GET OLD PATTERN

```



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 151-1  
 TEST 1: BUS ADDRESSING TEST

```

48 024222 005703          TST      R3          ; CHECK HIGH BIT
49 024224 002402          BLT      5#          ; IF BIT 15 SET, BRANCH
50 024226 000241          CLC          ; SHIFT 0 INTO PATTERN
51 024230 000401          BR      6#          ;
52 024232 000261          5#: SEC          ; SHIFT 1 INTO PATTERN
53 024234 006103          6#: ROL      R3          ; SHIFT PATTERN 1 BIT LEFT
54 024236 010337 002336  MOV      R3,T18PAT  ; SAVE PATTERN
55 024242 012703 000106  MOV      #MC.BSZ,R3 ; GET SIZE OF BUFFER (WORDS)
56 024246          PUSH     <R1,R4,R2> ; SAVE LOOP COUNT, MEMORY SIZE, ETC
57 024254 013722 002336  7#: MOV      T18PAT,(R2) ; MOVE DATA TO BUFFER
58 024260 005303          DEC      R3          ; DECREMENT WORDS REMAINING
59 024262 001374          BNE     7#          ; LOOP IF MORE WORDS TO LOAD
60 024264 012703 000106  MOV      #MC.BSZ,R3 ; GET SIZE OF BUFFER (WORDS)
61 024270 011602          MOV      (SP),R2    ; RESTORE BUFFER ADDRESS
62 024272 005001          CLR      R1          ; INIT CONTROLLER MEMORY OFFSET
63 024274 020403          T18NXT: CMP     R4,R3   ; CHECK REMAINING MEMORY
64 024276 002001          BGE     8#          ; IF MORE THAN BUFFER SIZE, SKIP
65 024300 010403          MOV     R4,R3       ; ELSE, USE REMAINING COUNT
66 024302 006303          8#: ASL     R3        ; CONVERT TO BYTES
67 024304 004737 017056  CALL    LOAD        ; LOAD INTO CONTROLLER
68 024310 001477          BEQ     T18EX1      ; IF ERROR, EXIT
69 024312 006203          ASR     R3          ; CONVERT BYTES TO WORDS
70 024314 060301          ADD     R3,R1       ; INCREASE OFFSET FOR NEXT BUFFER
71 024316 160304          SUB     R3,R4       ; REDUCE REMAINING BYTE COUNT
72 024320 001365          BNE     T18NXT     ; GET NEXT BUFFER
73          ;
74          ; READ DATA BACK FROM CONTROLLER MEMORY AND VERIFY IT
75          ;
76 024322 016604 000002  MOV     2(SP),R4    ; RESTORE CONTROLLER MEMORY SIZE
77 024326 012703 000106  MOV     #MC.BSZ,R3 ; GET SIZE OF BUFFER (WORDS)
78 024332 005001          CLR     R1          ; INIT CONTROLLER MEMORY OFFSET
79 024334 020403          T18RNK: CMP     R4,R3   ; CHECK REMAINING MEMORY
80 024336 002001          BGE     9#          ; IF MORE THAN BUFFER SIZE, SKIP
81 024340 010403          MOV     R4,R3       ; ELSE, USE REMAINING COUNT
82 024342          9#: PUSH     R3        ; SAVE BUFFER SIZE
83 024344 006303          ASL     R3          ; CONVERT WORD COUNT TO BYTES
84 024346 012700 000030  MOV     #OP.MRD,R0  ; GET MAINTENANCE READ COMMAND
85 024352 004737 017056  CALL    LOAD        ; READ DATA FROM CONTROLLER
86 024356 001453          BEQ     T18EX2      ; IF ERROR, GET OUT NOW
87 024360 006203          ASR     R3          ; CONVERT BYTE COUNT TO WORDS
88 024362 016602 000002  MOV     2(SP),R2    ; RESTORE DATA BUFFER ADDRESS
89 024366 023722 002336  10#: CMP     T18PAT,(R2) ; CHECK DATA READ FROM CONTROLLER
90 024372 001411          BEQ     11#         ; IF SAME, DATA OK
91 024374 162702 000002  SUB     #2.,R2      ; GO BACK TO BAD WORD
92 024400 062601          ADD     (SP),R1     ; POINT TO END OF DATA READ
93 024402 160301          SUB     R3,R1       ; POINT TO ADDRESS OF FAILING WORD
94 024404          ERRDF 39.,ERR039 ; PRINT COMPARE ERROR
   024404 104455          TRAP   C#ERRDF
   024406 000047          .WORD 39
   024410 000000          .WORD 0
   024412 011544          .WORD ERR039
95 024414 000435          BR     T18EX1      ; EXIT SUBTEST
96
97 024416 005303          11#: DEC     R3        ; DECREMENT WORDS REMAINING
98 024420 001362          BNE     10#        ; LOOP IF MORE WORDS TO READ
99
100 024422 011603          MOV     (SP),R3    ; RESTORE BUFFER SIZE

```



CZUDMA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 152  
 TEST 2: DISK RESIDENT DIAGNOSTIC TEST

```

1          .SBTTL TEST 2: DISK RESIDENT DIAGNOSTIC TEST
2
3 024524          BGNTST
4 024524          T2::
5 024524 012701 000002          MOV    #2,R1          ; INITIALIZE TEST PARAMETERS
9 024530 004737 013040          CALL   TINIT          ;
10 024534 013737 002172 002176  MOV    CTABS,TSTTAB  ; GET ADDRESS TO 1ST CONTROLLER TABLE
11
12 024542 004737 020744          1$:   CALL   RESET          ; RESET ALL CONTROLLERS
13 024546          PUSH   <FFREE,FSIZE>          ;
14 024556 012701 000001          MOV    #1,R1          ; RUN DM PROGRAM IN
15 024562 004737 013104          CALL   RUNDM          ; ONE CONTROLLER ONLY
16 024566 001402          BEQ    2$          ;
17 024570 004737 013200          CALL   RESPDM          ;
18 024574          2$:   POP    <FSIZE,FFREE>          ;
19 024604 062737 000034 002176  ADD    #C.SIZE,TSTTAB ; MOVE TO NEXT CONTROLLER
20 024612 005777 155360          TST   @TSTTAB          ; CHECK IF ANY MORE CONTROLLER TABLES
21 024616 001351          BNE   1$          ;
22
23 024620          ENDTST
   024620          L10057:
   024620 104401          TRAP   C#ETST
24

```

CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 153  
 TEST 3: DISK FUNCTION TEST

```

1          .SBTTL TEST 3: DISK FUNCTION TEST
2
3 024622          BGNTST
   024622          T3::
4 024622 012701 000003          MOV      @3,R1          ; INITIALIZE TEST PARAMETERS
8 024626 004737 013040          CALL    TINIT          ; INITIALIZE CONTROLLERS
9 024632 013737 002172 002176  MOV    CTABS,TSSTAB      ; GET ADDRESS OF 1ST TABLE ADDRESS
10 024640 013701 002174          MOV    CTRLRS,R1        ; GET COUNT OF CONTROLLERS TO TEST
11 024644 004737 013104          CALL    RUNDM          ; LOAD AND START DM PROGRAM
12 024650 001402          BEQ    14              ; EXIT ON ERROR
13 024652 004737 013200          CALL    RESPDM        ; WAIT FOR RESPONSE FROM CONTROLLERS
14 024656          14:
15          .EVEN
16
17 024656          ENDTST
   024656          L10060:
   024656 104401          TRAP   C4ETST
18
416

```

```

1      .SBTTL  HARDWARE PARAMETER CODING SECTION
2
3      ;**
4      ;      THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
5      ;      THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
6      ;      MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7      ;      INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8      ;      MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
9      ;      WITH THE OPERATOR.
10     ;--
11     024660      BGNHRD
12     024660      .WORD L10061-L#HARD/2
13     024662      000022
14     L#HARD::
15
16     TABLE                                     ;START A TABLE DEFINITION
17     ITEM H.UBA      2      <CSR ADDRESS>
18     ITEM H.VEC      2      <CONTROLLER VECTOR>
19     ITEM H.BRL      2      <BR LEVEL>
20     ITEM H.DRV      2      <DRIVE NUMBER>
21
22
23
24     HM.CYL  == BIT13                          ; TEST CUSTOMER DATA AREA
25     END
26
27     GPRMA  MSGUBA,H.UBA,0,160000,177774,YES   ; CSR ADDRESS OF CONTROLLER?
28     .WORD  T#CODE
29     .WORD  MSGUBA
30     .WORD  T#LOLIM
31     .WORD  T#HILIM
32
33     GPRMA  MSGVEC,H.VEC,0,4,774,YES           ; VECTOR?
34     .WORD  T#CODE
35     .WORD  MSGVEC
36     .WORD  T#LOLIM
37     .WORD  T#HILIM
38
39     GPRMD  MSGBRL,H.BRL,D,-1,4..7..YES       ; BR LEVEL?
40     .WORD  T#CODE
41     .WORD  MSGBRL
42     .WORD  -1
43     .WORD  T#LOLIM
44     .WORD  T#HILIM
45
46     GPRMD  MSGLDR,H.DRV,D,-1,0..255..YES     ; DRIVE #?
47     .WORD  T#CODE
48     .WORD  MSGLDR
49     .WORD  -1
50     .WORD  T#LOLIM
51     .WORD  T#HILIM
52
53     ENDHRD
54     .EVEN
55
56     L10061:
57
58     024726      103      123      122  MSGUBA: .ASCIZ  \CSR ADDRESS OF CONTROLLER\
59     024726      126      105      103  MSGVEC: .ASCIZ  \VECTOR\
60     024726      102      122      040  MSGBRL: .ASCIZ  \BR LEVEL\
61     025000      104      122      111  MSGLDR: .ASCIZ  \DRIVE #\

```

52  
53  
54  
55

.EVEN

ČZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 155  
SOFTWARE PARAMETER CODING SECTION

```

1      .SBTTL  SOFTWARE PARAMETER CODING SECTION
2
3      : **
4      :       THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
5      :       THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
6      :       MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7      :       INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8      :       MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
9      :       WITH THE OPERATOR.
10     : --
11
12     025010      BGNSFT
13     025010      000003      .WORD L10062-L$SOFT/2
14     025012
15     L$SOFT::
16     TABLE      ; START A TABLE DEFINITION
17     ITEM SO.BIT      2      <SINGLE BIT ANSWERS>
18     025012
19
20     23     000200      SM.MAN == BIT07      ; MANUAL INTERVENTION MODE
21     24     000400      SM.SSF  == BIT08      ; SUPPRESS SOFT ERRORS
22     25     001000      SM.LOG  == BIT09      ; ERROR LOG ENABLED
23     26     040000      SM.IW  == BIT14      ; INITIAL WRITE
24     27     025012      END
25
26
27
28
29
30
31
32     025012      GPRML S.MAN,SO.BIT,SM.MAN,YES
33     025012      000130      .WORD T$CODE
34     025014      025020      .WORD S.MAN
35     025016      000200      .WORD SM.MAN
36     55     025020      ENDSFT
37
38     025020      L10062:
39
40     56
41     58     025020      105      116      124      S.MAN: .ASCIZ \ENTER MANUAL INTERVENTION MODE IN TEST 2\
42     71
43     72

```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 156  
SOFTWARE PARAMETER CODING SECTION

```

1      ;**
2      ;      THIS IS WHERE THE DM PROGRAMS WILL BE LINKED
3      ;--
4
5      .DSABL AMA
6 000000 .PSECT DM,ABS          ; DM PROGRAMS ARE LINKED HERE
7 000000 .PSECT END
8
9      ;**
10     ;      THIS IS A PATCH AREA THAT SHOULD BE INCLUDED IN ALL DIAGNOSTICS.
11     ;      THE SIZE IS ADJUSTED AS NEEDED.
12     ;--
13
14 000000 $PATCH::
15 000000     .BLKW 16.
16
17 000040     LASTAD
18         .EVEN
19         .WORD T$FREE
20         .WORD T$SIZE
21
22     L$LAST::
23
24 000040 000060'
25 000042 000006
26 000044

```



```

1
2
3
4
5
6
7
8
9
10
11
12
13 000044
14 000044
   000044 000000
   000046 000004
   000050
15 000050 172150
17 000052 000154
18 000054 000005
20 000056 000000
24 000060
   000060
25 000060
26      000001

; **
;
;   HARDCODED P-TABLES MAY BE PLACED HERE BY USING THE SETUP MACROS.
;   THIS SECTION IS OPTIONAL AND SHOULD BE REMOVED IF IT IS NOT BEING
;   USED.  CHANGE THE POINTER MACRO ARGUMENT TO REFLECT THE REMOVAL.
;
;   THE P-TABLES ARE DELIMITED BY THE "BGNSETUP" AND "ENDSETUP" MACROS.
;   THE "BGNSETUP" MACRO HAS ONE ARGUMENT WHICH IS THE NUMBER OF
;   P-TABLE ENTRIES.  EACH ENTRY IS DELIMITED BY THE "BGNPTAB" AND
;   "ENDPTAB" MACROS.  NEITHER OF THESE MACROS REQUIRE AN ARGUMENT.
; --

      BGNSETUP      1
      BGNPTAB
      .WORD      0
      .WORD      L10065-./2-1
L10063:
      .WORD      172150          ; CSR ADDRESS
      .WORD      154           ; VECTOR ADDRESS
      .WORD      5.           ; BR LEVEL
      .WORD      0.           ; DRIVE NUMBER
      ENDPTAB
L10065:
      ENDSETUP
      .END

```

## Symbol table

ADR = 000020 G	CF.OTH= 000040	C#ERHR= 000056	DIV10 = 016052	EF.LOG= 000040
ALOCM = 012132	CF.SHC= 000002	C#ERRO= 000060	DMFRST= 001000	EF.NEM= 000035 G
ARG.CT= 000000	CF.THS= 000020	C#ERSF= 000054	DMMAIN= 000040	EF.PWR= 000034 G
ASS = 000001	CF.576= 000001	C#ERSO= 000057	JMOVRL= 000004	EF.RES= 000037 G
ASSEMB= 000010	CLOG = 020724	C#ESCA= 000010	DMPROG = 002200	EF.SEX= 000020
BAS = 004105	CLRBFL = 017364	C#ESEG= 000005	DMPTR = 002340	EF.STA= 000040 G
BASLN = 004106	CLRBUF = 017336	C#ESUB= 000003	DMRQA = 014564	ERRBLK = 002156 G
BASL1 = 004024	CNTINT = 017676	C#ETST= 000001	DMRQB = 014626	ERRC = 013006
BASL2 = 004042	CNTIST = 020272	C#EXIT= 000032	DMRQD = 014720	ERRD = 013020
BASL3 = 004070	CNTRSD = 002330	C#GETB= 000026	DMRQE = 015032	ERRLIM = 002643
BASNO = 003724	CNTRSP = 020566	C#GETW= 000027	DMRQO = 013770	ERRME1 = 002477
BASN1 = 003743	CNTRSRV = 017554 G	C#GMAN= 000043	DMRQ2 = 014110	ERRMSG = 002154 G
BASN2 = 003764	CON.A = 012566	C#GPHR= 000042	DMRQ2E = 014546	ERRMSX = 014714
BASN3 = 004004	CON.A1 = 012572	C#GPLD= 000030	DMRQ2M = 014124	ERRNBR = 002152 G
BELL = 000007 G	CON.A2 = 012614	C#GPRI= 000040	DMRQ2N = 014470	ERRTP = 002150 G
BIT0 = 000001 G	CON.D = 012616	C#INIT= 000011	DMRQ2Q = 014302	ERRVEC= 000004 G
BIT00 = 000001 G	CON.H = 012630	C#INLP= 000020	DMRQ2R = 014444	ERR.SZ= 000011
BIT01 = 000002 G	CON.N = 012654	C#MANI= 000050	DMRQ2V = 014370	ERR.TB = 012756
BIT02 = 000004 G	CON.N1 = 012660	C#MEM = 000031	DMRQ2W = 014520	ERR.TN = 011760 G
BIT03 = 000010 G	CON.O = 012642	C#MSG = 000023	DMRQ2X = 014542	ERR001 = 011002 G
BIT04 = 000020 G	CON.QU = 012546	C#OPEN= 000034	DMRQ20 = 014144	ERR002 = 011020 G
BIT05 = 000040 G	CON.QX = 012564	C#PNTB= 000014	DMRQ22 = 014216	ERR003 = 011036 G
BIT06 = 000100 G	CON.R = 012676	C#PNTF= 000017	DMRQ23 = 014400	ERR004 = 011054 G
BIT07 = 000200 G	CON.R1 = 012720	C#PNTS= 000016	DMRQ7 = 014560	ERR006 = 011066 G
BIT08 = 000400 G	CON.S = 012734	C#PNTX= 000015	DMTRLN= 000000	ERR008 = 011100 G
BIT09 = 001000 G	CON.S1 = 012740	C#QIO = 000377	DSPSIZ= 000017	ERR014 = 011116 G
BIT1 = 000002 G	CR = 000015 G	C#RDBU= 000007	DTABS = 002170 G	ERR021 = 011134 G
BIT10 = 002000 G	CRLF = 002440	C#REFG= 000047	DT.AVL = 100000	ERR022 = 011204 G
BIT11 = 004000 G	CTABER = 022606	C#RESE= 000033	DT.UNT = 000077	ERR023 = 011232 G
BIT12 = 010000 G	CTABS = 002172 G	C#REVI= 000003	DUP = 001000	ERR024 = 011312 G
BIT13 = 020000 G	CTRLRS = 002174	C#RFLA= 000021	DU.DFL = 020000	ERR025 = 011326 G
BIT14 = 040000 G	CT.AVL = 100000	C#RPT = 000025	DU.FTL = 050000	ERR026 = 011344 G
BIT15 = 100000 G	CT.BRL = 007000	C#SEFG= 000046	DU.INF = 030000	ERR027 = 011364 G
BIT2 = 000004 G	CT.CMD= 000004	C#SPRI= 000041	DU.QUE = 010000	ERR028 = 011402 G
BIT3 = 000010 G	CT.MSG= 000010	C#SVEC= 000037	DU.SPC = 060000	ERR029 = 011414 G
BIT4 = 000020 G	CT.REQ= 000020	C#TPRI= 000013	DU.TER = 040000	ERR030 = 011430 G
BIT5 = 000040 G	CT.RN = 000002	C.DR0 = 000016	D.BE = 000040	ERR031 = 011444 G
BIT6 = 000100 G	CT.UNT= 000077	C.DR1 = 000020	D.CYL = 000400	ERR032 = 011456 G
BIT7 = 000200 G	CT.VEC= 000777	C.DR2 = 000022	D.DC = 000002	ERR033 = 011474 G
BIT8 = 000400 G	CT.VER= 000100	C.DR3 = 000024	D.DCA = 000001	ERR035 = 011502 G
BIT9 = 001000 G	C#AU = 000052	C.FLG = 000012	D.DCY = 020000	ERR036 = 011520 G
BLDCMD = 017176	C#AUTO= 000061	C.HCOM= 000014	D.DRV = 000000	ERR038 = 011532 G
BLDC0 = 017244	C#BRK = 000022	C.JAD = 000010	D.ECC = 010000	ERR039 = 011544 G
BLDC1 = 017252	C#BSEG= 000004	C.JSR = 000006	D.IW = 040000	ERR21A = 011172
BNE = 000400 G	C#BSUB= 000002	C.REF = 000032	D.RET = 001000	ERR21E = 011202
BRLEV = 002334	C#CEFG= 000045	C.SIZE= 000034	D.RO = 004000	ERR23A = 011252
CALR1 = 015036	C#CLCK= 000062	C.TO = 000026	D.SEQ = 000100	ERR23B = 011272
CALR2 = 015064	C#CLEA= 000012	C.TOH = 000030	D.SIZE= 000004	EVL = 000004 G
CALR3 = 015162	C#CLOS= 000035	C.UADR= 000000	D.TR = 000020	E\$END = 002100
CALR4 = 015176	C#CLP1= 000006	C.UNIT= 000002	D.UNIT= 000002	E\$LOAD= 000035
CALR5 = 015252	C#CVEC= 000036	C.VEC = 000004	D.WC = 000010	FFREE = 002160 G
CALR6 = 015330	C#DCLN= 000044	DDEF = 011012	D.WCA = 000004	FMEM = 002164
CALR7 = 015344	C#DODU= 000051	DEBUG = 000000	D.WO = 002000	FMEMS = 002166
CALR8 = 015362	C#DRPT= 000024	DFPTBL = 002134 G	D.ZERO= 140200	FMERR = 012120
CALR9 = 015400	C#DU = 000053	DIAG = 177777	EF.BBR= 000200	FRMT = 002435
CF.ATN= 000200	C#EDIT= 000003	DIAGMC= 000000	EF.BBU= 000100	F\$SIZE = 002162 G
CF.MSC= 000100	C#ERDF= 000055	DIVIDE = 016014	EF.CON= 000036 G	F\$AU = 000015

Symbol table

F\$AUTO= 000020	HC.MPK= 000020	KW.VEC 002262	L\$LADP 002026 G	L10045 022722
F\$BGN = 000040	HC.MSG= 000004	KW11I 017572 G	L\$LAST 000044RG	003 L10046 024522
F\$CLEA= 000007	HC.PSZ= 000060	LDDM 013126	L\$LOAD 002100 G	L10047 023102
F\$DU = 000016	HC.RSZ= 000004	LDNEXT 013160	L\$LUN 002074 G	L10050 023146
F\$END = 000041	HC.SIZ= 000314	LF = 000012 G	L\$MREV 002050 G	L10051 023264
F\$HARD= 000004	HM.CYL= 020000 G	LOAD 017056	L\$NAME 002000 G	L10052 023300
F\$HW = 000013	HOE = 100000 G	LOADB 016744	L\$PRIO 002042 G	L10053 023730
F\$INIT= 000006	H.BRL = 000004	LOADDM 016524	L\$PROT 021472 G	L10054 023752
F\$JMP = 000050	H.DRV = 000006	LOADER 017052	L\$PRT 002112 G	L10055 024032
F\$MOD = 000000	H.UBA = 000000	LOADE1 017042	L\$REPP 002062 G	L10056 024200
F\$MSG = 000011	H.VEC = 000002	LOADTX 016740	L\$REV 002010 G	L10057 024620
F\$PROT= 000021	IBE = 010000 G	LOADX 017164	L\$RPT 021334 G	L10060 024656
F\$PWR = 000017	ICONT = 000002 G	LOADX1 017154	L\$SOFT 025012 G	L10061 024726
F\$RPT = 000012	IDU = 000040 G	LOE = 040000 G	L\$SPC 002056 G	L10062 025020
F\$SEG = 000003	IER = 020000 G	LOG = 000001	L\$SPCP 002020 G	L10063 000050R
F\$SOFT= 000005	IFLAGS 002202 G	LOT = 000010 G	L\$SPTP 002024 G	L10065 000060R
F\$SRV = 000010	INITBL 002312	LPNT 012430	L\$STA 002030 G	MD.CMP= 040000
F\$SUB = 000002	INITXX 022574	LPNTB 012402	L\$SW 002146 G	MD.CMB= 000010
F\$SW = 000014	INIT1 021772	LPNTF 012372	L\$TEST 002114 G	MD.ERR= 010000
F\$TEST= 000001	INIT2 022154	LPNTS 012422	L\$TIML 002014 G	MD.EXP= 100000
GETCDN 015540	INIT3 022212	LPNTX 012412	L\$UNIT 002012 G	MD.FEU= 000001
GETCNT 015474	INIT4 022246	LT1L1 016634	L10000 002144	MD.IMF= 000002
GETCNX 015500	INIT6 022560	LT1L2 016662	L10001 002150	MD.NXU= 000001
GETCXX 015546	INTRCV 002214	LT11 016646	L10002 011016	MD.PRI= 000001
GTDRVT 015404	INTSRV 017564 G	L\$ACP 002110 G	L10003 011034	MD.RIP= 000001
G\$CNTD= 000200	INTSTO 002736	L\$APT 002036 G	L10004 011052	MD.SCH= 004000
G\$DELM= 000372	INTST1 003042	L\$AU 022716 G	L10005 011064	MD.SCL= 002000
G\$DISP= 000003	IPADRS 002246	L\$AUT 002070 G	L10006 011076	MD.SEC= 000100
G\$EXCP= 000400	IREST = 000004 G	L\$AUTO 022674 G	L10007 011114	MD.SEQ= 000020
G\$HILI= 000002	ISR = 000100 G	L\$CCP 002106 G	L10010 011132	MD.SER= 000400
G\$LOLI= 000001	ISTRT = 000010 G	L\$CLEA 022676 G	L10011 011202	MD.SPD= 000001
G\$NO = 000000	ISTRTH= 000020 G	L\$CD 002032 G	L10012 011230	MD.SSH= 000200
G\$OFFS= 000400	IXE = 004000 G	L\$DEPD 002011 G	L10013 011310	MD.SWP= 000004
G\$OFSI= 000376	I\$AU = 000041	L\$DESC 002366 G	L10014 011324	MD.VOL= 000002
G\$PRMA= 000001	I\$AUTO= 000041	L\$DESP 002076 G	L10015 011342	MD.WBN= 000100
G\$PRMD= 000002	I\$CLN = 000041	L\$DEVP 002060 G	L10016 011362	MD.WBV= 000400
G\$PRML= 000000	I\$DU = 000041	L\$DISP 002124 G	L10017 011400	MEMFIL 014000
G\$RADA= 000140	I\$HRD = 000041	L\$DLY 002116 G	L10020 011412	MESSG 003574
G\$RADB= 000000	I\$INIT= 000041	L\$DTP 002040 G	L10021 011426	MLDRER 022622
G\$RADD= 000040	I\$MOD = 000041	L\$DTYP 002034 G	L10022 011442	MSCP = 000000
G\$RADL= 000120	I\$MSG = 000041	L\$DU 022710 G	L10023 011454	MSGBRL 024767
G\$RADO= 000020	I\$PROT= 000040	L\$DUT 002072 G	L10024 011472	MSGGLDR 025000
G\$XFER= 000004	I\$PTAB= 000041	L\$DVTY 002342 G	L10025 011500	MSGPKL 011704
G\$YES = 000010	I\$PWR = 000041	L\$EF 002052 G	L10026 011516	MSGPKT 011660
HCOMM 012156	I\$RPT = 000041	L\$ENVI 002044 G	L10027 011530	MSGUBA 024726
HC.BF1= 000100	I\$SEG = 000041	L\$ERRT 002150 G	L10030 011542	MSGVEC 024760
HC.BF2= 000206	I\$SETU= 000041	L\$ETP 002102 G	L10031 011564	MXFERP 002566
HC.BSZ= 000106	I\$SFT = 000041	L\$EXP1 002046 G	L10032 012116	NCON 012104
HC.CCT= 000012	I\$SRV = 000041	L\$EXP4 002064 G	L10033 017552	NCONF 012530
HC.CEV= 000014	I\$SUB = 000041	L\$EXP5 002066 G	L10034 017562	NCONS 012506
HC.CMD= 000010	I\$TST = 000041	L\$HARD 024662 G	L10035 017570	NOCLOC 003647
HC.CPK= 000020	J\$JMP = 000167	L\$HIME 002120 G	L10036 017612	NXMAD 002276
HC.ESZ= 000004	KW.BRL 002260	L\$HPCP 002016 G	L10037 021470	NXMI 017544 G
HC.INT= 000000	KW.CSR 002256	L\$HPTP 002022 G	L10041 022672	OP.ABO= 000001
HC.ISZ= 000004	KW.EL 002266	L\$HW 002134 G	L10042 022674	OP.ACC= 000020
HC.MCT= 000006	KW.HZ 002264	L\$ICP 002104 G	L10043 022706	OP.AVA= 000100
HC.MEV= 000014	KW.OUT= 000105 G	L\$INIT 021500 G	L10044 022714	OP.AVL= 000010

003  
003

Symbol table

OP.CCD= 000021	P.BUFF= 000020	RNTIM2 002474	SA.VCE = 000177	T\$FLAG= 000040
OP.CMP= 000040	P.CHVR= 000023	RPTMSG 021434	SA.VEC = 000177	T\$FREE= 000060R
OP.DUP= 000101	P.CMST= 000020	RPTXX 021414	SA.WRP = 040000	T\$GMAN= 000000
OP.ELP= 000003	P.CNTF= 000016	RSPJRP 013410	SET00 017626	T\$HILI= 000377
OP.END= 000200	P.CNTI= 000024	RSPDSP 013732	SET01 017634	T\$LAST= 000001
OP.ERS= 000022	P.CPSP= 000042	RSPERR 013474	SET02 017642	T\$LOLI= 000000
OP.ESP= 000002	P.CRF = 000000	RSPIN 013426	SETTO 017614	T\$LSYM= 010000
OP.FLU= 000023	P.CSVR= 000022	RSPMWR 013446	SFPTBL 002146 G	T\$LTND= 000003
OP.GCS= 000002	P.CTMO= 000020	RSPNRP 013374	SM.IW = 040000 G	T\$NEST= 177777
OP.GSS= 000001	P.CYLS= 000050	RSPNTO 013340	SM.LOG= 001000 G	T\$NS0 = 000005
OP.GUS= 000003	P.DEXT= 000014	RSPNXT 013342	SM.MAN= 000200 G	T\$NS1 = 000002
OP.MRD= 000030	P.DFLG= 000017	RSPOU 013514	SM.SSF = 000400 G	T\$PCNT= 000000
OP.MWR= 000031	P.DMDI= 000024	RSPOUT 013630	SNDCMD 017270	T\$PTAB= 010064
OP.ONL= 000011	P.DPRG= 000020	RSPOU2 013676	SND.S1 002314	T\$PTHV= 000001
OP.RD = 000041	P.DTMO= 000024	RSPOU3 013704	SND.S2 002320	T\$PTNU= 000001
OP.RLC= 000103	P.ELGF= 000034	RSPPTW 013506	SND.S3 002324	T\$SAVL= 177777
OP.RPL= 000024	P.FBBK= 000034	RSPPT2 013516	SO.BIT= 000000	T\$SEGL= 177777
OP.RSD= 000005	P.FLGS= 000011	RSPPT3 013566	SSTEP4 002326	T\$SIZE= 000006
OP.SCC= 000004	P.GRPS= 000046	RSPRPT 013372	STIME 002272	T\$SUBN= 000000
OP.SEX= 000007	P.HSTI= 000020	RSPTM 013276	ST.ABO= 000002	T\$TAGL= 177777
OP.SHC= 000102	P.HTMO= 000020	RSPTMO 013326	ST.AOL= 000400	T\$TAGN= 010066
OP.SSD= 000004	P.LBN = 000034	RSP.CK 020546	ST.AVL= 000004	T\$TEMP= 000005
OP.SUC= 000012	P.MEDI= 000034	RSP.S1 020500	ST.CMD= 000001	T\$TEST= 000003
OP.WR = 000042	P.MLUN= 000014	RSP.S2 020512	ST.CMP= 000007	T\$TSTM= 177777
OSTRE 012544	P.MOD = 000012	RSP.S3 020532	ST.CNT= 000012	T\$TSTS= 000001
OSTRNG 012476	P.OPCD= 000010	RUNDH 013104	ST.DAT= 000010	T\$\$AU = 010045
O\$APTS= 000000	P.OTRF= 000014	SAMVEC 022656	ST.DIA= 000037	T\$\$AUT= 010042
O\$AU = 000000	P.OVRL= 000034	SA.BST= 000374	ST.DRV= 000013	T\$\$CLE= 010043
O\$BGNR= 000001	P.RBN = 000014	SA.CMD= 034000	ST.HST= 000011	T\$\$DAT= 010065
O\$BGNS= 000001	P.RBNS= 000056	SA.CME= 000070	ST.MFE= 000005	T\$\$DU = 010044
O\$DU = 000000	P.RCTC= 000057	SA.CM1= 004000	ST.MSK= 000037	T\$\$HAR= 010061
O\$ERRT= 000001	P.RCTS= 000054	SA.CNT= 000360	ST.OFL= 000003	T\$\$HW = 010000
O\$GNSW= 000001	P.RGID= 000034	SA.CTP= 003400	ST.SUB= 000040	T\$\$INI= 010041
O\$POIN= 000001	P.RGOF= 0C J040	SA.DI = 000400	ST.SUC= 000000	T\$\$MSG= 010032
O\$SETU= 000001	P.SHST= 000042	SA.ERC= 003777	ST.WPR= 000006	T\$\$PC = 000001
PB 012276	P.SHUN= 000040	SA.ERR= 100000	SVCGBL= 000000	T\$\$PRO= 010040
PF 012252	P.STS = 000012	SA.GO = 000001	SVCINS= 000000	T\$\$PTA= 010064
PNT = 001000 G	P.TIME= 000024	SA.INE= 000200	SVCSUB= 000000	T\$\$RPT= 010037
PNTERR 016360	P.TRKS= 000044	SA.INT= 000200	SVCTAG= 000000	T\$\$SOF= 010062
PNTNUM 015552	P.UADR= 000020	SA.LFC= 000002	SVCTST= 000000	T\$\$SRV= 010036
PNTNUS 015560	P.UNFL= 000016	SA.MCV= 000017	S\$LSYM= 010000	T\$\$SUB= 010056
PNTPKL 011616	P.UNIT= 000004	SA.MP = 000100	S.MAN 025020	T\$\$SW = 010001
PNTPKT 011566	P.UNSZ= 000044	SA.MSE= 000007	TEMP 002216	T\$\$TES= 010060
PRI = 002000 G	P.UNTI= 000024	SA.MSG= 003400	TEST1 = ***** GX	T1 022724 G
PRINTC 012216	P.USEF= 000022	SA.MS1= 000400	TEST2 = ***** GX	T1NEXT 022742
PRI00 = 000000 G	P.VRSN= 000014	SA.NV = 002000	TEST3 = ***** GX	T1NOW 023002
PRI01 = 000040 G	P.VSER= 000050	SA.NVE= 000400	TINDEX= 000002	T1SKIP 022762
PRI02 = 000100 G	READDT 013076	SA.PRG= 000001	TINIT 013040	T1.1 023006
PRI03 = 000140 G	RESET 020744	SA.QB = 001000	TNAMES 013000	T1.2 023104
PRI04 = 000200 G	RESPCT 013212	SA.SM = 000040	TNUM 002204	T1.3 023150
PRI05 = 000240 G	RESPDM 013200	SA.STE= 000200	TOOMER 022640	T1.4 023266
PRI06 = 000300 G	RG.FLG= 040000	SA.STP= 100000	TSTTAB 002176	T1.5 023302
PRI07 = 000340 G	RG.OWN= 100000	SA.S1 = 004000	TTYOUT 002310	T1.6 023732
PS 012346	RNTIM 002443	SA.S2 = 010000	T\$ARGC= 000002	T1.7 023754
PTYPE 002306	RNTIME 021046	SA.S3 = 020000	T\$CODE= 000130	T1.8 024034
PX 012322	RNTIMX 021226	SA.S4 = 040000	T\$ERRN= 000047	T18BLD 024206
P.BCNT= 000014	RNTIM1 002466	SA.TST= 100000	T\$EXCP= 000000	T18BL1 024212

003

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 157-4

## Symbol table

T18EX	024516	T2PNTB	016150	UF.WBN=	000100	XPKT1	010301	X26	006331
T18EX1	024510	T2PNTD	016222	UF.WPH=	020000	XPKT2	010606	X27	006530
T18EX2	024506	T2PNT0	016200	UF.WPS=	001000	XSA	010635	X28	006652
T18MSZ=	036413	T2PNTW	016120	UF.576=	000004	X\$ALWA=	000000	X29	006727
T18NXT	024274	T2WARN	003057	URNING	002210	X\$FALS=	000040	X3	004340
T18PAT	002336	T2WRO	002302	URUN	002206	X\$OFFS=	000400	X3A	004117
T18RNX	024334	T2WRR	002300	UTOT1	014576	X\$TRUE=	000020	X31	007066
T2	024524 G	T3	024622 G	UTOT1A	014612	X1	004202	X32	007210
T2CMS1	003156	T4OPT7	002432	UTOT2	014616	X1A	004117	X35	007321
T2CMS5	003553	UAM	= 000200 G	WAITMS	017400	X14	004731	X36	007401
T2DR	002304	UCNT	002212	WCHNG	021240	X2	004271	X38	007526
T2GND1	016252	UF.CMR=	000001	WCHNGD	002332	X2A	004117	X39	007732
T2GND2	016254	UF.CMW=	000002	XFCE	010126	X22	005332	X4	004426
T2GND3	016314	UF.INA=	040000	XFRU	010672	X23A	005465	X6	004571
T2GNE	016350	UF.RPL=	100000	XFRUA	010741	X23B	005777	X8	004656
T2GNM	016232	UF.SCH=	004000	XMSG1	010200	X24	006013	X8A	004117
T2GNX	016344	UF.SCL=	002000	XMSG2	010234	X25	006200	\$PATCH	000000RG 003
T2PNT	016160								

. ABS. 025072 000 (RW,I,GBL,ABS,OVR)  
 000000 001 (RW,I,LCL,REL,CON)  
 DM 000000 002 (RW,I,LCL,ABS,CON)  
 END 000060 003 (RW,I,LCL,REL,CON)  
 Errors detected: 0

## \*\*\* Assembler statistics

Work file reads: 661  
 Work file writes: 595  
 Size of work file: 28754 Words ( 113 Pages)  
 Size of core pool: 14080 Words ( 55 Pages)  
 Operating system: RT-11 (Under RTEM-11)

Elapsed time: 00:04:23.00  
 ZUDHAO,ZUDHAO/C=SVC34R.MLB,ZUDHAO.MEM,CZUDHO.MAC











CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page S-5  
 Cross reference table (CREF V05.01)

DMRQ23	105-72	105-84#					
DMRQ2E	105-79	105-88	105-91	105-93	105-108	105-113#	118-49
DMRQ2M	105-34	105-36#					
DMRQ2N	105-98	105-100#					
DMRQ2Q	105-53	105-66#	105-114				
DMRQ2R	105-86	105-95#					
DMRQ2V	105-70	105-78#					
DMRQ2W	105-101	105-105	105-107#				
DMRQ2X	105-35	105-39	105-80	105-111#			
DMRQ7	102-159	106-10#					
DMRQA	102-152	107-18#					
DMRQB	102-163	102-164	108-26#				
DMRQD	102-165	109-19#					
DMRQE	102-166	110-14#					
DMTRLN	95-84#						
DSPSIZ	102-110	102-183#					
DT.AVL	95-15#	107-25	137-81	137-111			
DT.UNT	95-13#						
DTABS	96-14#	137-154*	137-155*	137-266	137-296*	137-297*	
DU.DFL	93-101#						
DU.FTL	93-104#						
DU.INF	93-102#						
DU.QUE	93-100#						
DU.SPC	93-105#	102-106					
DU.TER	93-103#						
DUP	90-17#	122-22					
E\$END	84-153#						
E\$LOAD	84-153#	84-201					
EF.BBR	91-80#						
EF.BBU	91-81#						
EF.CON	88-8#	137-58					
EF.LOG	91-82#						
EF.NEW	88-8#						
EF.PWR	88-8#	137-65					
EF.RES	88-8#	137-52					
EF.SEX	91-83#						
EF.STA	88-8#	137-46					
ERR.SZ	99-230	99-264#					
ERR.TB	99-233	99-254#	99-264				
ERR.TN	98-165#	119-40					
ERR001	98-10#	138-9					
ERR002	98-15#	138-19					
ERR003	98-19#	138-28					
ERR004	98-23#	99-9					
ERR006	98-27#	137-103					
ERR008	98-32#	138-37					
ERR014	98-37#	128-122	128-128	128-134			
ERR021	98-42#	130-51					
ERR022	98-55#	130-40					
ERR023	98-60#	128-96					
ERR024	98-75#	128-76					
ERR025	98-79#	129-93					
ERR026	98-83#	146-29					
ERR027	98-87#	134-22					
ERR028	98-91#	148-67					
ERR029	98-95#	148-77					

ERR030	98-100#	102-26	125-44												
ERR031	98-104#	102-44													
ERR032	98-108#	102-112													
ERR033	98-113#	102-94	120-91	121-35											
ERR035	98-117#	112-25													
ERR036	98-122#	125-35													
ERR038	98-126#	129-36	144-15												
ERR039	98-131#	151-94													
ERR21A	98-47	98-50#													
ERR21E	98-49	98-52#													
ERR23A	98-63#	98-70													
ERR23B	98-64	98-68#													
ERRBLK	96-8#														
ERRC	99-151	99-158	99-280#												
ERRD	99-160	99-296#													
ERRLIM	97-42#														
ERRME1	97-40#	99-156	99-236												
ERRMSG	96-8#														
ERRMSX	108-49	108-57	108-83#												
ERRNBR	96-8#														
ERRTYP	96-8#	119-31													
ERRVEC	88-12	129-30	129-33	132-14	144-8	144-12									
EVL	88-8#														
F\$AU	84-153#	142-9	142-14												
F\$AUTO	84-153#	139-10	139-12												
F\$BGN	84-153#	98-10	98-15	98-19	98-23	98-27	98-32	98-37	98-42	98-55	98-60	98-75	98-79	98-83	
	98-87	98-91	98-95	98-100	98-104	98-108	98-113	98-117	98-122	98-126	98-131	98-165	126-11	126-32	
	126-42	126-50	135-10	136-9	137-44	137-353	139-10	140-8	140-14	141-8	142-9	143-10	143-26	144-6	
	144-6	144-19	145-8	145-8	145-17	146-7	146-7	146-34	147-8	147-8	147-12	148-7	148-7	148-86	
	149-9	149-9	149-15	150-7	150-7	150-18	151-7	151-7	151-32	151-128	152-3	152-23	153-3	153-17	
	154-11	155-12	157-13	157-14	157-14	157-24	157-25								
F\$CLEA	84-153#	140-8	140-16												
F\$DU	84-153#	141-8	141-12												
F\$END	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153	84-153
	84-153	84-153	84-153#	98-12	98-17	98-21	98-25	98-29	98-34	98-39	98-53	98-58	98-73	98-77	
	98-81	98-85	98-89	98-93	98-97	98-102	98-106	98-111	98-115	98-120	98-124	98-128	98-133	98-186	
	126-13	126-35	126-44	126-54	135-93	135-107	137-353	138-42	139-12	140-14	140-16	141-10	141-12	142-12	
	142-14	143-10	143-10	143-10	143-26	144-6	144-6	144-19	144-19	145-8	145-8	145-17	145-17	146-7	
	146-7	146-34	146-34	147-8	147-8	147-12	147-12	148-7	148-7	148-86	148-86	149-9	149-9	149-15	
	149-15	150-7	150-7	150-18	150-18	151-7	151-7	151-32	151-32	151-128	151-128	152-3	152-3	152-3	
	152-23	152-23	153-3	153-3	153-3	153-17	153-17	154-41	155-55	157-13	157-14	157-24	157-25		
F\$HARD	84-153#	154-11	154-41												
F\$HW	84-153#	86-12	86-23												
F\$INIT	84-153#	137-44	138-42												
F\$JMP	84-153#	135-93	135-93	137-353	140-14	141-10	141-10	142-12	142-12	143-26					
F\$MOD	84-153#														
F\$MESSG	84-153#	98-10	98-12	98-15	98-17	98-19	98-21	98-23	98-25	98-27	98-29	98-32	98-34	98-37	
	98-39	98-42	98-53	98-55	98-58	98-60	98-73	98-75	98-77	98-79	98-81	98-83	98-85	98-87	
	98-89	98-91	98-93	98-95	98-97	98-100	98-102	98-104	98-106	98-108	98-111	98-113	98-115	98-117	
	98-120	98-122	98-124	98-126	98-128	98-131	98-133	98-165	98-186						
F\$PROT	84-153#	136-9	136-15												
F\$PWR	84-153#														
F\$RPT	84-153#	135-10	135-107												
F\$SEG	84-153#														
F\$SOFT	84-153#	155-12	155-55												
F\$SRV	84-153#	126-11	126-13	126-32	126-35	126-42	126-44	126-50	126-54						



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 5-8  
Cross reference table (CREF V05.01)

HC.RSZ	90-49#	90-59	90-62												
HC.SIZ	90-71#	99-51													
HCOMM	99-51#	120-25	151-19												
HM.CYL	154-24#														
HOE	88-8#														
I\$AU	84-153#	142-9#	142-14#												
I\$AUTO	84-153#	139-10#	139-12#												
I\$CLN	84-153#	140-8#	140-14	140-16#											
I\$DU	84-153#	141-8#	141-12#												
I\$HRD	154-11#	154-41#													
I\$INIT	84-153#	137-44#	137-353	138-42#											
I\$MOD	84-153#														
I\$MSG	84-153#	98-10#	98-12#	98-15#	98-17#	98-19#	98-21#	98-23#	98-25#	98-27#	98-29#	98-32#	98-34#	98-37#	
		98-39#	98-42#	98-53#	98-55#	98-58#	98-60#	98-73#	98-75#	98-77#	98-79#	98-81#	98-83#	98-85#	98-87#
		98-89#	98-91#	98-93#	98-95#	98-97#	98-100#	98-102#	98-104#	98-106#	98-108#	98-111#	98-113#	98-115#	98-117#
		98-120#	98-122#	98-124#	98-126#	98-128#	98-131#	98-133#	98-165#	98-186#					
I\$PROT	84-153#	136-9#													
I\$PTAB	84-153#	157-14	157-14#	157-24	157-24#										
I\$PWR	84-153#														
I\$RPT	84-153#	135-10#	135-107#												
I\$SEG	84-153#	143-10	144-6	145-8	146-7	147-8	148-7	149-9	150-7	151-7	152-3	153-3			
I\$SETU	84-153#	157-13	157-13#	157-14	157-25	157-25#									
I\$SFT	155-12#	155-55#													
I\$SRV	84-153#	126-11#	126-13#	126-32#	126-35#	126-42#	126-44#	126-50#	126-54#						
I\$SUB	84-153#	143-10	144-6	144-6#	144-19	144-19#	144-19#	145-8	145-8#	145-17	145-17#	145-17#	146-7	146-7#	
		146-34	146-34#	146-34#	147-8	147-8#	147-12	147-12#	147-12#	148-7	148-7#	148-86	148-86#	148-86#	149-9
		149-9#	149-15	149-15#	149-15#	150-7	150-7#	150-18	150-18#	150-18#	151-7	151-7#	151-32	151-32#	151-32#
		152-3	153-3												
I\$TST	84-153#	143-10	143-10#	143-26	144-6	145-8	146-7	147-8	148-7	149-9	150-7	151-7	151-128	151-128#	
		151-128#	152-3	152-3#	152-23	152-23#	152-23#	153-3	153-3#	153-17	153-17#	153-17#			
IBE	88-8#														
ICONT	96-22#	137-62													
IDU	88-8#	108-41													
IER	88-8#														
I\$FLAGS	96-20#	137-49#	137-55#	137-61#	137-62#										
INIT1	137-50	137-56	137-125#												
INIT2	137-154#														
INIT3	137-168#														
INIT4	137-181#														
INIT6	137-344#														
INITBL	96-89#	129-45													
INITXX	137-118	137-351#													
INTRCV	96-31#	126-43#	148-24#	148-34	148-47#	148-63									
INTSRV	126-42#	148-19													
INTSTO	97-52#	148-17													
INTST1	97-53#	148-82													
IPADRS	96-35#	132-16	137-171	137-227	137-228										
IREST	96-23#	137-55													
ISR	88-8#														
ISTRT	96-24#	137-49													
ISTRTH	96-25#	137-61													
IXE	88-8#														
J\$JMP	84-153#	135-93	141-10	142-12											
KW.BRL	96-66#	137-136#													
KW.CSR	96-65#	102-37	102-51	125-28	126-53#	130-31	132-22	132-24#	133-14	134-15	137-131#	137-135#	137-141#	148-38	
KW.EL	96-69#	102-39	102-42	102-53	102-56	125-30	125-33	126-51#	126-52#	127-29	127-30	127-31	130-33	130-36	

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page S-9  
 Cross reference table (CREF V05.01)

	133-17	133-18	134-17	134-20	137-125*	137-126*	148-40	148-43
KW.HZ	96-68#	127-16	133-19	137-138*				
KW.OUT	88-13	126-53	132-24	137-141				
KW.VEC	96-67#	137-137*	137-140					
KW11I	126-50#	137-140						
L\$ACP	84-201#							
L\$APT	84-201#							
L\$AU	142-9#							
L\$AUT	84-201#							
L\$AUTO	84-201	139-10#						
L\$CCP	84-201#							
L\$CLEA	84-201	140-8#						
L\$CO	84-201#							
L\$DEPO	84-201#							
L\$DESC	84-201	97-19#						
L\$DESP	84-201#							
L\$DEVP	84-201#							
L\$DISP	84-201	85-10#						
L\$DLY	84-201#							
L\$DTP	84-201#							
L\$DTYP	84-201#							
L\$DU	141-8#							
L\$DUT	84-201#							
L\$DVTY	84-201	97-13#						
L\$EF	84-201#							
L\$ENVI	84-201#							
L\$ERRT	84-201	96-8#						
L\$ETP	84-201#							
L\$EXP1	84-201#							
L\$EXP4	84-201#							
L\$EXP5	84-201#							
L\$HARD	84-201	154-11	154-11#					
L\$HIME	84-201#	104-55						
L\$HPCP	84-201#							
L\$HPTP	84-201#							
L\$HW	84-201	86-12	86-12#					
L\$ICP	84-201#							
L\$INIT	84-201	137-44#						
L\$LADP	84-201#							
L\$LAST	84-201	156-17#	157-25					
L\$LOAD	84-201#							
L\$LUN	84-201#	101-25*	102-15*	112-31*	119-17*	143-19*		
L\$MREV	84-201#							
L\$NAME	84-201#							
L\$PRIO	84-201#							
L\$PROT	84-201	136-9#						
L\$PRT	84-201#							
L\$REPP	84-201#							
L\$REV	84-201#							
L\$RPT	84-201	135-10#						
L\$SOFT	84-201	155-12	155-12#					
L\$SPC	84-201#							
L\$SPCP	84-201#							
L\$SPTP	84-201#							
L\$STA	84-201#							
L\$SW	84-201	87-11	87-11#					

CZUDHAO UDA50A/KDA50-G BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 5-10  
 Cross reference table (CREF V05.01)

L\$TEST	84-2010		
L\$TIML	84-2010		
L\$UNIT	84-2010	137-113	137-156 137-299
L10000	86-12	86-230	
L10001	87-11	87-210	
L10002	98-120		
L10003	98-170		
L10004	98-210		
L10005	98-250		
L10006	98-290		
L10007	98-340		
L10010	98-390		
L10011	98-530		
L10012	98-580		
L10013	98-730		
L10014	98-770		
L10015	98-810		
L10016	98-850		
L10017	98-890		
L10020	98-930		
L10021	98-970		
L10022	98-1020		
L10023	98-1060		
L10024	98-1110		
L10025	98-1150		
L10026	98-1200		
L10027	98-1240		
L10030	98-1280		
L10031	98-1330		
L10032	98-1860		
L10033	126-130		
L10034	126-350		
L10035	126-440		
L10036	126-540		
L10037	135-93	135-1070	
L10041	137-353	138-420	
L10042	139-120		
L10043	140-14	140-160	
L10044	141-10	141-120	
L10045	142-12	142-140	
L10046	143-26	151-1280	
L10047	144-190		
L10050	145-170		
L10051	146-340		
L10052	147-120		
L10053	148-860		
L10054	149-150		
L10055	150-180		
L10056	151-320		
L10057	152-230		
L10060	153-170		
L10061	154-11	154-410	
L10062	155-12	155-550	
L10063	157-140		
L10065	157-14	157-240	
LDDM	101-230	101-34	















CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 5-17  
 Cross reference table (CREF V05.01)

	138-19	138-19	138-19	138-19	138-19	138-19	138-19	138-19	138-21	138-21	138-28	138-28	138-28	138-28
	138-28	138-28	138-28	138-28	138-30	138-30	138-37	138-37	138-37	138-37	138-37	138-37	138-37	138-37
	138-39	138-39	138-42	138-42	139-12	139-12	140-14	140-14	140-14	140-14	140-14	140-16	140-16	141-10
	141-10	141-10	141-12	141-12	142-12	142-12	142-12	142-12	142-12	142-14	142-14	143-26	143-26	143-26
	144-6	144-6	144-8	144-8	144-8	144-8	144-8	144-8	144-8	144-8	144-8	144-8	144-8	144-8
	144-12	144-12	144-12	144-12	144-15	144-15	144-15	144-15	144-15	144-15	144-15	144-15	144-15	144-16
	144-19	144-19	145-8	145-8	145-17	145-17	146-7	146-7	146-7	146-7	146-29	146-29	146-29	146-29
	146-29	146-29	146-34	146-34	147-8	147-8	147-12	147-12	147-12	147-12	148-7	148-7	148-19	148-19
	148-19	148-19	148-19	148-19	148-19	148-19	148-19	148-19	148-19	148-20	148-20	148-20	148-20	148-36
	148-48	148-48	148-48	148-48	148-61	148-61	148-61	148-61	148-61	148-67	148-67	148-67	148-67	148-67
	148-67	148-67	148-70	148-70	148-70	148-70	148-77	148-77	148-77	148-77	148-77	148-77	148-77	148-77
	148-80	148-80	148-80	148-80	148-85	148-85	148-85	148-85	148-85	148-86	148-86	149-9	149-9	149-15
	150-7	150-7	150-18	150-18	151-7	151-7	151-14	151-14	151-14	151-14	151-14	151-14	151-14	151-14
	151-14	151-14	151-14	151-14	151-32	151-32	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94
	151-128	151-128	152-23	152-23	153-17	153-17	154-11	154-11	154-11	154-27	154-27	154-27	154-27	154-27
	154-27	154-27	154-30	154-30	154-30	154-30	154-30	154-30	154-30	154-30	154-30	154-32	154-32	154-32
	154-32	154-32	154-32	154-32	154-32	154-32	154-35	154-35	154-35	154-35	154-35	154-35	154-35	154-35
	154-35	154-35	154-41	154-41	155-12	155-12	155-32	155-32	155-32	155-32	155-32	155-32	155-32	155-32
	156-17	156-17	156-17	156-17	156-17	156-17	157-14	157-14	157-14	157-14	157-14	157-14	157-14	157-14
SVCSUB	84-153#	84-161#	144-6	144-6	144-6	145-8	145-8	145-8	145-8	146-7	146-7	146-7	147-8	147-8
	148-7	148-7	148-7	149-9	149-9	149-9	150-7	150-7	150-7	150-7	151-7	151-7	151-7	151-7
SVCTAG	84-153#	84-163#	86-23	86-23	86-23	87-21	87-21	87-21	87-21	98-12	98-12	98-12	98-17	98-17
	98-21	98-21	98-21	98-25	98-25	98-25	98-29	98-29	98-29	98-34	98-34	98-34	98-39	98-39
	98-39	98-53	98-53	98-53	98-58	98-58	98-58	98-73	98-73	98-73	98-73	98-77	98-77	98-81
	98-81	98-81	98-85	98-85	98-85	98-89	98-89	98-89	98-89	98-93	98-93	98-93	98-97	98-97
	98-102	98-102	98-102	98-106	98-106	98-106	98-111	98-111	98-111	98-115	98-115	98-115	98-120	98-120
	98-120	98-124	98-124	98-124	98-128	98-128	98-128	98-133	98-133	98-133	98-133	98-186	98-186	105-66
	105-66	105-66	126-13	126-13	126-13	126-35	126-35	126-35	126-35	126-44	126-44	126-44	126-54	126-54
	135-107	135-107	135-107	138-42	138-42	138-42	139-12	139-12	139-12	140-16	140-16	140-16	141-12	141-12
	141-12	142-14	142-14	142-14	144-19	144-19	144-19	145-17	145-17	145-17	146-34	146-34	146-34	147-12
	147-12	147-12	148-86	148-86	148-86	149-15	149-15	149-15	149-15	150-18	150-18	150-18	151-32	151-32
	151-128	151-128	151-128	152-23	152-23	152-23	153-17	153-17	153-17	153-17	154-41	154-41	154-41	155-55
	155-55	157-14	157-14	157-14	157-24	157-24	157-24	157-24	157-24	157-24	157-24	157-24	157-24	157-24
SVCTST	84-153#	84-160#	143-10	143-10	143-10	152-3	152-3	152-3	152-3	153-3	153-3	153-3	153-3	153-3
T\$AU	142-9#	142-12	142-14											
T\$AUT	139-10#	139-12												
T\$CLE	140-8#	140-14	140-16											
T\$DAT	157-14	157-14#	157-24											
T\$DU	141-8#	141-10	141-12											
T\$HAR	154-11	154-11#	154-41											
T\$HW	86-12	86-12#	86-23											
T\$INI	137-44#	137-353	138-42											
T\$MSG	98-10#	98-12	98-15#	98-17	98-19#	98-21	98-23#	98-25	98-27#	98-29	98-32#	98-34	98-37#	98-39
	98-42#	98-53	98-55#	98-58	98-60#	98-73	98-75#	98-77	98-79#	98-81	98-83#	98-85	98-87#	98-89
	98-91#	98-93	98-95#	98-97	98-100#	98-102	98-104#	98-106	98-108#	98-111	98-113#	98-115	98-117#	98-120
	98-122#	98-124	98-126#	98-128	98-131#	98-133	98-165#	98-186						
T\$PC	157-13#	157-25												
T\$PRO	136-9#													
T\$PTA	157-13#	157-14	157-14#											
T\$RPT	135-10#	135-93	135-107											
T\$SOF	155-12	155-12#	155-55											
T\$SRV	126-11#	126-13	126-32#	126-35	126-42#	126-44	126-50#	126-54						
T\$SUB	144-6#	144-19	145-8#	145-17	146-7#	146-34	147-8#	147-12	148-7#	148-86	149-9#	149-15	150-7#	150-18
	151-7#	151-32												
T\$SW	87-11	87-11#	87-21											
T\$TES	143-10#	143-26	151-128	152-3#	152-23	153-3#	153-17							

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 5-18  
Cross reference table (CREF V05.01)

T\$ARGC	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	99-78	99-78	99-78	99-78	99-78	99-78	99-80	99-80	99-80	99-80	99-80	99-82
	99-82	99-82	99-82	99-82	99-84	99-84	99-84	99-84	99-84	99-84	99-84	99-84	99-84	99-84	99-82
T\$CODE	105-66	105-66	105-66	105-66	105-66	105-66	105-66	154-27	154-27	154-27	154-27	154-27	154-27	154-30	154-30
	154-30	154-30	154-30	154-30	154-32	154-32	154-32	154-32	154-32	154-32	154-32	154-32	154-32	154-35	154-35
	154-35	154-35	155-32	155-32	155-32	155-32	155-32	155-32	155-32	155-32	155-32	155-32	155-32	154-35	154-35
T\$ERRN	84-153	99-9	99-9	102-26	102-26	102-26	102-44	102-44	102-94	102-94	102-112	102-112	102-112	112-25	112-25
	120-91	121-35	121-35	125-35	125-35	125-35	125-44	125-44	128-76	128-76	128-96	128-96	128-96	128-122	128-122
	128-128	128-134	128-134	129-36	129-36	129-36	129-93	129-93	130-40	130-40	130-51	130-51	130-51	134-22	134-22
	137-103	138-9	138-9	138-19	138-19	138-19	138-28	138-28	138-37	138-37	144-15	144-15	144-15	146-29	146-29
	148-67	148-77	148-77	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94
T\$EXCP	105-66	105-66	154-27	154-27	154-30	154-30	154-30	154-32	154-32	154-32	154-35	154-35	154-35	154-35	154-35
T\$FLAG	135-93	135-93	135-93	137-353	137-353	137-353	137-353	137-353	137-353	140-14	140-14	140-14	140-14	141-10	141-10
	142-12	142-12	142-12	143-26	143-26	143-26	143-26	143-26	143-26	143-26	143-26	143-26	143-26	143-26	143-26
T\$FREE	156-17	157-25	157-25	157-25	157-25	157-25	157-25	157-25	157-25	157-25	157-25	157-25	157-25	157-25	157-25
T\$GMAN	84-153	105-66	105-66	105-66	105-66	105-66	105-66	105-66	105-66	105-66	105-66	105-66	105-66	105-66	105-66
T\$HILI	105-66	105-66	154-27	154-27	154-30	154-30	154-30	154-32	154-32	154-32	154-35	154-35	154-35	154-35	154-35
T\$LAST	84-153	156-17	157-13	157-13	157-13	157-13	157-13	157-13	157-13	157-13	157-13	157-13	157-13	157-13	157-13
T\$LOLI	105-66	105-66	154-27	154-27	154-30	154-30	154-30	154-32	154-32	154-32	154-35	154-35	154-35	154-35	154-35
T\$LSYM	84-153	84-153	86-23	87-21	98-12	98-12	98-17	98-21	98-25	98-29	98-34	98-34	98-34	98-39	98-53
	98-77	98-81	98-85	98-89	98-93	98-93	98-97	98-102	98-106	98-111	98-115	98-115	98-115	98-120	98-124
	98-186	126-13	126-35	126-44	126-54	126-54	135-107	138-42	139-12	140-16	141-12	141-12	141-12	142-14	144-19
	147-12	148-86	149-15	150-18	151-32	151-32	151-128	152-23	153-17	154-41	155-55	155-55	155-55	155-55	155-55
T\$LTNO	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17	156-17
T\$NEST	84-153	86-12	86-12	86-12	86-23	86-23	86-23	86-23	86-23	87-11	87-11	87-11	87-11	87-21	87-21
	87-21	98-10	98-10	98-10	98-12	98-12	98-12	98-12	98-12	98-15	98-15	98-15	98-15	98-17	98-17
	98-17	98-19	98-19	98-19	98-21	98-21	98-21	98-21	98-21	98-23	98-23	98-23	98-23	98-25	98-25
	98-25	98-27	98-27	98-27	98-29	98-29	98-29	98-29	98-29	98-32	98-32	98-32	98-32	98-34	98-34
	98-34	98-37	98-37	98-37	98-39	98-39	98-39	98-39	98-39	98-42	98-42	98-42	98-42	98-53	98-53
	98-53	98-55	98-55	98-55	98-58	98-58	98-58	98-58	98-58	98-60	98-60	98-60	98-60	98-73	98-73
	98-73	98-75	98-75	98-75	98-77	98-77	98-77	98-77	98-77	98-79	98-79	98-79	98-79	98-81	98-81
	98-81	98-83	98-83	98-83	98-85	98-85	98-85	98-85	98-85	98-87	98-87	98-87	98-87	98-89	98-89
	98-89	98-91	98-91	98-91	98-93	98-93	98-93	98-93	98-93	98-95	98-95	98-95	98-95	98-97	98-97
	98-97	98-100	98-100	98-100	98-102	98-102	98-102	98-102	98-102	98-104	98-104	98-104	98-104	98-106	98-106
	98-106	98-108	98-108	98-108	98-111	98-111	98-111	98-111	98-111	98-113	98-113	98-113	98-113	98-115	98-115
	98-115	98-117	98-117	98-117	98-120	98-120	98-120	98-120	98-120	98-122	98-122	98-122	98-122	98-124	98-124
	98-124	98-126	98-126	98-126	98-128	98-128	98-128	98-128	98-128	98-131	98-131	98-131	98-131	98-133	98-133
	98-133	98-165	98-165	98-165	98-186	98-186	98-186	98-186	98-186	126-11	126-11	126-11	126-11	126-13	126-13
	126-13	126-32	126-32	126-32	126-35	126-35	126-35	126-35	126-35	126-42	126-42	126-42	126-42	126-44	126-44
	126-44	126-50	126-50	126-50	126-54	126-54	126-54	126-54	126-54	135-10	135-10	135-10	135-10	135-107	135-107
	135-107	136-9	136-9	136-9	136-15	136-15	136-15	136-15	136-15	137-44	137-44	137-44	137-44	138-42	138-42
	138-42	139-10	139-10	139-10	139-12	139-12	139-12	139-12	139-12	140-8	140-8	140-8	140-8	140-16	140-16
	140-16	141-8	141-8	141-8	141-12	141-12	141-12	141-12	141-12	142-9	142-9	142-9	142-9	142-14	142-14
	142-14	143-10	143-10	143-10	144-6	144-6	144-6	144-6	144-6	144-19	144-19	144-19	144-19	145-8	145-8
	145-17	145-17	145-17	145-17	146-7	146-7	146-7	146-7	146-7	146-34	146-34	146-34	146-34	147-8	147-8
	147-12	147-12	147-12	147-12	148-7	148-7	148-7	148-7	148-7	148-86	148-86	148-86	148-86	149-9	149-9
	149-15	149-15	149-15	149-15	150-7	150-7	150-7	150-7	150-7	150-18	150-18	150-18	150-18	151-7	151-7
	151-32	151-32	151-32	151-32	151-128	151-128	151-128	151-128	151-128	152-3	152-3	152-3	152-3	152-23	152-23
	152-23	153-3	153-3	153-3	153-17	153-17	153-17	153-17	153-17	154-11	154-11	154-11	154-11	154-41	154-41
	154-41	155-12	155-12	155-12	155-55	155-55	155-55	155-55	155-55	155-55	155-55	155-55	155-55	155-55	155-55
T\$NSO	86-12	86-23	87-11	87-21	98-10	98-12	98-15	98-17	98-19	98-21	98-23	98-25	98-27	98-29	98-29
	98-32	98-34	98-37	98-39	98-42	98-53	98-55	98-58	98-60	98-73	98-75	98-77	98-79	98-81	98-81
	98-83	98-85	98-87	98-89	98-91	98-93	98-95	98-97	98-100	98-102	98-104	98-106	98-108	98-111	98-111
	98-113	98-115	98-117	98-120	98-122	98-124	98-126	98-128	98-131	98-133	98-165	98-186	126-11	126-13	126-13
	126-32	126-35	126-42	126-44	126-50	126-54	135-10	135-107	136-9	136-15	137-44	138-42	139-10	139-12	139-12
	140-8	140-16	141-8	141-12	142-9	142-14	143-10	143-10	151-128	152-3	153-3	154-11	154-41	154-41	154-41







CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page 5-21  
 Cross reference table (CREF V05.01)

UF.RPL	91-102#				
UF.SCH	91-104#				
UF.SCL	91-105#				
UF.WBN	91-106#				
UF.WPH	91-107#				
UF.WPS	91-108#				
URNING	96-29#	101-17*	101-31*	101-38	102-71*
URUN	96-28#	101-16*	101-22	102-11	
UTOT1	107-21#	107-35			
UTOT1A	107-24	107-34#			
UTOT2	107-22	107-36#			
WAITMS	120-79	121-24	125-12#		
WCHNG	134-8#	146-16	146-24		
WCHNGD	96-100#	134-12	146-14*	146-22*	
X\$ALWA	84-153#				
X\$FALS	84-153#				
X\$OFFS	84-153#				
X\$TRUE	84-153#				
X1	97-90#	98-11			
X14	97-100#	98-38			
X1A	97-85#	98-11			
X2	97-92#	98-16			
X22	97-106#	98-57			
X23A	97-108#	98-61			
X23B	97-113#	98-65			
X24	97-114#	98-76			
X25	97-116#	98-80			
X26	97-118#	98-84			
X27	97-121#	98-88			
X28	97-123#	98-92			
X29	97-124#	98-96			
X2A	97-86#	98-16			
X3	97-93#	98-20			
X31	97-127#	98-105			
X32	97-129#	98-109			
X35	97-130#	98-118			
X36	97-131#	98-123			
X38	97-133#	98-127			
X39	97-138#	98-132			
X3A	97-87#	98-20			
X4	97-94#	98-24			
X6	97-96#	98-28			
X8	97-98#	98-33			
X8A	97-88#	98-33			
XFCE	97-146#	98-43	98-101		
XFRU	97-154#	98-51	98-72	111-106	
XFRUA	97-155#	98-48			
XMSG1	97-147#	98-155			
XMSG2	97-148#	98-159			
XPKT1	97-149#	98-141			
XPKT2	97-152#	98-148			
XSA	97-153#	111-118			



CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page M-2  
 Cross reference table (CREF V05.01)

ENDSET	1-5550	84-1530	157-25													
ENDSFT	1-5680	84-1530	155-55													
ENDSRV	1-5800	84-1530	126-13	126-35	126-44	126-54										
ENDSUB	1-5960	84-1530	144-19	145-17	146-34	147-12	148-86	149-15	150-18	151-32						
ENDSW	1-6140	84-1530	87-21													
ENDTST	1-6240	84-1530	151-128	152-23	153-17											
EQUALS	1-6420	84-1530	88-8													
ERRDF	1-7140	84-1530	102-26	102-44	102-94	102-112	112-25	120-91	121-35	125-35	125-44	128-76	128-96	128-122		
	129-36	129-93	130-40	130-51	134-22	144-15	146-29	148-67	148-77	151-94						
ERRHRD	1-7180	84-1530														
ERROR	1-7220	84-1530	119-41													
ERRSF	1-7260	84-1530	99-9	128-128	128-134	137-103	138-9	138-19	138-28	138-37						
ERRSOF	1-7300	84-1530														
ERRTBL	1-7340	84-1530	96-8													
ESCAPE	1-7440	84-1530														
EXIT	1-7710	84-1530	135-93	137-353	140-14	141-10	142-12	143-26								
FEQUAL	1-8100	84-1530														
GETBYT	1-8240	84-1530														
GETPRI	1-8340	84-1530														
GETWOR	1-8290	84-1530														
GMANIA	1-8390	84-1530														
GMANID	1-8480	84-1530	105-66													
GMANIL	1-8590	84-1530														
GPHARD	1-8680	84-1530	137-93	137-184												
GPRMA	1-8740	84-1530	154-27	154-30												
GPRMD	1-9030	84-1530	105-66	105-660	154-32	154-35										
GPRML	1-9340	84-1530	155-32													
HEADER	1-9540	84-1530	84-201													
INLOOP	1-9620	84-1530														
IOSETU	1-9660	84-1530														
IOSTAR	1-9740	84-1530														
ITEM	84-750	94-30	94-31	94-32	94-33	94-34	94-35	94-36	94-37	94-38	94-39	94-40	94-41	94-42		
	94-43	95-34	95-35	154-14	154-16	154-17	154-19	155-18								
KT11	1-9820	84-1530														
LASTAD	1-;470	84-1530	156-17													
M\$BYTE	1-D000	84-1530	84-201	84-201	84-201	84-2010										
M\$CHEC	1-E180	84-1530	135-93	135-930	137-353	137-3530	140-14	140-140	141-10	141-100	142-12	142-120	143-26	143-260		
M\$CNT0	1-E820	84-1530	105-66	105-660	154-27	154-270	154-30	154-300	154-32	154-320	154-35	154-350	155-32	155-320		
M\$COUN	1-D660	84-1530	99-78	99-780	99-80	99-800	99-82	99-820	99-84	99-840						
M\$DATA	1-B670	84-1530	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	97-130	97-19	97-190													
M\$DECR	1-D290	84-1530	86-23	86-230	87-21	87-210	98-12	98-120	98-17	98-170	98-21	98-210	98-25	98-250		
	98-29	98-290	98-34	98-340	98-39	98-390	98-53	98-530	98-58	98-580	98-73	98-730	98-77	98-770		
	98-81	98-810	98-85	98-850	98-89	98-890	98-93	98-930	98-97	98-970	98-102	98-1020	98-106	98-1060		
	98-111	98-1110	98-115	98-1150	98-120	98-1200	98-124	98-1240	98-128	98-1280	98-133	98-1330	98-186	98-1860		
	126-13	126-130	126-35	126-350	126-44	126-440	126-54	126-540	135-107	135-1070	136-15	136-150	138-42	138-420		
	139-12	139-120	140-16	140-160	141-12	141-120	142-14	142-140	144-19	144-190	145-17	145-170	146-34	146-340		
	147-12	147-120	148-86	148-860	149-15	149-150	150-18	150-180	151-32	151-320	151-128	151-1280	152-23	152-230		
	153-17	153-170	154-41	154-410	155-55	155-550	157-14	157-140								
M\$DEFA	1-E700	84-1530	105-66	105-660	154-27	154-270	154-30	154-300	154-32	154-320	154-35	154-350	155-32	155-320		
M\$ENDE	1-D740	84-1530	86-230	87-210	98-120	98-170	98-210	98-250	98-290	98-340	98-390	98-530	98-580	98-730		
	98-770	98-810	98-850	98-890	98-930	98-970	98-1020	98-1060	98-1110	98-1150	98-1200	98-1240	98-1280	98-1330		
	98-1860	126-130	126-350	126-440	126-540	135-1070	138-420	139-120	140-160	141-120	142-140	144-190	145-170	146-340		
	147-120	148-860	149-150	150-180	151-320	151-1280	152-230	153-170	154-410	155-550						

M\$ERRI	1-249#	84-153#	99-9	99-9#	102-26	102-26#	102-44	102-44#	102-94	102-94#	102-112	102-112#	112-25	112-25#
	120-91	120-91#	121-35	121-35#	125-35	125-35#	125-44	125-44#	128-76	128-76#	128-96	128-96#	128-122	128-122#
	128-128	128-128#	128-134	128-134#	129-36	129-36#	129-93	129-93#	130-40	130-40#	130-51	130-51#	134-22	134-22#
	137-103	137-103#	138-9	138-9#	138-19	138-19#	138-28	138-28#	138-37	138-37#	144-15	144-15#	146-29	146-29#
	148-67	148-67#	148-77	148-77#	151-94	151-94#								
M\$ESCA	1-D06#	84-153#												
M\$ESCS	1-D10#	84-153#												
M\$EXCP	1-E01#	84-153#	105-66	105-66	105-66#	154-27	154-27	154-27#	154-30	154-30	154-30#	154-32	154-32	154-32#
	154-35	154-35	154-35#											
M\$EXIT	1-D14#	84-153#	135-93#	137-353	137-353#	140-14	140-14#	141-10#	142-12#	143-26	143-26#			
M\$EXSE	1-D22#	84-153#	135-93#	137-353#	140-14#	141-10#	142-12#	143-26#						
M\$EXTJ	1-D18#	84-153#	135-93	135-93#	137-353#	140-14#	141-10	141-10#	142-12	142-12#	143-26#			
M\$GEN	1-D38#	84-153#	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	86-23	86-23#	87-11	87-11	87-11#	87-11#	87-21	87-21#	96-8	96-8#	97-13	97-13#	97-19	97-19#
	98-10	98-10#	98-12	98-12#	98-15	98-15#	98-17	98-17#	98-19	98-19#	98-21	98-21#	98-23	98-23#
	98-25	98-25#	98-27	98-27#	98-29	98-29#	98-32	98-32#	98-34	98-34#	98-37	98-37#	98-39	98-39#
	98-42	98-42#	98-53	98-53#	98-55	98-55#	98-58	98-58#	98-60	98-60#	98-73	98-73#	98-75	98-75#
	98-77	98-77#	98-79	98-79#	98-81	98-81#	98-83	98-83#	98-85	98-85#	98-87	98-87#	98-89	98-89#
	98-91	98-91#	98-93	98-93#	98-95	98-95#	98-97	98-97#	98-100	98-100#	98-102	98-102#	98-104	98-104#
	98-106	98-106#	98-108	98-108#	98-111	98-111#	98-113	98-113#	98-115	98-115#	98-117	98-117#	98-120	98-120#
	98-122	98-122#	98-124	98-124#	98-126	98-126#	98-128	98-128#	98-131	98-131#	98-133	98-133#	98-165	98-165#
	98-186	98-186#	105-66	105-66#	126-11	126-11#	126-13	126-13#	126-32	126-32#	126-35	126-35#	126-42	126-42#
	126-44	126-44#	126-50	126-50#	126-54	126-54#	135-10	135-10#	135-107	135-107#	136-9	136-9#	137-44	137-44#
	138-42	138-42#	139-10	139-10#	139-12	139-12#	140-8	140-8#	140-16	140-16#	141-8	141-8#	141-12	141-12#
	142-9	142-9#	142-14	142-14#	143-10	143-10#	144-6	144-6#	144-19	144-19#	145-8	145-8#	145-17	145-17#
	146-7	146-7#	146-34	146-34#	147-8	147-8#	147-12	147-12#	148-7	148-7#	148-86	148-86#	149-9	149-9#
	149-15	149-15#	150-7	150-7#	150-18	150-18#	151-7	151-7#	151-32	151-32#	151-128	151-128#	152-3	152-3#
	152-23	152-23#	153-3	153-3#	153-17	153-17#	154-11	154-11#	154-41	154-41#	155-12	155-12#	155-55	155-55#
	156-17	156-17#	157-14	157-14#	157-24	157-24#								
M\$GENB	1-C38#	84-153#	105-66	105-66#										
M\$GETS	1-D35#	84-153#	86-23	86-23#	87-21	87-21#	98-12	98-12#	98-17	98-17#	98-21	98-21#	98-25	98-25#
	98-29	98-29#	98-34	98-34#	98-39	98-39#	98-53	98-53#	98-58	98-58#	98-73	98-73#	98-77	98-77#
	98-81	98-81#	98-85	98-85#	98-89	98-89#	98-93	98-93#	98-97	98-97#	98-102	98-102#	98-106	98-106#
	98-111	98-111#	98-115	98-115#	98-120	98-120#	98-124	98-124#	98-126	98-126#	98-133	98-133#	98-186	98-186#
	126-13	126-13#	126-35	126-35#	126-44	126-44#	126-54	126-54#	135-107	135-107#	136-15	136-15#	138-42	138-42#
	139-12	139-12#	140-16	140-16#	141-12	141-12#	142-14	142-14#	144-19	144-19#	145-17	145-17#	146-34	146-34#
	147-12	147-12#	148-86	148-86#	149-15	149-15#	150-18	150-18#	151-32	151-32#	151-128	151-128#	152-23	152-23#
	153-17	153-17#	154-41	154-41#	155-55	155-55#								
M\$GETT	1-B77#	84-153#	135-93#	137-353#	140-14#	141-10#	142-12#	143-26#						
M\$GNGB	1-C02#	84-153#	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	87-11	87-11#	96-8	96-8#	97-13	97-13#	97-19	97-19#	98-10	98-10#	98-15	98-15#	98-19	98-19#
	98-23	98-23#	98-27	98-27#	98-32	98-32#	98-37	98-37#	98-42	98-42#	98-55	98-55#	98-60	98-60#
	98-75	98-75#	98-79	98-79#	98-83	98-83#	98-87	98-87#	98-91	98-91#	98-95	98-95#	98-100	98-100#
	98-104	98-104#	98-108	98-108#	98-113	98-113#	98-117	98-117#	98-122	98-122#	98-126	98-126#	98-131	98-131#
	98-165	98-165#	126-11	126-11#	126-32	126-32#	126-42	126-42#	126-50	126-50#	135-10	135-10#	136-9	136-9#
	137-44	137-44#	139-10	139-10#	140-8	140-8#	141-8	141-8#	142-9	142-9#	154-11	154-11#	155-12	155-12#



	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-128	151-128	152-23	152-23	153-17
	153-17	154-11	154-11	154-27	154-27	154-27	154-27	154-30	154-30	154-30	154-30	154-30	154-30	154-32
	154-32	154-32	154-32	154-32	154-32	154-35	154-35	154-35	154-35	154-35	154-35	154-41	154-41	155-12
	155-12	155-32	155-32	155-32	155-32	155-55	155-55	156-17	156-17	156-17	156-17	157-14	157-14	157-14
	157-14													
M\$GNLS	1-C13	84-153	105-66	105-66										
M\$GNSU	1-B98	84-153	144-6	144-6	145-8	145-8	146-7	146-7	147-8	147-8	148-7	148-7	149-9	149-9
	150-7	150-7	151-7	151-7										
M\$GNTA	1-B90	84-153	86-23	86-23	87-21	87-21	98-12	98-12	98-17	98-17	98-21	98-21	98-25	98-25
	98-29	98-29	98-34	98-34	98-39	98-39	98-53	98-53	98-58	98-58	98-73	98-73	98-77	98-77
	98-81	98-81	98-85	98-85	98-89	98-89	98-93	98-93	98-97	98-97	98-102	98-102	98-106	98-106
	98-111	98-111	98-115	98-115	98-120	98-120	98-124	98-124	98-128	98-128	98-133	98-133	98-186	98-186
	126-13	126-13	126-35	126-35	126-44	126-44	126-54	126-54	135-107	135-107	138-42	138-42	139-12	139-12
	140-16	140-16	141-12	141-12	142-14	142-14	144-19	144-19	145-17	145-17	146-34	146-34	147-12	147-12
	148-86	148-86	149-15	149-15	150-18	150-18	151-32	151-32	151-128	151-128	152-23	152-23	153-17	153-17
	154-41	154-41	155-55	155-55	157-14	157-14	157-24	157-24						
M\$GNTE	1-B94	84-153	143-10	143-10	152-3	152-3	153-3	153-3						
M\$HAPT	1-A39	84-153	84-201	84-201										
M\$HNAP	1-B24	84-153	84-201	84-201										
M\$INCR	1-D26	84-153	86-12	86-12	86-12	86-12	87-11	87-11	87-11	87-11	98-10	98-10	98-10	98-10
	98-12	98-15	98-15	98-15	98-15	98-17	98-19	98-19	98-19	98-19	98-21	98-23	98-23	98-23
	98-23	98-25	98-27	98-27	98-27	98-27	98-29	98-29	98-32	98-32	98-32	98-34	98-37	98-37
	98-37	98-37	98-39	98-42	98-42	98-42	98-42	98-42	98-53	98-55	98-55	98-55	98-53	98-60
	98-60	98-60	98-60	98-73	98-75	98-75	98-75	98-75	98-75	98-77	98-79	98-79	98-79	98-81
	98-83	98-83	98-83	98-83	98-85	98-87	98-87	98-87	98-87	98-89	98-91	98-91	98-91	98-91
	98-93	98-95	98-95	98-95	98-95	98-97	98-100	98-100	98-100	98-100	98-102	98-104	98-104	98-104
	98-104	98-106	98-108	98-108	98-108	98-108	98-111	98-111	98-113	98-113	98-113	98-115	98-117	98-117
	98-117	98-117	98-120	98-122	98-122	98-122	98-122	98-124	98-126	98-126	98-126	98-126	98-128	98-131
	98-131	98-131	98-131	98-133	98-165	98-165	98-165	98-165	98-186	99-9	99-11	99-78	99-80	99-82
	99-84	102-26	102-44	102-47	102-59	102-94	102-112	105-36	105-66	105-66	105-66	108-40	112-25	119-26
	119-41	120-20	120-91	121-35	125-27	125-35	125-44	126-11	126-11	126-11	126-11	126-32	126-32	126-32
	126-32	126-42	126-42	126-42	126-42	126-50	126-50	126-50	126-50	128-76	128-96	128-122	128-128	128-134
	129-17	129-30	129-33	129-36	129-93	130-30	130-40	130-51	132-14	134-14	134-22	135-10	135-10	135-10
	135-10	135-107	136-9	136-9	136-9	136-9	137-44	137-44	137-44	137-44	137-46	137-52	137-58	137-65
	137-93	137-103	137-105	137-127	137-129	137-140	137-146	137-184	137-351	137-353	138-9	138-11	138-19	138-21
	138-28	138-30	138-37	138-39	138-42	139-10	139-10	139-10	139-10	139-12	140-8	140-8	140-8	140-8
	140-14	140-16	141-8	141-8	141-8	141-8	141-12	142-9	142-9	142-9	142-9	142-14	143-10	143-10
	143-10	143-10	143-10	143-10	143-26	144-6	144-6	144-6	144-6	144-6	144-6	144-8	144-12	144-15
	144-16	144-19	145-8	145-8	145-8	145-8	145-8	145-8	145-8	145-17	146-7	146-7	146-7	146-7
	146-7	146-29	146-34	147-8	147-8	147-8	147-8	147-8	147-8	147-12	148-7	148-7	148-7	148-7
	148-7	148-7	148-19	148-20	148-36	148-48	148-61	148-67	148-70	148-77	148-80	148-85	148-86	149-9
	149-9	149-9	149-9	149-9	149-9	149-15	150-7	150-7	150-7	150-7	150-7	150-7	150-18	151-7
	151-7	151-7	151-7	151-7	151-7	151-14	151-32	151-94	151-128	152-3	152-3	152-3	152-3	152-3
	152-3	152-23	153-3	153-3	153-3	153-3	153-3	153-3	153-3	153-17	154-11	154-11	154-11	154-11
	155-12	155-12	155-12	157-13	157-13	157-14	157-14	157-14	157-14					
M\$IOSE	1-A00	84-153												
M\$LDRO	1-C42	84-153	129-33	129-33	137-46	137-46	137-52	137-52	137-58	137-58	137-65	137-65	137-93	137-93
	137-127	137-127	137-129	137-129	137-184	137-184	137-351	137-351	144-12	144-12	148-20	148-20	148-48	148-48
	148-61	148-61	148-70	148-70	148-80	148-80	148-85	148-85						
M\$MASK	1-071	84-153												
M\$MCHI	1-4	84-153	84-153	84-153										
M\$MCLO	1-024	84-153	84-153	84-153										
M\$MSK1	1-077	84-153												
M\$POP	1-B81	84-153	86-23	86-23	87-21	87-21	98-12	98-12	98-17	98-17	98-21	98-21	98-25	98-25
	98-29	98-29	98-34	98-34	98-39	98-39	98-53	98-53	98-58	98-58	98-73	98-73	98-77	98-77
	98-81	98-81	98-85	98-85	98-89	98-89	98-93	98-93	98-97	98-97	98-102	98-102	98-106	98-106

	98-111	98-111#	98-115	98-115#	98-120	98-120#	98-124	98-124#	98-128	98-128#	98-133	98-133#	98-186	98-186#
	126-13	126-13#	126-35	126-35#	126-44	126-44#	126-54	126-54#	135-107	135-107#	136-15	136-15#	138-42	138-42#
	139-12	139-12#	140-16	140-16#	141-12	141-12#	142-14	142-14#	144-19	144-19#	145-17	145-17#	146-34	146-34#
	147-12	147-12#	148-86	148-86#	149-15	149-15#	150-18	150-18#	151-32	151-32#	151-128	151-128#	152-23	152-23#
	153-17	153-17#	154-41	154-41#	155-55	155-55#								
M\$PRIN	1-#36#	84-153#	99-78	99-78#	99-80	99-80#	99-82	99-82#	99-84	99-84#				
M\$PUSH	1-#31#	84-153#	86-12	86-12#	87-11	87-11#	98-10	98-10#	98-15	98-15#	98-19	98-19#	98-23	98-23#
	98-27	98-27#	98-32	98-32#	98-37	98-37#	98-42	98-42#	98-55	98-55#	98-60	98-60#	98-75	98-75#
	98-79	98-79#	98-83	98-83#	98-87	98-87#	98-91	98-91#	98-95	98-95#	98-100	98-100#	98-104	98-104#
	98-108	98-108#	98-113	98-113#	98-117	98-117#	98-122	98-122#	98-126	98-126#	98-131	98-131#	98-165	98-165#
	126-11	126-11#	126-32	126-32#	126-42	126-42#	126-50	126-50#	135-10	135-10#	136-9	136-9#	137-44	137-44#
	139-10	139-10#	140-8	140-8#	141-8	141-8#	142-9	142-9#	143-10	143-10#	144-6	144-6#	145-8	145-8#
	146-7	146-7#	147-8	147-8#	148-7	148-7#	149-9	149-9#	150-7	150-7#	151-7	151-7#	152-3	152-3#
	153-3	153-3#	154-11	154-11#	155-12	155-12#								
M\$PUT	1-C72#	84-153#	99-78	99-78#	99-78	99-78#	99-80	99-80#	99-80	99-80#	99-82	99-82#	99-82	99-82#
	99-84	99-84#	99-84	99-84#	120-20	120-20#	120-20	120-20#	120-20	120-20#	129-30	129-30#	129-30	129-30#
	132-14	132-14#	132-14	132-14#	132-14#	132-14#	137-140	137-140#	137-140	137-140#	137-140#	144-8	144-8#	144-8#
	144-8#	148-19	148-19	148-19#	148-19	148-19#	148-19#	151-14	151-14#	151-14	151-14#	151-14#	151-14#	151-14#
M\$PUT1	1-C81#	84-153#	99-78	99-78#	99-78	99-78#	99-78#	99-78#	99-78#	99-78#	99-80	99-80#	99-80#	99-80#
	99-82	99-82#	99-82	99-82#	99-82#	99-82#	99-84	99-84#	99-84	99-84#	99-84#	99-84#	120-20	120-20#
	120-20	120-20#	120-20#	120-20#	120-20#	120-20#	129-30	129-30#	129-30	129-30#	129-30#	129-30#	129-30#	129-30#
	132-14	132-14#	132-14	132-14#	132-14#	132-14#	132-14#	132-14#	137-140	137-140#	137-140	137-140#	137-140#	137-140#
	137-140#	137-140#	144-8	144-8#	144-8	144-8#	144-8#	144-8#	144-8#	144-8#	148-19	148-19#	148-19	148-19#
	148-19#	148-19#	148-19#	148-19#	151-14	151-14#	151-14	151-14#	151-14#	151-14#	151-14#	151-14#	151-14#	151-14#
M\$RADI	1-D77#	84-153#	105-66	105-66#	154-27	154-27#	154-30	154-30#	154-32	154-32#	154-35	154-35#	155-32	155-32#
M\$RBRO	1-C52#	84-153#												
M\$RNRO	1-C62#	84-153#	108-40	108-40#	137-93	137-93#	137-127	137-127#	137-129	137-129#	137-146	137-146#	137-184	137-184#
M\$SETS	1-D32#	84-153#	86-12	86-12#	87-11	87-11#	98-10	98-10#	98-15	98-15#	98-19	98-19#	98-23	98-23#
	98-27	98-27#	98-32	98-32#	98-37	98-37#	98-42	98-42#	98-55	98-55#	98-60	98-60#	98-75	98-75#
	98-79	98-79#	98-83	98-83#	98-87	98-87#	98-91	98-91#	98-95	98-95#	98-100	98-100#	98-104	98-104#
	98-108	98-108#	98-113	98-113#	98-117	98-117#	98-122	98-122#	98-126	98-126#	98-131	98-131#	98-165	98-165#
	126-11	126-11#	126-32	126-32#	126-42	126-42#	126-50	126-50#	135-10	135-10#	136-9	136-9#	137-44	137-44#
	139-10	139-10#	140-8	140-8#	141-8	141-8#	142-9	142-9#	143-10	143-10#	144-6	144-6#	145-8	145-8#
	146-7	146-7#	147-8	147-8#	148-7	148-7#	149-9	149-9#	150-7	150-7#	151-7	151-7#	152-3	152-3#
	153-3	153-3#	154-11	154-11#	155-12	155-12#								
M\$STAR	1-A33#	84-153#												
M\$SVC	1-C33#	84-153#	98-12	98-12#	98-17	98-17#	98-21	98-21#	98-25	98-25#	98-29	98-29#	98-34	98-34#
	98-39	98-39#	98-53	98-53#	98-58	98-58#	98-73	98-73#	98-77	98-77#	98-81	98-81#	98-85	98-85#
	98-89	98-89#	98-93	98-93#	98-97	98-97#	98-102	98-102#	98-106	98-106#	98-111	98-111#	98-115	98-115#
	98-120	98-120#	98-124	98-124#	98-128	98-128#	98-133	98-133#	98-186	98-186#	99-9	99-9#	99-11	99-11#
	99-78#	99-80	99-80#	99-82	99-82#	99-84	99-84#	102-26	102-44	102-47	102-47#	102-59	102-59#	102-94
	102-112	105-36	105-36#	105-66	105-66#	108-40	108-40#	112-25	119-26	119-26#	119-41	119-41#	120-20	120-20#
	120-91	121-35	125-27	125-27#	125-35	125-44	128-76	128-96	128-122	128-128	128-134	129-17	129-17#	129-30
	129-30#	129-33	129-33#	129-36	129-93	130-30	130-30#	130-40	130-51	132-14	132-14#	134-14	134-14#	134-22
	135-93#	135-107	135-107#	137-46	137-46#	137-52	137-52#	137-58	137-58#	137-65	137-65#	137-93	137-93#	137-103
	137-105	137-105#	137-127	137-127#	137-129	137-129#	137-140	137-140#	137-146	137-146#	137-184	137-184#	137-351	137-351#
	137-353	137-353#	138-9	138-11	138-11#	138-19	138-21	138-21#	138-28	138-30	138-30#	138-37	138-39	138-39#
	138-42	138-42#	139-12	139-12#	140-14	140-14#	140-16	140-16#	141-10#	141-12	141-12#	142-12#	142-14	142-14#
	143-26	143-26#	144-6	144-6#	144-8	144-8#	144-12	144-12#	144-15	144-16	144-16#	144-19	144-19#	145-8
	145-8#	145-17	145-17#	146-7	146-7#	146-29	146-34	146-34#	147-8	147-8#	147-12	147-12#	148-7	148-7#
	148-19	148-19#	148-20	148-20#	148-36	148-36#	148-48	148-48#	148-61	148-61#	148-67	148-70	148-70#	148-77
	148-80	148-80#	148-85	148-85#	148-86	148-86#	149-9	149-9#	149-15	149-15#	150-7	150-7#	150-18	150-18#
	151-7	151-7#	151-14	151-14#	151-32	151-32#	151-94	151-128	151-128#	152-23	152-23#	153-17	153-17#	153-17#
M\$TLAB	1-C29#	84-153#	98-12#	98-17#	98-21#	98-25#	98-29#	98-34#	98-39#	98-53#	98-58#	98-73#	98-77#	98-81#
	98-85#	98-89#	98-93#	98-97#	98-102#	98-106#	98-111#	98-115#	98-120#	98-124#	98-128#	98-133#	98-186#	99-9#
	99-11#	99-78#	99-80#	99-82#	99-84#	102-26#	102-44#	102-47#	102-59#	102-94#	102-112#	105-36#	105-66#	108-40#



	112-25@	119-26@	119-41@	120-20@	120-91@	121-35@	125-27@	125-35@	125-44@	128-76@	128-96@	128-122@	128-128@	128-134@
	129-17@	129-30@	129-33@	129-36@	129-93@	130-30@	130-40@	130-51@	132-14@	134-14@	134-22@	135-107@	137-46@	137-52@
	137-58@	137-65@	137-93@	137-103@	137-105@	137-127@	137-129@	137-140@	137-146@	137-184@	137-351@	137-353@	138-9@	138-11@
	138-19@	138-21@	138-28@	138-30@	138-37@	138-39@	138-42@	139-12@	140-14@	140-16@	141-12@	142-14@	143-26@	144-6@
	144-8@	144-12@	144-15@	144-16@	144-19@	145-8@	145-17@	146-7@	146-29@	146-34@	147-8@	147-12@	148-7@	148-19@
	148-20@	148-36@	148-48@	148-61@	148-67@	148-70@	148-77@	148-80@	148-85@	148-86@	149-9@	149-15@	150-7@	150-18@
	151-7@	151-14@	151-32@	151-94@	151-128@	152-23@	153-17@							
M&TSTL	1-C21@	84-153@	98-12	98-12@	98-17	98-17@	98-21	98-21@	98-25	98-25@	98-29	98-29@	98-34	98-34@
	98-39	98-39@	98-53	98-53@	98-58	98-58@	98-73	98-73@	98-77	98-77@	98-81	98-81@	98-85	98-85@
	98-89	98-89@	98-93	98-93@	98-97	98-97@	98-102	98-102@	98-106	98-106@	98-111	98-111@	98-115	98-115@
	98-120	98-120@	98-124	98-124@	98-128	98-128@	98-133	98-133@	98-186	98-186@	99-9	99-9@	99-9@	99-11
	99-11@	99-78	99-78@	99-80	99-80@	99-82	99-82@	99-84	99-84@	102-26	102-26@	102-26@	102-44	102-44@
	102-44@	102-47	102-47@	102-59	102-59@	102-94	102-94@	102-94@	102-112	102-112@	102-112@	105-36	105-36@	105-66
	105-66@	108-40	108-40@	112-25	112-25@	112-25@	119-26	119-26@	119-41	119-41@	120-20	120-20@	120-91	120-91@
	120-91@	121-35	121-35@	121-35@	125-27	125-27@	125-35	125-35@	125-35@	125-44	125-44@	125-44@	128-76	128-76@
	128-76@	128-96	128-96@	128-96@	128-122	128-122@	128-122@	128-128	128-128@	128-128@	128-134	128-134@	128-134@	129-17
	129-17@	129-30	129-30@	129-33	129-33@	129-36	129-36@	129-36@	129-93	129-93@	129-93@	130-30	130-30@	130-40
	130-40@	130-40@	130-51	130-51@	130-51@	132-14	132-14@	134-14	134-14@	134-22	134-22@	134-22@	135-107	135-107@
	137-46	137-46@	137-52	137-52@	137-58	137-58@	137-65	137-65@	137-93	137-93@	137-103	137-103@	137-103@	137-105
	137-105@	137-127	137-127@	137-129	137-129@	137-140	137-140@	137-146	137-146@	137-184	137-184@	137-351	137-351@	137-353
	137-353@	138-9	138-9@	138-9@	138-11	138-11@	138-19	138-19@	138-19@	138-21	138-21@	138-28	138-28@	138-28@
	138-30	138-30@	138-37	138-37@	138-37@	138-39	138-39@	138-42	138-42@	139-12	139-12@	140-14	140-14@	140-16
	140-16@	141-12	141-12@	142-14	142-14@	143-26	143-26@	144-6	144-6@	144-8	144-8@	144-12	144-12@	144-15
	144-15@	144-15@	144-16	144-16@	144-19	144-19@	145-8	145-8@	145-17	145-17@	146-7	146-7@	146-29	146-29@
	146-29@	146-34	146-34@	147-8	147-8@	147-12	147-12@	148-7	148-7@	148-19	148-19@	148-20	148-20@	148-36
	148-36@	148-48	148-48@	148-61	148-61@	148-67	148-67@	148-67@	148-70	148-70@	148-77	148-77@	148-77@	148-80
	148-80@	148-85	148-85@	148-86	148-86@	149-9	149-9@	149-15	149-15@	150-7	150-7@	150-18	150-18@	151-7
	151-7@	151-14	151-14@	151-32	151-32@	151-94	151-94@	151-94@	151-128	151-128@	152-23	152-23@	153-17	153-17@
M\$WORD	1-C94@	84-153@	84-201	84-201@	85-10	85-10	85-10	85-10	85-10@	99-9	99-9	99-9	99-9@	102-26
	102-26	102-26	102-26@	102-44	102-44	102-44	102-44@	102-94	102-94	102-94	102-94@	102-112	102-112	102-112
	102-112@	105-66	105-66	105-66@	105-66@	112-25	112-25	112-25	112-25@	120-91	120-91	120-91	120-91@	121-35
	121-35	121-35	121-35@	125-35	125-35	125-35	125-35@	125-44	125-44	125-44	125-44@	128-76	128-76	128-76
	128-76@	128-96	128-96	128-96	128-96@	128-122	128-122	128-122	128-122@	128-128	128-128	128-128	128-128@	128-134
	128-134	128-134	128-134@	129-36	129-36	129-36	129-36@	129-93	129-93	129-93	129-93@	130-40	130-40	130-40
	130-40@	130-51	130-51	130-51	130-51@	134-22	134-22	134-22	134-22@	135-93	135-93@	137-103	137-103	137-103
	137-103@	137-353@	138-9	138-9	138-9	138-9@	138-19	138-19	138-19	138-19@	138-28	138-28	138-28	138-28@
	138-37	138-37	138-37	138-37@	140-14@	141-10	141-10@	142-12	142-12@	143-26@	144-15	144-15	144-15	144-15@
	146-29	146-29	146-29	146-29@	148-67	148-67	148-67	148-67@	148-77	148-77	148-77	148-77@	151-94	151-94
	151-94	151-94@	154-27	154-27@	154-30	154-30@	154-32	154-32@	154-35	154-35@	155-32	155-32@	157-14	157-14@
M\$XFER	1-82@	84-153@												
MANUAL	1-;62@	84-153@	105-36											
MEMORY	1-;66@	84-153@	137-146											
OPEN	1-;71@	84-153@												
PNT	84-146@	133-25	133-29	133-34										
PNT...	84-107@	98-11	98-16	98-20	98-24	98-28	98-33	98-38	98-43	98-48	98-51	98-57	98-61	98-65
	98-72	98-76	98-80	98-84	98-88	98-92	98-96	98-101	98-105	98-109	98-118	98-123	98-127	98-132
	98-141	98-148	98-155	98-159	98-172	99-156	99-236	105-38	105-47	105-113	109-26	111-68	111-81	133-25
	133-29	133-34	135-13	137-132	148-17	148-82								
PNTB	84-134@	98-11	98-16	98-20	98-24	98-28	98-33	98-38	98-43	98-48	98-51	98-57	98-61	98-65
	98-72	98-76	98-80	98-84	98-88	98-92	98-96	98-101	98-105	98-109	98-118	98-123	98-127	98-132
	98-141	98-148	98-155	98-159	98-172	111-68	111-81							
PNTF	84-130@	99-156	99-236	105-38	105-47	105-113	137-132							
PNTS	84-142@	135-13												
PNTX	84-138@	109-26	148-17	148-82										
POINTE	1-;76@	84-153@	84-192											
POP	84-31@	99-35	99-76	99-110	99-237	109-28	109-39	111-15	111-35	111-41	111-96	111-108	111-120	112-26

