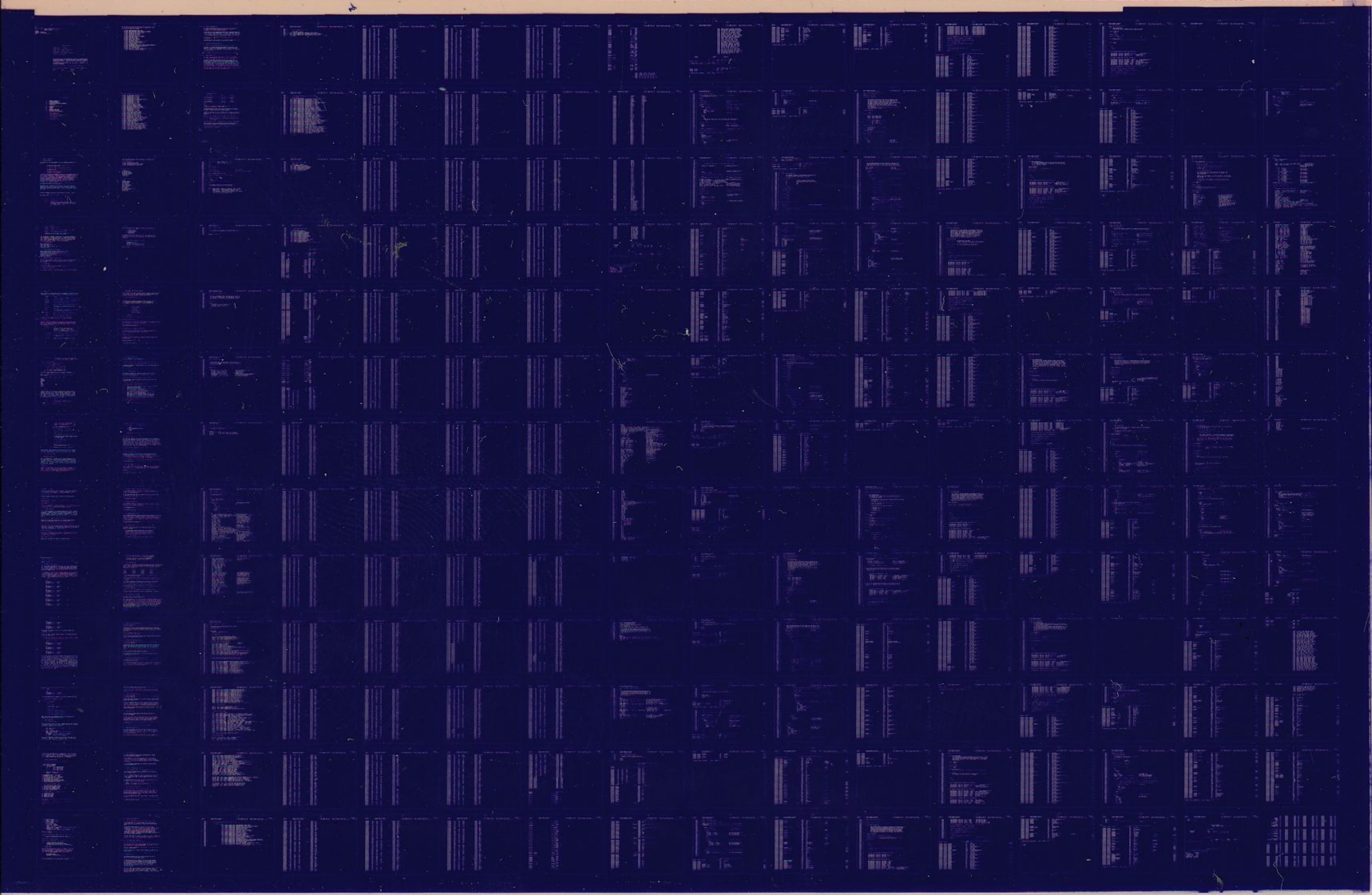


RC25

RC25 FR END TEST
CZRCFC0

AH-T271C-MC
1 OF 3 JUL 1985
COPYRIGHT© 1983-85

digital
MADE IN USA

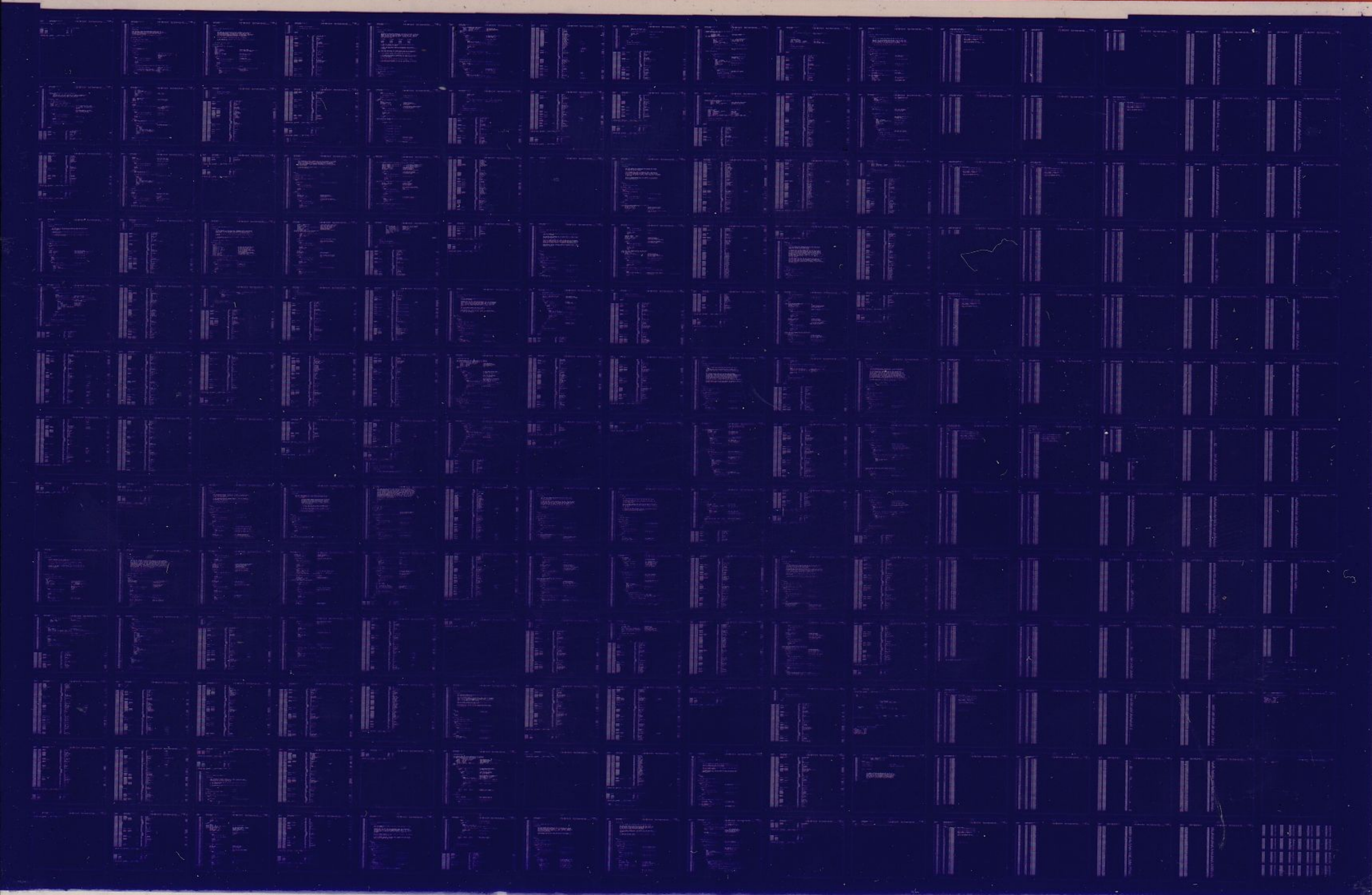


RC25

RC25 FR END TEST
CZRCFC0

AH-T271C-MC
2 OF 3 JUL 1985
COPYRIGHT © 1983-85

digital
MADE IN USA



RC25

RC25 FR END TEST
CZRCFC0

AH-T271C-MC

3 OF 3 JUL 1985

COPYRIGHT © 1983-85

digital

MADE IN USA

b 3 w
A 11
1
MODULE AZTECO (#TITLE 'CZRCFC0 RC25 FR END TEST'
IDENT = 'V03.0',
ADDRESSING_MODE (RELATIVE))=

BEGIN
LIBRARY 'library',
REQUIRE 'BLSMAC.REQ',
#SBTTL 'USER DOCUMENTATION'
#(

IDENTIFICATION

PRODUCT CODE: AC-T270C-MC
PRODUCT NAME: CZRCFC0 RC25 FR END TEST
PRODUCT DATE: MARCH 29, 1985
MAINTAINER: SMALL STORAGE ENGINEERING
AUTHOR: SING LAKSHMANAN

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) 1983,1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES
7.0	MAINTENANCE HISTORY

1.1 PROGRAM ABSTRACT

The aztec front-end host diagnostic is a diagnostic program to test the aztec disk drive subsystem. Tests are performed to verify that:

- a. The processor can properly communicate with the aztec through the adapter card.
- b. The aztec can seek and head select properly.
- c. The aztec conforms to the specified seek and rotational times.
- d. The aztec can perform certain basic functions in response to mscp commands.

The aztec front-end/host diagnostic consists of one program that runs in the host processor and programs that run in the aztec controller's buffer memory through an interpreter called the "diagnostic machine" which resides in the aztec. The host processor program will be responsible for testing the aztec adapter, testing some of the drive functions, downline loading the "diagnostic machine" programs into the aztec and starting their execution. When the "diagnostic machine" programs are running, they will control the testing by requesting the host processor to supply information and print error messages. The "diagnostic machine" programs will inform the processor when a test is complete.

Up to four (4) aztec controllers with one or two spindles each may be selected for test by this diagnostic.

One aztec "unit" is defined as a single platter. There are two platters on one spindle in an aztec drive. An aztec controller may have either one or two drives (two or four platters). The unit numbers for the aztec platters come in pairs. The removable media has an even number and the fixed media has the sequentially following odd number.

Software parameter questions include number of retries in case of an error, whether to continue execution after failures, select seek area in the disk, select manual intervention test and set trace mode.

This diagnostic is divided into 6 modules:

- module 0 - documentation
- module 1 - literals, format statements, ascii text, global data, hardware configuration questions and default tables, software parameter questions and default table, initialization code, cleanup code, summary report code

module 2 - global routines
module 3 - tests 1 - 29
module 4 - tests 9,10,11,12,13,19,21,26,27 (dm code)
module 5 - Last address and setup section
AZTECO.R16 is a file containing literals and field
declarations used throughout the program.

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. This program can be used with XXDP+, ACT, APT, slide and paper tape. 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.

1.2 SYSTEM REQUIREMENTS

PDP-11 Processor
28K Words of memory (minimum)
XXDP+ Load media
One or more aztec disk drive subsystems
Line clock - either type L or P
Console terminal

1.3 RELATED DOCUMENTS AND STANDARDS

AZTEC - RC25 Functional specification Rev 5, 3/9/82
Mass storage control protocol (MSCP) (version 1.0)
Unibus/Q-bus storage systems port (version 1.3)
Diagnostics and utilities protocol (R. Lary, May 1981)
Aztec diagnostic project plan
Diagnostic engineering functional specification for aztec
Resident diagnostics
XXDP+ User's manual

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

The bus, host processor, memory, system clocks and console terminal are all assumed to be functioning properly when this diagnostic is run. If they are not, the result of running this program is unpredictable.

1.5 ASSUMPTIONS

An aztec that meets the specifications for diagnostic machine timing will meet the specifications for MSCP timing.

2.0 OPERATING INSTRUCTIONS

This section contains a brief description of the runtime services. For detailed information, refer to the XXDP+ user's manual (CHQUS).

2.1 COMMANDS

There are eleven legal commands for the diagnostic runtime services (supervisor). This section lists the commands and gives a very brief description of them. The XXDP+ user's manual has more details.

COMMAND	EFFECT
START	Start the diagnostic from an initial state
RESTART	Start the diagnostic without initializing
CONTINUE	Continue at test that was interrupted (after ↑C)
PROCEED	Continue from an error halt
EXIT	Return to XXDP+ monitor (XXDP+ operation only!)
ADD	Activate a unit for testing (all units are considered to be active at start time)
DROP	Deactivate a unit
PRINT	Print statistical information (if implemented by the diagnostic - 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 "DDDD".

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:DDDD	Execute DDDDD passes (DDDD = 1 to 64000)
/FLAGS:FLGS	Set specified flags. flags are described in section 2.3.
/EOP:DDDD	Report end of pass message after every DDDDD passes only. (DDDD = 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.
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 command unless set using the flag switch. The ZFLAGS command may also be used to clear all flags, with the exception of the START 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

IBR*	Inhibit all error reports except first level (first level contains error type, number, PC, test and unit)
IXR*	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 (does not apply to diagnostics which do not support statistical reporting)
IDR	Inhibit program dropping of units
ADR	Execute autodrop code
LOT	Loop on test
EVL	Execute evaluation (on diagnostics which have evaluation support)

*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) ?" you must answer "Y" after a start command unless the hardware information has been "preloaded" using the setup utility (see chapter 6 of the XXDP* user's manual). 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.

UNITS (D) ?

Answer with the number of units to be tested (no default). This answer will determine how many times the following questions are asked. A unit is a logical disk (single platter) on an aztec. One to sixteen units may be specified (maximum configuration of four controllers with four platters per controller).

IP ADDRESS (O) 172150 ?

I1

SEQ 0008

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

VECTOR (O) 154 ?

Answer with the interrupt vector address of the aztec controller. A vector address in the range of 4 to 774 may be specified.

BR LEVEL (D) 5?

Answer with the interrupt priority used by the aztec. levels 4 to 7 are accepted.

UNIT NUMBER(S) (D) 0 ?

answer with the physical platter number(s) for the platter(s) you wish to test (NO DEFAULT). The removable platter is an even number and the fixed platter is the sequentially following odd number.

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. These parameters will govern some diagnostic specific operation modes. 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 paragraph(s).

Use top surface for all single surface tests (L) Y ?

Answer yes to use top surface for all single surface testing.
answer no to use bottom surface for all single surface testing.
The tests affected are 15 thru 18, 21 thru 23 and 25.

Do you wish to limit the area tested in tests 15-18 (L) N ?

Answer yes if you wish to specify a starting and ending track for the test area. this limitation applies only to seek verification testing. (tests #15 through #18). The following two questions will be asked only if this one is answered yes. The limits will be 0 and 820 for top surface and 821 and 1641 for bottom surface.

Starting track (D) 0 ?

Answer with the beginning track number of the area you wish to select for testing. This applies to tests #15 through #18 only. Test 22 will also use this starting track, if you answer this question instead of track 0 or 821.

Ending track (D) 820 ?

Answer with the last track number in the area you wish to select for testing. This applies to tests #15 through #18 only.

Do you wish to do the manual intervention test (L) Y ?

Answer yes to do the test of the write protect switches. Answer no to omit this test.

Do you wish trace mode (L) Y ?

Answer no if you do not like the test names to be printed out. Default is yes.

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 have a multiplexed device such as a mass storage controller with several drives or a communication device with several lines, this becomes tedious since most of the answers are repetitious.

To illustrate a more efficient method, suppose you are testing a fictional device, the XY11. Suppose this device consists of a control module with eight units (sub-devices) attached to it. These units are described by the octal numbers 0 through 7. There is one hardware parameter that can vary among units called the Q-FACTOR. This Q-FACTOR may be 0 or 1. Below is a simple way to build a table for one xy11 with eight units.

UNITS (D) ? 8<CR>

UNIT 1

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 0<CR>

Q-FACTOR (0) 0 ? 1<CR>

UNIT 2

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 1<CR>

Q-FACTOR (0) 1 ? 0<CR>

UNIT 3

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 2<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 4

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 3<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 5

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 4<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 6

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 5<CR>

Q-FACTOR (0) 0 ? <CR>

```

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

```

```

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

```

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

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

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

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

```

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

```

```

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

```

```

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

```

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

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

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

UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0-7<CR>
Q-FACTOR (0) 0 ? 0.1,0,....,1.1<CR>
```

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

2.7 QUICK START-UP PROCEDURE (XXDP.)

To start-up this program:

1. Boot XXDP.
2. Give the date
3. Type "R Name", where name is the name of the bin or bic file for this program
4. Type "START"
5. Answer the "CHANGE HW" question with "Y"
6. Answer all the hardware questions
7. Answer the "CHANGE SW" question with "N"

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

```
Name type number on unit number tst number PC:XXXXXX
error message
```

where; NAME = Diagnostic name
 TYPE = Error type (SYS FATAL, DEV FATAL, HARD or SOFT)
 NUMBER = Error number
 UNIT NUMBER = 0 - N (N is last unit in ptable)
 TST NUMBER = Test and subtest where error occurred
 PC:XXXXXX = Address of error message call

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

3.2 SPECIFIC ERROR MESSAGES

The following are device fatal error messages:

- 1) RCSA FAILED TO RESPOND
- 2) RCIP FAILED TO RESPOND
- 3) INIT STEP READ ERROR

STEP MASK = XX	FAILING REGISTER =	DATA =
XX = 1	- STEP 1 READ FAILURE	
XX = 2	- STEP 2 READ FAILURE	
XX = 4	- STEP 3 READ FAILURE	
XX = 10	- STEP 4 READ FAILURE	
- 4) STEP READ DATA DOES NOT MATCH

ADDRESS:	EXPECTED:	READ:
----------	-----------	-------
- 5) VECTOR AND BR LEVEL TEST FAILURE
- 6) INTERRUPT AT VEC= BR LEVEL=
- 7) NO INTERRUPT FROM PORT / CONTROLLER
- 8) BR LEVEL RECEIVED/TYPED IS INCORRECT !
- 9) HOST DETECTED TIME OUT ERROR
- 10) RING BUFFERS NOT CLEARED BY THE PORT
- 11) DATA ECHOED FROM RCSA DOES NOT MATCH
- 12) MEMORY BUFFER DOES NOT CONTAIN EXPECTED DATA
- 13) DM CODE RETURNED FAILURE CODE

The following are DUP/MSCP command failure messages:

- 14) RC25 UNIT DOES NOT COME ONLINE
- 15) EX_SUP PROG DUP COMMAND FAILURE
- 16) SEND DATA DUP COMMAND FAILURE
- 17) REC DATA DUP COMMAND FAILURE
- 18) GET UNIT STATUS COMMAND FAILURE
- 19) AVAILABLE COMMAND FAILURE

The following seek error messages are used.

- 20) FORWARD SEEK ERROR
- 21) REVERSE SEEK ERROR
- 22) TOGGLE SEEK ERROR
- 23) RANDOM SEEK ERROR
- 24) RC25 SEEK FAILURE

Also, one of the following will be printed as extended information:

STARTING TRACK: ENDING TRACK: DESIRED LBN:

NUMBER OF SEEKS (D): LBN:

- 25) HEAD SWITCH FAILURE
 UNIT: HEAD: TRACK:
- 26) SECTOR READ FAILURE
 UNIT: HEAD: TRACK:
- 27) OFFSET READ ERROR
 MAX. OFFSET VALUE:
- 28) WRITE DATA TEST IN ERROR
 WRITE DATA: READ DATA:
 TRACK: SECTOR: HEAD:
 ERROR STATUS: (NONZERO WILL INDICATE MICROCODE ERROR INFO.)
- 29) WRITE PROTECT TEST FAILURE
 EXPECTED SW = OFF ACTUAL SW = ON PLATTER # = (D)
 EXPECTED SW = ON ACTUAL SW = OFF PLATTER # = (D)

Note: All numbers displayed are octal unless (D) is indicated for decimal number.

LBN means logical block number from 0 to 143325 (octal)

TRACK refer to LBN tracks from 0 to 3151 (octal)
 except in DM code tests where this means DBN tracks.

UNIT refers to platter number.

SECTOR refers to DBN sector for DM code tests.

HEAD = 0 means top surface top platter
 40 means bottom surface top platter
 100 means top surface bottom (fixed) platter
 140 means bottom surface bottom (fixed) platter

3.2.1 ERROR CODES :

Whenever RCSA data contains fatal error codes or there was an error in end packet status code received for any of the MSCP commands used or if there was an error log message then the error code received from port will be given by one of the following messages with 6 octal digits:

RCSA ERROR STATUS:
 END PACKET ERROR STATUS:
 UNEXPECTED LOG PACKET ERROR STATUS:

Also, an explanation of the error code in the form \$FTLERR- will be printed out as an extended error message.

The following are self-detected fatal port/controller errors. These will be reported as extended error messages when RCSA data contains fatal error codes:

- \$FTLERR- UNRECOGNIZABLE ERROR CODE
- \$FTLERR- ENVELOPE/PACKET READ (PARITY OR TIMEOUT)
- \$FTLERR- ENVELOPE/PACKET WRITE (PARITY OR TIMEOUT)
- \$FTLERR- CONTROLLER ROM AND RAM PARITY
- \$FTLERR- CONTROLLER RAM PARITY
- \$FTLERR- CONTROLLER ROM PARITY
- \$FTLERR- RING READ (PARITY OR TIMEOUT)
- \$FTLERR- RING WRITE (PARITY OR TIMEOUT)
- \$FTLERR- INTERRUPT MASTER
- \$FTLERR- HOST ACCESS TIMEOUT
- \$FTLERR- CREDIT LIMIT EXCEEDED
- \$FTLERR- BUS MASTER ERROR
- \$FTLERR- DIAGNOSTIC CONTROLLER FATAL ERROR
- \$FTLERR- INSTRUCTION LOOP TIMEOUT
- \$FTLERR- INVALID CONNECTION IDENTIFIER
- \$FTLERR- INTERRUPT WRITE
- \$FTLERR- MAINTENANCE READ/WRITE INVALID REGION IDENTIFIER
- \$FTLERR- MAINTENANCE WRITE LOAD TO NON-LOADABLE CONTROLLER
- \$FTLERR- CONTROLLER RAM ERROR (NON-PARITY)
- \$FTLERR- INIT SEQUENCE ERROR
- \$FTLERR- HIGH LEVEL PROTOCOL INCOMPATIBILITY ERROR
- \$FTLERR- PURGE/POLL HARDWARE FAILURE
- \$FTLERR- MAPPING REGISTER READ ERROR (PARITY OR TIMEOUT)

: Self-detected fatal port/controller errors

\$FTLERR- VAX READ/WRITE ERROR ON INTERRUPT
\$FTLERR- INCONSISTENCY AT U.BFIL
\$FTLERR- INCONSISTENCY AT U.BMTY
\$FTLERR- INCONSISTENCY AT U.ALOC
\$FTLERR- INCONSISTENCY AT SERVO ENTRY (PIP SET)
\$FTLERR- INCONSISTENCY AT SERVO ENTRY (ERR SET)
\$FTLERR- INCONSISTENCY AT U.SEND
\$FTLERR- INCONSISTENCY AT U.RECV
\$FTLERR- INCONSISTENCY AT U.ATTN
\$FTLERR- INCONSISTENCY AT U.ONLN
\$FTLERR- ILLEGAL D REQUEST (U.QDRQ)
\$FTLERR- FENCE-POST ERROR AT PROTAB
\$FTLERR- BAD PACKET DEQUEUED AT U.DONE
\$FTLERR- UNEXPLAINED D-PROC SUSPENSION (U..TDS)
\$FTLERR- DUP PACKET D-Q FAILED (XFC 34/35)
\$FTLERR- INCONSISTENCY AT U.HTST
\$FTLERR- INCONSISTENCY AT U.SEKO
\$FTLERR- INCONSISTENCY AT U.CKSV
\$FTLERR- D.OPCD FOUND ILLEGAL OPCODE
\$FTLERR- D.CSF FOUND ILLEGAL OPCODE
\$FTLERR- UNKNOWN BAD DRIVE STATUS AT D.DSTS
\$FTLERR- ILLEGAL XFC EXECUTED BY DM
\$FTLERR- D PICKED UP A ZERO SCB.DB
\$FTLERR- INCONSISTENCY AT D IDLE LOOP
\$FTLERR- DM WORD COUNT ERROR ON HOST DMA/SEND/RECV
\$FTLERR- UNKNOWN DISPLAY FAULT CODE AT D.DFLT
\$FTLERR- DRIVE NOT FAULTING IN P.OFLN STATE
\$FTLERR- U POWER UP DIAGNOSTICS FAILED
\$FTLERR- D POWER UP DIAGNOSTICS FAILED
\$FTLERR- ADAPTER CARD FAILURE
\$FTLERR- EC.TMR TIMED OUT
\$FTLERR- U.SEND/U.RECV RING READ INCONSISTENCY
\$FTLERR- UNKNOWN WAITRY REASON AT D.RVCT
\$FTLERR- D.ARCS DID NOT FIND CLOSEST UNDONE ZONE
\$FTLERR- U.SEEK FOUND SEEK TO ILLEGAL TRACK
\$FTLERR- U.HTST INIT DIAG DMA WRITE FAILED
\$FTLERR- U.HTST INIT DIAG DMA COMPARE FAILED
\$FTLERR- U.SYDR FOUND SS.DER SET AND SS.SPN NOT SET
\$FTLERR- MASTER DRIVES ACLO ASSERTED

The following are return status messages. If response status error, then one of DUP return status codes or MSCP codes will be printed out.

\$FTLERR- RESPONSE STATUS ERROR:
\$FTLERR- SUPERVISOR SERVICE CALL FAILED
\$FTLERR- PORT/CONTROLLER TIMEOUT ERROR
\$FTLERR- UNKNOWN RETURN STATUS CODE

.....
Dup return status codes

SUCCESSFUL
INVALID COMMAND
NO REGION AVAILABLE
NO REGION SUITABLE
PROGRAM NOT KNOWN
ALOAD FAILURE
STANDALONE

.....
MSCP return status codes

SUCCESS
INVALID COMMAND
COMMAND ABORTED
UNIT-OFFLINE
UNIT-AVAILABLE
MEDIA FORMAT ERROR
WRITE PROTECTED
COMPARE ERROR
DATA ERROR
HOST BUFFER ACCESS ERROR
CONTROLLER ERROR
DRIVE ERROR
MESSAGE FROM AN INTERNAL DIAGNOSTIC

FAILING FRU

One or more of the four different module will be called out some times based on the major error code received from port.

- 1) ADAPTER BOARD
- 2) CONTROLLER BOARD
- 3) DRIVE BOARD
- 4) MECHANIC SET

For detailed information about the error code displayed and possible failing logic/function call-out, the RC25 controller manual that deals with error/status condition codes should be consulted.

⋮ The following are system error messages:
⋮

POWER DELAY - WAITING
TOO MANY UNITS
NO CLOCK WAS FOUND IN THE SYSTEM
INCORRECT TRACK NUMBERS SELECTED

Note: If there was no clock in the system, then the diagnostic will not run.

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.

5.0 DEVICE INFORMATION TABLES

The Supervisor builds one Hardware P_Table for every logical unit tested while answering Hardware P_table questions. This diagnostic gets one table at a time in sequence and runs diagnostic tests as selected. The P_table looks like this:

HWP_TABLE:

```

0      :-----:
      :HWP_IP_ADDRESS :
      :-----:
2      :HWP_VECTOR    :
      :-----:
4      :HWP_BR_LEVEL  :
      :-----:
6      :HWP_UNIT_NUMBER:
      :-----:

```

6.0 TEST SUMMARIES

A brief description of the tests done are described below:

TEST #1 REGISTER EXISTENCE TEST

This test will first check for the existence of the address of the IP and SA registers for the device under test. If these memory addresses are non-existent, the error will be reported.

If the operator has specified loop on error, looping will be from the beginning of each sub test.

TEST #2 INITIALIZATION TEST (POWER UP DIAGNOSTICS)

This test init's the aztec and runs the power up diagnostics by writing with step 1 data. Then it will check for errors and report if aztec does not come upto step 2 read.

TEST #3 DIAGNOSTIC WRAP TEST

The aztec will be initialized in diagnostic wrap mode and a one bit and also zero bit floated through the SA register to see that it echoes properly.

A failure to echo what was written will result in a callout to the adapter card fru.

If the operator has specified loop on error, the program will loop on the failing write and read.

TEST #4 - VECTOR AND BR LEVEL TEST

The init sequence will be started with the interrupt enable bit set to verify the aztec's vector and BR level.

This test assumes the vector given by the operator is correct.

The priority level of the interrupt request will be verified.

Failure of the aztec to vector properly will necessitate that this program be restarted. A completed interrupt at the wrong BR level will be reported.

Loop on error will restart this test if the error is recoverable.

TEST #5 STEP 1 -3 INITIALZATION TEST

This test will check for information echoed from the port at each step read coming upto that step from scratch. If there was an error reported or echoed information was incorrect the error will be reported.

Loop on error will be from the beginning of sub test.

Port gives some information about the Port at every step read in RCSA Register. This information will be printed out to the operator as follows:

1) At step 1 read the following will be given:

PORT SPECIFIC INFO: /NV/QB/DI/OD/MP/ = xx (0)

NV = 1 means that the port does not support a host settable interrupt vector address

QB = 1 means that the Port supports a 22-bit host bus. This bit will be a 0 for unibus.

DI = 1 means that the Port implements enhanced diagnostics, i.e. wraparound, purge and poll tests.

OD = 1 means that the Port allows odd host address to be specified in the buffer descriptor.

MP = 1 means that the Port supports address mapping. The host supplies a virtual data address in the buffer descriptor which is mapped to a resultant address using mapping registers maintained in host memory.

xx Two digit octal value of the above right justified.

2) At step 2 read the following will be given:

PORT TYPE NUMBER = xx (0)

xx 0 means UNIBUS/QBUS storage systems port.

3) At step 4 read the following will be given:

MICRO CODE: MODEL = xx (0) VERSION = yy (0)

xx = 0 UDA50
 1 RC25 Integrated Controller
 5 TUB1 Integrated Controller
 6 UDA50A
 7 QDRX01

yy = Mod 16 value of the actual controller microcode
 version.

TEST #6 PURGE AND POLL TEST

This test will perform the first three steps of the init sequence. When the host responds to the step 3 transition it will write a one bit to bit 15 of the SA register, thereby requesting the execution of purge and poll testing. The host then waits for the SA register to transition to a zero value. The host then writes zeroes to the SA register simulating a "purge completed" host action. The host then reads the IP register to simulate a "start polling" command from the host to the port. The test is complete when the controller announces the transition to step 4 in the SA register.

Failure to properly complete this test will be reported.

Loop on error will restart the test.

TEST #7 - SMALL RING BUFFER INIT TEST

The aztec will be initialized without interrupts and using the smallest ring buffer. This will be the first time that the initialization sequence is carried out to completion. Initializing with the smallest ring buffer minimizes the host memory area with which the aztec controller must be able to communicate.

Failure to properly initialize the aztec and com_area will be reported.

If the operator has specified loop on error, looping will be from the start of this test.

TEST #8 - LARGE RING BUFFER INIT TEST

The init sequence is executed without interrupts with a ring buffer large enough to cover the normal host communications area packet and buffer space (a 16 in message length and a 16 in command length).

A failure to complete the initialization sequence without error will be reported.

If the operator has specified loop on error, looping will be from the beginning of this test.

TEST #9 - "DIAGNOSTIC MACHINE" CODE DOWN LINE LOAD TEST

This "Diagnostic Machine" program will attempt to transfer a block of data from host memory to an area in the controller and then examine the transferred data.

If the transferred data does not compare correctly, then an error will be reported. This test also reports errors if any of the routines used returned failure code.

If the operator has specified loop on error, looping will be from the start of this test.

TEST #10 - NONEXISTENT MEMORY TEST

This "Diagnostic Machine" program will attempt to read the first address of the I/O page of the host CPU. This location is reserved for diagnostics and a nxm should occur.

If the controller does not see the nxm, there will be a fru callout of the adapter card.

If the operator has specified loop on error, looping will be from the start of this test.

TEST #11 - BUS ADDRESSING/DATA TEST A

This "Diagnostic Machine" program asks the PDP-11 program to fill free memory (that memory available to the PDP-11 program that is not being used by the program or the PDP-11 supervisor) with an addressing pattern (write address with address) and report the location and size of the free memory. Every location of free memory will be read and the data checked.

If the data does not compare correctly, the address, data expected and data received are reported.

TEST #12 - BUS ADDRESSING/DATA TEST B

This test first brings aztec drive Ready and Online and then loads DM_12 program vector to port controller memory, then does the following:

- a. Give free memory address and buffer size to DM code and ask DM code write a pattern of one's complement of address at the address and expects to receive success or failure code from DM program. Then checks memory buffer for the expected pattern and reports error if encountered.
- b. If success, asks DM code to write to memory a pattern of all ones and checks for the pattern in memory.

- c. If success, asks DM code to write to memory a pattern of all zeroes and checks for the pattern in memory.
- d. If failure, retries will be done as controlled by a software question. Loop on error flag will loop from beginning of test to the point of failure.

TEST #13 - BLOCK TRANSFER TEST

The ability of the Aztec controller to do block transfers to and from memory will be tested with different data patterns. The "write host memory" XFC and the "read host memory" XFC will be used. The host memory buffer is 256 words in size. 4 different data patterns as given below are used.

Pattern 0	Pattern 1	Pattern 2	Pattern 3
-----	-----	-----	-----
111111	177400	155555	000377
044444	007760	133333	170017
022222	000377	066666	177400

- 1) This test brings RC25 controller online and loads DM code program to controller's memory.
- 2) First the host memory buffer is initialized with pattern 0. A send data command with host buffer addresses (transmit and receive) is issued.
- 3) DM code then reads host memory buffer and puts in controller's memory and writes back in host memory receive buffer using XFC's.
- 4) Host program compares both buffers for data pattern 0.
- 5) If there was an error in comparison the error will be reported. If there was error in the MSCP DUP calls or initialization, this will also be reported.

Steps 2 thru 5 will be repeated for data patterns 1,2 and 3.

If an error was encountered the test will be aborted. If operator chose for retries, retries will be done from the start of the test.

TEST #14 - SPIN UP/HEAD LOAD SEQUENCE

This test first initializes RC25 controller, initializes com_area, and does set control characteristics.

Then, this test will first issue the mscp "available" command with the spin down modifier set. It will then wait for 30 seconds to insure that the drive has had time to spin down. It will then issue the MSCP "online" command to spin the drive up. This operation will be timed and the time will be reported to the operator so that this time can be verified to make sure it is within limits. The run/start and head load internal diagnostics will run during this time. If an error is encountered the returned status of the "online" command will be something other than "success" and this status will be decoded and reported with error message.

If the operator has specified retries on error, the test will be repeated.

TEST #15 - SEQUENTIAL SEEK AND VERIFY TEST

This test brings RC25 controller and unit online and ready to accept MSCP DUP commands.

Starting with the user specified beginning track and incrementing through every track to the user specified ending track, this test will seek from track to track in a forward direction, then it will repeat the operation in the reverse direction, from the ending track to the beginning.

This is a single surface test and is done on top surface. The operator can select bottom surface also.

A failure report includes starting track, ending track and desired track. After reporting the failure, the program will abort current seek and will jump to reverse seek.

TEST #16 - SAWTOOTH SEEK AND VERIFY TEST

This test brings RC25 controller and the unit online and ready to take mscp commands.

Starting with the user specified beginning track and incrementing through every track in the selected range, this test will perform a seek to the selected track and then a seek back to the beginning track. When all tracks have been covered, it will do the same operation in the reverse direction with the ending track as the base.

This is a single surface test and is done on top surface. The operator can select bottom surface also.

error reports will state starting, ending and desired tracks. If there was an error the test will be aborted unless the operator has selected for retries.

TEST #17 - CONVERGING/DIVERGING SEEK AND VERIFY TEST

This test first brings RC25 controller and unit online so that MSCP commands can be issued.

This test performs seeks to the beginning track, then to the ending track, then to the beginning track + 1, ending track - 1, beginning track + 2, etc. until the tracks converge and then diverge again back to the beginning and ending tracks.

This is a single surface test and is done on top surface. The operator can select bottom surface also.

Error reports will include starting, ending and desired tracks. If failure in seek the test will be aborted unless the operator selects retries.

TEST #18 - TOGGLE SEEK AND VERIFY TEST

This test brings RC25 controller and the unit on line and ready to accept MSCP commands.

One thousand seek commands will be issued one at a time to toggle between the beginning track of 0 (lbn = 0) and the ending track of 820 (lbn = 820 + 31).

This is a single surface test. seek is done only on top surface unless the operator chose to seek on bottom surface by answering one of the software questions. The operator has control over the beginning and ending tracks, if desired by answering questions.

Error reports include starting, ending and desired tracks. After reporting the failure the diagnostic will abort the test, unless retries is enabled.

TEST #19 - HEAD SWITCH TEST

This test will bring RC25 controller and the unit online and will load dm code program to controller's memory using EX_SUP_PROG command.

DM code will seek to both surfaces of the unit. The XFC status will be used to verify that the proper track has been reached. Block headers will be read to verify that the proper heads are selected. DM code will retry if there was any error in seek. DM code will give success or failure code to the host.

If failure, the track, head and unit will be reported as received from DM code.

If retries are turned on the test will be repeated.

TEST #20 - RANDOM SEEK AND VERIFY TEST

This test brings RC25 controller and the selected unit online and then issues 1000 seeks one at a time to randomly selected LBN tracks between the range of 0 - 1641. This will ensure head switch as well because tracks over 820 will be in the bottom surface of selected unit.

Error reports include seek count and failing track number. If loop on error flag is set, failing track will be retried for ever.

TEST #21 - SECTOR ACCESS TEST

This test brings RC25 controller and selected unit online and then loads DM 21 vector array into controller's memory by giving EX_SUP_PROG command.

The DM program will seek to diagnostic track 0 and read 32 blocks after making sure that good header is found. DM code will retry if any error was found. dm code will send status back to host with failing unit, head and track. Error will be reported by host code.

This is a single surface test. top surface will be accessed unless the operator chose bottom surface by answering one of the software questions.

TEST #22 - CONTROLLER PROCESSING TIME TEST

This test brings RC25 controller and selected unit online.

The controller processing time is measured by averaging the time it takes to do 100 zero length seeks, that is, seeks that are zero tracks long.

This is a single surface test. seek will be done on top surface unless the operator chose to seek on bottom surface. Track 0 will be used or the starting track number as given by the operator will be used.

If there was any error in seek, this will be reported with the the number of seeks completed and desired track. The test will be aborted unless retries are enabled.

If success, the average time will be reported. Controller processing time expected will be around 2 ms.

TEST #23 - ONE TRACK SEEK TIMING TEST

This test brings RC25 controller and selected unit online.

One track seek time is the average of all one track seeks that do not include a head switch. all forward one track seeks will be done and timed and then reverse one track seeks will be done and timed. Average time will be reported. The expected time will be around 6 to 7 ms.

This is a single surface test. Top surface will be used unless the operator chose otherwise. Seeks will be from start to the end of tracks.

If there was an error, error will be reported and the test aborted unless retries are turned on.

TEST #24 - AVERAGE SEEK TIMING TEST

This test brings RC25 controller and selected unit online.

The average seek time is the average time it takes to do a seek given that it is equally likely to start on any track and any head, and equally likely to end on any track and any head.

One thousand random seeks will be done over the range of LBN track 0 thru LBN track 1641 to cover both surfaces of the selected unit. First time express bit in command modifier field for READ_CMD will be set so that random seeks are timed and in the second time express bit will be reset so that the random LBN available to the controller are ordered by the controller for seeks. Average time for both cases will be reported. The expected time will be around 32 ms for random LBN seeks and 17 ms for ordered LBN seeks.

An error report for this test will report the number of seeks and desired track number. After reporting a failure, the diagnostic will proceed to the next test unless retries is turned on.

TEST #25 - FULL STROKE SEEK TIMING TEST

This test brings RC25 controller and unit online.

The full stroke seek time is the average time of 1000 full stroke seeks that do not involve head switches. The average time will be reported and is expected around 55 ms.

This is a single surface test. top surface will be used unless the operator chose otherwise.

The error report will include number of seeks and desired track number. After failure, the test will be aborted unless retries are turned on.

TEST #26 - WRITE DATA TEST

This test brings RC25 controller and selected unit online, then loads dm code vector array DM_26 to the controllers memory by issuing EX_SUP_PROG command.

The dmcode gets the unit number from the host and attempts to find at least one good diagnostic block on each surface of the platter specified and make sure that dmcode can read and write to the block in order to verify that the heads are working properly. First top surface will be attempted with all ones data and second all zero data. This will be repeated for bottom surface as well. The data written will be read and compared.

The error report on this test will include data written, data read plus the track, head and sector number. Also error status from the micro code if any will be reported. Error status of zero will mean other errors trapped in dmcode. After reporting the error the rest of the test will be aborted unless the operator selects retries.

TEST #27 - OFFSET TOLERANCE TEST

This test brings RC25 controller and the unit online and loads DM program DM_27 vector array into controller's memory for execution by issuing EX_SUP_PROG command.

The DM code will do an offset tolerance test. a good odd block will be found in track 829 (DBN track). It will be read with increasing + and - offset, until a hard error is forced. The offset value used in the last good read will be sent to host program. The host will give the maximum offset as a percentage of track to track distance.

This test will be performed on top surface of the unit being tested.

A message report on this test will include the largest offset value used in order to read the block without forcing errors.

TEST #28 - AVERAGE ROTATIONAL TIMING TEST

This test will bring RC25 controller and the unit online.

This test will be performed from the host using the MSCP "read" command. An LBN will be selected randomly. One thousand two byte count reads of the same LBN will be performed. This operation will be timed and the average time will be reported. The expected time will be 21 ms.

If the operator has selected retries, the test will be repeated.

TEST #29 - WRITE PROTECT TEST

This test requires manual intervention. It will be executed if the software parameter questions do not cause it to be omitted.

This test brings RC25 controller and the unit online first. The test is done from the host using the mscp command "GET UNIT STATUS" (gus). The test will ask the operator to make sure the write protect switch for the unit is in the off position. It will do the gus for the unit to verify that the controller knows it is not write protected. Then the operator will be asked to put the write protect switch in the on position and a gus will be done to make sure the controller recognizes that the unit is write protected.

The error report for this test will contain the unit number, expected and actual positions of the write protect switch.

7.0 MAINTENANCE HISTORY

Modified By:	Date:	Version:
***** SING LAKSHMANAN	JULY 83	CZRCFA0
SING LAKSHMANAN	OCT 83	CZRCFB0
SING LAKSHMANAN *****	JAN 85	CZRCFC0

NOTE :

CZRCFB0 is a release of complete tests for RC25 FR END TESTS, following the base level release CZRCFA0.

CZRCFB0 contains 29 tests. The first 12 tests are functionally the same as CZRCFA0. All source modules to make up this diagnostics have been revised, appended to produce CZRCFB0.

CZRCFC0 is modified version of CZRCFB0 with the following corrections. source modules have indications with VER:C in comments wherever needed.

Patch B1: Test 10 hangs with C15 microde in the controller.

Patch B2: Time-out error for an indefinite pass of any one test.
Test 14 is done only in first pass. corrected to run in all passes.

Patch B3: Test 4 error 7 time out occurs in orion 11/73 processors.

Test 14 does not wait long enough to spindown completely. 30 sec. realtime timer included to avoid false spin up time reports. This problem was specific to Orion 11/73 processor only as of date.

Test 26 contains a revised dm code. The test description will explain the test better now.

)
ELUDOM

ZRCFB1

CZRCFC0 RC25 FR END TEST

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC1.B16;1SEQ 0029
Page 1
(1)

```

: 0001 0  MODULE ZRCFB1 (*TITLE 'CZRCFC0 RC25 FR END TEST'
: 0002 0  IDENT = 'V03.0',
: 0003 0  ADDRESSING_MODE (RELATIVE)
: 0004 0  ) =
: 0005 1  BEGIN
: 0006 1  !
: 0007 1  !<BLF/LOWERCASE_KEY>
: 0008 1  !
: 0009 1  !
: 0010 1  library 'AZTECO';           ! AZTEC LIBRARY
: 0011 1  !
: 0012 1  require 'BLSMAC.REQ';     ! DIAGNOSTIC SUPERVISOR LIBRARY
: 1501 1  !
: 1502 1  *sbttl 'PROGRAM HEADER AND TABLES'
: 1503 1  !
: 1504 1  ! DEFINE THE NUMBER OF TESTS IN THIS DIAGNOSTIC
: 1505 1  !
: 1506 1  psect
: 1507 1  code = AA$CODE;
: 1508 1  !
: 1509 1  literal
: 1510 1  DS$NBR_OF_TESTS = 29;
: 1511 1  !
: 1512 1  POINTER (ALL);
: 1513 1  !
: 1514 1  !**
: 1515 1  ! THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: 1516 1  ! THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
: 1517 1  !--
: 1518 1  !
: 1519 1  HEADER (*ascii'CZRCF ', *ascii'A', *ascii'O', 120, 0, PRI00);
: 1520 1  !: ARGUMENTS ARE: NAME,REV,PATCH,LONGEST TEST TIME,TYPE
: 1521 1  !: WHERE "TYPE" = 0 FOR SEQUENTIAL DIAGNOSTIC AND =1
: 1522 1  !: FOR EXERCISER. THERE IS ALSO AN OPTIONAL SIXTH ARGUMENT
: 1523 1  !: WHICH SPECIFIES THE PROCESSOR PRIORITY TO BE SET WHEN
: 1524 1  !: STARTING THE DIAGNOSTIC (DEFAULT IS 0).

```


E3

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
DISPATCH TABLE

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0030
Page 2
(2)

```
: 1525 1 %sbttl 'DISPATCH TABLE'  
: 1526 1  
: 1527 1  
: 1528 1 !*  
: 1529 1 ! THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.  
: 1530 1 ! IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.  
: 1531 1 !--  
: 1532 1 DISPATCH (DS#NBR_OF_TESTS);  
: 1533 1 ERRTAB;
```

ZRCFB1
V03.0CZRCFC0 RC25 FR END TEST
DEFAULT HARDWARE P-TABLE27-Mar-1985 15:21:49
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1SEQ 0031
Page 3
(3)

```
: 1534 1  %sbttl 'DEFAULT HARDWARE P-TABLE'
: 1535 1
: 1536 1  !**
: 1537 1  ! THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: 1538 1  ! THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: 1539 1  ! IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
: 1540 1  ! AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
: 1541 1  !--
: 1542 1
: 1543 1  BGNHW (DFPTBL);
: 1544 1
: 1545 1  global
: 1546 1  P_IP_ADDRESS : word initial ('172150'),
: 1547 1  P_VECTOR : word initial ('154'),
: 1548 1  P_BR_LEVEL : word initial (5),
: 1549 1  P_UNIT_NUMBER : word initial (0);
: 1550 1
: 1551 1  ENDHW;
```

ZRCFB1
V03.0CZRCFC0 RC25 FR END TEST
SOFTWARE P-TABLE27-Mar-1985 15:21:49
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1SEQ 0032
Page 4
(4)

```

: 1552 1 #sbttl 'SOFTWARE P-TABLE'
: 1553 1
: 1554 1
: 1555 1 !**
: 1556 1 ! THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
: 1557 1 ! PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE
: 1558 1 ! SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
: 1559 1 ! AT RUN TIME.
: 1560 1 !--
: 1561 1 BGNSW (SFPTBL);
: 1562 1
: 1563 1 global
: 1564 1 SWP_TOP : word initial (YES), !USE TOP SURFACE FOR SINGLE SURFACE TESTS
: 1565 1 SWP_LIMIT : word initial (NO), !LIMIT AREA TESTED
: 1566 1 SWP_START : word initial (0), !STARTING TRACK
: 1567 1 SWP_END : word initial (820), !ENDING TRACK
: 1568 1 SWP_RETRIES : word initial (0), !NUMBER OF RETRIES BEFORE DROPPING UNIT
: 1569 1 SWP_CONTINUE : word initial (NO), !DO YOU NEED TO CONTINUE TESTING?
: 1570 1 SWP_MANUAL : word initial (NO), !DO MANUAL INTERVENTION TEST
: 1571 1 SWP_TRACE : word initial (YES); !DO YOU NEED TRACE MODE?
: 1572 1
: 1573 1 ENDSW;

```

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
PROTECTION TABLE

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

```

: 1574 1 #sbttl 'PROTECTION TABLE'
: 1575 1
: 1576 1
: 1577 1 !**
: 1578 1 ! THIS TABLE IS USED BY THE RUNTIME SERVICES
: 1579 1 ! TO PROTECT THE LOAD MEDIA.
: 1580 1 !--
: 1581 1 BGNPROT (-1, -1, -1);
: 1582 1 !1ST ARG = OFFSET INTO P-TABLE FOR CSR ADDRESS
: 1583 1 !2ND ARG = OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
: 1584 1 !3RD ARG = OFFSET INTO P-TABLE FOR DRIVE NUMBER
: 1585 1 ENDPROT;

```

ZRCFB1
V03.0CZRCFC0 RC25 FR END TEST
GLOBAL DATA SECTION27-Mar-1985 15:21:49
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1SEQ 0034
Page 6
(6)

```

: 1586 1  *sbttl 'GLOBAL DATA SECTION'
: 1587 1
: 1588 1
: 1589 1  !**
: 1590 1  ! THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
: 1591 1  ! IN MORE THAN ONE TEST.
: 1592 1  !--
: 1593 1
: 1594 1  psect
: 1595 1      split = $split$( global),
: 1596 1      global = $GLOB$(nowrite, noexecute, global, concatenate),
: 1597 1      own = $own$;
: 1598 1
: 1599 1  structure                                ! DEFINE ACCESS ALGORITHM
: 1600 1      RC25 [0, P, S, E] =                  ! TO ALLOW FIELD REFERANCES
: 1601 2      begin                                ! TO THE AZTEC
: 1602 2
: 1603 2          local
: 1604 2              RC_REG;
: 1605 2
: 1606 2              RC_REG = .(RC25 + $upval*0)<0, $bpval, 0>;
: 1607 2              RC_REG
: 1608 1          end
: 1609 1          <P, S, E>;
: 1610 1
: 1611 1  global
: 1612 1      RT : vector [WORD1_IN_RT_TAB, word],      !RUNTIME TABLE STORAGE
: 1613 1      RT_TABLE : ref block [WORD1_IN_RT_TAB, word] field (RT_FIELDS),      !RUNTIME TABLE POINTER
: 1614 1      HWP_TABLE : ref block [WORD2_IN_HWP_TAB, word] field (HWP_FIELDS),
: 1615 1      XMT_DATA_BUF : vector [256, word],      ! TRANSMITTING DATA BUFFER 1
: 1616 1      RCV_DATA_BUF : vector [256, word],      ! RECEIVING DATA BUFFER 2
: 1617 1      CLK_ADR : word,      ! LOC. TO RETURN CLOCK ADDR.
: 1618 1      CLK_TYPE : word,      ! TYPE OF CLOCK ON SYSTEM
: 1619 1      CLK_CSR : word,      ! STORE CSR ADDRESS FOR CLOCK HERE
: 1620 1      CLK_HERTZ : word,      ! STORE CLOCK HERTZ RATE
: 1621 1      CLK_START : word,      ! STORE CLOCK START VALUE
: 1622 1      UNIT : word,      ! UNIT UNDER TEST THIS PASS
: 1623 1      LOG_UNIT : word,
: 1624 1      OUT_BOUND : word,      ! COMMAND COUNT
: 1625 1      IN_BOUND : word,      ! RECEIVE COUNT
: 1626 1      VEC_AD : byte volatile,      ! VECTOR ADDRESS OF AZTEC
: 1627 1      RC25_ADDR : ref RC25 field (RC_REG),      ! DEFINE REFERANCE TO AZTEC FIELDS
: 1628 1      RC25_DATA : block [2, word] field (RC_REG),
: 1629 1      COM_AREA : blockvector [REC_ALLOCATE + SND_ALLOCATE + HDR_SIZ, 2, word],
: 1630 1      HEAD_AREA : ref block [4, word] field (HDR_FIELD),
: 1631 1      RECEIVE_RING : ref blockvector [REC_ALLOCATE, 2, word] field (DSC_FIELD),
: 1632 1      SEND_RING : ref blockvector [SND_ALLOCATE, 2, word] field (DSC_FIELD),
: 1633 1      REC_ENVELOPE : blockvector [REC_ALLOCATE, RB_SIZE + 2, word] field (ENV_FIELD),
: 1634 1      SND_ENVELOPE : blockvector [SND_ALLOCATE, SB_SIZE + 2, word] field (ENV_FIELD),
: 1635 1      BUF_DESCRPTR : word volatile,      ! BUFFER DESCRIPTOR AREA
: 1636 1      CMD_REF : word volatile,      ! COMMAND REFERENCE BUFFER
: 1637 1      BYTE_COUNT : word volatile,      ! BYTE COUNT BUFFER
: 1638 1      TICKS : word initial (0) volatile,      ! SOTRE THE NUMBERS OF CLOCK INTERRUPTED
: 1639 1      SECONDS : word initial (0) volatile,      ! STORE SECONDS
: 1640 1      MINUTES : word initial (0) volatile,      ! STORE MINUTES
: 1641 1      TIP : word,      ! STORAGE FOR NUMBER OF TEST IN PROGRESS
: 1642 1      DATA1 : word volatile,      ! AZTEC STEP 1 WRITE DATA
: 1642 1      DATA2 : word volatile,      ! AZTEC STEP 2 WRITE DATA

```

ZRCFB1
V03.0CZRCFC0 RC25 FR END TEST
GLOBAL DATA SECTION27-Mar-1985 15:21:49
11-Jan-1985 08:19:19VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1SEQ 0035
Page 7
(6)

```

: 1643 1      DATA3 : word volatile,      ! AZTEC STEP 3 WRITE DATA
: 1644 1      DATA4 : word volatile,      ! AZTEC STEP 4 WRITE DATA
: 1645 1      I_AM_NEX : word initial (0) volatile, ! INTERRUPT FLAG
: 1646 1      MSGADR : word volatile,
: 1647 1      END_LBN : word initial (50901) volatile, ! ENDING LBN
: 1648 1      P_MASK : byte volatile,
: 1649 1      B_MASK : byte volatile,
: 1650 1      MANU_SW : word volatile,
: 1651 1      SWITCH2 : word volatile,
: 1652 1      RET_UNIT_FLAG : word volatile,
: 1653 1      P1 : word volatile,
: 1654 1      P2 : word volatile,
: 1655 1      P3 : word volatile,
: 1656 1      P4 : word volatile,
: 1657 1      P5 : word volatile,
: 1658 1      P6 : word volatile,
: 1659 1      RET_STATUS : word volatile,      ! SAVES VARIOUS RETURN STATUS
: 1660 1      ER_STATUS : word initial (0),      ! SAVES ERROR STATUS CODE
: 1661 1      CANCEL_TIMER : word volatile,      ! INIT SEQUENCE INTERRUPT
: 1662 1      CMD_SLOT : word volatile,      ! COMMAND DESCRIPTOR SLOT
: 1663 1      RES_SLOT : word volatile,      ! RECEIVE DESCRIPTOR SLOT
: 1664 1      LBN : word volatile,
: 1665 1      LBN_ST : word volatile,      ! STARTING LOGICAL BLOCK #
: 1666 1      LBN_ED : word volatile,      ! ENDING LOGICAL BLOCK #
: 1667 1      LBN_SZ : word volatile,      ! INCREMENTING LBN SIZE
: 1668 1      FREE_MEM_ADDR,      ! STARTING FREE MEMORY ADDR.
: 1669 1      MEM_SIZE : word volatile,      ! FREE MEMORY SIZE
: 1670 1      H_SADD : word volatile,      ! LOW-BYTE FREE MEMORY ADDR.
: 1671 1      H_EADD : word volatile,      ! HIGH-BYTE FREE MEMORY ADDR.
: 1672 1      BUF_LENGTH : word volatile,      ! BUFFER LENGTH
: 1673 1      CMOD : word initial (0),      ! COMMAND MODIFIER.
: 1674 1      NUM_RETRIES : word volatile,
: 1675 1      RETRIES : word initial (FALSE),
: 1676 1      FAL_CODE : word initial (1),      ! FAIL STATUS
: 1677 1      DMC_TEST : word,
: 1678 1      BYT_CNT : word,
: 1679 1      DM_REC : word,
: 1680 1      DM_XMT : word,
: 1681 1      SIZ_LBN : word initial (31),      ! SIZE OF LBN TO GET TO NEXT TRACK
: 1682 1      OFFSET : word initial (0),      ! USED TO GET TO BOTTOM SURFACE
: 1683 1      PASS0 : word,      ! FLAG FOR FIRST PASS
: 1684 1      TEMP : word volatile;
: 1685 1

```

```

: 1686 1 #sbttl 'GLOBAL TEXT SECTION'
: 1687 1
: 1688 1
: 1689 1 :..
: 1690 1 : THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: 1691 1 : MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: 1692 1 : MORE THAN ONE TEST.
: 1693 1 :--
: 1694 1
: 1695 1 global bind
: 1696 1 RINGBASE = COM AREA [REC_BASE],
: 1697 1 TIME = plit (P4, P5),
: 1698 1 :
: 1699 1 : FAILING FRU'S
: 1700 1 FRU = uplit (#esciz'#AFAILING FRU = #T#D3#N'),
: 1701 1 ADAPTO = uplit (#esciz'#ADAPTOR BOARD FOR UNIT #:'),
: 1702 1 CONTRO = uplit (#esciz'#CONTROLLER BOARD FOR UNIT #:'),
: 1703 1 DRIVE = uplit (#esciz'#DRIVE BOARD FOR UNIT #:'),
: 1704 1 MECHAN = uplit (#esciz'#MECHANIC SET FOR UNIT #:'),
: 1705 1 :
: 1706 1 : HARDWARE AND SOFTWARE QUESTIONS
: 1707 1 :
: 1708 1 QST1 = uplit (#esciz'#IP ADDRESS'),
: 1709 1 QST2 = uplit (#esciz'#VECTOR'),
: 1710 1 QST3 = uplit (#esciz'#BR LEVEL'),
: 1711 1 QST4 = uplit (#esciz'#PLATTER ADDRESS[ES]'),
: 1712 1 QST6 = uplit (#esciz'#USE TOP SURFACE FOR SINGLE SURFACE TESTS'),
: 1713 1 QST7 = uplit (#esciz'#DO YOU WISH TO LIMIT AREA TESTED IN TESTS 15-18'),
: 1714 1 QST8 = uplit (#esciz'#STARTING TRACK'),
: 1715 1 QST9 = uplit (#esciz'#ENDING TRACK'),
: 1716 1 QST10 = uplit (#esciz'#DO YOU WISH TO DO THE MANUAL INTERVENTION TEST?'),
: 1717 1 QS10_1 = uplit (#esciz'#DO YOU WISH TRACE MODE?'),
: 1718 1 QS10_2 = uplit (#esciz'#DO YOU WISH TO CONTINUE TESTING AFTER RETRIES?'),
: 1719 1 QST11 = uplit (#esciz'#NUMBER OF RETRIES FOR TEST IF ERROR OCCURED'),
: 1720 1 QST14 = uplit (#esciz'#TURN OFF WRITE PROTECT SWITCH AND DO <CR>'),
: 1721 1 QST15 = uplit (#esciz'#TURN ON WRITE PROTECT SWITCH AND DO <CR>'),
: 1722 1
: 1723 1 :..
: 1724 1 : THE FOLLOWING MESSAGES INCLUDE THE NAMES OF EACH ROUTINE, PLUS
: 1725 1 : FORMAT STATEMENTS FOR PRINTING OUT OTHER INFORMATION.
: 1726 1 :--
: 1727 1
: 1728 1 DBM1 = uplit (#esciz'#N#N#N#ATESTING UNIT#:#D3#A IP REGISTER:#06#A PLATTER#:#D3#N'),
: 1729 1 DBM7 = uplit (#esciz'#N#ATEST 1 REGISTER EXISTENCE TEST'),
: 1730 1 DBM8 = uplit (#esciz'#N#ATEST 2 STEP 1 READ/WRITE POWERUP DIAGNOSTICS'),
: 1731 1 DBM9 = uplit (#esciz'#N#ATEST 5 STEP 1 THROUGH STEP 3 READ/WRITE TEST'),
: 1732 1 DBM10 = uplit (#esciz'#N#ATEST 3 DIAGNOSTIC WRAP TEST'),
: 1733 1 DBM11 = uplit (#esciz'#N#ATEST 4 VECTOR AND BR LEVEL TEST'),
: 1734 1 DBM12 = uplit (#esciz'#N#ATEST 6 PURGE AND POLL TEST'),
: 1735 1 DBM13 = uplit (#esciz'#N#ATEST 7 SMALL RING TEST'),
: 1736 1 DBM14 = uplit (#esciz'#N#ATEST 8 LARGE RING TEST'),
: 1737 1 DBM15 = uplit (#esciz'#N#ATEST 9 DM CODE OVERLAY TEST'),
: 1738 1 DBM16 = uplit (#esciz'#N#ATEST 10 NONEXISTENT MEMORY TEST'),
: 1739 1 DBM17 = uplit (#esciz'#N#ATEST 11 BUS ADDRESSING/DATA TEST A'),
: 1740 1 DBM18 = uplit (#esciz'#N#ATEST 12 BUS ADDRESSING/DATA TEST B'),
: 1741 1 DBM19 = uplit (#esciz'#N#ATEST 13 BLOCK TRANSFER TEST'),
: 1742 1 DBM20 = uplit (#esciz'#N#ATEST 14 SPIN UP HEAD LOAD SEQUENCE'),

```

```

: 1743 1 DBM21 = uplit (#esciz'#N#ATEST 15 SEQUENTIAL SEEK AND VERIFY'),
: 1744 1 DBM22 = uplit (#esciz'#N#ATEST 16 SAWTOOTH SEEK AND VERIFY'),
: 1745 1 DBM23 = uplit (#esciz'#N#ATEST 17 CONVERGING/DIVERGING SEEK AND VERIFY'),
: 1746 1 DBM24 = uplit (#esciz'#N#ATEST 18 TOGGLE SEEK AND VERIFY'),
: 1747 1 DBM25 = uplit (#esciz'#N#ATEST 19 HEAD SWITCH TEST'),
: 1748 1 DBM26 = uplit (#esciz'#N#ATEST 20 RANDOM SEEK AND VERIFY'),
: 1749 1 DBM27 = uplit (#esciz'#N#ATEST 21 SECTOR ACCESS TEST'),
: 1750 1 DBM28 = uplit (#esciz'#N#ATEST 22 CONTROLLER PROCESSING TIME'),
: 1751 1 DBM29 = uplit (#esciz'#N#ATEST 23 ONE TRACK SEEK TIME'),
: 1752 1 DBM30 = uplit (#esciz'#N#ATEST 24 AVERAGE SEEK TIME'),
: 1753 1 DBM31 = uplit (#esciz'#N#ATEST 25 FULL STROKE SEEK TIME'),
: 1754 1 DBM32 = uplit (#esciz'#N#ATEST 26 WRITE DATA TEST'),
: 1755 1 DBM36 = uplit (#esciz'#N#ATEST 27 OFFSET TOLERANCE TEST'),
: 1756 1 DBM37 = uplit (#esciz'#N#ATEST 28 AVERAGE ROTATIONAL TIME'),
: 1757 1 DBM38 = uplit (#esciz'#N#ATEST 29 WRITE PROTECT TEST'),
: 1758 1 DBM39 = uplit (#esciz'#N#A MANUAL INTERVENTION TEST NOT PERFORMED'),
: 1759 1 :
: 1760 1 : SYSTEM ERROR MESSAGES
: 1761 1 :
: 1762 1 MSG_01 = uplit (#esciz'#N#APOWER DELAY - WAITING'),
: 1763 1 ERR_01 = uplit (#esciz'#N#ATOO MANY UNITS'),
: 1764 1 ERR_02 = uplit (#esciz'#N#AND CLOCK WAS FOUND IN THE SYSTEM'),
: 1765 1 ERR_03 = uplit (#esciz'#N#AINCORRECT TRACK NUMBERS SELECTED'),
: 1766 1 :
: 1767 1 : FORMATTED ASCII STRINGS
: 1768 1 :
: 1769 1 FMT#C = uplit (#esciz'#N#N'),
: 1770 1 FMT1 = uplit (#esciz'#N#A REGISTER FAILED TO RESPOND AT ADDRESS: #06#N'),
: 1771 1 FMT2 = uplit (#esciz'#N#AADDRESS: #06#A EXPECTED: #06#A READ: #06#N'),
: 1772 1 FMT3 = uplit (#esciz'#N#ASTEP MASK = #02#A FAILING REGISTER = #06#A DATA = #06#N'),
: 1773 1 FMT4 = uplit (#esciz'#N#A PORT TYPE NUMBER = #02'),
: 1774 1 FMT5 = uplit (#esciz'#N#A PORT SPECIFIC INFO:/NV/QB/DI/OD/MP/ = #02'),
: 1775 1 FMT6 = uplit (#esciz'#N#A MICRO CODE: MODEL = #02#A VERSION = #02'),
: 1776 1 FMT7 = uplit (#esciz'#N#A XMT_BUF: #06#A REC_BUF: #06#N'),
: 1777 1 FMT7A = uplit (#esciz'#A XMT_DATA: #06#A REC_DATA: #06#N'),
: 1778 1 FMT8 = uplit (#esciz'#N#A UNIT COMES ONLINE IN: #02#A min. #02#A.#02#A sec.'),
: 1779 1 FMT9 = uplit (#esciz'#N#A STARTING TRACK: #04#A ENDING TRACK: #04#A DESIRED LBN: #06#N'),
: 1780 1 FMT10 = uplit (#esciz'#N#A UNIT: #04#A HEAD: #04#A TRACK: #04#N'),
: 1781 1 FMT11 = uplit (#esciz'#N#A NUMBER OF SEEKS (D): #06#A LBN: #06#N'),
: 1782 1 FMT12 = uplit (#esciz'#N#A MAX. OFFSET VALUE: #02#A.#01#A percent'),
: 1783 1 FMT13 = uplit (#esciz'#N#A RCSA ERROR STATUS: #06#N'),
: 1784 1 FMT14 = uplit (#esciz'#N#A END PACKET ERROR STATUS: #06#N'),
: 1785 1 FMT15 = uplit (#esciz'#N#A UNEXPECTED LOG PACKET ERROR STATUS: #06#N'),
: 1786 1 FMT16 = uplit (#esciz'#N#A WRITE DATA: #06#A READ DATA: #06'),
: 1787 1 FMT17 = uplit (#esciz'#N#A TRACK: #04#A SECTOR: #04#A HEAD: #04#N'),
: 1788 1 FMT18 = uplit (#esciz'#N#A EXPECTED SW = OFF ACTUAL SW = ON PLATTER # = #03'),
: 1789 1 FMT19 = uplit (#esciz'#N#A EXPECTED SW = ON ACTUAL SW = OFF PLATTER # = #03'),
: 1790 1 :VER:C
: 1791 1 FMT20 = uplit (#esciz'#N#A ERROR STATUS: #06#N'),
: 1792 1 FMT#A = uplit (#esciz'#N#A NUMBER OF RETRIES (D) =#04'),
: 1793 1 :
: 1794 1 : INIT ERROR MESSAGES
: 1795 1 :
: 1796 1 MSG_PWR = uplit (#esciz'#N#A WAIT - POWER FAIL RECOVERY'),
: 1797 1 MSG_1 = uplit (#esciz'#N#A RCSA FAILED TO RESPOND'),
: 1798 1 MSG_2 = uplit (#esciz'#N#A RCIP FAILED TO RESPOND'),
: 1799 1 MSG_7 = uplit (#esciz'#N#A TEST PATTERN ECHOED IN RCSA IS INCORRECT'),

```



```

: 1800 1 MSG_8 = uplit (#esciz'VECTOR AND BR LEVEL TEST FAILURE'),
: 1801 1 MSG_9 = uplit (#esciz'MOST DETECTED TIME OUT ERROR'),
: 1802 1 MSG_10 = uplit (#esciz'RING BUFFERS NOT CLEARED BY THE PORT'),
: 1803 1 MSG_11 = uplit (#esciz'STEP READ DATA DOES NOT MATCH'),
: 1804 1 MSG_13 = uplit (#esciz'PORT FATAL ERROR'),
: 1805 1 MSG_14 = uplit (#esciz'INIT STEP READ ERROR'),
: 1806 1 BUFF_ERR = uplit (#esciz'MEMORY BUFFER DOES NOT CONTAIN EXPECTED DATA'),
: 1807 1 DMC_ERR = uplit (#esciz'DM CODE RETURNED FAILURE CODE'),
: 1808 1 INI_MSG = uplit (#esciz'#N#A INTERRUPT AT VEC= #03#A BR LEVEL= #01'),
: 1809 1 END_MSG = uplit (#esciz'NO INTERRUPT FROM PORT / CONTROLLER'),
: 1810 1 BRERR = uplit (#esciz'#N#A BR LEVEL RECEIVED/TYPED IS INCORRECT !'),
: 1811 1 MSG_SEEK_ERR = uplit (#esciz'RC25 SEEK FAILURE'),
: 1812 1 MSG_HSWICH_ERR = uplit (#esciz'HEAD SWITCH FAILURE'),
: 1813 1 MSG_SAC_ERR = uplit (#esciz'SECTOR READ FAILURE'),
: 1814 1 MSG_COM_WPT = uplit (#esciz'WRITE PROTECT TEST FAILURE'),
: 1815 1 SK_FOR_ERR = uplit (#esciz'FORWARD SEEK ERROR'),
: 1816 1 SK_REV_ERR = uplit (#esciz'REVERSE SEEK ERROR'),
: 1817 1 SK_TOG_ERR = uplit (#esciz'TOGGLE SEEK ERROR'),
: 1818 1 SK_RAN_ERR = uplit (#esciz'RANDOM SEEK ERROR'),
: 1819 1 MSG_WRITE_ERR = uplit (#esciz'READ/WRITE TEST IN ERROR'),
: 1820 1 MSG_READ_ERR = uplit (#esciz'OFFSET READ ERROR'),
: 1821 1 MSG_GUS_ERR = uplit (#esciz'GET UNIT STATUS COMMAND FAILURE'),
: 1822 1 AVAIL_ERR = uplit (#esciz'AVAILABLE COMMAND FAILURE'),
: 1823 1
: 1824 1 MSG_AVE_TIME = uplit (#esciz'#N#AAVERAGE SEEK TIME WITH RANDOM LBN (ms) = #D3#A.#D2'),
: 1825 1 MES_SKO_TIME = uplit (#esciz'#N#AAVERAGE SEEK TIME WITH ORDERED LBN (ms) = #D3#A.#D2'),
: 1826 1 MSG_PRO_TIME = uplit (#esciz'#N#APROCESSING TIME (ms) = #D3#A.#D2'),
: 1827 1 MSG_SK_TIME = uplit (#esciz'#N#AONE TRACK SEEK TIME (ms) = #D3#A.#D2'),
: 1828 1 MG_SKF_TIME = uplit (#esciz'#N#AFULL TRACK SEEK TIME (ms) = #D3#A.#D2'),
: 1829 1 MSG_ROT_TIME = uplit (#esciz'#N#AAVERAGE ROTATIONAL TIME (ms) = #D3#A.#D2'),
: 1830 1
: 1831 1 AZT_READY_ERR = uplit (#esciz'RC25 UNIT DOES NOT COME ONLINE'),
: 1832 1 EXE_SUP_ERR = uplit (#esciz'EX_SUP PROG DUP COMMAND FAILURE'),
: 1833 1 SND_DATA_ERR = uplit (#esciz'SEND DATA DUP COMMAND FAILURE'),
: 1834 1 RE_DATA_ERR = uplit (#esciz'REC_DATA DUP COMMAND FAILURE'),
: 1835 1 !<BLF/PAGE>

```

```

: 1836 1      |
: 1837 1      |
: 1838 1      |
: 1839 1      |
: 1840 1      |
: 1841 1      |
: 1842 1      |
: 1843 1      |
: 1844 1      |
: 1845 1      |
: 1846 1      |
: 1847 1      |
: 1848 1      |
: 1849 1      |
: 1850 1      |
: 1851 1      |
: 1852 1      |
: 1853 1      |
: 1854 1      |
: 1855 1      |
: 1856 1      |
: 1857 1      |
: 1858 1      |
: 1859 1      |
: 1860 1      |
: 1861 1      |
: 1862 1      |
: 1863 1      |
: 1864 1      |

```

```

: |
: | Self-detected fatal port/controller errors
: |
: | PFE_STRUCT = uplit (
: |   uplit ( %esc:z' %N% A%FTLERR- UNRECOGNIZABLE ERROR CODE' ),
: |   uplit ( %esc:z' %N% A%FTLERR- ENVELOPE/PACKET READ (PARITY OR TIMEOUT)' ),
: |   uplit ( %esc:z' %N% A%FTLERR- ENVELOPE/PACKET WRITE (PARITY OR TIMEOUT)' ),
: |   uplit ( %esc:z' %N% A%FTLERR- CONTROLLER ROM AND RAM PARITY' ),
: |   uplit ( %esc:z' %N% A%FTLERR- CONTROLLER RAM PARITY' ),
: |   uplit ( %esc:z' %N% A%FTLERR- CONTROLLER ROM PARITY' ),
: |   uplit ( %esc:z' %N% A%FTLERR- RING READ (PARITY OR TIMEOUT)' ),
: |   uplit ( %esc:z' %N% A%FTLERR- RING WRITE (PARITY OR TIMEOUT)' ),
: |   uplit ( %esc:z' %N% A%FTLERR- INTERRUPT MASTER' ),
: |   uplit ( %esc:z' %N% A%FTLERR- HOST ACCESS TIMEOUT' ),
: |   uplit ( %esc:z' %N% A%FTLERR- CREDIT LIMIT EXCEEDED' ),
: |   uplit ( %esc:z' %N% A%FTLERR- BUS MASTER ERROR' ),
: |   uplit ( %esc:z' %N% A%FTLERR- DIAGNOSTIC CONTROLLER FATAL ERROR' ),
: |   uplit ( %esc:z' %N% A%FTLERR- INSTRUCTION LOOP TIMEOUT' ),
: |   uplit ( %esc:z' %N% A%FTLERR- INVALID CONNECTION IDENTIFIER' ),
: |   uplit ( %esc:z' %N% A%FTLERR- INTERRUPT WRITE' ),
: |   uplit ( %esc:z' %N% A%FTLERR- MAINTENANCE READ/WRITE INVALID REGION IDENTIFIER' ),
: |   uplit ( %esc:z' %N% A%FTLERR- MAINTENANCE WRITE LOAD TO NON-LOADABLE CONTROLLER' ),
: |   uplit ( %esc:z' %N% A%FTLERR- CONTROLLER RAM ERROR (NON-PARITY)' ),
: |   uplit ( %esc:z' %N% A%FTLERR- INIT SEQUENCE ERROR' ),
: |   uplit ( %esc:z' %N% A%FTLERR- HIGH LEVEL PROTOCOL INCOMPATIBILITY ERROR' ),
: |   uplit ( %esc:z' %N% A%FTLERR- PURGE/POLL HARDWARE FAILURE ' ),
: |   uplit ( %esc:z' %N% A%FTLERR- MAPPING REGISTER READ ERROR (PARITY OR TIMEOUT)' )
: |   ) : vector [23],
: |
: | ! <BLF/PAGE>

```

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0040
Page 12
(9)

```

: 1865 1      !
: 1866 1      ! Error message structure
: 1867 1      !
: 1868 1      ! MSG_STRUCT = uplit (
: 1869 1      ! uplit (%esciz'%N%A$FTLERR- RESPONSE STATUS ERROR:%S'),
: 1870 1      ! uplit (%esciz'%N%A$FTLERR- SUPERVISOR SERVICE CALL FAILED'),
: 1871 1      ! uplit (%esciz'%N%A$FTLERR- PORT/CONTROLLER TIMEOUT ERROR'),
: 1872 1      ! uplit (%esciz'%N%A$FTLERR- UNKNOWN RETURN STATUS CODE')) : vector [4],
: 1873 1      !<blf/page>

```

```

: 1874 1      :
: 1875 1      : Self-detected fatal port/controller errors
: 1876 1      :
: 1877 1      : RC STRUCTURE = uplit (
: 1878 1      : uplit (#asciz' #N#A$FTLERR- VAX READ/WRITE ERROR ON INTERRUPT'),
: 1879 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.BFIL'),
: 1880 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.BMTY'),
: 1881 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.ALOC'),
: 1882 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT SERVO ENTRY (PIP SET)'),
: 1883 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT SERVO ENTRY (ERR SET)'),
: 1884 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.SEND'),
: 1885 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.RECV'),
: 1886 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.ATTN'),
: 1887 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.ONLN'),
: 1888 1      : uplit (#asciz' #N#A$FTLERR- ILLEGAL D REQUEST (U.QDRQ)'),
: 1889 1      : uplit (#asciz' #N#A$FTLERR- FENCE-POST ERROR AT PROTAB'),
: 1890 1      : uplit (#asciz' #N#A$FTLERR- BAD PACKET DEQUEUED AT U.DONE'),
: 1891 1      : uplit (#asciz' #N#A$FTLERR- UNEXPLAINED D-PROC SUSPENSION (U..TDS)'),
: 1892 1      : uplit (#asciz' #N#A$FTLERR- DUP PACKET D-Q FAILED (XFC 34/35)'),
: 1893 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.HTST'),
: 1894 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.SEKO'),
: 1895 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT U.CKSV'),
: 1896 1      : uplit (#asciz' #N#A$FTLERR- D.OPCD FOUND ILLEGAL OPCODE'),
: 1897 1      : uplit (#asciz' #N#A$FTLERR- D.CSF FOUND ILLEGAL OPCODE'),
: 1898 1      : uplit (#asciz' #N#A$FTLERR- UNKNOWN BAD DRIVE STATUS AT D.DSTS'),
: 1899 1      : uplit (#asciz' #N#A$FTLERR- ILLEGAL XFC EXECUTED BY DM'),
: 1900 1      : uplit (#asciz' #N#A$FTLERR- D PICKED UP A ZERO SCB.DB'),
: 1901 1      : uplit (#asciz' #N#A$FTLERR- INCONSISTENCY AT D IDLE LOOP'),
: 1902 1      : uplit (#asciz' #N#A$FTLERR- DM WORD COUNT ERROR ON HOST DMA/SEND/RECV'),
: 1903 1      : uplit (#asciz' #N#A$FTLERR- UNKNOWN DISPLAY FAULT CODE AT D.DFLT'),
: 1904 1      : uplit (#asciz' #N#A$FTLERR- DRIVE NOT FAULTING IN P.OFLN STATE'),
: 1905 1      : uplit (#asciz' #N#A$FTLERR- U POWER UP DIAGNOSTICS FAILED'),
: 1906 1      : uplit (#asciz' #N#A$FTLERR- D POWER UP DIAGNOSTICS FAILED'),
: 1907 1      : uplit (#asciz' #N#A$FTLERR- ADAPTER CARD FAILURE'),
: 1908 1      : uplit (#asciz' #N#A$FTLERR- EC.TMR TIMED OUT'),
: 1909 1      : uplit (#asciz' #N#A$FTLERR- U.SEND/U.RECV RING READ INCONSISTENCY'),
: 1910 1      : uplit (#asciz' #N#A$FTLERR- UNKNOWN WAITRV REASON AT D.RVCT'),
: 1911 1      : uplit (#asciz' #N#A$FTLERR- D.ARCS DID NOT FIND CLOSEST UNDONE ZONE'),
: 1912 1      : uplit (#asciz' #N#A$FTLERR- U.SEEK FOUND SEEK TO ILLEGAL TRACK'),
: 1913 1      : uplit (#asciz' #N#A$FTLERR- U.HTST INIT DIAG DMA WRITE FAILED'),
: 1914 1      : uplit (#asciz' #N#A$FTLERR- U.HTST INIT DIAG DMA COMPARE FAILED'),
: 1915 1      : uplit (#asciz' #N#A$FTLERR- U.SYDR FOUND SS.DER SET AND SS.SPN NOT SET'),
: 1916 1      : uplit (#asciz' #N#A$FTLERR- MASTER DRIVES ACLO ASSERTED')
: 1917 1      : ) : vector [39].
: 1918 1      : !<blf/page>

```

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0042
Page 14
(11)

```
: 1919 1      |
: 1920 1      |      | Dup return status codes
: 1921 1      |      |
: 1922 1      |      | SDUP_STRUCT = uplit (
: 1923 1      |      | uplit (%asciz'%A SUCCESSFUL%N'),
: 1924 1      |      | uplit (%asciz'%INVALID COMMAND%N'),
: 1925 1      |      | uplit (%asciz'%NO REGION AVAILABLE%N'),
: 1926 1      |      | uplit (%asciz'%NO REGION SUITABLE%N'),
: 1927 1      |      | uplit (%asciz'%PROGRAM NOT KNOWN%N'),
: 1928 1      |      | uplit (%asciz'%LOAD FAILURE%N'),
: 1929 1      |      | uplit (%asciz'%STANDALONE%N')
: 1930 1      |      | ) : vector [7].
: 1931 1      |      | !<blf/page>
```

```

: 1932 1      |
: 1933 1      |      | MSCP return status codes
: 1934 1      |      |
: 1935 1      |      | SMSCP_STRUCT = uplit (
: 1936 1      |      | uplit (%asciz'ASUCCESS#N'),
: 1937 1      |      | uplit (%asciz'AINVALID COMMAND#N'),
: 1938 1      |      | uplit (%asciz'ACOMMAND ABORTED#N'),
: 1939 1      |      | uplit (%asciz'AUNIT-OFFLINE#N'),
: 1940 1      |      | uplit (%asciz'AUNIT-AVAILABLE#N'),
: 1941 1      |      | uplit (%asciz'AMEDIA FORMAT ERROR#N'),
: 1942 1      |      | uplit (%asciz'AWRITE PROTECTED#N'),
: 1943 1      |      | uplit (%asciz'ACOMPARE ERROR#N'),
: 1944 1      |      | uplit (%asciz'ADATA ERROR#N'),
: 1945 1      |      | uplit (%asciz'AHOST BUFFER ACCESS ERROR#N'),
: 1946 1      |      | uplit (%asciz'ACONTROLLER ERROR#N'),
: 1947 1      |      | uplit (%asciz'ADRIIVE ERROR#N'),
: 1948 1      |      | uplit (%asciz'AMESSAGE FROM AN INTERNAL DIAGNOSTIC#N')
: 1949 1      |      | ) : vector [13];
: 1950 1      |
: 1951 1      |      | end
: 1952 1      |
: 1953 0      |      | eludom

```

```

.TITLE ZRCFB1 CZRCFC0 RC25 FR END TEST
.IDENT /V03.0/

```

000000				.PSECT	AA\$CODE, RO
000000	103	132	122	L\$NAME::	.ASCII /CZR/
000003	103	106	040		.ASCII /CF /
000006	000				.BYTE 0
000007	000				.BYTE 0
000010				L\$REV::	
000010	101				.ASCII /A/
000011	060				.ASCII /O/
000012	000000G			L\$UNIT::	.WORD T\$PTHV
000014	000170			L\$TIML::	.WORD 170
000016	000000G			L\$HPCP::	.WORD L\$HARD
000020	000000G			L\$SPCP::	.WORD L\$SOFT
000022	000230'			L\$HPTP::	.WORD L\$HW
000024	000244'			L\$SPTP::	.WORD L\$SW
000026	000000G			L\$LADP::	.WORD L\$LAST
000030	000000			L\$STA::	.WORD 0
000032	000000			L\$CO::	.WORD 0
000034	000000			L\$DTYP::	.WORD 0
000036	000000			L\$APT::	.WORD 0
000040	000124'			L\$DTP::	.WORD L\$DISPATCH
000042	000000			L\$PRIO::	.WORD 0
000044	000000			L\$ENVI::	.WORD 0
000046	000000			L\$EXP1::	.WORD 0
000050				L\$MREV::	
000050	003				.BYTE 3
000051	003				.BYTE 3
000052	000000			L\$EF::	.WORD 0
000054	000000				.WORD 0
000056	000000			L\$SPC::	.WORD 0

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0044
Page 16
(12)

000060	000000G	L\$DEVP::WORD	L\$DVTYP
000062	000000G	L\$REPP::WORD	L\$RPT
000064	000000	L\$EXP4::WORD	0
000066	000000	L\$EXP5::WORD	0
000070	000000G	L\$AUT::WORD	L\$AU
000072	000000G	L\$DUT::WORD	L\$DU
000074	000000	L\$LUN::WORD	0
000076	000000G	L\$DESP::WORD	L\$DESC
000100	104035	L\$LOAD::WORD	-73743
000102	000216'	L\$ETP::WORD	L\$ERRTBL
000104	000000G	L\$ICP::WORD	L\$INIT
000106	000000G	L\$CCP::WORD	L\$CLEAN
000110	000000G	L\$ACP::WORD	L\$AUTO
000112	000266'	L\$PRT::WORD	L\$PROT
000114	000000	L\$TEST::WORD	0
000116	000000	L\$DLY::WORD	0
000120	000000	L\$HIME::WORD	0
000122	000035	D\$PCNT::WORD	35
000124	000000G	L\$DISPATCH::	
		.WORD	T1
000126	000000G	.WORD	T2
000130	000000G	.WORD	T3
000132	000000G	.WORD	T4
000134	000000G	.WORD	T5
000136	000000G	.WORD	T6
000140	000000G	.WORD	T7
000142	000000G	.WORD	T8
000144	000000G	.WORD	T9
000146	000000G	.WORD	T10
000150	000000G	.WORD	T11
000152	000000G	.WORD	T12
000154	000000G	.WORD	T13
000156	000000G	.WORD	T14
000160	000000G	.WORD	T15
000162	000000G	.WORD	T16
000164	000000G	.WORD	T17
000166	000000G	.WORD	T18
000170	000000G	.WORD	T19
000172	000000G	.WORD	T20
000174	000000G	.WORD	T21
000176	000000G	.WORD	T22
000200	000000G	.WORD	T23
000202	000000G	.WORD	T24
000204	000000G	.WORD	T25
000206	000000G	.WORD	T26
000210	000000G	.WORD	T27
000212	000000G	.WORD	T28
000214	000000G	.WORD	T29
000216		ERRTYP::BLKW	1
000220		ERRNBR::BLKW	1
000222		ERRMSG::BLKW	1
000224		ERRBLK::BLKW	1
000226	000000C	L\$HWLEN::	
		.WORD	<<L\$NDHW-L\$HWLEN>/2>
000230	172150	P.IP.ADDRESS::	
		.WORD	-5630
000232	000154	P.VECTOR::	

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0045
Page 17
(12)

000234	000005	P.BR.LEVEL::	.WORD	154
000236	000000	P.UNIT.NUMBER::	.WORD	5
000240		L\$NDHW::	.WORD	0
000242	000000C	L\$SWLEN::	.WORD	1
000244	000001	SWP.TOP::	.WORD	<<L\$NDSW-L\$SWLEN>/2>
000246	000000	SWP.LIMIT::	.WORD	1
000250	000000	SWP.START::	.WORD	0
000252	001464	SWP.END::	.WORD	0
000254	000000	SWP.RETRIES::	.WORD	1464
000256	000000	SWP.CONTINUE::	.WORD	0
000260	000000	SWP.MANUAL::	.WORD	0
000262	000001	SWP.TRACE::	.WORD	0
000264		L\$NDSW::	.WORD	1
000266	177777	L\$PROT::	.WORD	1
000270	177777		.WORD	-1
000272	177777		.WORD	-1

000000				.PSECT	\$PLIT\$, RO, D, GBL	
000000	000002			.WORD	2	
000002	005640'			P.AAA: .WORD	P4	; Plit count word
000004	005642'			.WORD	P5	
000006	045	101	106	P.AAB: .ASCII	/AF/	
000011	101	111	114	.ASCII	/AIL/	
000014	111	116	107	.ASCII	/ING/	
000017	040	106	122	.ASCII	/FR/	
000022	125	040	075	.ASCII	/U =/	
000025	040	045	124	.ASCII	/T/	
000030	045	104	063	.ASCII	/D3/	
000033	045	116	000	.ASCII	/N/<00>	
000036	101	104	101	P.AAC: .ASCII	/ADA/	
000041	120	124	117	.ASCII	/PTQ/	
000044	122	040	102	.ASCII	/R B/	
000047	117	101	122	.ASCII	/OAR/	
000052	104	040	106	.ASCII	/D F/	
000055	117	122	040	.ASCII	/OR /	
000060	125	116	111	.ASCII	/UNI/	
000063	124	040	043	.ASCII	/T #/	
000066	072	000		.ASCII	/:/<00>	
000070	103	117	116	P.AAD: .ASCII	/CON/	
000073	124	122	117	.ASCII	/TRO/	
000076	114	105	122	.ASCII	/LER/	
000101	040	102	117	.ASCII	/BO/	
000104	101	122	104	.ASCII	/ARD/	

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0046
Page 18
(12)

000107	040	106	117	.ASCII	/ FO/
000112	122	040	125	.ASCII	/R U/
000115	116	111	124	.ASCII	/NIT/
000120	040	043	072	.ASCII	/ #:/
000123	000			.ASCII	<00>
000124	104	122	111	P.AAE:	.ASCII /DRI/
000127	126	105	040	.ASCII	/VE /
000132	102	117	101	.ASCII	/BOA/
000135	122	104	040	.ASCII	/RD /
000140	106	117	122	.ASCII	/FOR/
000143	040	125	116	.ASCII	/ UN/
000146	111	124	040	.ASCII	/IT /
000151	043	072	000	.ASCII	/#:/<00>
000154	115	105	103	P.AAF:	.ASCII /MEC/
000157	110	101	116	.ASCII	/HAN/
000162	111	103	040	.ASCII	/IC /
000165	123	105	124	.ASCII	/SET/
000170	040	106	117	.ASCII	/ FO/
000173	122	040	125	.ASCII	/R U/
000176	116	111	124	.ASCII	/NIT/
000201	040	043	072	.ASCII	/ #:/
000204	000	000		.ASCII	<00><00>
000206	111	120	040	P.AAG:	.ASCII /IP /
000211	101	104	104	.ASCII	/ADD/
000214	122	105	123	.ASCII	/RES/
000217	123	000	000	.ASCII	/S/<00><00>
000222	126	105	103	P.AAH:	.ASCII /VEC/
000225	124	117	122	.ASCII	/TOR/
000230	000	000		.ASCII	<00><00>
000232	102	122	040	P.AAI:	.ASCII /BR /
000235	114	105	126	.ASCII	/LEV/
000240	105	114	000	.ASCII	/EL/<00>
000243	000			.ASCII	<00>
000244	120	114	101	P.AAJ:	.ASCII /PLA/
000247	124	124	105	.ASCII	/TTE/
000252	122	040	101	.ASCII	/R A/
000255	104	104	122	.ASCII	/DDR/
000260	105	123	123	.ASCII	/ESS/
000263	133	105	123	.ASCII	/[ES/
000266	135	000		.ASCII	/]/<00>
000270	125	123	105	P.AAK:	.ASCII /USE/
000273	040	124	117	.ASCII	/ TO/
000276	120	040	123	.ASCII	/P S/
000301	125	122	106	.ASCII	/URF/
000304	101	103	105	.ASCII	/ACE/
000307	040	106	117	.ASCII	/ FO/
000312	122	040	123	.ASCII	/R S/
000315	111	116	107	.ASCII	/ING/
000320	114	105	040	.ASCII	/LE /
000323	123	125	122	.ASCII	/SUR/
000326	106	101	103	.ASCII	/FAC/
000331	105	040	124	.ASCII	/E T/
000334	105	123	124	.ASCII	/EST/
000337	123	000	000	.ASCII	/S/<00><00>
000342	104	117	040	P.AAL:	.ASCII /DO /
000345	131	117	125	.ASCII	/YOU/
000350	040	127	111	.ASCII	/ WI/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.816;1

SEQ 0047
Page 19
(12)

000353	123	110	040	.ASCII	/SH /
000356	124	117	040	.ASCII	/TO /
000361	114	111	115	.ASCII	/LIM/
000364	111	124	040	.ASCII	/IT /
000367	101	122	105	.ASCII	/ARE/
000372	101	040	124	.ASCII	/A T/
000375	105	123	124	.ASCII	/EST/
000400	105	104	040	.ASCII	/ED /
000403	111	116	040	.ASCII	/IN /
000406	124	105	123	.ASCII	/TES/
000411	124	123	040	.ASCII	/TS /
000414	061	065	055	.ASCII	/15-/
000417	061	070	000	.ASCII	/18/<00>
000422	123	124	101	P.AAM:	.ASCII /STA/
000425	122	124	111	.ASCII	/RTI/
000430	116	107	040	.ASCII	/NG /
000433	124	122	101	.ASCII	/TRA/
000436	103	113	000	.ASCII	/CK/<00>
000441	000			.ASCII	<00>
000442	105	116	104	P.AAN:	.ASCII /END/
000445	111	116	107	.ASCII	/ING/
000450	040	124	122	.ASCII	/ TR/
000453	101	103	113	.ASCII	/ACK/
000456	000	000		.ASCII	<00><00>
000460	104	117	040	P.AAQ:	.ASCII /DO /
000463	131	117	125	.ASCII	/YOU/
000466	040	127	111	.ASCII	/ WI/
000471	123	110	040	.ASCII	/SH /
000474	124	117	040	.ASCII	/TO /
000477	104	117	040	.ASCII	/DO /
000502	124	110	105	.ASCII	/THE/
000505	040	115	101	.ASCII	/ MA/
000510	116	125	101	.ASCII	/NUA/
000513	114	040	111	.ASCII	/L I/
000516	116	124	105	.ASCII	/NTE/
000521	122	126	105	.ASCII	/RVE/
000524	116	124	111	.ASCII	/NTI/
000527	117	116	040	.ASCII	/ON /
000532	124	105	123	.ASCII	/TES/
000535	124	077	000	.ASCII	/T?/<00>
000540	104	117	040	P.AAP:	.ASCII /DO /
000543	131	117	125	.ASCII	/YOU/
000546	040	127	111	.ASCII	/ WI/
000551	123	110	040	.ASCII	/SH /
000554	124	122	101	.ASCII	/TRA/
000557	103	105	040	.ASCII	/CE /
000562	115	117	104	.ASCII	/MOD/
000565	105	077	000	.ASCII	/E?/<00>
000570	104	117	040	P.AAQ:	.ASCII /DO /
000573	131	117	125	.ASCII	/YOU/
000576	040	127	111	.ASCII	/ WI/
000601	123	110	040	.ASCII	/SH /
000604	124	117	040	.ASCII	/TO /
000607	103	117	116	.ASCII	/CON/
000612	124	111	116	.ASCII	/TIN/
000615	125	105	040	.ASCII	/UE /
000620	124	105	123	.ASCII	/TES/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0048
Page 20
(12)

000623	124	111	116	.ASCII	/TIN/
000626	107	040	101	.ASCII	/G A/
000631	106	124	105	.ASCII	/FTE/
000634	122	040	122	.ASCII	/R R/
000637	105	124	122	.ASCII	/ETR/
000642	111	105	123	.ASCII	/IES/
000645	077	000	000	.ASCII	/?/<00><00>
000650	116	125	115	P.AAR: .ASCII	/NUM/
000653	102	105	122	.ASCII	/BER/
000656	040	117	106	.ASCII	/ OF/
000661	040	122	105	.ASCII	/ RE/
000664	124	122	111	.ASCII	/TRI/
000667	105	123	040	.ASCII	/ES /
000672	106	117	122	.ASCII	/FOR/
000675	040	124	105	.ASCII	/ TE/
000700	123	124	040	.ASCII	/ST /
000703	111	106	040	.ASCII	/IF /
000706	105	122	122	.ASCII	/ERR/
000711	117	122	040	.ASCII	/OR /
000714	117	103	103	.ASCII	/OCC/
000717	125	122	105	.ASCII	/URE/
000722	104	000		.ASCII	/D/<00>
000724	124	125	122	P.AAS: .ASCII	/TUR/
000727	116	040	117	.ASCII	/N O/
000732	106	106	040	.ASCII	/FF /
000735	127	122	111	.ASCII	/WRI/
000740	124	105	040	.ASCII	/TE /
000743	120	122	117	.ASCII	/PRO/
000746	124	105	103	.ASCII	/TEC/
000751	124	040	123	.ASCII	/T S/
000754	127	111	124	.ASCII	/WIT/
000757	103	110	040	.ASCII	/CH /
000762	101	116	104	.ASCII	/AND/
000765	040	104	117	.ASCII	/ DO/
000770	040	074	103	.ASCII	/ <C/
000773	122	076	000	.ASCII	/R>/<00>
000776	124	125	122	P.AAT: .ASCII	/TUR/
001001	116	040	117	.ASCII	/N O/
001004	116	040	127	.ASCII	/N W/
001007	122	111	124	.ASCII	/RIT/
001012	105	040	120	.ASCII	/E P/
001015	122	117	124	.ASCII	/ROT/
001020	105	103	124	.ASCII	/ECT/
001023	040	123	127	.ASCII	/ SW/
001026	111	124	103	.ASCII	/ITC/
001031	110	040	101	.ASCII	/H A/
001034	116	104	040	.ASCII	/ND /
001037	104	117	040	.ASCII	/DO /
001042	074	103	122	.ASCII	/<CR/
001045	076	000	000	.ASCII	/>/<00><00>
001050	045	116	045	P.AAU: .ASCII	/N#N/
001053	116	045	116	.ASCII	/N#N/
001056	045	101	124	.ASCII	/#AT/
001061	105	123	124	.ASCII	/EST/
001064	111	116	107	.ASCII	/ING/
001067	040	125	116	.ASCII	/ UN/
001072	111	124	043	.ASCII	/IT#/

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0049
Page 21
(12)

001075	072	045	104	.ASCII	/: #D/
001100	063	045	101	.ASCII	/3 #A/
001103	040	040	111	.ASCII	/ I/
001106	120	137	122	.ASCII	/P R/
001111	105	107	111	.ASCII	/EGI/
001114	123	124	105	.ASCII	/STE/
001117	122	072	045	.ASCII	/R: #/
001122	117	066	045	.ASCII	/06 #/
001125	101	040	040	.ASCII	/A /
001130	120	114	101	.ASCII	/PLA/
001133	124	124	105	.ASCII	/TTE/
001136	122	043	072	.ASCII	/R#: /
001141	045	104	063	.ASCII	/ #D3/
001144	045	116	000	.ASCII	/ #N/ <00>
001147	000			.ASCII	<00>
001150	045	116	045	P.AAV: .ASCII	/ #N #/
001153	101	124	105	.ASCII	/ATE/
001156	123	124	040	.ASCII	/ST /
001161	040	061	040	.ASCII	/ 1 /
001164	122	105	107	.ASCII	/REG/
001167	111	123	124	.ASCII	/IST/
001172	105	122	040	.ASCII	/ER /
001175	105	130	111	.ASCII	/EXI/
001200	123	124	105	.ASCII	/STE/
001203	116	103	105	.ASCII	/NCE/
001206	040	124	105	.ASCII	/ TE/
001211	123	124	000	.ASCII	/ST/ <00>
001214	045	116	045	P.AAW: .ASCII	/ #N #/
001217	101	124	105	.ASCII	/ATE/
001222	123	124	040	.ASCII	/ST /
001225	040	062	040	.ASCII	/ 2 /
001230	123	124	105	.ASCII	/STE/
001233	120	040	061	.ASCII	/P 1/
001236	040	122	105	.ASCII	/ RE/
001241	101	104	057	.ASCII	/AD/ <57>
001244	127	122	111	.ASCII	/WRI/
001247	124	105	040	.ASCII	/TE /
001252	120	117	127	.ASCII	/POW/
001255	105	122	125	.ASCII	/ERU/
001260	120	040	104	.ASCII	/P D/
001263	111	101	107	.ASCII	/IAG/
001266	116	117	123	.ASCII	/NOS/
001271	124	111	103	.ASCII	/TIC/
001274	123	000		.ASCII	/S/ <00>
001276	045	116	045	P.AAX: .ASCII	/ #N #/
001301	101	124	105	.ASCII	/ATE/
001304	123	124	040	.ASCII	/ST /
001307	040	065	040	.ASCII	/ 5 /
001312	123	124	105	.ASCII	/STE/
001315	120	040	061	.ASCII	/P 1/
001320	040	124	110	.ASCII	/ TH/
001323	122	117	125	.ASCII	/ROU/
001326	107	110	040	.ASCII	/GH /
001331	123	124	105	.ASCII	/STE/
001334	120	040	063	.ASCII	/P 3/
001337	040	122	105	.ASCII	/ RE/
001342	101	104	057	.ASCII	/AD/ <57>

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.816;1

SEQ 0050
Page 22
(12)

001345	127	122	111	.ASCII	/WRI/	
001350	124	105	040	.ASCII	/TE /	
001353	124	105	123	.ASCII	/TES/	
001356	124	000		.ASCII	/T/<00>	
001360	045	116	045	P.AAY:	.ASCII	/NNS/
001363	101	124	105	.ASCII	/ATE/	
001366	123	124	040	.ASCII	/ST /	
001371	040	063	040	.ASCII	/ 3 /	
001374	104	111	101	.ASCII	/DIA/	
001377	107	116	117	.ASCII	/GNO/	
001402	123	124	111	.ASCII	/STI/	
001405	103	040	127	.ASCII	/C W/	
001410	122	101	120	.ASCII	/RAP/	
001413	040	124	105	.ASCII	/ TE/	
001416	123	124	000	.ASCII	/ST/<00>	
001421	000			.ASCII	<00>	
001422	045	116	045	P.AAZ:	.ASCII	/NNS/
001425	101	124	105	.ASCII	/ATE/	
001430	123	124	040	.ASCII	/ST /	
001433	040	064	040	.ASCII	/ 4 /	
001436	126	105	103	.ASCII	/VEC/	
001441	124	117	122	.ASCII	/TOR/	
001444	040	101	116	.ASCII	/ AN/	
001447	104	040	102	.ASCII	/D B/	
001452	122	040	114	.ASCII	/R L/	
001455	105	126	105	.ASCII	/EVE/	
001460	114	040	124	.ASCII	/L T/	
001463	105	123	124	.ASCII	/EST/	
001466	000	000		.ASCII	<00><00>	
001470	045	116	045	P.ABA:	.ASCII	/NNS/
001473	101	124	105	.ASCII	/ATE/	
001476	123	124	040	.ASCII	/ST /	
001501	040	066	040	.ASCII	/ 6 /	
001504	120	125	122	.ASCII	/PUR/	
001507	107	105	040	.ASCII	/GE /	
001512	101	116	104	.ASCII	/AND/	
001515	040	120	117	.ASCII	/ PO/	
001520	114	114	040	.ASCII	/LL /	
001523	124	105	123	.ASCII	/TES/	
001526	124	000		.ASCII	/T/<00>	
001530	045	116	045	P.ABB:	.ASCII	/NNS/
001533	101	124	105	.ASCII	/ATE/	
001536	123	124	040	.ASCII	/ST /	
001541	040	067	040	.ASCII	/ 7 /	
001544	123	115	101	.ASCII	/SMA/	
001547	114	114	040	.ASCII	/LL /	
001552	122	111	116	.ASCII	/RIN/	
001555	107	040	124	.ASCII	/G T/	
001560	105	123	124	.ASCII	/EST/	
001563	000			.ASCII	<00>	
001564	045	116	045	P.ABC:	.ASCII	/NNS/
001567	101	124	105	.ASCII	/ATE/	
001572	123	124	040	.ASCII	/ST /	
001575	040	070	040	.ASCII	/ 8 /	
001600	114	101	122	.ASCII	/LAR/	
001603	107	105	040	.ASCII	/GE /	
001606	122	111	116	.ASCII	/RIN/	

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC1.B16;1

SEQ 0051
Page 23
(12)

001611	107	040	124		.ASCII	/G T/
001614	105	123	124		.ASCII	/EST/
001617	000				.ASCII	<00>
001620	045	116	045	P.ABD:	.ASCII	/NNS/
001623	101	124	105		.ASCII	/ATE/
001626	123	124	040		.ASCII	/ST /
001631	040	071	040		.ASCII	/ 9 /
001634	104	115	040		.ASCII	/DM /
001637	103	117	104		.ASCII	/COD/
001642	105	040	117		.ASCII	/E O/
001645	126	105	122		.ASCII	/VER/
001650	114	101	131		.ASCII	/LAY/
001653	040	124	105		.ASCII	/ TE/
001656	123	124	000		.ASCII	/ST/<00>
001661	000				.ASCII	<00>
001662	045	116	045	P.ABE:	.ASCII	/NNS/
001665	101	124	105		.ASCII	/ATE/
001670	123	124	040		.ASCII	/ST /
001673	061	060	040		.ASCII	/10 /
001676	116	117	116		.ASCII	/NON/
001701	105	130	111		.ASCII	/EXI/
001704	123	124	105		.ASCII	/STE/
001707	116	124	040		.ASCII	/NT /
001712	115	105	115		.ASCII	/MEM/
001715	117	122	131		.ASCII	/ORY/
001720	040	124	105		.ASCII	/ TE/
001723	123	124	000		.ASCII	/ST/<00>
001726	045	116	045	P.ABF:	.ASCII	/NNS/
001731	101	124	105		.ASCII	/ATE/
001734	123	124	040		.ASCII	/ST /
001737	061	061	040		.ASCII	/11 /
001742	102	125	123		.ASCII	/BUS/
001745	040	101	104		.ASCII	/ AD/
001750	104	122	105		.ASCII	/DRE/
001753	123	123	111		.ASCII	/SSI/
001756	116	107	057		.ASCII	/NG/<57>
001761	104	101	124		.ASCII	/DAT/
001764	101	040	124		.ASCII	/A T/
001767	105	123	124		.ASCII	/EST/
001772	040	101	000		.ASCII	/ A/<00>
001775	000				.ASCII	<00>
001776	045	116	045	P.ABG:	.ASCII	/NNS/
002001	101	124	105		.ASCII	/ATE/
002004	123	124	040		.ASCII	/ST /
002007	061	062	040		.ASCII	/12 /
002012	102	125	123		.ASCII	/BUS/
002015	040	101	104		.ASCII	/ AD/
002020	104	122	105		.ASCII	/DRE/
002023	123	123	111		.ASCII	/SSI/
002026	116	107	057		.ASCII	/NG/<57>
002031	104	101	124		.ASCII	/DAT/
002034	101	040	124		.ASCII	/A T/
002037	105	123	124		.ASCII	/EST/
002042	040	102	000		.ASCII	/ B/<00>
002045	000				.ASCII	<00>
002046	045	116	045	P.ABH:	.ASCII	/NNS/
002051	101	124	105		.ASCII	/ATE/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0052
Page 24
(12)

002054	123	124	040	.ASCII	/ST /	
002057	061	063	040	.ASCII	/13 /	
002062	102	114	117	.ASCII	/BLO/	
002065	103	113	040	.ASCII	/CK /	
002070	124	122	101	.ASCII	/TRA/	
002073	116	123	106	.ASCII	/NSF/	
002076	105	122	040	.ASCII	/ER /	
002101	124	105	123	.ASCII	/TES/	
002104	124	000		.ASCII	/T/<00>	
002106	045	116	045	P.ABI:	.ASCII	/N# /
002111	101	124	105	.ASCII	/ATE/	
002114	123	124	040	.ASCII	/ST /	
002117	061	064	040	.ASCII	/14 /	
002122	123	120	111	.ASCII	/SPI/	
002125	116	040	125	.ASCII	/N U/	
002130	120	040	110	.ASCII	/P H/	
002133	105	101	104	.ASCII	/EAD/	
002136	040	114	117	.ASCII	/ LO/	
002141	101	104	040	.ASCII	/AD /	
002144	123	105	121	.ASCII	/SEQ/	
002147	125	105	116	.ASCII	/UEN/	
002152	103	105	000	.ASCII	/CE/<00>	
002155	000			.ASCII	<00>	
002156	045	116	045	P.ABJ:	.ASCII	/N# /
002161	101	124	105	.ASCII	/ATE/	
002164	123	124	040	.ASCII	/ST /	
002167	061	065	040	.ASCII	/15 /	
002172	123	105	121	.ASCII	/SEQ/	
002175	125	105	116	.ASCII	/UEN/	
002200	124	111	101	.ASCII	/TIA/	
002203	114	040	123	.ASCII	/L S/	
002206	105	105	113	.ASCII	/EEK/	
002211	040	101	116	.ASCII	/ AN/	
002214	104	040	126	.ASCII	/D V/	
002217	105	122	111	.ASCII	/ERI/	
002222	106	131	000	.ASCII	/FY/<00>	
002225	000			.ASCII	<00>	
002226	045	116	045	P.ABK:	.ASCII	/N# /
002231	101	124	105	.ASCII	/ATE/	
002234	123	124	040	.ASCII	/ST /	
002237	061	066	040	.ASCII	/16 /	
002242	123	101	127	.ASCII	/SAW/	
002245	124	117	117	.ASCII	/TOO/	
002250	124	110	040	.ASCII	/TH /	
002253	123	105	105	.ASCII	/SEE/	
002256	113	040	101	.ASCII	/K A/	
002261	116	104	040	.ASCII	/ND /	
002264	126	105	122	.ASCII	/VER/	
002267	111	106	131	.ASCII	/IFY/	
002272	000	000		.ASCII	<00><00>	
002274	045	116	045	P.ABL:	.ASCII	/N# /
002277	101	124	105	.ASCII	/ATE/	
002302	123	124	040	.ASCII	/ST /	
002305	061	067	040	.ASCII	/17 /	
002310	103	117	116	.ASCII	/CON/	
002313	126	105	122	.ASCII	/VER/	
002316	107	111	116	.ASCII	/GIN/	

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC1.B16;1

SEQ 0053
Page 25
(12)

002321	107	057	104	.ASCII	/G/<57>/D/
002324	111	126	105	.ASCII	/IVE/
002327	122	107	111	.ASCII	/RGI/
002332	116	107	040	.ASCII	/NG /
002335	123	105	105	.ASCII	/SEE/
002340	113	040	101	.ASCII	/K A/
002343	116	104	040	.ASCII	/ND /
002346	126	105	122	.ASCII	/VER/
002351	111	106	131	.ASCII	/IFY/
002354	000	000		.ASCII	<00><00>
002356	045	116	045	P.ABM:	.ASCII /%N%/
002361	101	124	105		.ASCII /ATE/
002364	123	124	040		.ASCII /ST /
002367	061	070	040		.ASCII /18 /
002372	124	117	107		.ASCII /TOG/
002375	107	114	105		.ASCII /GLE/
002400	040	123	105		.ASCII / SE/
002403	105	113	040		.ASCII /EK /
002406	101	116	104		.ASCII /AND/
002411	040	126	105		.ASCII / VE/
002414	122	111	106		.ASCII /RIF/
002417	131	000	000		.ASCII /Y/<00><00>
002422	045	116	045	P.ABN:	.ASCII /%N%/
002425	101	124	105		.ASCII /ATE/
002430	123	124	040		.ASCII /ST /
002433	061	071	040		.ASCII /19 /
002436	110	105	101		.ASCII /HEA/
002441	104	040	123		.ASCII /D S/
002444	127	111	124		.ASCII /WIT/
002447	103	110	040		.ASCII /CH /
002452	124	105	123		.ASCII /TES/
002455	124	000	000		.ASCII /T/<00><00>
002460	045	116	045	P.ABO:	.ASCII /%N%/
002463	101	124	105		.ASCII /ATE/
002466	123	124	040		.ASCII /ST /
002471	062	060	040		.ASCII /20 /
002474	122	101	116		.ASCII /RAN/
002477	104	117	115		.ASCII /DOM/
002502	040	123	105		.ASCII / SE/
002505	105	113	040		.ASCII /EK /
002510	101	116	104		.ASCII /AND/
002513	040	126	105		.ASCII / VE/
002516	122	111	106		.ASCII /RIF/
002521	131	000	000		.ASCII /Y/<00><00>
002524	045	116	045	P.ABP:	.ASCII /%N%/
002527	101	124	105		.ASCII /ATE/
002532	123	124	040		.ASCII /ST /
002535	062	061	040		.ASCII /21 /
002540	123	105	103		.ASCII /SEC/
002543	124	117	122		.ASCII /TOR/
002546	040	101	103		.ASCII / AC/
002551	103	105	123		.ASCII /CES/
002554	123	040	124		.ASCII /S T/
002557	105	123	124		.ASCII /EST/
002562	000	000			.ASCII <00><00>
002564	045	116	045	P.ABQ:	.ASCII /%N%/
002567	101	124	105		.ASCII /ATE/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 B16-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0054
Page 26
(12)

002572	123	124	040	.ASCII	/ST /
002575	062	062	040	.ASCII	/22 /
002600	103	117	116	.ASCII	/CON/
002603	124	122	117	.ASCII	/TRO/
002606	114	114	105	.ASCII	/LLE/
002611	122	040	120	.ASCII	/R P/
002614	122	117	103	.ASCII	/ROC/
002617	105	123	123	.ASCII	/ESS/
002622	111	116	107	.ASCII	/ING/
002625	040	124	111	.ASCII	/ TI/
002630	115	105	000	.ASCII	/ME/<00>
002633	000			.ASCII	<00>
002634	045	116	045	P.ABR: .ASCII	/N#/
002637	101	124	105	.ASCII	/ATE/
002642	123	124	040	.ASCII	/ST /
002645	062	063	040	.ASCII	/23 /
002650	117	116	105	.ASCII	/ONE/
002653	040	124	122	.ASCII	/ TR/
002656	101	103	113	.ASCII	/ACK/
002661	040	123	105	.ASCII	/ SE/
002664	105	113	040	.ASCII	/EK /
002667	124	111	115	.ASCII	/TIM/
002672	105	000		.ASCII	/E/<00>
002674	045	116	045	P.ABS: .ASCII	/N#/
002677	101	124	105	.ASCII	/ATE/
002702	123	124	040	.ASCII	/ST /
002705	062	064	040	.ASCII	/24 /
002710	101	126	105	.ASCII	/AVE/
002713	122	101	107	.ASCII	/RAG/
002716	105	040	123	.ASCII	/E S/
002721	105	105	113	.ASCII	/EEK/
002724	040	124	111	.ASCII	/ TI/
002727	115	105	000	.ASCII	/ME/<00>
002732	045	116	045	P.ABT: .ASCII	/N#/
002735	101	124	105	.ASCII	/ATE/
002740	123	124	040	.ASCII	/ST /
002743	062	065	040	.ASCII	/25 /
002746	106	125	114	.ASCII	/FUL/
002751	114	040	123	.ASCII	/L S/
002754	124	122	117	.ASCII	/TRO/
002757	113	105	040	.ASCII	/KE /
002762	123	105	105	.ASCII	/SEE/
002765	113	040	124	.ASCII	/K T/
002770	111	115	105	.ASCII	/IME/
002773	000			.ASCII	<00>
002774	045	116	045	P.ABU: .ASCII	/N#/
002777	101	124	105	.ASCII	/ATE/
003002	123	124	040	.ASCII	/ST /
003005	062	066	040	.ASCII	/26 /
003010	127	122	111	.ASCII	/WRI/
003013	124	105	040	.ASCII	/TE /
003016	104	101	124	.ASCII	/DAT/
003021	101	040	124	.ASCII	/A T/
003024	105	123	124	.ASCII	/EST/
003027	000			.ASCII	<00>
003030	045	116	045	P.ABV: .ASCII	/N#/
003033	101	124	105	.ASCII	/ATE/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0055
Page 27
(12)

003036	123	124	040	.ASCII	/ST /
003041	062	067	040	.ASCII	/27 /
003044	117	106	106	.ASCII	/OFF/
003047	123	105	124	.ASCII	/SET/
003052	040	124	117	.ASCII	/ TO/
003055	114	105	122	.ASCII	/LER/
003060	101	116	103	.ASCII	/ANC/
003063	105	040	124	.ASCII	/E T/
003066	105	123	124	.ASCII	/EST/
003071	000			.ASCII	<00>
003072	045	116	045	P.ABW:	.ASCII /%N%/
003075	101	124	105	.ASCII	/ATE/
003100	123	124	040	.ASCII	/ST /
003103	062	070	040	.ASCII	/28 /
003106	101	126	105	.ASCII	/AVE/
003111	122	101	107	.ASCII	/RAG/
003114	105	040	122	.ASCII	/E R/
003117	117	124	101	.ASCII	/OTA/
003122	124	111	117	.ASCII	/TIO/
003125	116	101	114	.ASCII	/NAL/
003130	040	124	111	.ASCII	/ TI/
003133	115	105	000	P.ABX:	.ASCII /ME/<00>
003136	045	116	045	.ASCII	/%N%/
003141	101	124	105	.ASCII	/ATE/
003144	123	124	040	.ASCII	/ST /
003147	062	071	040	.ASCII	/29 /
003152	127	122	111	.ASCII	/WRI/
003155	124	105	040	.ASCII	/TE /
003160	120	122	117	.ASCII	/PRO/
003163	124	105	103	.ASCII	/TEC/
003166	124	040	124	.ASCII	/T T/
003171	105	123	124	.ASCII	/EST/
003174	000	000		P.ABY:	.ASCII <00><00>
003176	045	116	045	.ASCII	/%N%/
003201	101	011	115	.ASCII	/A/<11>/M/
003204	101	116	125	.ASCII	/ANU/
003207	101	114	040	.ASCII	/AL /
003212	111	116	124	.ASCII	/INT/
003215	105	122	126	.ASCII	/ERV/
003220	105	116	124	.ASCII	/ENT/
003223	111	117	116	.ASCII	/ION/
003226	040	124	105	.ASCII	/ TE/
003231	123	124	040	.ASCII	/ST /
003234	116	117	124	.ASCII	/NOT/
003237	040	120	105	.ASCII	/ PE/
003242	122	106	117	.ASCII	/RFO/
003245	122	115	105	.ASCII	/RME/
003250	104	000		P.ABZ:	.ASCII /D/<00>
003252	045	116	045	.ASCII	/%N%/
003255	101	120	117	.ASCII	/APO/
003260	127	105	122	.ASCII	/WER/
003263	040	104	105	.ASCII	/ DE/
003266	114	101	131	.ASCII	/LAY/
003271	040	055	040	.ASCII	/ - /
003274	127	101	111	.ASCII	/WAI/
003277	124	111	116	.ASCII	/TIN/
003302	107	000		.ASCII	/G/<00>

ZRCFB1
V03.0CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION27-Mar-1985 15:21:49
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1SEQ 0056
Page 28
(12)

003304	045	116	045	P.ACA:	.ASCII	/#N#/ <00><00>
003307	101	124	117		.ASCII	/ATO/ <00><00>
003312	117	040	115		.ASCII	/O M/ <00><00>
003315	101	116	131		.ASCII	/ANY/ <00><00>
003320	040	125	116		.ASCII	/ UN/ <00><00>
003323	111	124	123		.ASCII	/ITS/ <00><00>
003326	000	000		P.ACB:	.ASCII	/#N#/ <00><00>
003330	045	116	045		.ASCII	/ANO/ <00><00>
003333	101	116	117		.ASCII	/ANO/ <00><00>
003336	040	103	114		.ASCII	/ CL/ <00><00>
003341	117	103	113		.ASCII	/OCK/ <00><00>
003344	040	127	101		.ASCII	/ WA/ <00><00>
003347	123	040	106		.ASCII	/S F/ <00><00>
003352	117	125	116		.ASCII	/OUN/ <00><00>
003355	104	040	111		.ASCII	/D I/ <00><00>
003360	116	040	124		.ASCII	/N T/ <00><00>
003363	110	105	040		.ASCII	/HE / <00><00>
003366	123	131	123		.ASCII	/SYS/ <00><00>
003371	124	105	115		.ASCII	/TEM/ <00><00>
003374	000	000		P.ACC:	.ASCII	/#N#/ <00><00>
003376	045	116	045		.ASCII	/#N#/ <00><00>
003401	101	111	116		.ASCII	/AIN/ <00><00>
003404	103	117	122		.ASCII	/COR/ <00><00>
003407	122	105	103		.ASCII	/REC/ <00><00>
003412	124	040	124		.ASCII	/T T/ <00><00>
003415	122	101	103		.ASCII	/RAC/ <00><00>
003420	113	040	116		.ASCII	/K N/ <00><00>
003423	125	115	102		.ASCII	/UMB/ <00><00>
003426	105	122	123		.ASCII	/ENS/ <00><00>
003431	040	123	105		.ASCII	/ SE/ <00><00>
003434	114	105	103		.ASCII	/LEC/ <00><00>
003437	124	105	104		.ASCII	/TED/ <00><00>
003442	000	000		P.ACD:	.ASCII	/#N#/ <00><00>
003444	045	116	045		.ASCII	/#N#/ <00><00>
003447	116	000	000		.ASCII	/N/<00><00> <00><00>
003452	045	116	045	P.ACE:	.ASCII	/#N#/ <00><00>
003455	101	011	040		.ASCII	/A/<11>/ / <00><00>
003460	122	105	107		.ASCII	/REG/ <00><00>
003463	111	123	124		.ASCII	/IST/ <00><00>
003466	105	122	040		.ASCII	/ER / <00><00>
003471	106	101	111		.ASCII	/FAI/ <00><00>
003474	114	105	104		.ASCII	/LED/ <00><00>
003477	040	124	117		.ASCII	/ TO/ <00><00>
003502	040	122	105		.ASCII	/ RE/ <00><00>
003505	123	120	117		.ASCII	/SPO/ <00><00>
003510	116	104	040		.ASCII	/ND / <00><00>
003513	101	124	040		.ASCII	/AT / <00><00>
003516	101	104	104		.ASCII	/ADD/ <00><00>
003521	122	105	123		.ASCII	/RES/ <00><00>
003524	123	072	040		.ASCII	/S: / <00><00>
003527	040	045	117		.ASCII	/ #O/ <00><00>
003532	066	045	116		.ASCII	/6#N/ <00><00>
003535	000			P.ACF:	.ASCII	<00> <00><00>
003536	045	116	045		.ASCII	/#N#/ <00><00>
003541	101	101	104		.ASCII	/AAD/ <00><00>
003544	104	122	105		.ASCII	/DRE/ <00><00>
003547	123	123	072		.ASCII	/SS:/ <00><00>

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0057
Page 29
(12)

003552	040	045	117	.ASCII	/ #0/
003555	066	045	101	.ASCII	/6%A/
003560	011	105	130	.ASCII	<11>/EX/
003563	120	105	103	.ASCII	/PEC/
003566	124	105	104	.ASCII	/TED/
003571	072	040	045	.ASCII	/: #/
003574	117	066	045	.ASCII	/06#/
003577	101	011	122	.ASCII	/A/<11>/R/
003602	105	101	104	.ASCII	/EAD/
003605	072	040	045	.ASCII	/: #/
003610	117	066	045	.ASCII	/06#/
003613	116	000	000	.ASCII	/N/<00><00>
003616	045	116	045	P.ACG: .ASCII	/#N#/
003621	101	123	124	.ASCII	/AST/
003624	105	120	040	.ASCII	/EP /
003627	115	101	123	.ASCII	/MAS/
003632	113	040	075	.ASCII	/K =/
003635	040	045	117	.ASCII	/ #0/
003640	062	045	101	.ASCII	/2%A/
003643	011	106	101	.ASCII	<11>/FA/
003646	111	114	111	.ASCII	/ILI/
003651	116	107	040	.ASCII	/NG /
003654	122	105	107	.ASCII	/REG/
003657	111	123	124	.ASCII	/IST/
003662	105	122	040	.ASCII	/ER /
003665	075	040	045	.ASCII	/= #/
003670	117	066	045	.ASCII	/06#/
003673	101	040	104	.ASCII	/A D/
003676	101	124	101	.ASCII	/ATA/
003701	040	075	040	.ASCII	/ = /
003704	045	117	066	.ASCII	/#06/
003707	045	116	000	.ASCII	/#N/<00>
003712	045	116	045	P.ACH: .ASCII	/#N#/
003715	101	011	040	.ASCII	/A/<11>/ /
003720	120	117	122	.ASCII	/POR/
003723	124	040	124	.ASCII	/T T/
003726	131	120	105	.ASCII	/YPE/
003731	040	116	125	.ASCII	/ NU/
003734	115	102	105	.ASCII	/MBE/
003737	122	040	075	.ASCII	/R =/
003742	040	045	117	.ASCII	/ #0/
003745	062	000	000	.ASCII	/2/<00><00>
003750	045	116	045	P.ACI: .ASCII	/#N#/
003753	101	011	040	.ASCII	/A/<11>/ /
003756	120	117	122	.ASCII	/POR/
003761	124	040	123	.ASCII	/T S/
003764	120	105	103	.ASCII	/PEC/
003767	111	106	111	.ASCII	/IFI/
003772	103	040	111	.ASCII	/C I/
003775	116	106	117	.ASCII	/NFO/
004000	072	057	116	.ASCII	/:/<57>/N/
004003	126	057	121	.ASCII	/V/<57>/Q/
004006	102	057	104	.ASCII	/B/<57>/D/
004011	111	057	117	.ASCII	/I/<57>/O/
004014	104	057	115	.ASCII	/D/<57>/M/
004017	120	057	040	.ASCII	/P/<57>/ /
004022	075	040	045	.ASCII	/= #/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0058
Page 30
(12)

004025	117	062	000		.ASCII	/02/<00>
004030	045	116	045	P.ACJ:	.ASCII	/N%/
004033	101	011	040		.ASCII	/A/<11>/ /
004036	115	111	103		.ASCII	/MIC/
004041	122	117	040		.ASCII	/RO /
004044	103	117	104		.ASCII	/COD/
004047	105	072	040		.ASCII	/E: /
004052	115	117	104		.ASCII	/MOD/
004055	105	114	040		.ASCII	/EL /
004060	075	040	045		.ASCII	/= %/
004063	117	062	045		.ASCII	/02%/
004066	101	040	040		.ASCII	/A /
004071	040	126	105		.ASCII	/ VE/
004074	122	123	111		.ASCII	/RSI/
004077	117	116	040		.ASCII	/ON /
004102	075	040	045		.ASCII	/= %/
004105	117	062	000		.ASCII	/02/<00>
004110	045	116	045	P.ACK:	.ASCII	/N%/
004113	101	011	130		.ASCII	/A/<11>/X/
004116	115	124	137		.ASCII	/MT /
004121	102	125	106		.ASCII	/BUF/
004124	072	040	045		.ASCII	/: %/
004127	117	066	045		.ASCII	/06%/
004132	101	011	122		.ASCII	/A/<11>/R/
004135	105	103	137		.ASCII	/EC /
004140	102	125	106		.ASCII	/BUF/
004143	072	040	045		.ASCII	/: %/
004146	117	066	045		.ASCII	/06%/
004151	116	000	000	P.ACL:	.ASCII	/N/<00><00>
004154	045	101	011		.ASCII	/A/<11>
004157	130	115	124		.ASCII	/XMT/
004162	137	104	101		.ASCII	/ DA/
004165	124	101	072		.ASCII	/TA:/
004170	040	045	117		.ASCII	/ %0/
004173	066	045	101		.ASCII	/6%A/
004176	011	122	105		.ASCII	<11>/RE/
004201	103	137	104		.ASCII	/C D/
004204	101	124	101		.ASCII	/ATA/
004207	072	040	045		.ASCII	/: %/
004212	117	066	045		.ASCII	/06%/
004215	116	000	000	P.ACM:	.ASCII	/N/<00><00>
004220	045	116	045		.ASCII	/N%/
004223	101	011	125		.ASCII	/A/<11>/U/
004226	116	111	124		.ASCII	/NIT/
004231	040	103	117		.ASCII	/ CO/
004234	115	105	123		.ASCII	/MES/
004237	040	117	116		.ASCII	/ ON/
004242	114	111	116		.ASCII	/LIN/
004245	105	040	111		.ASCII	/E I/
004250	116	040	072		.ASCII	/N :/
004253	040	045	104		.ASCII	/ %D/
004256	062	045	101		.ASCII	/2%A/
004261	040	155	151		.ASCII	/ mi/
004264	156	056	040		.ASCII	/n. /
004267	045	104	062		.ASCII	/D2/
004272	045	101	056		.ASCII	/A./
004275	045	104	062		.ASCII	/D2/

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.816;1

004300	045	101	040	.ASCII	/#A /
004303	163	145	143	.ASCII	/sec/
004306	056	000		.ASCII	./<00>
004310	045	116	045	P.ACN: .ASCII	/#N#/
004313	101	011	123	.ASCII	/A/<11>/S/
004316	124	101	122	.ASCII	/TAR/
004321	124	111	116	.ASCII	/TIN/
004324	107	040	124	.ASCII	/G T/
004327	122	101	103	.ASCII	/RAC/
004332	113	072	040	.ASCII	/K: /
004335	045	117	064	.ASCII	/#04/
004340	045	101	011	.ASCII	/#A/<11>
004343	105	116	104	.ASCII	/END/
004346	111	116	107	.ASCII	/ING/
004351	040	124	122	.ASCII	/ TR/
004354	101	103	113	.ASCII	/ACK/
004357	072	040	045	.ASCII	/: #/
004362	117	064	045	.ASCII	/04#/
004365	101	011	104	.ASCII	/A/<11>/D/
004370	105	123	111	.ASCII	/ESI/
004373	122	105	104	.ASCII	/RED/
004376	040	114	102	.ASCII	/ LB/
004401	116	072	040	.ASCII	/N: /
004404	045	117	066	.ASCII	/#06/
004407	045	116	000	.ASCII	/#N/<00>
004412	045	116	045	P.ACO: .ASCII	/#N#/
004415	101	011	125	.ASCII	/A/<11>/U/
004420	116	111	124	.ASCII	/NIT/
004423	072	040	045	.ASCII	/: #/
004426	117	064	045	.ASCII	/04#/
004431	101	011	110	.ASCII	/A/<11>/H/
004434	105	101	104	.ASCII	/EAD/
004437	072	040	045	.ASCII	/: #/
004442	117	064	045	.ASCII	/04#/
004445	101	011	124	.ASCII	/A/<11>/T/
004450	122	101	103	.ASCII	/RAC/
004453	113	072	040	.ASCII	/K: /
004456	045	117	064	.ASCII	/#04/
004461	045	116	000	.ASCII	/#N/<00>
004464	045	116	045	P.ACP: .ASCII	/#N#/
004467	101	011	116	.ASCII	/A/<11>/N/
004472	125	115	102	.ASCII	/UMB/
004475	105	122	040	.ASCII	/ER /
004500	117	106	040	.ASCII	/OF /
004503	123	105	105	.ASCII	/SEE/
004506	113	123	040	.ASCII	/KS /
004511	050	104	051	.ASCII	/(D)/
004514	072	040	045	.ASCII	/: #/
004517	104	066	045	.ASCII	/D6#/
004522	101	011	114	.ASCII	/A/<11>/L/
004525	102	116	072	.ASCII	/BN:/
004530	040	045	117	.ASCII	/ #0/
004533	066	045	116	.ASCII	/6#N/
004536	000	000		.ASCII	<00><00>
004540	045	116	045	P.ACQ: .ASCII	/#N#/
004543	101	011	115	.ASCII	/A/<11>/M/
004546	101	130	056	.ASCII	/AX./

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0060
Page 32
(12)

004551	040	117	106	.ASCII	/ OF/
004554	106	123	105	.ASCII	/FSE/
004557	124	040	126	.ASCII	/T V/
004562	101	114	125	.ASCII	/ALU/
004565	105	072	040	.ASCII	/E: /
004570	045	104	062	.ASCII	/#D2/
004573	045	101	056	.ASCII	/#A./
004576	045	104	061	.ASCII	/#D1/
004601	045	101	040	.ASCII	/#A /
004604	160	145	162	.ASCII	/per/
004607	143	145	156	.ASCII	/cen/
004612	164	000		.ASCII	/t/<00>
004614	045	116	045	P.ACR: .ASCII	/#N#/
004617	101	011	122	.ASCII	/A/<11>/R/
004622	103	123	101	.ASCII	/CSA/
004625	040	105	122	.ASCII	/ ER/
004630	122	117	122	.ASCII	/ROR/
004633	040	123	124	.ASCII	/ ST/
004636	101	124	125	.ASCII	/ATU/
004641	123	072	040	.ASCII	/S: /
004644	045	117	066	.ASCII	/#06/
004647	045	116	000	.ASCII	/#N/<00>
004652	045	116	045	P.ACS: .ASCII	/#N#/
004655	101	011	105	.ASCII	/A/<11>/E/
004660	116	104	040	.ASCII	/ND /
004663	120	101	103	.ASCII	/PAC/
004666	113	105	124	.ASCII	/KET/
004671	040	105	122	.ASCII	/ ER/
004674	122	117	122	.ASCII	/ROR/
004677	040	123	124	.ASCII	/ ST/
004702	101	124	125	.ASCII	/ATU/
004705	123	072	040	.ASCII	/S: /
004710	045	117	066	.ASCII	/#06/
004713	045	116	000	.ASCII	/#N/<00>
004716	045	116	045	P.ACT: .ASCII	/#N#/
004721	101	011	125	.ASCII	/A/<11>/U/
004724	116	105	130	.ASCII	/NEX/
004727	120	105	103	.ASCII	/PEC/
004732	124	105	104	.ASCII	/TED/
004735	040	114	117	.ASCII	/ LO/
004740	107	040	120	.ASCII	/G P/
004743	101	103	113	.ASCII	/ACK/
004746	105	124	040	.ASCII	/ET /
004751	105	122	122	.ASCII	/ERR/
004754	117	122	040	.ASCII	/OR /
004757	123	124	101	.ASCII	/STA/
004762	124	125	123	.ASCII	/TUS/
004765	072	040	045	.ASCII	/: #/
004770	117	066	045	.ASCII	/06#/
004773	116	000	000	.ASCII	/N/<00><00>
004776	045	116	045	P.ACU: .ASCII	/#N#/
005001	101	011	127	.ASCII	/A/<11>/W/
005004	122	111	124	.ASCII	/RIT/
005007	105	040	104	.ASCII	/E D/
005012	101	124	101	.ASCII	/ATA/
005015	072	040	045	.ASCII	/: #/
005020	117	066	045	.ASCII	/06#/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.816;1

SEQ 0061
Page 33
(12)

005023	101	011	122	.ASCII	/A/<11>/R/	
005026	105	101	104	.ASCII	/EAD/	
005031	040	104	101	.ASCII	/ DA/	
005034	124	101	072	.ASCII	/TA:/	
005037	040	045	117	.ASCII	/ #0/	
005042	066	000		.ASCII	/6/<00>	
005044	045	116	045	P.ACW:	.ASCII	/#N#/
005047	101	011	124	.ASCII	/A/<11>/T/	
005052	122	101	103	.ASCII	/RAC/	
005055	113	072	040	.ASCII	/K: /	
005060	045	117	064	.ASCII	/#04/	
005063	045	101	011	.ASCII	/#A/<11>	
005066	123	105	103	.ASCII	/SEC/	
005071	124	117	122	.ASCII	/TOR/	
005074	072	040	045	.ASCII	/: #/	
005077	117	064	045	.ASCII	/04#/	
005102	101	011	110	.ASCII	/A/<11>/H/	
005105	105	101	104	.ASCII	/EAD/	
005110	072	040	045	.ASCII	/: #/	
005113	117	064	045	.ASCII	/04#/	
005116	116	000		P.ACW:	.ASCII	/N/<00>
005120	045	116	045	.ASCII	/#N#/	
005123	101	011	105	.ASCII	/A/<11>/E/	
005126	130	120	105	.ASCII	/XPE/	
005131	103	124	105	.ASCII	/CTE/	
005134	104	040	123	.ASCII	/D S/	
005137	127	040	075	.ASCII	/W =/	
005142	040	117	106	.ASCII	/ OF/	
005145	106	040	040	.ASCII	/F /	
005150	101	103	124	.ASCII	/ACT/	
005153	125	101	114	.ASCII	/UAL/	
005156	040	123	127	.ASCII	/ SW/	
005161	040	075	040	.ASCII	/ = /	
005164	117	116	040	.ASCII	/ON /	
005167	040	120	114	.ASCII	/ PL/	
005172	101	124	124	.ASCII	/ATT/	
005175	105	122	040	.ASCII	/ER /	
005200	043	040	075	.ASCII	/# =/	
005203	040	045	104	.ASCII	/ #D/	
005206	063	000		P.ACX:	.ASCII	/3/<00>
005210	045	116	045	.ASCII	/#N#/	
005213	101	011	105	.ASCII	/A/<11>/E/	
005216	130	120	105	.ASCII	/XPE/	
005221	103	124	105	.ASCII	/CTE/	
005224	104	040	123	.ASCII	/D S/	
005227	127	040	075	.ASCII	/W =/	
005232	040	117	116	.ASCII	/ ON/	
005235	040	040	101	.ASCII	/ A/	
005240	103	124	125	.ASCII	/CTU/	
005243	101	114	040	.ASCII	/AL /	
005246	123	127	040	.ASCII	/SW /	
005251	075	040	117	.ASCII	/= 0/	
005254	106	106	040	.ASCII	/FF /	
005257	040	120	114	.ASCII	/ PL/	
005262	101	124	124	.ASCII	/ATT/	
005265	105	122	040	.ASCII	/ER /	
005270	043	040	075	.ASCII	/# =/	

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0062
Page 34
(12)

005273	040	045	104		.ASCII	/ #D/
005276	063	000			.ASCII	/3/<00>
005300	045	116	045	P.ACY:	.ASCII	/#N#/
005303	101	011	105		.ASCII	/A/<11>/E/
005306	122	122	117		.ASCII	/RRO/
005311	122	040	123		.ASCII	/R S/
005314	124	101	124		.ASCII	/TAT/
005317	125	123	072		.ASCII	/US:/
005322	040	045	117		.ASCII	/ #0/
005325	066	045	116		.ASCII	/6#N/
005330	000	000			.ASCII	<00><00>
005332	045	116	045	P.ACZ:	.ASCII	/#N#/
005335	101	011	040		.ASCII	/A/<11>/ /
005340	116	125	115		.ASCII	/NUM/
005343	102	105	122		.ASCII	/BER/
005346	040	117	106		.ASCII	/ OF/
005351	040	122	105		.ASCII	/ RE/
005354	124	122	111		.ASCII	/TRI/
005357	105	123	040		.ASCII	/ES /
005362	050	104	051		.ASCII	/(D)/
005365	040	075	045		.ASCII	/ =#/
005370	104	064	000		.ASCII	/D4/<00>
005373	000				.ASCII	<00>
005374	011	127	101	P.ADA:	.ASCII	<11>/WA/
005377	111	124	040		.ASCII	/IT /
005402	137	040	120		.ASCII	/ P/
005405	117	127	105		.ASCII	/ÖWE/
005410	122	040	106		.ASCII	/R F/
005413	101	111	114		.ASCII	/AIL/
005416	040	122	105		.ASCII	/ RE/
005421	103	117	126		.ASCII	/COV/
005424	105	122	131		.ASCII	/ERY/
005427	000				.ASCII	<00>
005430	122	103	123	P.ADB:	.ASCII	/RCS/
005433	101	040	106		.ASCII	/A F/
005436	101	111	114		.ASCII	/AIL/
005441	105	104	040		.ASCII	/ED /
005444	124	117	040		.ASCII	/TO /
005447	122	105	123		.ASCII	/RES/
005452	120	117	116		.ASCII	/PON/
005455	104	000	000		.ASCII	/D/<00><00>
005460	122	103	111	P.ADC:	.ASCII	/RCI/
005463	120	040	106		.ASCII	/P F/
005466	101	111	114		.ASCII	/AIL/
005471	105	104	040		.ASCII	/ED /
005474	124	117	040		.ASCII	/TO /
005477	122	105	123		.ASCII	/RES/
005502	120	117	116		.ASCII	/PON/
005505	104	000	000		.ASCII	/D/<00><00>
005510	124	105	123	P.ADD:	.ASCII	/TES/
005513	124	040	120		.ASCII	/T P/
005516	101	124	124		.ASCII	/ATT/
005521	105	122	116		.ASCII	/ERN/
005524	040	105	103		.ASCII	/ EC/
005527	110	117	105		.ASCII	/HOE/
005532	104	040	111		.ASCII	/D I/
005535	116	040	122		.ASCII	/N R/

ZRCFR1
V03.0CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION27-Mar-1985 15:21:49
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1SEQ 0063
Page 35
(12)

005540	103	123	101	.ASCII	/CSA/
005543	040	111	123	.ASCII	/ IS/
005546	040	111	116	.ASCII	/ IN/
005551	103	117	122	.ASCII	/COR/
005554	122	105	103	.ASCII	/REC/
005557	124	000	000	.ASCII	/T/<00><00>
005562	126	105	103	P.ADE: .ASCII	/VEC/
005565	124	117	122	.ASCII	/TOR/
005570	040	101	116	.ASCII	/ AN/
005573	104	040	102	.ASCII	/D B/
005576	122	040	114	.ASCII	/R L/
005601	105	126	105	.ASCII	/EVE/
005604	114	040	124	.ASCII	/L T/
005607	105	123	124	.ASCII	/EST/
005612	040	106	101	.ASCII	/ FA/
005615	111	114	125	.ASCII	/ILU/
005620	122	105	000	.ASCII	/RE/<00>
005623	000			.ASCII	<00>
005624	110	117	123	P.ADF: .ASCII	/HOS/
005627	124	040	104	.ASCII	/T D/
005632	105	124	105	.ASCII	/ETE/
005635	103	124	105	.ASCII	/CTE/
005640	104	040	124	.ASCII	/D T/
005643	111	115	105	.ASCII	/IME/
005646	040	117	125	.ASCII	/ OU/
005651	124	040	105	.ASCII	/T E/
005654	122	122	117	.ASCII	/RRO/
005657	122	000	000	.ASCII	/R/<00><00>
005662	122	111	116	P.ADG: .ASCII	/RIN/
005665	107	040	102	.ASCII	/G B/
005670	125	106	106	.ASCII	/UFF/
005673	105	122	123	.ASCII	/ERS/
005676	040	116	117	.ASCII	/ NO/
005701	124	040	103	.ASCII	/T C/
005704	114	105	101	.ASCII	/LEA/
005707	122	105	104	.ASCII	/RED/
005712	040	102	131	.ASCII	/ BY/
005715	040	124	110	.ASCII	/ TH/
005720	105	040	120	.ASCII	/E P/
005723	117	122	124	.ASCII	/ORT/
005726	000	000		.ASCII	<00><00>
005730	123	124	105	P.ADH: .ASCII	/STE/
005733	120	040	122	.ASCII	/P R/
005736	105	101	104	.ASCII	/EAD/
005741	040	104	101	.ASCII	/ DA/
005744	124	101	040	.ASCII	/TA /
005747	104	117	105	.ASCII	/DOE/
005752	123	040	116	.ASCII	/S N/
005755	117	124	040	.ASCII	/OT /
005760	115	101	124	.ASCII	/MAT/
005763	103	110	000	.ASCII	/CH/<00>
005766	120	117	122	P.ADI: .ASCII	/POR/
005771	124	040	106	.ASCII	/T F/
005774	101	124	101	.ASCII	/ATA/
005777	114	040	105	.ASCII	/L E/
006002	122	122	117	.ASCII	/RRO/
006005	122	000	000	.ASCII	/R/<00><00>

ZRCFR1
V03.0

CZRCFCO RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC1.B16;1

006010	111	116	111	P.ADJ:	.ASCII	/INI/
006013	124	040	123		.ASCII	/T S/
006016	124	105	120		.ASCII	/TEP/
006021	040	122	105		.ASCII	/RE/
006024	101	104	040		.ASCII	/AD /
006027	105	122	122		.ASCII	/ERR/
006032	117	122	000		.ASCII	/OR/<00>
006035	000				.ASCII	<00>
006036	115	105	115	P.ADK:	.ASCII	/MEM/
006041	117	122	131		.ASCII	/ORY/
006044	040	102	125		.ASCII	/BU/
006047	106	106	105		.ASCII	/FFE/
006052	122	040	104		.ASCII	/R D/
006055	117	105	123		.ASCII	/OES/
006060	040	116	117		.ASCII	/NO/
006063	124	040	103		.ASCII	/T C/
006066	117	116	124		.ASCII	/ONT/
006071	101	111	116		.ASCII	/AIN/
006074	040	105	130		.ASCII	/EX/
006077	120	105	103		.ASCII	/PEC/
006102	124	105	104		.ASCII	/TED/
006105	040	104	101		.ASCII	/DA/
006110	124	101	000		.ASCII	/TA/<00>
006113	000				.ASCII	<00>
006114	104	115	040	P.ADL:	.ASCII	/DM /
006117	103	117	104		.ASCII	/COD/
006122	105	040	122		.ASCII	/E R/
006125	105	124	125		.ASCII	/ETU/
006130	122	116	105		.ASCII	/RNE/
006133	104	040	106		.ASCII	/D F/
006136	101	111	114		.ASCII	/AIL/
006141	125	122	105		.ASCII	/URE/
006144	040	103	117		.ASCII	/CO/
006147	104	105	000	P.ADM:	.ASCII	/DE/<00>
006152	045	116	045		.ASCII	/ENB/
006155	101	040	040		.ASCII	/A /
006160	040	040	040		.ASCII	/ /
006163	040	040	040		.ASCII	/ /
006166	111	116	124		.ASCII	/INT/
006171	105	122	122		.ASCII	/ERR/
006174	125	120	124		.ASCII	/UPT/
006177	040	101	124		.ASCII	/ AT/
006202	040	126	105		.ASCII	/ VE/
006205	103	075	040		.ASCII	/C = /
006210	045	117	063		.ASCII	/#03/
006213	045	101	040		.ASCII	/#A /
006216	102	122	040		.ASCII	/BR /
006221	114	105	126		.ASCII	/LEV/
006224	105	114	075		.ASCII	/EL = /
006227	040	045	117		.ASCII	/ #0/
006232	061	000			.ASCII	/1/<00>
006234	116	117	040	P.ADN:	.ASCII	/NO /
006237	111	116	124		.ASCII	/INT/
006242	105	122	122		.ASCII	/ERR/
006245	125	120	124		.ASCII	/UPT/
006250	040	106	122		.ASCII	/ FR/
006253	117	115	040		.ASCII	/OM /

006256	120	117	122	.ASCII	/POR/
006261	124	040	057	.ASCII	/T /<57>
006264	040	103	117	.ASCII	/ CO/
006267	116	124	122	.ASCII	/NTR/
006272	117	114	114	.ASCII	/OLL/
006275	105	122	000	.ASCII	/ER/<00>
006300	045	116	045	P.ADO: .ASCII	/MMS/
006303	101	011	011	.ASCII	/A/<11><11>
006306	102	122	040	.ASCII	/BR /
006311	114	105	126	.ASCII	/LEV/
006314	105	114	040	.ASCII	/EL /
006317	122	105	103	.ASCII	/REC/
006322	105	111	126	.ASCII	/EIV/
006325	105	104	057	.ASCII	/ED/<57>
006330	124	131	120	.ASCII	/TYP/
006333	105	104	040	.ASCII	/ED /
006336	111	123	040	.ASCII	/IS /
006341	111	116	103	.ASCII	/INC/
006344	117	122	122	.ASCII	/ORR/
006347	105	103	124	.ASCII	/ECT/
006352	040	041	000	.ASCII	/ !/<00>
006355	000			.ASCII	<00>
006356	122	103	062	P.ADP: .ASCII	/RC2/
006361	065	040	123	.ASCII	/S S/
006364	105	105	113	.ASCII	/EEK/
006367	040	106	101	.ASCII	/ FA/
006372	111	114	125	.ASCII	/ILU/
006375	122	105	000	.ASCII	/RE/<00>
006400	110	105	101	P.ADQ: .ASCII	/HEA/
006403	104	040	123	.ASCII	/D S/
006406	127	111	124	.ASCII	/WIT/
006411	103	110	040	.ASCII	/CH /
006414	106	101	111	.ASCII	/FAI/
006417	114	125	122	.ASCII	/LUR/
006422	105	000		.ASCII	/E/<00>
006424	123	105	103	P.ADR: .ASCII	/SEC/
006427	124	117	122	.ASCII	/TOR/
006432	040	122	105	.ASCII	/ RE/
006435	101	104	040	.ASCII	/AD /
006440	106	101	111	.ASCII	/FAI/
006443	114	125	122	.ASCII	/LUR/
006446	105	000		.ASCII	/E/<00>
006450	127	122	111	P.ADS: .ASCII	/WRI/
006453	124	105	040	.ASCII	/TE /
006456	120	122	117	.ASCII	/PRO/
006461	124	105	103	.ASCII	/TEC/
006464	124	040	124	.ASCII	/T T/
006467	105	123	124	.ASCII	/EST/
006472	040	106	101	.ASCII	/ FA/
006475	111	114	125	.ASCII	/ILU/
006500	122	105	000	.ASCII	/RE/<00>
006503	000			.ASCII	<00>
006504	106	117	122	P.ADT: .ASCII	/FOR/
006507	127	101	122	.ASCII	/WAR/
006512	104	040	123	.ASCII	/D S/
006515	105	105	113	.ASCII	/EEK/
006520	040	105	122	.ASCII	/ ER/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Blise-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.016;1

SEQ 0066
Page 38
(12)

006523	122	117	122		.ASCII	/ROR/
006526	000	000			.ASCII	<00><00>
006530	122	105	126	P.ADU:	.ASCII	/REV/
006533	105	122	123		.ASCII	/ERS/
006536	105	040	123		.ASCII	/E S/
006541	105	105	113		.ASCII	/EEK/
006544	040	105	122		.ASCII	/ ER/
006547	122	117	122		.ASCII	/ROR/
006552	000	000			.ASCII	<00><00>
006554	124	117	107	P.ADV:	.ASCII	/TOG/
006557	107	114	105		.ASCII	/GLE/
006562	040	123	105		.ASCII	/ SE/
006565	105	113	040		.ASCII	/EK /
006570	105	122	122		.ASCII	/ERR/
006573	117	122	000		.ASCII	/OR/<00>
006576	122	101	116	P.ADW:	.ASCII	/RAN/
006601	104	117	115		.ASCII	/DOM/
006604	040	123	105		.ASCII	/ SE/
006607	105	113	040		.ASCII	/EK /
006612	105	122	122		.ASCII	/ERR/
006615	117	122	000		.ASCII	/OR/<00>
006620	122	105	101	P.ADX:	.ASCII	/REA/
006623	104	057	127		.ASCII	/D/<57>/W/
006626	122	111	124		.ASCII	/RIT/
006631	105	040	124		.ASCII	/E T/
006634	105	123	124		.ASCII	/EST/
006637	040	111	116		.ASCII	/ IN/
006642	040	105	122		.ASCII	/ ER/
006645	122	117	122		.ASCII	/ROR/
006650	000	000			.ASCII	<00><00>
006652	117	106	106	P.ADY:	.ASCII	/OFF/
006655	123	105	124		.ASCII	/SET/
006660	040	122	105		.ASCII	/ RE/
006663	101	104	040		.ASCII	/AD /
006666	105	122	122		.ASCII	/ERR/
006671	117	122	000		.ASCII	/OR/<00>
006674	107	105	124	P.ADZ:	.ASCII	/GET/
006677	137	125	116		.ASCII	/ UN/
006702	111	124	137		.ASCII	/IT /
006705	123	124	101		.ASCII	/STA/
006710	124	125	123		.ASCII	/TUS/
006713	040	103	117		.ASCII	/ CO/
006716	115	115	101		.ASCII	/MMA/
006721	116	104	040		.ASCII	/ND /
006724	106	101	111		.ASCII	/FAI/
006727	114	125	122		.ASCII	/LUR/
006732	105	000			.ASCII	/E/<00>
006734	101	126	101	P.AEA:	.ASCII	/AVA/
006737	111	114	101		.ASCII	/ILA/
006742	102	114	105		.ASCII	/BLE/
006745	040	103	117		.ASCII	/ CO/
006750	115	115	101		.ASCII	/MMA/
006753	116	104	040		.ASCII	/ND /
006756	106	101	111		.ASCII	/FAI/
006761	114	125	122		.ASCII	/LUR/
006764	105	000			.ASCII	/E/<00>
006766	045	116	045	P.AEB:	.ASCII	/N% /

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

006771	101	101	126	.ASCII	/AAV/
006774	105	122	101	.ASCII	/ERA/
006777	107	105	040	.ASCII	/GE /
007002	123	105	105	.ASCII	/SEE/
007005	113	040	124	.ASCII	/K T/
007010	111	115	105	.ASCII	/IME/
007013	040	127	111	.ASCII	/ WI/
007016	124	110	040	.ASCII	/TH /
007021	122	101	116	.ASCII	/RAN/
007024	104	117	115	.ASCII	/DOM/
007027	040	114	102	.ASCII	/ LB/
007032	116	040	050	.ASCII	/N (/
007035	155	163	051	.ASCII	/ms)/
007040	040	075	040	.ASCII	/ = /
007043	045	104	063	.ASCII	/#D3/
007046	045	101	056	.ASCII	/#A./
007051	045	104	062	.ASCII	/#D2/
007054	000	000		.ASCII	<00><00>
007056	045	116	045	P.AEC: .ASCII	/#N#/
007061	101	101	126	.ASCII	/AAV/
007064	105	122	101	.ASCII	/ERA/
007067	107	105	040	.ASCII	/GE /
007072	123	105	105	.ASCII	/SEE/
007075	113	040	124	.ASCII	/K T/
007100	111	115	105	.ASCII	/IME/
007103	040	127	111	.ASCII	/ WI/
007106	124	110	040	.ASCII	/TH /
007111	117	122	104	.ASCII	/ORD/
007114	105	122	105	.ASCII	/ERE/
007117	104	040	114	.ASCII	/D L/
007122	102	116	040	.ASCII	/BN /
007125	050	155	163	.ASCII	/(ms/
007130	051	040	075	.ASCII	/) = /
007133	040	045	104	.ASCII	/ #D/
007136	063	045	101	.ASCII	/3#A/
007141	056	045	104	.ASCII	/.#D/
007144	062	000		.ASCII	/2/<00>
007146	045	116	045	P.AED: .ASCII	/#N#/
007151	101	120	122	.ASCII	/APR/
007154	117	103	105	.ASCII	/OCE/
007157	123	123	111	.ASCII	/SSI/
007162	116	107	040	.ASCII	/NG /
007165	124	111	115	.ASCII	/TIM/
007170	105	040	050	.ASCII	/E (/
007173	155	163	051	.ASCII	/ms)/
007176	040	075	040	.ASCII	/ = /
007201	045	104	063	.ASCII	/#D3/
007204	045	101	056	.ASCII	/#A./
007207	045	104	062	.ASCII	/#D2/
007212	000	000		.ASCII	<00><00>
007214	045	116	045	P.AEE: .ASCII	/#N#/
007217	101	117	116	.ASCII	/AON/
007222	105	040	124	.ASCII	/E T/
007225	122	101	103	.ASCII	/RAC/
007230	113	040	123	.ASCII	/K S/
007233	105	105	113	.ASCII	/EEK/
007236	040	124	111	.ASCII	/ TI/

007241	115	105	040	.ASCII	/ME /	
007244	050	155	163	.ASCII	/(ms/	
007247	051	040	075	.ASCII	/) =/	
007252	040	045	104	.ASCII	/ #D/	
007255	063	045	101	.ASCII	/3#A/	
007260	056	045	104	.ASCII	/.#D/	
007263	062	000	000	.ASCII	/2/<00><00>	
007266	045	116	045	P.AEF:	.ASCII	/#N#/
007271	101	106	125	.ASCII	/AFU/	
007274	114	114	040	.ASCII	/LL /	
007277	124	122	101	.ASCII	/TRA/	
007302	103	113	040	.ASCII	/CK /	
007305	123	105	105	.ASCII	/SEE/	
007310	113	040	124	.ASCII	/K T/	
007313	111	115	105	.ASCII	/IME/	
007316	040	050	155	.ASCII	/(m/	
007321	163	051	040	.ASCII	/s) /	
007324	075	040	045	.ASCII	/= #/	
007327	104	063	045	.ASCII	/D3#/	
007332	101	056	045	.ASCII	/A.#/	
007335	104	062	000	P.AEG:	.ASCII	/D2/<00>
007340	045	116	045	.ASCII	/#N#/	
007343	101	101	126	.ASCII	/AAV/	
007346	105	122	101	.ASCII	/ERA/	
007351	107	105	040	.ASCII	/GE /	
007354	122	117	124	.ASCII	/ROT/	
007357	101	124	111	.ASCII	/ATI/	
007362	117	116	101	.ASCII	/ONA/	
007365	114	040	124	.ASCII	/L T/	
007370	111	115	105	.ASCII	/IME/	
007373	040	050	155	.ASCII	/(m/	
007376	163	051	040	.ASCII	/s) /	
007401	075	040	045	.ASCII	/= #/	
007404	104	063	045	.ASCII	/D3#/	
007407	101	056	045	.ASCII	/A.#/	
007412	104	062	000	P.AEH:	.ASCII	/D2/<00>
007415	000			.ASCII	<00>	
007416	122	103	062	.ASCII	/RC2/	
007421	065	040	125	.ASCII	/5 U/	
007424	116	111	124	.ASCII	/NIT/	
007427	040	104	117	.ASCII	/ DO/	
007432	105	123	040	.ASCII	/ES /	
007435	116	117	124	.ASCII	/NOT/	
007440	040	103	117	.ASCII	/ CO/	
007443	115	105	040	.ASCII	/ME /	
007446	117	116	114	.ASCII	/ONL/	
007451	111	116	105	.ASCII	/INE/	
007454	000	000		P.AEI:	.ASCII	<00><00>
007456	105	130	137	.ASCII	/EX /	
007461	123	125	120	.ASCII	/SUP/	
007464	137	120	122	.ASCII	/ PR/	
007467	117	107	040	.ASCII	/OG /	
007472	104	125	120	.ASCII	/DUP/	
007475	040	103	117	.ASCII	/ CO/	
007500	115	115	101	.ASCII	/MMA/	
007503	116	104	040	.ASCII	/ND /	
007506	106	101	111	.ASCII	/FAI/	

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 B1iss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0069
Page 41
(12)

007511	114	125	122		.ASCII	/LUR/
007514	105	000			.ASCII	/E/<00>
007516	123	105	116	P.AEJ:	.ASCII	/SEN/
007521	104	137	104		.ASCII	/D D/
007524	101	124	101		.ASCII	/ATA/
007527	040	104	125		.ASCII	/ DU/
007532	120	040	103		.ASCII	/P C/
007535	117	115	115		.ASCII	/OMM/
007540	101	116	104		.ASCII	/AND/
007543	040	106	101		.ASCII	/ FA/
007546	111	114	125		.ASCII	/ILU/
007551	122	105	000		.ASCII	/RE/<00>
007554	122	105	103	P.AEK:	.ASCII	/REC/
007557	137	104	101		.ASCII	/ DA/
007562	124	101	040		.ASCII	/TA /
007565	104	125	120		.ASCII	/DUP/
007570	040	103	117		.ASCII	/ CO/
007573	115	115	101		.ASCII	/MMA/
007576	116	104	040		.ASCII	/ND /
007601	106	101	111		.ASCII	/FAI/
007604	114	125	122		.ASCII	/LUR/
007607	105	000	000		.ASCII	/E/<00><00>
007612	045	116	045	P.AEM:	.ASCII	/N#/
007615	101	044	106		.ASCII	/A\$F/
007620	124	114	105		.ASCII	/TLE/
007623	122	122	055		.ASCII	/RR-/
007626	040	125	116		.ASCII	/ UN/
007631	122	105	103		.ASCII	/REC/
007634	117	107	116		.ASCII	/OGN/
007637	111	132	101		.ASCII	/IZA/
007642	102	114	105		.ASCII	/BLE/
007645	040	105	122		.ASCII	/ ER/
007650	122	117	122		.ASCII	/ROR/
007653	040	103	117		.ASCII	/ CO/
007656	104	105	000		.ASCII	/DE/<00>
007661	000				.ASCII	<00>
007662	045	116	045	P.AEN:	.ASCII	/N#/
007665	101	044	106		.ASCII	/A\$F/
007670	124	114	105		.ASCII	/TLE/
007673	122	122	055		.ASCII	/RR-/
007676	040	105	116		.ASCII	/ EN/
007701	126	105	114		.ASCII	/VEL/
007704	117	120	105		.ASCII	/OPE/
007707	057	120	101		.ASCII	<57>/PA/
007712	103	113	105		.ASCII	/CKE/
007715	124	040	122		.ASCII	/T R/
007720	105	101	104		.ASCII	/EAD/
007723	040	050	120		.ASCII	/ (P/
007726	101	122	111		.ASCII	/ARI/
007731	124	131	040		.ASCII	/TY /
007734	117	122	040		.ASCII	/OR /
007737	124	111	115		.ASCII	/TIM/
007742	105	117	125		.ASCII	/EQU/
007745	124	051	000		.ASCII	/T)/<00>
007750	045	116	045	P.AEO:	.ASCII	/N#/
007753	101	044	106		.ASCII	/A\$F/
007756	124	114	105		.ASCII	/TLE/

007761	122	122	055	.ASCII	/RR-/	
007764	040	105	116	.ASCII	/ EN/	
007767	126	105	114	.ASCII	/VEL/	
007772	117	120	105	.ASCII	/OPE/	
007775	057	120	101	.ASCII	<57>/PA/	
010000	103	113	105	.ASCII	/CKE/	
010003	124	040	127	.ASCII	/T W/	
010006	122	111	124	.ASCII	/RIT/	
010011	105	040	050	.ASCII	/E (/	
010014	120	101	122	.ASCII	/PAR/	
010017	111	124	131	.ASCII	/ITY/	
010022	040	117	122	.ASCII	/ OR/	
010025	040	124	111	.ASCII	/ TI/	
010030	115	105	117	.ASCII	/MEQ/	
010033	125	124	051	.ASCII	/UT)/	
010036	000	000		.ASCII	<00><00>	
010040	045	116	045	P.AEP:	.ASCII	/N#/
010043	101	044	106	.ASCII	/A\$F/	
010046	124	114	105	.ASCII	/TLE/	
010051	122	122	055	.ASCII	/RR-/	
010054	040	103	117	.ASCII	/ CO/	
010057	116	124	122	.ASCII	/NTR/	
010062	117	114	114	.ASCII	/OLL/	
010065	105	122	040	.ASCII	/ER /	
010070	122	117	115	.ASCII	/ROM/	
010073	040	101	116	.ASCII	/ AN/	
010076	104	040	122	.ASCII	/D R/	
010101	101	115	040	.ASCII	/AM /	
010104	120	101	122	.ASCII	/PAR/	
010107	111	124	131	.ASCII	/ITY/	
010112	000	000		.ASCII	<00><00>	
010114	045	116	045	P.AEQ:	.ASCII	/N#/
010117	101	044	106	.ASCII	/A\$F/	
010122	124	114	105	.ASCII	/TLE/	
010125	122	122	055	.ASCII	/RR-/	
010130	040	103	117	.ASCII	/ CO/	
010133	116	124	122	.ASCII	/NTR/	
010136	117	114	114	.ASCII	/OLL/	
010141	105	122	040	.ASCII	/ER /	
010144	122	101	115	.ASCII	/RAM/	
010147	040	120	101	.ASCII	/ PA/	
010152	122	111	124	.ASCII	/RIT/	
010155	131	000	000	.ASCII	/Y/<00><00>	
010160	045	116	045	P.AER:	.ASCII	/N#/
010163	101	044	106	.ASCII	/A\$F/	
010166	124	114	105	.ASCII	/TLE/	
010171	122	122	055	.ASCII	/RR-/	
010174	040	103	117	.ASCII	/ CO/	
010177	116	124	122	.ASCII	/NTR/	
010202	117	114	114	.ASCII	/OLL/	
010205	105	122	040	.ASCII	/ER /	
010210	122	117	115	.ASCII	/ROM/	
010213	040	120	101	.ASCII	/ PA/	
010216	122	111	124	.ASCII	/RIT/	
010221	131	000	000	.ASCII	/Y/<00><00>	
010224	045	116	045	P.AES:	.ASCII	/N#/
010227	101	044	106	.ASCII	/A\$F/	

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0071
Page 43
(12)

010232	124	114	105	.ASCII	/TLE/	
010235	122	122	055	.ASCII	/RR-/	
010240	040	122	111	.ASCII	/RI/	
010243	116	107	040	.ASCII	/NG /	
010246	122	105	101	.ASCII	/REA/	
010251	104	040	050	.ASCII	/D (/	
010254	120	101	122	.ASCII	/PAR/	
010257	111	124	131	.ASCII	/ITY/	
010262	040	117	122	.ASCII	/OR/	
010265	040	124	111	.ASCII	/TI/	
010270	115	105	117	.ASCII	/MEO/	
010273	125	124	051	.ASCII	/UT)/	
010276	000	000		.ASCII	<00><00>	
010300	045	116	045	P.AET:	.ASCII	/N#/
010303	101	044	106	.ASCII	/A\$F/	
010306	124	114	105	.ASCII	/TLE/	
010311	122	122	055	.ASCII	/RR-/	
010314	040	122	111	.ASCII	/RI/	
010317	116	107	040	.ASCII	/NG /	
010322	127	122	111	.ASCII	/WRI/	
010325	124	105	040	.ASCII	/TE /	
010330	050	120	101	.ASCII	/(PA/	
010333	122	111	124	.ASCII	/RIT/	
010336	131	040	117	.ASCII	/Y O/	
010341	122	040	124	.ASCII	/R T/	
010344	111	115	105	.ASCII	/IME/	
010347	117	125	124	.ASCII	/OUT/	
010352	051	000		.ASCII	/)/<00>	
010354	045	116	045	P.AEU:	.ASCII	/N#/
010357	101	044	106	.ASCII	/A\$F/	
010362	124	114	105	.ASCII	/TLE/	
010365	122	122	055	.ASCII	/RR-/	
010370	040	111	116	.ASCII	/IN/	
010373	124	105	122	.ASCII	/TER/	
010376	122	125	120	.ASCII	/RUP/	
010401	124	040	115	.ASCII	/T M/	
010404	101	123	124	.ASCII	/AST/	
010407	105	122	000	.ASCII	/ER/<00>	
010412	045	116	045	P.AEV:	.ASCII	/N#/
010415	101	044	106	.ASCII	/A\$F/	
010420	124	114	105	.ASCII	/TLE/	
010423	122	122	055	.ASCII	/RR-/	
010426	040	110	117	.ASCII	/HO/	
010431	123	124	040	.ASCII	/ST /	
010434	101	103	103	.ASCII	/ACC/	
010437	105	123	123	.ASCII	/ESS/	
010442	040	124	111	.ASCII	/TI/	
010445	115	105	117	.ASCII	/MEO/	
010450	125	124	000	.ASCII	/UT/<00>	
010453	000			.ASCII	<00>	
010454	045	116	045	P.AEW:	.ASCII	/N#/
010457	101	044	106	.ASCII	/A\$F/	
010462	124	114	105	.ASCII	/TLE/	
010465	122	122	055	.ASCII	/RR-/	
010470	040	103	122	.ASCII	/CR/	
010473	105	104	111	.ASCII	/EDI/	
010476	124	040	114	.ASCII	/T L/	

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0072
Page 44
(12)

010501	111	115	111	.ASCII	/IMI/	
010504	124	040	105	.ASCII	/T E/	
010507	130	103	105	.ASCII	/XCE/	
010512	105	104	105	.ASCII	/EDE/	
010515	104	000	000	.ASCII	/D/<00><00>	
010520	045	116	045	P.AEX:	.ASCII	/N#/
010523	101	044	106	.ASCII	/A\$F/	
010526	124	114	105	.ASCII	/TLE/	
010531	122	122	055	.ASCII	/RR-/	
010534	040	102	125	.ASCII	/BU/	
010537	123	040	115	.ASCII	/S M/	
010542	101	123	124	.ASCII	/AST/	
010545	105	122	040	.ASCII	/ER /	
010550	105	122	122	.ASCII	/ERR/	
010553	117	122	000	.ASCII	/OR/<00>	
010556	045	116	045	P.AEY:	.ASCII	/N#/
010561	101	044	106	.ASCII	/A\$F/	
010564	124	114	105	.ASCII	/TLE/	
010567	122	122	055	.ASCII	/RR-/	
010572	040	104	111	.ASCII	/DI/	
010575	101	107	116	.ASCII	/AGN/	
010600	117	123	124	.ASCII	/OST/	
010603	111	103	040	.ASCII	/IC /	
010606	103	117	116	.ASCII	/CON/	
010611	124	122	117	.ASCII	/TRO/	
010614	114	114	105	.ASCII	/LLE/	
010617	122	040	106	.ASCII	/R F/	
010622	101	124	101	.ASCII	/ATA/	
010625	114	040	105	.ASCII	/L E/	
010630	122	122	117	.ASCII	/RRO/	
010633	122	000	000	.ASCII	/R/<00><00>	
010636	045	116	045	P.AEZ:	.ASCII	/N#/
010641	101	044	106	.ASCII	/A\$F/	
010644	124	114	105	.ASCII	/TLE/	
010647	122	122	055	.ASCII	/RR-/	
010652	040	111	116	.ASCII	/IN/	
010655	123	124	122	.ASCII	/STR/	
010660	125	103	124	.ASCII	/UCT/	
010663	111	117	116	.ASCII	/ION/	
010666	040	114	117	.ASCII	/LO/	
010671	117	120	040	.ASCII	/OP /	
010674	124	111	115	.ASCII	/TIM/	
010677	105	117	125	.ASCII	/EQU/	
010702	124	000		.ASCII	/T/<00>	
010704	045	116	045	P.AFA:	.ASCII	/N#/
010707	101	044	106	.ASCII	/A\$F/	
010712	124	114	105	.ASCII	/TLE/	
010715	122	122	055	.ASCII	/RR-/	
010720	040	111	116	.ASCII	/IN/	
010723	126	101	114	.ASCII	/VAL/	
010726	111	104	040	.ASCII	/ID /	
010731	103	117	116	.ASCII	/CON/	
010734	116	105	103	.ASCII	/NEC/	
010737	124	111	117	.ASCII	/TIO/	
010742	116	040	111	.ASCII	/N I/	
010745	104	105	116	.ASCII	/DEN/	
010750	124	111	106	.ASCII	/TIF/	

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0073
Page 45
(12)

010753	111	105	122		.ASCII	/IER/
010756	000	000			.ASCII	<00><00>
010760	045	116	045	P.AFB:	.ASCII	/N#/
010763	101	044	106		.ASCII	/A\$F/
010766	124	114	105		.ASCII	/TLE/
010771	122	122	055		.ASCII	/RR-/
010774	040	111	116		.ASCII	/ IN/
010777	124	105	122		.ASCII	/TER/
011002	122	125	120		.ASCII	/RUP/
011005	124	040	127		.ASCII	/T W/
011010	122	111	124		.ASCII	/RIT/
011013	105	000	000		.ASCII	/E/<00><00>
011016	045	116	045	P.AFC:	.ASCII	/N#/
011021	101	044	106		.ASCII	/A\$F/
011024	124	114	105		.ASCII	/TLE/
011027	122	122	055		.ASCII	/RR-/
011032	040	115	101		.ASCII	/ MA/
011035	111	116	124		.ASCII	/INT/
011040	105	116	101		.ASCII	/ENA/
011043	116	103	105		.ASCII	/NCE/
011046	040	122	105		.ASCII	/ RE/
011051	101	104	057		.ASCII	/AD/<57>
011054	127	122	111		.ASCII	/WRI/
011057	124	105	040		.ASCII	/TE /
011062	111	116	126		.ASCII	/INV/
011065	101	114	111		.ASCII	/ALI/
011070	104	040	122		.ASCII	/D R/
011073	105	107	111		.ASCII	/EGI/
011076	117	116	040		.ASCII	/ON /
011101	111	104	105		.ASCII	/IDE/
011104	116	124	111		.ASCII	/NTI/
011107	106	111	105		.ASCII	/FIE/
011112	122	000			.ASCII	/R/<00>
011114	045	116	045	P.AFD:	.ASCII	/N#/
011117	101	044	106		.ASCII	/A\$F/
011122	124	114	105		.ASCII	/TLE/
011125	122	122	055		.ASCII	/RR-/
011130	040	115	101		.ASCII	/ MA/
011133	111	116	124		.ASCII	/INT/
011136	105	116	101		.ASCII	/ENA/
011141	116	103	105		.ASCII	/NCE/
011144	040	127	122		.ASCII	/ WR/
011147	111	124	105		.ASCII	/ITE/
011152	040	114	117		.ASCII	/ LO/
011155	101	104	040		.ASCII	/AD /
011160	124	117	040		.ASCII	/TO /
011163	116	117	116		.ASCII	/NON/
011166	055	114	117		.ASCII	/-LO/
011171	101	104	101		.ASCII	/ADA/
011174	102	114	105		.ASCII	/BLE/
011177	040	103	117		.ASCII	/ CO/
011202	116	124	122		.ASCII	/NTR/
011205	117	114	114		.ASCII	/OLL/
011210	105	122	000		.ASCII	/ER/<00>
011213	000				.ASCII	<00>
011214	045	116	045	P.AFE:	.ASCII	/N#/
011217	101	044	106		.ASCII	/A\$F/

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

011222	124	114	105	.ASCII	/TLE/
011225	122	122	055	.ASCII	/RR-/
011230	040	103	117	.ASCII	/ CO/
011233	116	124	122	.ASCII	/NTR/
011236	117	114	114	.ASCII	/OLL/
011241	105	122	040	.ASCII	/ER /
011244	122	101	115	.ASCII	/RAM/
011247	040	105	122	.ASCII	/ ER/
011252	122	117	122	.ASCII	/ROR/
011255	040	050	116	.ASCII	/ (N/
011260	117	116	055	.ASCII	/ON-/
011263	120	101	122	.ASCII	/PAR/
011266	111	124	131	.ASCII	/ITY/
011271	051	000	000	.ASCII	/)/<00><00>
011274	045	116	045	P.AFF: .ASCII	/N#/
011277	101	044	106	.ASCII	/A\$F/
011302	124	114	105	.ASCII	/TLE/
011305	122	122	055	.ASCII	/RR-/
011310	040	111	116	.ASCII	/ IN/
011313	111	124	040	.ASCII	/IT /
011316	123	105	121	.ASCII	/SEQ/
011321	125	105	116	.ASCII	/UEN/
011324	103	105	040	.ASCII	/CE /
011327	105	122	122	.ASCII	/ERR/
011332	117	122	000	.ASCII	/OR/<00>
011335	000			.ASCII	<00>
011336	045	116	045	P.AFG: .ASCII	/N#/
011341	101	044	106	.ASCII	/A\$F/
011344	124	114	105	.ASCII	/TLE/
011347	122	122	055	.ASCII	/RR-/
011352	040	110	111	.ASCII	/ HI/
011355	107	110	040	.ASCII	/GH /
011360	114	105	126	.ASCII	/LEV/
011363	105	114	040	.ASCII	/EL /
011366	120	122	117	.ASCII	/PRO/
011371	124	117	103	.ASCII	/TOC/
011374	117	114	040	.ASCII	/OL /
011377	111	116	103	.ASCII	/INC/
011402	117	115	120	.ASCII	/OMP/
011405	101	124	111	.ASCII	/ATI/
011410	102	111	114	.ASCII	/BIL/
011413	111	124	131	.ASCII	/ITY/
011416	040	105	122	.ASCII	/ ER/
011421	122	117	122	.ASCII	/ROR/
011424	000	000		.ASCII	<00><00>
011426	045	116	045	P.AFH: .ASCII	/N#/
011431	101	044	106	.ASCII	/A\$F/
011434	124	114	105	.ASCII	/TLE/
011437	122	122	055	.ASCII	/RR-/
011442	040	120	125	.ASCII	/ PU/
011445	122	107	105	.ASCII	/RGE/
011450	057	120	117	.ASCII	<57>/PO/
011453	114	114	040	.ASCII	/LL /
011456	110	101	122	.ASCII	/HAR/
011461	104	127	101	.ASCII	/DWA/
011464	122	105	040	.ASCII	/RE /
011467	106	101	111	.ASCII	/FAI/

011472	114	125	122	.ASCII	/LUR/
011475	105	040	000	.ASCII	/E /<00>
011500	045	116	045	P.AFI: .ASCII	/N#/
011503	101	044	106	.ASCII	/A\$F/
011506	124	114	105	.ASCII	/TLE/
011511	122	122	055	.ASCII	/RR-/
011514	040	115	101	.ASCII	/ MA/
011517	120	120	111	.ASCII	/PPI/
011522	116	107	040	.ASCII	/NG /
011525	122	105	107	.ASCII	/REG/
011530	111	123	124	.ASCII	/IST/
011533	105	122	040	.ASCII	/ER /
011536	122	105	101	.ASCII	/REA/
011541	104	040	105	.ASCII	/D E/
011544	122	122	117	.ASCII	/RRO/
011547	122	040	050	.ASCII	/R (/
011552	120	101	122	.ASCII	/PAR/
011555	111	124	131	.ASCII	/ITY/
011560	040	117	122	.ASCII	/ OR/
011563	040	124	111	.ASCII	/ TI/
011566	115	105	117	.ASCII	/MEO/
011571	125	124	051	.ASCII	/UT)/
011574	000	000		.ASCII	<00><00>
011576	007612'			P.AEL: .WORD	P.AEM
011600	007662'			.WORD	P.AEN
011602	007750'			.WORD	P.AEO
011604	010040'			.WORD	P.AEP
011606	010114'			.WORD	P.AEQ
011610	010160'			.WORD	P.AER
011612	010224'			.WORD	P.AES
011614	010300'			.WORD	P.AET
011616	010354'			.WORD	P.AEU
011620	010412'			.WORD	P.AEV
011622	010454'			.WORD	P.AEW
011624	010520'			.WORD	P.AEX
011626	010556'			.WORD	P.AEY
011630	010636'			.WORD	P.AEZ
011632	010704'			.WORD	P.AFA
011634	010760'			.WORD	P.AFB
011636	011016'			.WORD	P.AFC
011640	011114'			.WORD	P.AFD
011642	011214'			.WORD	P.AFE
011644	011274'			.WORD	P.AFF
011646	011336'			.WORD	P.AFG
011650	011426'			.WORD	P.AFH
011652	011500'			.WORD	P.AFI
011654	045	116	045	P.AFK: .ASCII	/N#/
011657	101	044	106	.ASCII	/A\$F/
011662	124	114	105	.ASCII	/TLE/
011665	122	122	055	.ASCII	/RR-/
011670	040	122	105	.ASCII	/ RE/
011673	123	120	117	.ASCII	/SPO/
011676	116	123	105	.ASCII	/NSE/
011701	040	123	124	.ASCII	/ ST/
011704	101	124	125	.ASCII	/ATU/
011707	123	040	105	.ASCII	/S E/
011712	122	122	117	.ASCII	/RRO/

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0076
Page 48
(12)

011715	122	072	045		.ASCII	/R:\$/
011720	123	000			.ASCII	/S/<00>
011722	045	116	045	P.AFL:	.ASCII	/N\$/
011725	101	044	106		.ASCII	/A\$F/
011730	124	114	105		.ASCII	/TLE/
011733	122	122	055		.ASCII	/RR-/
011736	040	123	125		.ASCII	/ SU/
011741	120	105	122		.ASCII	/PER/
011744	126	111	123		.ASCII	/VIS/
011747	117	122	040		.ASCII	/OR /
011752	123	105	122		.ASCII	/SER/
011755	126	111	103		.ASCII	/VIC/
011760	105	040	103		.ASCII	/E C/
011763	101	114	114		.ASCII	/ALL/
011766	040	106	101		.ASCII	/ FA/
011771	111	114	105		.ASCII	/ILE/
011774	104	000			.ASCII	/D/<00>
011776	045	116	045	P.AFM:	.ASCII	/N\$/
012001	101	044	106		.ASCII	/A\$F/
012004	124	114	105		.ASCII	/TLE/
012007	122	122	055		.ASCII	/RR-/
012012	040	120	117		.ASCII	/ PO/
012015	122	124	057		.ASCII	/RT/<57>
012020	103	117	116		.ASCII	/CON/
012023	124	122	117		.ASCII	/TRO/
012026	114	114	105		.ASCII	/LLE/
012031	122	040	124		.ASCII	/R T/
012034	111	115	105		.ASCII	/IME/
012037	117	125	124		.ASCII	/OUT/
012042	040	105	122		.ASCII	/ ER/
012045	122	117	122		.ASCII	/ROR/
012050	000	000			.ASCII	<00><00>
012052	045	116	045	P.AFN:	.ASCII	/N\$/
012055	101	044	106		.ASCII	/A\$F/
012060	124	114	105		.ASCII	/TLE/
012063	122	122	055		.ASCII	/RR-/
012066	040	125	116		.ASCII	/ UN/
012071	113	116	117		.ASCII	/KNO/
012074	127	116	040		.ASCII	/WN /
012077	122	105	124		.ASCII	/RET/
012102	125	122	116		.ASCII	/URN/
012105	040	123	124		.ASCII	/ ST/
012110	101	124	125		.ASCII	/ATU/
012113	123	040	103		.ASCII	/S C/
012116	117	104	105		.ASCII	/ODE/
012121	000				.ASCII	<00>
012122	011654'			P.AFJ:	.WORD	P.AFK
012124	011722'				.WORD	P.AFL
012126	011776'				.WORD	P.AFM
012130	012052'				.WORD	P.AFN
012132	045	116	045	P.AFP:	.ASCII	/N\$/
012135	101	044	106		.ASCII	/A\$F/
012140	124	114	105		.ASCII	/TLE/
012143	122	122	055		.ASCII	/RR-/
012146	040	126	101		.ASCII	/ VA/
012151	130	040	122		.ASCII	/X R/
012154	105	101	104		.ASCII	/EAD/

012157	057	127	122	.ASCII	<57>/WR/
012162	111	124	105	.ASCII	/ITE/
012165	040	105	122	.ASCII	/ER/
012170	122	117	122	.ASCII	/ROR/
012173	040	117	116	.ASCII	/ON/
012176	040	111	116	.ASCII	/IN/
012201	124	105	122	.ASCII	/TER/
012204	122	125	120	.ASCII	/RUP/
012207	124	000	000	.ASCII	/T/<00><00>
012212	045	116	045	P.AFQ: .ASCII	/N#/
012215	101	044	106	.ASCII	/A#F/
012220	124	114	105	.ASCII	/TLE/
012223	122	122	055	.ASCII	/RR-/
012226	040	111	116	.ASCII	/IN/
012231	103	117	116	.ASCII	/CON/
012234	123	111	123	.ASCII	/SIS/
012237	124	105	116	.ASCII	/TEN/
012242	103	131	040	.ASCII	/CY /
012245	101	124	040	.ASCII	/AT /
012250	125	056	102	.ASCII	/U.B/
012253	106	111	114	.ASCII	/FIL/
012256	000	000		.ASCII	<00><00>
012260	045	116	045	P.AFR: .ASCII	/N#/
012263	101	044	106	.ASCII	/A#F/
012266	124	114	105	.ASCII	/TLE/
012271	122	122	055	.ASCII	/RR-/
012274	040	111	116	.ASCII	/IN/
012277	103	117	116	.ASCII	/CON/
012302	123	111	123	.ASCII	/SIS/
012305	124	105	116	.ASCII	/TEN/
012310	103	131	040	.ASCII	/CY /
012313	101	124	040	.ASCII	/AT /
012316	125	056	102	.ASCII	/U.B/
012321	115	124	131	.ASCII	/MTY/
012324	000	000		.ASCII	<00><00>
012326	045	116	045	P.AFS: .ASCII	/N#/
012331	101	044	106	.ASCII	/A#F/
012334	124	114	105	.ASCII	/TLE/
012337	122	122	055	.ASCII	/RR-/
012342	040	111	116	.ASCII	/IN/
012345	103	117	116	.ASCII	/CON/
012350	123	111	123	.ASCII	/SIS/
012353	124	105	116	.ASCII	/TEN/
012356	103	131	040	.ASCII	/CY /
012361	101	124	040	.ASCII	/AT /
012364	125	056	101	.ASCII	/U.A/
012367	114	117	103	.ASCII	/LOC/
012372	000	000		.ASCII	<00><00>
012374	045	116	045	P.AFT: .ASCII	/N#/
012377	101	044	106	.ASCII	/A#F/
012402	124	114	105	.ASCII	/TLE/
012405	122	122	055	.ASCII	/RR-/
012410	040	111	116	.ASCII	/IN/
012413	103	117	116	.ASCII	/CON/
012416	123	111	123	.ASCII	/SIS/
012421	124	105	116	.ASCII	/TEN/
012424	103	131	040	.ASCII	/CY /

ZRCFR1
V03.0

CZRCFCO RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 B1100-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16:1

SEQ 0078
Page 50
(12)

012427	101	124	040	.ASCII	/AT /
012432	123	105	122	.ASCII	/SER/
012435	126	117	040	.ASCII	/VO /
012440	105	116	124	.ASCII	/ENT/
012443	122	131	040	.ASCII	/RY /
012446	050	120	111	.ASCII	/(PI/
012451	120	040	123	.ASCII	/P S/
012454	105	124	051	.ASCII	/ET)/
012457	000			.ASCII	<00>
012460	045	116	045	P.AFU: .ASCII	/N#/
012463	101	044	106	.ASCII	/A#F/
012466	124	114	105	.ASCII	/TLE/
012471	122	122	055	.ASCII	/RR-/
012474	040	111	116	.ASCII	/ IN/
012477	103	117	116	.ASCII	/CON/
012502	123	111	123	.ASCII	/SIS/
012505	124	105	116	.ASCII	/TEN/
012510	103	131	040	.ASCII	/CY /
012513	101	124	040	.ASCII	/AT /
012516	123	105	122	.ASCII	/SER/
012521	126	117	040	.ASCII	/VO /
012524	105	116	124	.ASCII	/ENT/
012527	122	131	040	.ASCII	/RY /
012532	050	105	122	.ASCII	/(ER/
012535	122	040	123	.ASCII	/R S/
012540	105	124	051	.ASCII	/ET)/
012543	000			.ASCII	<00>
012544	045	116	045	P.AFV: .ASCII	/N#/
012547	101	044	106	.ASCII	/A#F/
012552	124	114	105	.ASCII	/TLE/
012555	122	122	055	.ASCII	/RR-/
012560	040	111	116	.ASCII	/ IN/
012563	103	117	116	.ASCII	/CON/
012566	123	111	123	.ASCII	/SIS/
012571	124	105	116	.ASCII	/TEN/
012574	103	131	040	.ASCII	/CY /
012577	101	124	040	.ASCII	/AT /
012602	125	056	123	.ASCII	/U.S/
012605	105	116	104	.ASCII	/END/
012610	000	000		.ASCII	<00><00>
012612	045	116	045	P.AFW: .ASCII	/N#/
012615	101	044	106	.ASCII	/A#F/
012620	124	114	105	.ASCII	/TLE/
012623	122	122	055	.ASCII	/RR-/
012626	040	111	116	.ASCII	/ IN/
012631	103	117	116	.ASCII	/CON/
012634	123	111	123	.ASCII	/SIS/
012637	124	105	116	.ASCII	/TEN/
012642	103	131	040	.ASCII	/CY /
012645	101	124	040	.ASCII	/AT /
012650	125	056	122	.ASCII	/U.R/
012653	105	103	126	.ASCII	/ECV/
012656	000	000		.ASCII	<00><00>
012660	045	116	045	P.AFX: .ASCII	/N#/
012663	101	044	106	.ASCII	/A#F/
012666	124	114	105	.ASCII	/TLE/
012671	122	122	055	.ASCII	/RR-/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 B1100-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC1.B16;1

SEQ 0079
Page 51
(12)

012674	040	111	116	.ASCII	/ IN/
012677	103	117	116	.ASCII	/CON/
012702	123	111	123	.ASCII	/SIS/
012705	124	105	116	.ASCII	/TEN/
012710	103	131	040	.ASCII	/CY /
012713	101	124	040	.ASCII	/AT /
012716	125	056	101	.ASCII	/U.A/
012721	124	124	116	.ASCII	/TTN/
012724	000	000		.ASCII	<00><00>
012726	045	116	045	P.AFY:	.ASCII /#N#/
012731	101	044	106	.ASCII	/A#F/
012734	124	114	105	.ASCII	/TLE/
012737	122	122	055	.ASCII	/RR-/
012742	040	111	116	.ASCII	/ IN/
012745	103	117	116	.ASCII	/CON/
012750	123	111	123	.ASCII	/SIS/
012753	124	105	116	.ASCII	/TEN/
012756	103	131	040	.ASCII	/CY /
012761	101	124	040	.ASCII	/AT /
012764	125	056	117	.ASCII	/U.O/
012767	116	114	116	.ASCII	/NLN/
012772	000	000		.ASCII	<00><00>
012774	045	116	045	P.AFZ:	.ASCII /#N#/
012777	101	044	106	.ASCII	/A#F/
013002	124	114	105	.ASCII	/TLE/
013005	122	122	055	.ASCII	/RR-/
013010	040	111	114	.ASCII	/ IL/
013013	114	105	107	.ASCII	/LEG/
013016	101	114	040	.ASCII	/AL /
013021	104	040	122	.ASCII	/D R/
013024	105	121	125	.ASCII	/EQU/
013027	105	123	124	.ASCII	/EST/
013032	040	050	125	.ASCII	/ (U/
013035	056	121	104	.ASCII	/.QD/
013040	122	121	051	.ASCII	/RQ)/
013043	000			.ASCII	<00>
013044	045	116	045	P.AGA:	.ASCII /#N#/
013047	101	044	106	.ASCII	/A#F/
013052	124	114	105	.ASCII	/TLE/
013055	122	122	055	.ASCII	/RR-/
013060	040	106	105	.ASCII	/ FE/
013063	116	103	105	.ASCII	/NCE/
013066	055	120	117	.ASCII	/-PO/
013071	123	124	040	.ASCII	/ST /
013074	105	122	122	.ASCII	/ERR/
013077	117	122	040	.ASCII	/OR /
013102	101	124	040	.ASCII	/AT /
013105	120	122	117	.ASCII	/PRO/
013110	124	101	102	.ASCII	/TAB/
013113	000			.ASCII	<00>
013114	045	116	045	P.AGB:	.ASCII /#N#/
013117	101	044	106	.ASCII	/A#F/
013122	124	114	105	.ASCII	/TLE/
013125	122	122	055	.ASCII	/RR-/
013130	040	102	101	.ASCII	/ BA/
013133	104	040	120	.ASCII	/D P/
013136	101	103	113	.ASCII	/ACK/

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

013141	105	124	040	.ASCII	/ET /
013144	104	105	121	.ASCII	/DEQ/
013147	125	105	125	.ASCII	/UEU/
013152	105	104	040	.ASCII	/ED /
013155	101	124	040	.ASCII	/AT /
013160	125	056	104	.ASCII	/U.D/
013163	117	116	105	.ASCII	/ONE/
013166	000	000		.ASCII	<00><00>
013170	045	116	045	P.AGC:	.ASCII /%N%/
013173	101	044	106	.ASCII	/A\$F/
013176	124	114	105	.ASCII	/TLE/
013201	122	122	055	.ASCII	/RR-/
013204	040	125	116	.ASCII	/ UN/
013207	105	130	120	.ASCII	/EXP/
013212	114	101	111	.ASCII	/LAI/
013215	116	105	104	.ASCII	/NED/
013220	040	104	055	.ASCII	/ D-/
013223	120	122	117	.ASCII	/PRO/
013226	103	040	123	.ASCII	/C S/
013231	125	123	120	.ASCII	/USP/
013234	105	116	123	.ASCII	/ENS/
013237	111	117	116	.ASCII	/ION/
013242	040	050	125	.ASCII	/ (U/
013245	056	056	124	.ASCII	/ .T/
013250	104	123	051	.ASCII	/DS)/
013253	000			.ASCII	<00>
013254	045	116	045	P.AGD:	.ASCII /%N%/
013257	101	044	106	.ASCII	/A\$F/
013262	124	114	105	.ASCII	/TLE/
013265	122	122	055	.ASCII	/RR-/
013270	040	104	125	.ASCII	/ DU/
013273	120	040	120	.ASCII	/P P/
013276	101	103	113	.ASCII	/ACK/
013301	105	124	040	.ASCII	/ET /
013304	104	055	121	.ASCII	/D-Q/
013307	040	106	101	.ASCII	/ FA/
013312	111	114	105	.ASCII	/ILE/
013315	104	040	050	.ASCII	/D (/
013320	130	106	103	.ASCII	/XFC/
013323	040	063	064	.ASCII	/ 34/
013326	057	063	065	.ASCII	<57>/35/
013331	051	000	000	.ASCII	/)/<00><00>
013334	045	116	045	P.AGE:	.ASCII /%N%/
013337	101	044	106	.ASCII	/A\$F/
013342	124	114	105	.ASCII	/TLE/
013345	122	122	055	.ASCII	/RR-/
013350	040	111	116	.ASCII	/ IN/
013353	103	117	116	.ASCII	/CON/
013356	123	111	123	.ASCII	/SIS/
013361	124	105	116	.ASCII	/TEN/
013364	103	131	040	.ASCII	/CY /
013367	101	124	040	.ASCII	/AT /
013372	125	056	110	.ASCII	/U.H/
013375	124	123	124	.ASCII	/TST/
013400	000	000		.ASCII	<00><00>
013402	045	116	045	P.AGF:	.ASCII /%N%/
013405	101	044	106	.ASCII	/A\$F/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

013410	124	114	105	.ASCII	/TLE/
013413	122	122	055	.ASCII	/RR-/
013416	040	111	116	.ASCII	/ IN/
013421	103	117	116	.ASCII	/CON/
013424	123	111	123	.ASCII	/SIS/
013427	124	105	116	.ASCII	/TEN/
013432	103	131	040	.ASCII	/CY /
013435	101	124	040	.ASCII	/AT /
013440	125	056	123	.ASCII	/U.S/
013443	105	113	117	.ASCII	/EKO/
013446	000	000		.ASCII	<00><00>
013450	045	116	045	P.AGG:	.ASCII /N#/
013453	101	044	106	.ASCII	/A\$F/
013456	124	114	105	.ASCII	/TLE/
013461	122	122	055	.ASCII	/RR-/
013464	040	111	116	.ASCII	/ IN/
013467	103	117	116	.ASCII	/CON/
013472	123	111	123	.ASCII	/SIS/
013475	124	105	116	.ASCII	/TEN/
013500	103	131	040	.ASCII	/CY /
013503	101	124	040	.ASCII	/AT /
013506	125	056	103	.ASCII	/U.C/
013511	113	123	126	.ASCII	/KSV/
013514	000	000		.ASCII	<00><00>
013516	045	116	045	P.AGH:	.ASCII /N#/
013521	101	044	106	.ASCII	/A\$F/
013524	124	114	105	.ASCII	/TLE/
013527	122	122	055	.ASCII	/RR-/
013532	040	104	056	.ASCII	/ D./
013535	117	120	103	.ASCII	/OPC/
013540	104	040	106	.ASCII	/D F/
013543	117	125	116	.ASCII	/OUN/
013546	104	040	111	.ASCII	/D I/
013551	114	114	105	.ASCII	/LLE/
013554	107	101	114	.ASCII	/GAL/
013557	040	117	120	.ASCII	/ OP/
013562	103	117	104	.ASCII	/COD/
013565	105	000	000	.ASCII	/E/<00><00>
013570	045	116	045	P.AGI:	.ASCII /N#/
013573	101	044	106	.ASCII	/A\$F/
013576	124	114	105	.ASCII	/TLE/
013601	122	122	055	.ASCII	/RR-/
013604	040	104	056	.ASCII	/ D./
013607	103	123	106	.ASCII	/CSF/
013612	040	106	117	.ASCII	/ FO/
013615	125	116	104	.ASCII	/UND/
013620	040	111	114	.ASCII	/ IL/
013623	114	105	107	.ASC*I	/LEG/
013626	101	114	040	.ASCII	/AL /
013631	117	120	103	.ASCII	/OPC/
013634	117	104	105	.ASCII	/ODE/
013637	000			.ASCII	<00>
013640	045	116	045	P.AGJ:	.ASCII /N#/
013643	101	044	106	.ASCII	/A\$F/
013646	124	114	105	.ASCII	/TLE/
013651	122	122	055	.ASCII	/RR-/
013654	040	125	116	.ASCII	/ UN/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.816;1

013657	113	116	117	.ASCII	/KNO/
013662	127	116	040	.ASCII	/WN /
013665	102	101	104	.ASCII	/BAD/
013670	040	104	122	.ASCII	/ DR/
013673	111	126	105	.ASCII	/IVE/
013676	040	123	124	.ASCII	/ ST/
013701	101	124	125	.ASCII	/ATU/
013704	123	040	101	.ASCII	/S A/
013707	124	040	104	.ASCII	/T D/
013712	056	104	123	.ASCII	/.DS/
013715	124	123	000	.ASCII	/TS/<00>
013720	045	116	045	P.AGK:	.ASCII /#N#/
013723	101	044	106	.ASCII	/A\$F/
013726	124	114	105	.ASCII	/TLE/
013731	122	122	055	.ASCII	/RR-/
013734	040	111	114	.ASCII	/ IL/
013737	114	105	107	.ASCII	/LEG/
013742	101	114	040	.ASCII	/AL /
013745	130	106	103	.ASCII	/XFC/
013750	040	105	130	.ASCII	/ EX/
013753	105	103	125	.ASCII	/ECU/
013756	124	105	104	.ASCII	/TED/
013761	040	102	131	.ASCII	/ BY/
013764	040	104	115	.ASCII	/ DM/
013767	000			.ASCII	<00>
013770	045	116	045	P.AGL:	.ASCII /#N#/
013773	101	044	106	.ASCII	/A\$F/
013776	124	114	105	.ASCII	/TLE/
014001	122	122	055	.ASCII	/RR-/
014004	040	104	040	.ASCII	/ D /
014007	120	111	103	.ASCII	/PIC/
014012	113	105	104	.ASCII	/KED/
014015	040	125	120	.ASCII	/ UP/
014020	040	101	040	.ASCII	/ A /
014023	132	105	122	.ASCII	/ZER/
014026	117	040	123	.ASCII	/O S/
014031	103	102	056	.ASCII	/CB./
014034	104	102	000	.ASCII	/DB/<00>
014037	000			.ASCII	<00>
014040	045	116	045	P.AGM:	.ASCII /#N#/
014043	101	044	106	.ASCII	/A\$F/
014046	124	114	105	.ASCII	/TLE/
014051	122	122	055	.ASCII	/RR-/
014054	040	111	116	.ASCII	/ IN/
014057	103	117	116	.ASCII	/CON/
014062	123	111	123	.ASCII	/SIS/
014065	124	105	116	.ASCII	/TEN/
014070	103	131	040	.ASCII	/CY /
014073	101	124	040	.ASCII	/AT /
014076	104	040	111	.ASCII	/D I/
014101	104	114	105	.ASCII	/DLE/
014104	040	114	117	.ASCII	/ LO/
014107	117	120	000	.ASCII	/OP/<00>
014112	045	116	045	P.AGN:	.ASCII /#N#/
014115	101	044	106	.ASCII	/A\$F/
014120	124	114	105	.ASCII	/TLE/
014123	122	122	055	.ASCII	/RR-/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0083
Page 55
(12)

014126	040	104	115	.ASCII	/ DM/
014131	040	127	117	.ASCII	/ WO/
014134	122	104	040	.ASCII	/RD /
014137	103	117	125	.ASCII	/COU/
014142	116	124	040	.ASCII	/NT /
014145	105	122	122	.ASCII	/ERR/
014150	117	122	040	.ASCII	/OR /
014153	117	116	040	.ASCII	/ON /
014156	110	117	123	.ASCII	/HOS/
014161	124	040	104	.ASCII	/T D/
014164	115	101	057	.ASCII	/MA/<57>
014167	123	105	116	.ASCII	/SEN/
014172	104	057	122	.ASCII	/D/<57>/R/
014175	105	103	126	.ASCII	/ECV/
014200	000	000		.ASCII	<00><00>
014202	045	116	045	P.AGO: .ASCII	/N#/
014205	101	044	106	.ASCII	/A\$F/
014210	124	114	105	.ASCII	/TLE/
014213	122	122	055	.ASCII	/RR-/
014216	040	125	116	.ASCII	/ UN/
014221	113	116	117	.ASCII	/KNO/
014224	127	116	040	.ASCII	/WN /
014227	104	111	123	.ASCII	/DIS/
014232	120	114	101	.ASCII	/PLA/
014235	131	040	106	.ASCII	/Y F/
014240	101	125	114	.ASCII	/AUL/
014243	124	040	103	.ASCII	/T C/
014246	117	104	105	.ASCII	/ODE/
014251	040	101	124	.ASCII	/ AT/
014254	040	104	056	.ASCII	/ D./
014257	104	106	114	.ASCII	/DFL/
014262	124	000		.ASCII	/T/<00>
014264	045	116	045	P.AGP: .ASCII	/N#/
014267	101	044	106	.ASCII	/A\$F/
014272	124	114	105	.ASCII	/TLE/
014275	122	122	055	.ASCII	/RR-/
014300	040	104	122	.ASCII	/ DR/
014303	111	126	105	.ASCII	/IVE/
014306	040	116	117	.ASCII	/ NO/
014311	124	040	106	.ASCII	/T F/
014314	101	125	114	.ASCII	/AUL/
014317	124	111	116	.ASCII	/TIN/
014322	107	040	111	.ASCII	/G I/
014325	116	040	120	.ASCII	/N P/
014330	056	117	106	.ASCII	/.OF/
014333	114	116	040	.ASCII	/LN /
014336	123	124	101	.ASCII	/STA/
014341	124	105	000	.ASCII	/TE/<00>
014344	045	116	045	P.AGQ: .ASCII	/N#/
014347	101	044	106	.ASCII	/A\$F/
014352	124	114	105	.ASCII	/TLE/
014355	122	122	055	.ASCII	/RR-/
014360	040	125	040	.ASCII	/ U /
014363	120	117	127	.ASCII	/POW/
014366	105	122	040	.ASCII	/ER /
014371	125	120	040	.ASCII	/UP /
014374	104	111	101	.ASCII	/DTA/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.816;1

SEQ 0084
Page 56
(12)

014377	107	116	117	.ASCII	/GNO/
014402	123	124	111	.ASCII	/STI/
014405	103	123	040	.ASCII	/CS /
014410	106	101	111	.ASCII	/FAI/
014413	114	105	104	.ASCII	/LED/
014416	000	000		.ASCII	<00><00>
014420	045	116	045	P.AGR:	.ASCII /#N#/
014423	101	044	106		.ASCII /A\$F/
014426	124	114	105		.ASCII /TLE/
014431	122	122	055		.ASCII /RR-/
014434	040	104	040		.ASCII / D /
014437	120	117	127		.ASCII /POW/
014442	105	122	040		.ASCII /ER /
014445	125	120	040		.ASCII /UP /
014450	104	111	101		.ASCII /DIA/
014453	107	116	117		.ASCII /GNO/
014456	123	124	111		.ASCII /STI/
014461	103	123	040		.ASCII /CS /
014464	106	101	111		.ASCII /FAI/
014467	114	105	104		.ASCII /LED/
014472	000	000			.ASCII <00><00>
014474	045	116	045	P.AGS:	.ASCII /#N#/
014477	101	044	106		.ASCII /A\$F/
014502	124	114	105		.ASCII /TLE/
014505	122	122	055		.ASCII /RR-/
014510	040	101	104		.ASCII / AD/
014513	101	120	124		.ASCII /APT/
014516	105	122	040		.ASCII /ER /
014521	103	101	122		.ASCII /CAR/
014524	104	040	106		.ASCII /D F/
014527	101	111	114		.ASCII /AIL/
014532	125	122	105		.ASCII /URE/
014535	000				.ASCII <00>
014536	045	116	045	P.AGT:	.ASCII /#N#/
014541	101	044	106		.ASCII /A\$F/
014544	124	114	105		.ASCII /TLE/
014547	122	122	055		.ASCII /RR-/
014552	040	105	103		.ASCII / EC/
014555	056	124	115		.ASCII / .TM/
014560	122	040	124		.ASCII /R T/
014563	111	115	105		.ASCII /IME/
014566	104	040	117		.ASCII /D O/
014571	125	124	000		.ASCII /UT/<00>
014574	045	116	045	P.AGU:	.ASCII /#N#/
014577	101	044	106		.ASCII /A\$F/
014602	124	114	105		.ASCII /TLE/
014605	122	122	055		.ASCII /RR-/
014610	040	125	056		.ASCII / U./
014613	123	105	116		.ASCII /SEN/
014616	104	057	125		.ASCII /D/<57>/U/
014621	056	122	105		.ASCII / .RE/
014624	103	126	040		.ASCII /CV /
014627	122	111	116		.ASCII /RIN/
014632	107	040	122		.ASCII /G R/
014635	105	101	104		.ASCII /EAD/
014640	040	111	116		.ASCII / IN/
014643	103	117	116		.ASCII /CON/

014646	123	111	123	.ASCII	/SIS/
014651	124	105	116	.ASCII	/TEN/
014654	103	131	000	.ASCII	/CY/<00>
014657	000			.ASCII	<00>
014660	045	116	045	P.AGV:	.ASCII /%N%/
014663	101	044	106	.ASCII	/A\$F/
014666	124	114	105	.ASCII	/TLE/
014671	122	122	055	.ASCII	/RR-/
014674	040	125	116	.ASCII	/ UN/
014677	113	116	117	.ASCII	/KNO/
014702	127	116	040	.ASCII	/WN /
014705	127	101	111	.ASCII	/WAI/
014710	124	122	126	.ASCII	/TRV/
014713	040	122	105	.ASCII	/ RE/
014716	101	123	117	.ASCII	/ASO/
014721	116	040	101	.ASCII	/N A/
014724	124	040	104	.ASCII	/T D/
014727	056	122	126	.ASCII	/.RV/
014732	103	124	000	.ASCII	/CT/<00>
014735	000			.ASCII	<00>
014736	045	116	045	P.AGW:	.ASCII /%N%/
014741	101	044	106	.ASCII	/A\$F/
014744	124	114	105	.ASCII	/TLE/
014747	122	122	055	.ASCII	/RR-/
014752	040	104	056	.ASCII	/ D./
014755	101	122	103	.ASCII	/ARC/
014760	123	040	104	.ASCII	/S D/
014763	111	104	040	.ASCII	/ID /
014766	116	117	124	.ASCII	/NOT/
014771	040	106	111	.ASCII	/ FI/
014774	116	104	040	.ASCII	/ND /
014777	103	114	117	.ASCII	/CLO/
015002	123	105	123	.ASCII	/SES/
015005	124	040	125	.ASCII	/T U/
015010	116	104	117	.ASCII	/NDO/
015013	116	105	040	.ASCII	/NE /
015016	132	117	116	.ASCII	/ZON/
015021	105	000	000	P.AGX:	.ASCII /E/<00><00>
015024	045	116	045	.ASCII	/%N%/
015027	101	044	106	.ASCII	/A\$F/
015032	124	114	105	.ASCII	/TLE/
015035	122	122	055	.ASCII	/RR-/
015040	040	125	056	.ASCII	/ U./
015043	123	105	105	.ASCII	/SEE/
015046	113	040	106	.ASCII	/K F/
015051	117	125	116	.ASCII	/OUN/
015054	104	040	123	.ASCII	/D S/
015057	105	105	113	.ASCII	/EEK/
015062	040	124	117	.ASCII	/ TO/
015065	040	111	114	.ASCII	/ IL/
015070	114	105	107	.ASCII	/LEG/
015073	101	114	040	.ASCII	/AL /
015076	124	122	101	.ASCII	/TRA/
015101	103	113	000	P.AGY:	.ASCII /CK/<00>
015104	045	116	045	.ASCII	/%N%/
015107	101	044	106	.ASCII	/A\$F/
015112	124	114	105	.ASCII	/TLE/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.816;1

SEQ 0086
Page 58
(12)

015115	122	122	055	.ASCII	/RR-/
015120	040	125	056	.ASCII	/ U./
015123	110	124	123	.ASCII	/HTS/
015126	124	040	111	.ASCII	/T I/
015131	116	111	124	.ASCII	/NIT/
015134	040	104	111	.ASCII	/ DI/
015137	101	107	040	.ASCII	/AG /
015142	104	115	101	.ASCII	/DMA/
015145	040	127	122	.ASCII	/ WR/
015150	111	124	105	.ASCII	/ITE/
015153	040	106	101	.ASCII	/ FA/
015156	111	114	105	.ASCII	/ILE/
015161	104	000	000	.ASCII	/D/<00><00>
015164	045	116	045	P.AGZ: .ASCII	/N#/
015167	101	044	106	.ASCII	/A\$F/
015172	124	114	105	.ASCII	/TLE/
015175	122	122	055	.ASCII	/RR-/
015200	040	125	056	.ASCII	/ U./
015203	110	124	123	.ASCII	/HTS/
015206	124	040	111	.ASCII	/T I/
015211	116	111	124	.ASCII	/NIT/
015214	040	104	111	.ASCII	/ DI/
015217	101	107	040	.ASCII	/AG /
015222	104	115	101	.ASCII	/DMA/
015225	040	103	117	.ASCII	/ CO/
015230	115	120	101	.ASCII	/MPA/
015233	122	105	040	.ASCII	/RE /
015236	106	101	111	.ASCII	/FAI/
015241	114	105	104	.ASCII	/LED/
015244	000	000		.ASCII	<00><00>
015246	045	116	045	P.AHA: .ASCII	/N#/
015251	101	044	106	.ASCII	/A\$F/
015254	124	114	105	.ASCII	/TLE/
015257	122	122	055	.ASCII	/RR-/
015262	040	125	056	.ASCII	/ U./
015265	123	131	104	.ASCII	/SYD/
015270	122	040	106	.ASCII	/R F/
015273	117	125	116	.ASCII	/OUN/
015276	104	040	123	.ASCII	/D S/
015301	123	056	104	.ASCII	/S.D/
015304	105	122	040	.ASCII	/ER /
015307	123	105	124	.ASCII	/SET/
015312	040	101	116	.ASCII	/ AN/
015315	104	040	123	.ASCII	/D S/
015320	123	056	123	.ASCII	/S.S/
015323	120	116	040	.ASCII	/PN /
015326	116	117	124	.ASCII	/NOT/
015331	040	123	105	.ASCII	/ SE/
015334	124	000		.ASCII	/T/<00>
015336	045	116	045	P.AHB: .ASCII	/N#/
015341	101	044	106	.ASCII	/A\$F/
015344	124	114	105	.ASCII	/TLE/
015347	122	122	055	.ASCII	/RR-/
015352	040	115	101	.ASCII	/ MA/
015355	123	124	105	.ASCII	/STE/
015360	122	040	104	.ASCII	/R D/
015363	122	111	126	.ASCII	/RIV/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0087
Page 59
(12)

015366	105	123	040	.ASCII	/ES /
015371	101	103	114	.ASCII	/ACL/
015374	117	040	101	.ASCII	/O A/
015377	123	123	105	.ASCII	/SSE/
015402	122	124	105	.ASCII	/RTE/
015405	104	000	000	.ASCII	/D/<00><00>
015410	012132'			P.AFO: .WORD	P.AFP
015412	012212'			.WORD	P.AFQ
015414	012260'			.WORD	P.AFR
015416	012326'			.WORD	P.AFS
015420	012374'			.WORD	P.AFT
015422	012460'			.WORD	P.AFU
015424	012544'			.WORD	P.AFV
015426	012612'			.WORD	P.AFW
015430	012660'			.WORD	P.AFX
015432	012726'			.WORD	P.AFY
015434	012774'			.WORD	P.AFZ
015436	013044'			.WORD	P.AGA
015440	013114'			.WORD	P.AGB
015442	013170'			.WORD	P.AGC
015444	013254'			.WORD	P.AGD
015446	013334'			.WORD	P.AGE
015450	013402'			.WORD	P.AGF
015452	013450'			.WORD	P.AGG
015454	013516'			.WORD	P.AGH
015456	013570'			.WORD	P.AGI
015460	013640'			.WORD	P.AGJ
015462	013720'			.WORD	P.AGK
015464	013770'			.WORD	P.AGL
015466	014040'			.WORD	P.AGM
015470	014112'			.WORD	P.AGN
015472	014202'			.WORD	P.AGO
015474	014264'			.WORD	P.AGP
015476	014344'			.WORD	P.AGQ
015500	014420'			.WORD	P.AGR
015502	014474'			.WORD	P.AGS
015504	014536'			.WORD	P.AGT
015506	014574'			.WORD	P.AGU
015510	014660'			.WORD	P.AGV
015512	014736'			.WORD	P.AGW
015514	015024'			.WORD	P.AGX
015516	015104'			.WORD	P.AGY
015520	015164'			.WORD	P.AGZ
015522	015246'			.WORD	P.AHA
015524	015336'			.WORD	P.AHB
015526	045	101	040	P.AHD: .ASCII	/#A /
015531	123	125	103	.ASCII	/SUC/
015534	103	105	123	.ASCII	/CES/
015537	123	106	125	.ASCII	/SFU/
015542	114	045	116	.ASCII	/L#N/
015545	000			.ASCII	<00>
015546	045	101	111	P.AHE: .ASCII	/#AI/
015551	116	126	101	.ASCII	/NVA/
015554	114	111	104	.ASCII	/LID/
015557	040	103	117	.ASCII	/CO/
015562	115	115	101	.ASCII	/MMA/
015565	116	104	045	.ASCII	/ND#/

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0088
Page 60
(12)

015570	116	000			.ASCII	/N/<00>
015572	045	101	116	P.AHF:	.ASCII	/AN/
015575	117	040	122		.ASCII	/O R/
015600	105	107	111		.ASCII	/EGI/
015603	117	116	040		.ASCII	/ON /
015606	101	126	101		.ASCII	/AVA/
015611	111	114	101		.ASCII	/ILA/
015614	102	114	105		.ASCII	/BLE/
015617	045	116	000		.ASCII	/N/<00>
015622	045	101	116	P.AHG:	.ASCII	/AN/
015625	117	040	122		.ASCII	/O R/
015630	105	107	111		.ASCII	/EGI/
015633	117	116	040		.ASCII	/ON /
015636	123	125	111		.ASCII	/SUI/
015641	124	101	102		.ASCII	/TAB/
015644	114	105	045		.ASCII	/LE#/
015647	116	000	000		.ASCII	/N/<00><00>
015652	045	101	120	P.AHH:	.ASCII	/AP/
015655	122	117	107		.ASCII	/ROG/
015660	122	101	115		.ASCII	/RAM/
015663	040	116	117		.ASCII	/ NO/
015666	124	040	113		.ASCII	/T K/
015671	116	117	127		.ASCII	/NOW/
015674	116	045	116		.ASCII	/N#/
015677	000				.ASCII	<00>
015700	045	101	114	P.AHI:	.ASCII	/AL/
015703	117	101	104		.ASCII	/OAD/
015706	040	106	101		.ASCII	/ FA/
015711	111	114	125		.ASCII	/ILU/
015714	122	105	045		.ASCII	/RE#/
015717	116	000	000		.ASCII	/N/<00><00>
015722	045	101	123	P.AHJ:	.ASCII	/AS/
015725	124	101	116		.ASCII	/TAN/
015730	104	101	114		.ASCII	/DAL/
015733	117	116	105		.ASCII	/ONE/
015736	045	116	000		.ASCII	/N/<00>
015741	000				.ASCII	<00>
015742	015526'			P.AHC:	.WORD	P.AHD
015744	015546'				.WORD	P.AHE
015746	015572'				.WORD	P.AHF
015750	015622'				.WORD	P.AHG
015752	015652'				.WORD	P.AHH
015754	015700'				.WORD	P.AHI
015756	015722'				.WORD	P.AHJ
015760	045	101	123	P.AHL:	.ASCII	/AS/
015763	125	103	103		.ASCII	/UCC/
015766	105	123	123		.ASCII	/ESS/
015771	045	116	000		.ASCII	/N/<00>
015774	045	101	111	P.AHM:	.ASCII	/AI/
015777	116	126	101		.ASCII	/NVA/
016002	114	111	104		.ASCII	/LID/
016005	040	103	117		.ASCII	/ CO/
016010	115	115	101		.ASCII	/MMA/
016013	116	104	045		.ASCII	/ND#/
016016	116	000			.ASCII	/N/<00>
016020	045	101	103	P.AHN:	.ASCII	/AC/
016023	117	115	115		.ASCII	/OMM/

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0089
Page 61
(12)

016026	101	116	104	.ASCII	/AND/	
016031	040	101	102	.ASCII	/ AB/	
016034	117	122	124	.ASCII	/ORT/	
016037	105	104	045	.ASCII	/ED#/	
016042	116	000		.ASCII	/N/<00>	
016044	045	101	125	P.AHO:	.ASCII	/#AU/
016047	116	111	124	.ASCII	/NIT/	
016052	055	117	106	.ASCII	/-OF/	
016055	106	114	111	.ASCII	/FLI/	
016060	116	105	045	.ASCII	/NE#/	
016063	116	000	000	.ASCII	/N/<00><00>	
016066	045	101	125	P.AHP:	.ASCII	/#AU/
016071	116	111	124	.ASCII	/NIT/	
016074	055	101	126	.ASCII	/-AV/	
016077	101	111	114	.ASCII	/AIL/	
016102	101	102	114	.ASCII	/ABL/	
016105	105	045	116	.ASCII	/E#N/	
016110	000	000		.ASCII	<00><00>	
016112	045	101	115	P.AHQ:	.ASCII	/#AM/
016115	105	104	111	.ASCII	/EDI/	
016120	101	040	106	.ASCII	/A F/	
016123	117	122	115	.ASCII	/ORM/	
016126	101	124	040	.ASCII	/AT /	
016131	105	122	122	.ASCII	/ERR/	
016134	117	122	045	.ASCII	/OR#/	
016137	116	000	000	.ASCII	/N/<00><00>	
016142	045	101	127	P.AHR:	.ASCII	/#AW/
016145	122	111	124	.ASCII	/RIT/	
016150	105	040	120	.ASCII	/E P/	
016153	122	117	124	.ASCII	/ROT/	
016156	105	103	124	.ASCII	/ECT/	
016161	105	104	045	.ASCII	/ED#/	
016164	116	000		.ASCII	/N/<00>	
016166	045	101	103	P.AHS:	.ASCII	/#AC/
016171	117	115	120	.ASCII	/OMP/	
016174	101	122	105	.ASCII	/ARE/	
016177	040	105	122	.ASCII	/ ER/	
016202	122	117	122	.ASCII	/ROR/	
016205	045	116	000	.ASCII	/#N/<00>	
016210	045	101	104	P.AHT:	.ASCII	/#AD/
016213	101	124	101	.ASCII	/ATA/	
016216	040	105	122	.ASCII	/ ER/	
016221	122	117	122	.ASCII	/ROR/	
016224	045	116	000	.ASCII	/#N/<00>	
016227	000			.ASCII	<00>	
016230	045	101	110	P.AHU:	.ASCII	/#AH/
016233	117	123	124	.ASCII	/OST/	
016236	040	102	125	.ASCII	/ BU/	
016241	106	106	105	.ASCII	/FFE/	
016244	122	040	101	.ASCII	/R A/	
016247	103	103	105	.ASCII	/CCE/	
016252	123	123	040	.ASCII	/SS /	
016255	105	122	122	.ASCII	/ERR/	
016260	117	122	045	.ASCII	/OR#/	
016263	116	000	000	.ASCII	/N/<00><00>	
016266	045	101	103	P.AHV:	.ASCII	/#AC/
016271	117	116	124	.ASCII	/ONT/	

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0090
Page 62
(12)

016274	122	117	114	.ASCII	/ROL/
016277	114	105	122	.ASCII	/LER/
016302	040	105	122	.ASCII	/ER/
016305	122	117	122	.ASCII	/ROR/
016310	045	116	000	.ASCII	/#N/<00>
016313	000			.ASCII	<00>
016314	045	101	104	P.AHW: .ASCII	/#AD/
016317	122	111	126	.ASCII	/RIV/
016322	105	040	105	.ASCII	/E E/
016325	122	122	117	.ASCII	/RRO/
016330	122	045	116	.ASCII	/R#N/
016333	000			.ASCII	<00>
016334	045	101	115	P.AHX: .ASCII	/#AM/
016337	105	123	123	.ASCII	/ESS/
016342	101	107	105	.ASCII	/AGE/
016345	040	106	122	.ASCII	/FR/
016350	117	115	040	.ASCII	/OM /
016353	101	116	040	.ASCII	/AN /
016356	111	116	124	.ASCII	/INT/
016361	105	122	116	.ASCII	/ERN/
016364	101	114	040	.ASCII	/AL /
016367	104	111	101	.ASCII	/DIA/
016372	107	116	117	.ASCII	/GNO/
016375	123	124	111	.ASCII	/STI/
016400	103	045	116	.ASCII	/C#N/
016403	000			.ASCII	<00>
016404	015760			P.AHK: .WORD	P.AHL
016406	015774			.WORD	P.AHM
016410	016020			.WORD	P.AHN
016412	016044			.WORD	P.AHO
016414	016066			.WORD	P.AHP
016416	016112			.WORD	P.AHQ
016420	016142			.WORD	P.AHR
016422	016166			.WORD	P.AHS
016424	016210			.WORD	P.AHT
016426	016230			.WORD	P.AHU
016430	016266			.WORD	P.AHV
016432	016314			.WORD	P.AHW
016434	016334			.WORD	P.AHX

000000	.PSECT	\$GLOB\$,	RO	.	D	.	GBL
000000	RT::	.BLKW	5				
000012	RT.TABLE::	.BLKW	1				
000014	HWP.TABLE::	.BLKW	1				
000016	XMT.DATA.BUF::	.BLKW	400				
001016	RCV.DATA.BUF::	.BLKW	400				
002016	CLK.ADR::	.BLKW	1				
002020	CLK.TYPE::	.BLKW	1				
002022	CLK.CSR::	.BLKW	1				

ZRCFR1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC1.B16;1

SEQ 0091
Page 63
(12)

002024		CLK.HERTZ::	.BLKW	1
002026		CLK.START::	.BLKW	1
002030		UNIT::	.BLKW	1
002032		LOG.UNIT::	.BLKW	1
002034		OUT.BOUND::	.BLKW	1
002036		IN.BOUND::	.BLKW	1
002040		VEC.AD::	.BLKB	1
002042		RC25.ADDR::	EVEN	
002044		RC25.DATA::	.BLKW	1
002050		COM.AREA::	.BLKW	2
002260		HEAD.AREA::	.BLKW	104
002262		RECEIVE.RING::	.BLKW	1
002264		SEND.RING::	.BLKW	1
002266		REC.ENVELOPE::	.BLKW	1
004266		SND.ENVELOPE::	.BLKW	1000
005566		BUF.DESCRPTR::	.BLKW	540
005570		CMD.REF::	.BLKW	1
005572		BYTE.COUNT::	.BLKW	1
005574	000000	TICKS::	.WORD	0
005576	000000	SECONDS::	.WORD	0
005600	000000	MINUTES::	.WORD	0
005602		TIP::	.BLKW	1
005604		DATA1::	.BLKW	1
005606		DATA2::	.BLKW	1
005610		DATA3::	.BLKW	1
005612		DATA4::	.BLKW	1
005614	000000	I.AM.NEX::	.WORD	0
005616		MSGADR::	.BLKW	1
005620	143325	END.LBN::	.WORD	-34453
005622		P.MASK::	.BLKB	1
005623		B.MASK::	.BLKB	1
005624		MANU.SW::	.BLKW	1
005626		SWITCH2::	.BLKW	1

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0092
Page 64
(12)

005630		RET.UNIT.FLAG::	
		.BLKW	1
005632		P1::	.BLKW 1
005634		P2::	.BLKW 1
005636		P3::	.BLKW 1
005640		P4::	.BLKW 1
005642		P5::	.BLKW 1
005644		P6::	.BLKW 1
005646		RET.STATUS::	
		.BLKW	1
005650	000000	ER.STATUS::	
		.WORD	0
005652		CANCEL.TIMER::	
		.BLKW	1
005654		CMD.SLOT::	
		.BLKW	1
005656		RES.SLOT::	
		.BLKW	1
005660		LBN::	.BLKW 1
005662		LBN.ST::	.BLKW 1
005664		LBN.ED::	.BLKW 1
005666		LBN.SZ::	.BLKW 1
005670		FREE.MEM.ADDR::	
		.BLKW	1
005672		MEM.SIZE::	
		.BLKW	1
005674		H.SADD::	.BLKW 1
005676		H.EADD::	.BLKW 1
005700		BUF.LENGTH::	
		.BLKW	1
005702	000000	CMOD::	.WORD 0
005704		NUM.RETRIES::	
		.BLKW	1
005706	000000	RETRIES::	
		.WORD	0
005710	000001	FAL.CODE::	
		.WORD	1
005712		DMC.TEST::	
		.BLKW	1
005714		BYT.CNT::	
		.BLKW	1
005716		DM.REC::	.BLKW 1
005720		DM.XMT::	.BLKW 1
005722	000037	SIZ.LBN::	
		.WORD	37
005724	000000	OFFSET::	.WORD 0
005726		PASSO::	.BLKW 1
005730		TEMP::	.BLKW 1

```
.GLOBL L$SOFT, T$PTHV, L$RPT, L$INIT
.GLOBL L$CLEAN, L$LAST, L$HARD, L$DVTYP
.GLOBL L$DESC, L$DU, L$AU, L$AUTO, T1
.GLOBL T2, T3, T4, T5, T6, T7, T8, T9
.GLOBL T10, T11, T12, T13, T14, T15, T16
.GLOBL T17, T18, T19, T20, T21, T22, T23
.GLOBL T24, T25, T26, T27, T28, T29
```

000216'	L\$ERRTBL==	ERRTYP
000244'	L\$SW==	L\$SWLEN*2
000230'	L\$HW==	L\$HWLEN*2
000011'	L\$DEPO==	L\$REV*1
000230'	DFPTBL==	L\$HWLEN*2
000244'	SFPTBL==	L\$SWLEN*2
002060'	RINGBASE==	COM.AREA*10
000002'	TIME==	P.AAA
000006'	FRU==	P.AAB
000036'	ADAPTO==	P.AAC
000070'	CONTRO==	P.AAD
000124'	DRIVE.==	P.AAE
000154'	MECHAN==	P.AAF
000206'	QST1==	P.AAG
000222'	QST2==	P.AAH
000232'	QST3==	P.AAI
000244'	QST4==	P.AAJ
000270'	QST6==	P.AAK
000342'	QST7==	P.AAL
000422'	QST8==	P.AAM
000442'	QST9==	P.AAN
000460'	QST10==	P.AAO
000540'	QS10.1==	P.AAP
000570'	QS10.2==	P.AAQ
000650'	QST11==	P.AAR
000724'	QST14==	P.AAS
000776'	QST15==	P.AAT
001050'	DBM1==	P.AAU
001150'	DBM7==	P.AAV
001214'	DBM8==	P.AAW
001276'	DBM9==	P.AAX
001360'	DBM10==	P.AAY
001422'	DBM11==	P.AAZ
001470'	DBM12==	P.ABA
001530'	DBM13==	P.ABB
001564'	DBM14==	P.ABC
001620'	DBM15==	P.ABD
001662'	DBM16==	P.ABE
001726'	DBM17==	P.ABF
001776'	DBM18==	P.ABG
002046'	DBM19==	P.ABH
002106'	DBM20==	P.ABI
002156'	DBM21==	P.ABJ
002226'	DBM22==	P.ABK
002274'	DBM23==	P.ABL
002356'	DBM24==	P.ABM
002422'	DBM25==	P.ABN
002460'	DBM26==	P.ABO
002524'	DBM27==	P.ABP
002564'	DBM28==	P.ABQ
002634'	DBM29==	P.ABR
002674'	DBM30==	P.ABS
002732'	DBM31==	P.ABT
002774'	DBM32==	P.ABU
003030'	DBM36==	P.ABV

003072'	DBM37==	P.ABW
003136'	DBM38==	P.ABX
003176'	DBM39==	P.ABY
003252'	MSG.01==	P.ABZ
003304'	ERR.01==	P.ACA
003330'	ERR.02==	P.ACB
003376'	ERR.03==	P.ACC
003444'	FMT#C==	P.ACD
003452'	FMT1==	P.ACE
003536'	FMT2==	P.ACF
003616'	FMT3==	P.ACG
003712'	FMT4==	P.ACH
003750'	FMT5==	P.ACI
004030'	FMT6==	P.ACJ
004110'	FMT7==	P.ACK
004154'	FMT7A==	P.ACL
004220'	FMT8==	P.ACM
004310'	FMT9==	P.ACN
004412'	FMT10==	P.ACO
004464'	FMT11==	P.ACP
004540'	FMT12==	P.ACQ
004614'	FMT13==	P.ACR
004652'	FMT14==	P.ACS
004716'	FMT15==	P.ACT
004776'	FMT16==	P.ACU
005044'	FMT17==	P.ACV
005120'	FMT18==	P.ACW
005210'	FMT19==	P.ACX
005300'	FMT20==	P.ACY
005332'	FMT#A==	P.ACZ
005374'	MSG.PWR==	P.ADA
005430'	MSG.1==	P.ADB
005460'	MSG.2==	P.ADC
005510'	MSG.7==	P.ADD
005562'	MSG.8==	P.ADE
005624'	MSG.9==	P.ADF
005662'	MSG.10==	P.ADG
005730'	MSG.11==	P.ADH
005766'	MSG.13==	P.ADI
006010'	MSG.14==	P.ADJ
006036'	BUFF.ERR==	P.ADK
006114'	DMC.ERR==	P.ADL
006152'	INI.MSG==	P.ADM
006234'	END.MSG==	P.ADN
006300'	BRERR==	P.ADO
006356'	MSG.SEEK.ERR==	P.ADP
006400'	MSG.HSWICH.ERR==	P.ADQ
006424'	MSG.SAC.ERR==	P.ADR
006450'	MSG.COM.WPT==	P.ADS
006504'	SK.FOR.ERR==	P.ADT
006530'	SK.REV.ERR==	P.ADU
006554'	SK.TOG.ERR==	P.ADV
006576'	SK.RAN.ERR==	P.ADW
006620'	MSG.WRITE.ERR==	P.ADX
006652'	MSG.READ.ERR==	P.ADY
006674'	MSG.GUS.ERR==	P.ADZ
006734'	AVAIL.ERR==	P.AEA

ZRCFB1
V03.0

CZRCFC0 RC25 FR END TEST
GLOBAL TEXT SECTION

27-Mar-1985 15:21:49
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC1.B16;1

SEQ 0095
Page 67
(12)

006766'	MSG.AVE.TIME==	P.AEB
007056'	MES.SKO.TIME==	P.AEC
007146'	MSG.PRO.TIME==	P.AED
007214'	MSG.SK.TIME==	P.AEE
007266'	MG.SKF.TIME==	P.AEF
007340'	MSG.ROT.TIME==	P.AEG
007416'	AZT.READY.ERR==	P.AEH
007456'	EXE.SUP.ERR==	P.AEI
007516'	SND.DATA.ERR==	P.AEJ
007554'	RE.DATA.ERR==	P.AEK
011576'	PFE.STRUCT==	P.AEL
012122'	EMSG.STRUCT==	P.AFJ
015410'	RC.STRUCTURE==	P.AFO
015742'	SDUP.STRUCT==	P.AHC
016404'	SMSCP.STRUCT==	P.AHK

PSECT SUMMARY

:						
:						
:	Psect Name	Words	Attributes			
:	AA\$CODE	94	RO , I ,	LCL, REL, CON		
:	\$GLOB\$	1517	RO , D ,	GBL, REL, CON		
:	\$PLIT\$	3727	RO , D ,	GBL, REL, CON		

Library Statistics

:						
:						
:						
:	File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
:	USER#1:[AZTEC.CZRCFC]AZTECO.L16;2	485	154	31	24	00:00.2

COMMAND QUALIFIERS

```

;
; BLISS/PDP11/LIST ZRCFC1.B16/EN:NOEIS
; Size: 0 code + 5338 data words
; Run Time: 01:33.8
; Elapsed Time: 01:39.6
; Lines/CPU Min: 1248
; Lexemes/CPU-Min: 10517
; Memory Used: 299 pages
; Compilation Complete

```

F8

ZRCFB2

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0096
Page 1
(1)

: 0001 0 MODULE ZRCFB2 (

ZRCFR2

CZRCFC0 RC25 FR END TEST

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1SEQ 0097
Page 2
(2)

```

: 0002 0  *TITLE 'CZRCFC0 RC25 FR END TEST'
: 0003 0      IDENT = 'V03.0',
: 0004 0      OPTLEVEL = 0,
: 0005 0      ADDRESSING_MODE (RELATIVE)
: 0006 0      ) =
: 0007 1  BEGIN
: 0008 1  !
: 0009 1  !<BLF/LOWERCASE_KEY>
: 0010 1  !
: 0011 1
: 0012 1  library 'AZTECO';
: 0013 1
: 0014 1  require 'BLSMAC.REQ';
: 1503 1
: 1504 1  !
: 1505 1  !
: 1506 1
: 1507 1  structure
: 1508 1      RC25 [0, P, S, E] =
: 1509 2      begin
: 1510 2
: 1511 2      local
: 1512 2      RC_REG;
: 1513 2
: 1514 2      RC_REG = .(RC25 + %upval*0)<0, %bpval, 0>;
: 1515 2      RC_REG
: 1516 2      end
: 1517 1      <P, S, E>;
: 1518 1
: 1519 1  psect
: 1520 1      code = AA$CODE;
: 1521 1
: 1522 1  forward routine
: 1523 1      FIND_CLOCK : novalue,
: 1524 1      CLOCK_INIT : novalue,
: 1525 1      RC25$ERR_RPT : novalue,
: 1526 1      AZT_INIT,
: 1527 1      AZP_INIT,
: 1528 1      PRT$FRU_CALLOUT : novalue,
: 1529 1      INIT_COM_AREA,
: 1530 1      NXMI : L$ISR novalue,
: 1531 1      CLK_INT_SERV : L$ISR novalue,
: 1532 1      REC_STATUS,
: 1533 1      SET_CNTRLR_CHAR,
: 1534 1      AVAILABLE,
: 1535 1      ON_LINE,
: 1536 1      READ_CMD,
: 1537 1      READ_FILL_CMD : novalue,
: 1538 1      GET_UNIT_STATUS,
: 1539 1      RANDOM_NUM : novalue,
: 1540 1      GET_CMD_SLOT,
: 1541 1      GET_RES_SLOT : novalue,
: 1542 1      EXAM_DATA,
: 1543 1      AZTEC_READY,
: 1544 1      DO_RETRIES : novalue,
: 1545 1      decode : novalue;
: 1546 1

```

```

! DEFINE ACCESS ALGORITHM
! TO ALLOW FIELD REFERENCE

! TO THE AZTEC

```

```

: 1547 1 external
: 1548 1 ADAPTO,
: 1549 1 CONTRO,
: 1550 1 COM_AREA : blockvector [REC_ALLOCATE + SND_ALLOCATE + HDR_SIZ, 2, word],
: 1551 1 HEAD_AREA : ref block [4, word] field (HDR_FIELD),
: 1552 1 RECEIVE_RING : ref blockvector [REC_ALLOCATE, 2, word] field (DSC_FIELD),
: 1553 1 SEND_RING : ref blockvector [SND_ALLOCATE, 2, word] field (DSC_FIELD),
: 1554 1 REC_ENVELOPE : blockvector [REC_ALLOCATE, RB_SIZE + 2, word] field (ENV_FIELD),
: 1555 1 SND_ENVELOPE : blockvector [SND_ALLOCATE, SB_SIZE + 2, word] field (ENV_FIELD),
: 1556 1 BUF_DESCRPTR : word volatile, ! BUFFER DESCRIPTOR AREA
: 1557 1 BYTE_COUNT : word volatile, ! BYTE COUNT BUFFER
: 1558 1 CLK_ADR : word, ! LOCATION TO RETURN CLOCK ADDRESS
: 1559 1 CLK_TYPE : word, ! TYPE OF CLOCK ON SYSTEM
: 1560 1 ! (0=NO CLOCK, -1= L-CLOCK, 1=P-CLOCK)
: 1561 1 CLK_CSR : word, ! STORE CSR ADDRESS FOR CLOCK HERE
: 1562 1 CLK_HERTZ : word, ! CLOCK RATE
: 1563 1 CLK_START : word, ! STORE CLOCK START VALUE
: 1564 1 TICKS : word volatile, ! STORE NUMBERS OF CLOCK INT. OCCURED
: 1565 1 SECONDS : word volatile, ! STORE SECONDS
: 1566 1 MINUTES : word volatile, ! STORE MINUTES
: 1567 1 MSGADR : word volatile, ! STORE MESSAGE ADDRESS
: 1568 1 DATA1 : word, ! STEP 1 WRITE DATA TO AZTEC_INIT
: 1569 1 DATA2 : word volatile, ! STEP 2 WRITE DATA TO AZTEC_INIT
: 1570 1 DATA3 : word volatile, ! STEP 3 WRITE DATA TO AZTEC_INIT
: 1571 1 DATA4 : word volatile, ! STEP 4 WRITE DATA TO AZTEC_INIT
: 1572 1 B_MASK : byte volatile, ! MASK FOR WITCH STEP TO DO
: 1573 1 ! IN AZTEC_INIT.
: 1574 1 LBN : word volatile, ! LOGICAL BLOCK NUMBER BUFFER
: 1575 1 LBN_ST : word volatile, ! START LOGICAL BLOCK NUMBER
: 1576 1 LBN_ED : word volatile, ! ENDING LOGICAL BLOCK NUMBER
: 1577 1 CMD_REF : word volatile, ! COMMAND REFERENCE
: 1578 1 RES_SLOT : word volatile, ! RECEIVING RING SLOT
: 1579 1 CMD_SLOT : word volatile, ! SENDING RING SLOT
: 1580 1 VEC_AD : byte, ! INIT INTERRUPT VECTOR
: 1581 1 RET_STATUS : word volatile, ! RETURN STATUS
: 1582 1 ER_STATUS : word, ! SAVES ERROR CODE
: 1583 1 TEMP : word volatile,
: 1584 1 PASS0 : word, ! FLAG FOR FIRST PASS
: 1585 1 CMOD : word, ! COMMAND MODIFIER
: 1586 1 IN_BOUND : word,
: 1587 1 FREE_MEM_ADDR, ! STARING FREE MEMORY ADDRESS
: 1588 1 MEM_SIZE, ! FREE MEMORY SIZE
: 1589 1 RINGBASE,
: 1590 1 DRIVE_,
: 1591 1 DBM1,
: 1592 1 ERR_01,
: 1593 1 ERR_02,
: 1594 1 ERR_03,
: 1595 1 FMT3C,
: 1596 1 FRU,
: 1597 1 FMT2,
: 1598 1 FMT3,
: 1599 1 FMT13,
: 1600 1 FMT14,
: 1601 1 FMT15,
: 1602 1 DMC_TEST,
: 1603 1 BYT_CNT,

```

```

: 1604 1 DM_XMT,
: 1605 1 DM_REC,
: 1606 1 H_SADD,
: 1607 1 H_EADD,
: 1608 1 BUF_LENGTH,
: 1609 1 MANU_SW,
: 1610 1 SWITCH2,
: 1611 1 TIP,
: 1612 1 SWP_CONTINUE,
: 1613 1 FMT$A,
: 1614 1 QST15,
: 1615 1 QST14,
: 1616 1 ! RUN TIME TABLE STORAGE
: 1617 1 HWP_TABLE : ref block [WORD2_IN_HWP_TAB, word] field (HWP_FIELDS),
: 1618 1 RT_TABLE : ref block [WORD1_IN_RT_TAB, word] field (RT_FIELDS),
: 1619 1 RT : vector [WORD1_IN_RT_TAB, word],
: 1620 1 I AM NEX : word volatile,
: 1621 1 CANCEL_TIMER : word volatile,
: 1622 1 RETRIES,
: 1623 1 SWP_RETRIES,
: 1624 1 NUM_RETRIES,
: 1625 1 SWP_TRACE,
: 1626 1 SWP_START,
: 1627 1 SWP_END,
: 1628 1 SWP_TOP,
: 1629 1 SWP_LIMIT,
: 1630 1 L$UNIT,
: 1631 1 MECHAN,
: 1632 1 MSG_PWR,
: 1633 1 MSG_14,
: 1634 1 FAL_CODE,
: 1635 1 END_LBN : word volatile,
: 1636 1 P_MASK : byte volatile,
: 1637 1 RET_UNIT_FLAG : word,
: 1638 1 P1 : word volatile,
: 1639 1 P2 : word volatile,
: 1640 1 P3 : word volatile,
: 1641 1 P4 : word volatile,
: 1642 1 P5 : word volatile,
: 1643 1 P6 : word volatile,
: 1644 1 QST1,
: 1645 1 QST2,
: 1646 1 QST3,
: 1647 1 QST4,
: 1648 1 QST6,
: 1649 1 QST7,
: 1650 1 QST8,
: 1651 1 QST9,
: 1652 1 QST10,
: 1653 1 QS10_1,
: 1654 1 QS10_2,
: 1655 1 QST11,
: 1656 1 RC25_ADDR : ref RC25 field (RC_REG),
: 1657 1 RC25_DATA : block [2, word] field (RC_REG),
: 1658 1 EMSG_STRUCT : vector [4],
: 1659 1 PFE_STRUCT : vector [23],
: 1660 1 RC_STRUCTURE : vector [39],

```

ZRCFB2
V03.0

CZRCFC0 RC25 FR END TEST

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

```
: 1661 1 SDUP_STRUCT : vector [7],  
: 1662 1 SMSCP_STRUCT : vector [13],  
: 1663 1 XMT_DATA_BUF : vector [256, word],  
: 1664 1 RCV_DATA_BUF : vector [256, word],  
: 1665 1 UNIT : word,  
: 1666 1 OFFSET : word,  
: 1667 1 LOG_UNIT : word;  
: 1668 1
```

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19VAX-11 B1116-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1SEQ 0101
Page 6
(3)

```

: 1669 1 *title 'MISCELLANEOUS SECTIONS'
: 1670 1 *sbttl 'TYPE AND DESCRIPTION'
: 1671 1 !: NAMES OF DEVICES SUPPORTED BY PROGRAM
: 1672 1 DEVTYP (*esciz'AZTEC RC25 PLATTER');
: 1673 1 !: TEST DESCRIPTION
: C 1674 1 DESCRIPT (*esciz'RC25 FRONT END/HOST DIAGNOSTIC');*(
: C 1675 1 ;**
: C 1676 1 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
: C 1677 1 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: C 1678 1 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: C 1679 1 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: C 1680 1 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: C 1681 1 ; WITH THE OPERATOR.
: C 1682 1 ;-
: 1683 1 )*
: 1684 1 BGNHRD;
: 1685 1 GPRMA (QST1, %'0', 0, %'00000', %'177777', YES, 1); !IP ADDRESS?
: 1686 1 GPRMA (QST2, %'2', 0, %'4', %'774', YES, 1); !VECTOR?
: 1687 1 GPRMD (QST3, %'4', 0, %'177777', %'4', %'7', YES, 1); !BR LEVEL
: 1688 1 GPRMD (QST4, %'6', D, %'377', %'0', %decimal'253', NO, 1); !UNIT NUMBER(S)
: 1689 1 ENDHRD;

```


ZRCFR2
V03.0MISCELLANEOUS SECTIONS
SOFTWARE PARAMETER CODING SECTION27-Mar-1985 15:23:34
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC2.B16;1SEQ 0102
Page 7
(4)

```

: 1690 1  %bttl 'SOFTWARE PARAMETER CODING SECTION'
: C 1691 1  %(
: C 1692 1  ;**
: C 1693 1  ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
: C 1694 1  ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: C 1695 1  ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: C 1696 1  ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: C 1697 1  ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: C 1698 1  ; WITH THE OPERATOR.
: C 1699 1  ;--
: 1700 1  )%
: 1701 1  BGNSFT;
: 1702 1  GPRML (QST6, %'0', %'177777', YES, 1);          !USE TOP SURFACE FOR SINGLE SURFACE TESTS?
: 1703 1  XFERF (BOTTOM);          !IF NO GO TO LABEL BOTTOM
: 1704 1  GPRML (QST7, %'2', %'177777', YES, 1);          !DO YOU WISH TO LIMIT THE AREA TESTED
: 1705 1  !IN TESTS #15 - #18?
: 1706 1  XFERF (MANINT);          !IF NO, SKIP NEXT TWO QUESTIONS
: 1707 1  GPRMD (QST8, %'4', D, %'177777', %decimal'0', %decimal'820', NO, 1); !STARTING TRACK?
: 1708 1  GPRMD (QST9, %'6', D, %'177777', %decimal'0', %decimal'820', NO, 1) !ENDING TRACK?
: 1709 1  XFER(LAST);
: 1710 1  $L (BOTTOM);
: 1711 1  GPRML (QST7, %'2', %'177777', YES, 1);          !DO YOU WISH TO LIMIT THE AREA TESTED
: 1712 1  !IN TESTS #15 - #18?
: 1713 1  XFERF (VLAST);          !IF NO GO TO LABEL VLAST
: 1714 1  GPRMD (QST8, %'4', D, %'177777', %decimal'821', %decimal'1641', NO, 1); !STARTING TRACK?
: 1715 1  GPRMD (QST9, %'6', D, %'177777', %decimal'821', %decimal'1641', NO, 1); !ENDING TRACK?
: 1716 1  $L (MANINT);          !LABEL THIS QUESTION
: 1717 1  $L (LAST);
: 1718 1  $L (VLAST);
: 1719 1  GPRMD (QST11, %'10', D, %'177777', %'0', %'177777', YES, 1);          !NUMBER OF RETRIES FOR TEST
: 1720 1  GPRML (QST10_2, %'12', 1, YES, 1);          !DO YOU WISH TO CONTINUE TESTING?
: 1721 1  GPRML (QST10, %'14', 1, YES, 1);          !DO YOU WANT TO DO THE MANUAL
: 1722 1  !INTERVENSON TEST?
: 1723 1  GPRML (QST10_1, %'16', 1, YES, 1);          !DO YOU NEED TRACE MODE?
: 1724 1  ENDSFT;

```


000136	000130				
000140	000000G	GP#5::	.WORD	<<<L#NDSFT-L#SFTLN>/2>-1>	
000142	177777		.WORD	130	
000144	000000C		.WORD	QST6	
000146	001130	\$BOTTOM:	.WORD	-1	
000150	000000G	GP#6::	.WORD	<<<<\$LBOTTOM-\$BOTTOM>*400>.4>.40>	
000152	177777		.WORD	1130	
000154	000000C		.WORD	QST7	
000156	002042	\$MANINT:	.WORD	-1	
000160	000000G	GP#7::	.WORD	<<<<\$LMANINT-\$MANINT>*400>.4>.40>	
000162	177777		.WORD	2042	
000164	000000		.WORD	QST8	
000166	001464		.WORD	-1	
000170	003042	GP#8::	.WORD	0	
000172	000000G		.WORD	1464	
000174	177777		.WORD	3042	
000176	000000		.WORD	QST9	
000200	001464		.WORD	-1	
000202	000000C	\$LAST:	.WORD	0	
000204	001004	\$LBOTTOM:	.WORD	1464	
			.WORD	<<<\$LLAST-\$LAST>*400>.4>	
000206	001130	GP#9::	.WORD	1004	
000210	000000G		.WORD	1130	
000212	177777		.WORD	QST7	
000214	000000C		.WORD	-1	
000216	002042	\$VLAST:	.WORD	<<<<\$LVLAST-\$VLAST>*400>.4>.40>	
000220	000000G	GP#10::	.WORD	2042	
000222	177777		.WORD	QST8	
000224	001465		.WORD	-1	
000226	003151		.WORD	1465	
000230	003042	GP#11::	.WORD	3151	
000232	000000G		.WORD	3042	
000234	177777		.WORD	QST9	
000236	001465		.WORD	-1	
000240	003151		.WORD	1465	
000242	001004		.WORD	3151	
		\$LMANINT:	.WORD	1004	
000244	001004	\$LLAST:	.WORD	1004	
000246	001004	\$LVLAST:	.WORD	1004	
000250	004052	GP#12::	.WORD	4052	
000252	000000G		.WORD	QST11	
000254	177777		.WORD	-1	
000256	000000		.WORD	0	
000260	177777		.WORD	-1	
000262	005130	GP#13::	.WORD	5130	
000264	000000G		.WORD	QS10.2	
000266	000001		.WORD	1	
000270	006130	GP#14::	.WORD	6130	
000272	000000G		.WORD	QST10	
000274	000001		.WORD	1	
000276	007130	GP#15::	.WORD	7130	
000300	000000G		.WORD	QS10.1	
000302	000001		.WORD	1	
000304		L#NDSFT::	.WORD	1	
			.BLKW	1	

```

.GLOBAL ADAPTO, CONTRO, COM.AREA, HEAD.AREA
.GLOBAL RECEIVE.RING, SEND.RING, REC.ENVELOPE
.GLOBAL SND.ENVELOPE, BUF.DESCRPTR, BYTE.COUNT
.GLOBAL CLK.ADR, CLK.TYPE, CLK.CSR, CLK.HERTZ
.GLOBAL CLK.START, TICKS, SECONDS, MINUTES
.GLOBAL MSGADR, DATA1, DATA2, DATA3, DATA4
.GLOBAL B.MASK, LBN, LBN.ST, LBN.ED, CMD.REF
.GLOBAL RES.SLOT, CMD.SLOT, VEC.AD, RET.STATUS
.GLOBAL ER.STATUS, TEMP, PASSO, CMOD, IN.BOUND
.GLOBAL FREE.MEM.ADDR, MEM.SIZE, RINGBASE
.GLOBAL DRIVE., DBM1, ERR.01, ERR.02, ERR.03
.GLOBAL FMT#C, FRU, FMT2, FMT3, FMT13
.GLOBAL FMT14, FMT15, DMC.TEST, BYT.CNT
.GLOBAL DM.XMT, DM.REC, H.SADD, H.EADD
.GLOBAL BUF.LENGTH, MANU.SW, SWITCH2, TIP
.GLOBAL SWP.CONTINUE, FMT#A, QST15, QST14
.GLOBAL HWP.TABLE, RT.TABLE, RT, I.AM.NEX
.GLOBAL CANCEL.TIMER, RETRIES, SWP.RETRIES
.GLOBAL NUM.RETRIES, SWP.TRACE, SWP.START
.GLOBAL SWP.END, SWP.TOP, SWP.LIMIT, L#UNIT
.GLOBAL MECHAN, MSG.PWR, MSG.14, FAL.CODE
.GLOBAL END.LBN, P.MASK, RET.UNIT.FLAG
.GLOBAL P1, P2, P3, P4, P5, P6, QST1, QST2
.GLOBAL QST3, QST4, QST6, QST7, QST8, QST9
.GLOBAL QST10, QST10.1, QST10.2, QST11, RC25.ADDR
.GLOBAL RC25.DATA, EMSG.STRUCT, PFE.STRUCT
.GLOBAL RC.STRUCTURE, SDUP.STRUCT, SMSCP.STRUCT
.GLOBAL XMT.DATA.BUF, RCV.DATA.BUF, UNIT
.GLOBAL OFFSET, LOG.UNIT

```

000066'
000136'

L#HARD==
L#SOFT==

L#HRDLN-2
L#SFTLN-2

```

000000 000207          LRPT:  .SBTTL  LRPT REPORT CODING SECTION          1724
                        RTS      PC
; Routine Size: 1 word,      Routine Base: AA#CODE * 0306
; Maximum stack depth per invocation: 0 words

```

```

000000 004767 177772    L#RPT:: .SBTTL  L#RPT REPORT CODING SECTION          1731
000004 104425          JSR      PC,LRPT
000006 000207          TRAP    25
                        RTS      PC
; Routine Size: 4 words,      Routine Base: AA#CODE * 0310
; Maximum stack depth per invocation: 2 words

```

```

: 1733 1 #sbttl 'INITIALIZE SECTION'
: 1734 1 !*
: 1735 1 ! THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
: 1736 1 ! AT THE BEGINNING OF EACH PASS.
: 1737 1 !-
: 1738 2 BGNINIT;
: 1739 2
: 1740 2 local
: 1741 2 DELAY_MULT; !CONTAINS DELAY FACTOR
: 1742 2
: 1743 2 SETPRI (PRI00); !PRIORITY 0
: 1744 2
: 1745 2 if READEF (EF_PWR) !ARE WE HERE BECAUSE OF POWER FAIL?
: 1746 2 then
: 1747 3 begin
: 1748 3 PRINTF (MSG_PWR); !"POWER DELAY - WAITING"
: 1749 3
: 1750 3 incru COUNT from 0 to 60 do ! WAIT APPROX. 60 SECONDS
: 1751 4 begin
: 1752 4 DELAY_MULT = 10000;
: 1753 4 DELAY (.DELAY_MULT);
: 1754 4 BREAK; ! BREAK FOR ACT
: 1755 3 end;
: 1756 3
: 1757 3 DOCLN;
: 1758 2 end;
: 1759 2
: 1760 2 ! *
: 1761 2 ! MAKE SURE NOT MORE THAN 16 UNITS (PLATTERS) HAVE BEEN SPECIFIED.
: 1762 2 ! IF THERE ARE TOO MANY, NOTIFY USER AND RETURN TO SUPERVISOR.
: 1763 2 ! -
: 1764 2
: 1765 2 if .L$UNIT gequ 16
: 1766 2 then !MORE THAN 16 UNITS?
: 1767 3 begin
: 1768 3 PRINTF (ERR_01); !ERROR - TOO MANY UNITS
: 1769 3 DOCLN; !RETURN TO SUPERVISOR AND CLEAN UP
: 1770 2 end;
: 1771 2
: 1772 2 if READEF (EF_CONTINUE) then return; !IF CONTINUE GETS YOU HERE SKIP INIT.
: 1773 2
: 1774 2 PASSO = READEF (EF_START); ! SAVE START FLAG
: 1775 2
: 1776 2 if .PASSO or READEF (EF_RESTART) or READEF (EF_NEW)
: 1777 2 then
: 1778 3 begin
: 1779 3 LOG_UNIT = -1;
: 1780 3 NUM_RETRIES = ZERO;
: 1781 3 RETRIES = FALSE;
: 1782 3 FIND_CLOCK ();
: 1783 3
: 1784 3 if CLK_TYPE eqlu NO_CLOCK ! IF THERE IS NO CLOCK
: 1785 3 then ! IN THE SYSTEM TELL THE
: 1786 4 begin ! OPERATOR
: 1787 4 PRINTF (ERR_02);
: 1788 4 DOCLN;
: 1789 4 end

```

```

: 1790 3      else
: 1791 3        .CLK_CSR = ZERO;           ! STOP THE CLOCK
: 1792 3
: 1793 3      !+
: 1794 3        DETERMINE THE FREE MEMORY STARTING ADDRESS AND IT SIZE
: 1795 3      !-
: 1796 3      MEMORY (FREE_MEM_ADDR);      !FIND THE STARTING ADDR
: 1797 3      MEM_SIZE = ..FREE_MEM_ADDR;  !DETERMINE THE SIZE
: 1798 3      !+
: 1799 3      !-
: 1800 2      end;
: 1801 2
: 1802 2      do                          !OTHERWISE, INCREMENT LOGICAL UNIT
: 1803 3      begin                       !AND CHECK FOR HIGH LIMIT.
: 1804 3      LOG_UNIT = .LOG_UNIT + 1;
: 1805 3
: 1806 3      if .LOG_UNIT gequ .L$UNIT then DOCLN;      !IF SO QUIT INIT AND DO CLEANUP.
: 1807 3
: 1808 3      end
: 1809 2      until (GPHARD (.LOG_UNIT, HWP_TABLE)) nequ 0;  !GET HARDWARE P_TABLE POINTER
: 1810 2
: 1811 2      RT_TABLE = RT [0];           !AND LOAD RT TABLE WITH THE
: 1812 2      RT_TABLE [RT_IP_ADDRESS] = .HWP_TABLE [HWP_IP_ADDRESS]; !HARDWARE P_TABLE INFO.
: 1813 2      RT_TABLE [RT_VECTOR] = .HWP_TABLE [HWP_VECTOR];
: 1814 2      RT_TABLE [RT_BR_LEVEL] = .HWP_TABLE [HWP_BR_LEVEL];
: 1815 2      RT_TABLE [RT_UNIT 1] = .HWP_TABLE [HWP_UNIT_NUMBER];      !PLATTER #
: 1816 2      RC25_ADDR = .RT_TABLE [RT_IP_ADDRESS];      !IP ADDRESS FOR THE CONTROLLER
: 1817 2      UNIT = .RT_TABLE [RT_UNIT 1];              !AND PLATTER # UNDER TEST
: 1818 2      SETVEC (.RT_TABLE [RT_VECTOR], NXMI, PRI07); !SET UP INTERRUPT ROUTINE
: 1819 2      PRINTB (DBM1, .LOG_UNIT, .RC25_ADDR, .UNIT); !GIVE THIS INFO TO OPERATOR.
: 1820 2
: 1821 2      if .SWP_TOP
: 1822 2      then
: 1823 2      OFFSET = 0                      ! SELECT OFFSET BASED IN SURFACE
: 1824 2      else
: 1825 2      OFFSET = 821;                  ! SELECTED.
: 1826 2
: 1827 2      if not .SWP_LIMIT
: 1828 2      then
: 1829 2      !
: 1830 2      ! IF LIMITS NOT PROVIDED THEN
: 1831 2      ! SELECT TRACK NUMBERS
: 1832 2      ! ACCORDINGLY.
: 1833 3      begin
: 1834 3      SWP_START = 0 + .OFFSET;
: 1835 3      SWP_END = 820 + .OFFSET;
: 1836 3      end;
: 1837 2
: 1838 2      if .SWP_START gequ .SWP_END
: 1839 2      then
: 1840 3      begin
: 1841 3      PRINTB (ERR_03);
: 1842 3      DOCLN;
: 1843 3      end;
: 1844 2
: 1845 2      !-
: 1846 2      ENDINIT;

```

.GLOBL L\$DLY

Address	Offset	Label	Instruction	Comment	Line No.
000000	004167	000000G	LINIT: JSR	R1,\$SAVE3	1732
000004	005746		TST	-(SP)	
000006	005000		CLR	R0	1743
000010	104441		TRAP	41	
000012	012700	000034	MOV	#34,R0	1745
000016	104447		TRAP	47	
000020	103033		BHIS	6\$	
000022	012746	000000G	MOV	#MSG.PWR,-(SP)	1748
000026	012746	000001	MOV	#1,-(SP)	
000032	010600		MOV	SP,R0	
000034	104417		TRAP	17	
000036	005002		CLR	R2	1750
000040	012703	023420	1\$: MOV	#23420,R3	1752
000044	010301		MOV	R3,R1	1753
000046	001411		2\$: BEQ	5\$	
000050	016700	000000G	MOV	L\$DLY,R0	
000054	001404		BEQ	4\$	
000056	005066	000004	3\$: CLR	4(SP)	
000062	005300		DEC	R0	
000064	001374		BNE	3\$	
000066	005301		4\$: DEC	R1	
000070	000766		BR	2\$	
000072	104422		5\$: TRAP	22	
000074	005202		INC	R2	1750
000076	020227	000074	CMP	R2,#74	
000102	101756		BLOS	1\$	
000104	104444		TRAP	44	1755
000106	022626		CMP	(SP),-(SP)	1747
000110	026727	000000G 000020	6\$: CMP	L\$UNIT,#20	1765
000116	103410		BLO	7\$	
000120	012746	000000G	MOV	#ERR.01,-(SP)	1768
000124	012746	000001	MOV	#1,-(SP)	
000130	010600		MOV	SP,R0	
000132	104417		TRAP	17	
000134	104444		TRAP	44	
000136	022626		CMP	(SP),-(SP)	1767
000140	012700	000036	7\$: MOV	#36,R0	1772
000144	104447		TRAP	47	
000146	103002		BHIS	8\$	
000150	000167	000426	JMP	18\$	
000154	012700	000040	8\$: MOV	#40,R0	1774
000160	104447		TRAP	47	
000162	040101		BIC	R1,R1	
000164	005501		ADC	R1	
000166	010167	000000G	MOV	R1,PASS0	
000172	032701	000001	BIT	#1,R1	1776
000176	001010		BNE	9\$	
000200	012700	000037	MOV	#37,R0	
000204	104447		TRAP	47	
000206	103404		BCS	9\$	
000210	012700	000035	MOV	#35,R0	
000214	104447		TRAP	47	
000216	103034		BHIS	12\$	
000220	012767	177777 000000G	9\$: MOV	#-1,LOG.UNIT	1779

000226	005067	000000G		CLR	NUM.RETRIES	:	1780
000232	005067	000000G		CLR	RETRIES	:	1781
000236	004767	000000V		JSR	PC,FIND.CLOCK	:	1782
000242	005727	000000G		TST	#CLK.TYPE	:	1784
000246	001011			BNE	10\$		
000250	012746	000000G		MOV	#ERR.02,-(SP)	:	1787
000254	012746	0000001		MOV	#1,-(SP)		
000260	010600			MOV	SP,R0	: SP,*	
000262	104417			TRAP	17		
000264	104444			TRAP	44		
000266	022626			CMP	(SP)*,(SP)*	:	1786
000270	000402			BR	11\$:	1784
000272	005077	000000G	10\$:	CLR	@CLK.CSR	:	1791
000276	104431		11\$:	TRAP	31	:	1796
000300	010067	000000G		MOV	R0,FREE.MEM.ADDR		
000304	011067	000000G		MOV	(R0),MEM.SIZE	: FREE.MEM.ADDR,*	1797
000310	005267	000000G	12\$:	INC	LOG.UNIT	:	1804
000314	026767	000000G	000000G	CMP	LOG.UNIT,L\$UNIT	:	1806
000322	103401			BLO	13\$		
000324	104444			TRAP	44		
000326	016700	000000G	13\$:	MOV	LOG.UNIT,R0	:	1809
000332	104442			TRAP	42		
000334	010067	000000G		MOV	R0,HWP.TABLE		
000340	001763			BEQ	12\$		
000342	012767	000000G	000000G	MOV	#RT,RT.TABLE	:	1811
000350	011067	000000G		MOV	(R0),RT	: HWP.TABLE,*	1812
000354	012701	000000G		MOV	#RT,R1	:	1813
000360	016061	0000002	0000002	MOV	2(R0),2(R1)		
000366	016061	0000004	0000004	MOV	4(R0),4(R1)	:	1814
000374	016061	0000006	0000006	MOV	6(R0),6(R1)	:	1815
000402	011067	000000G		MOV	(R0),RC25.ADDR	: RT,*	1816
000406	010100			MOV	R1,R0	: RT,*	1817
000410	016067	0000006	000000G	MOV	6(R0),UNIT		
000416	012746	000340		MOV	#340,-(SP)	:	1818
000422	012746	000000V		MOV	#NXMI,-(SP)		
000426	016046	0000002		MOV	2(R0)-,(SP)		
000432	012746	0000003		MOV	#3,-(SP)		
000436	104437			TRAP	37		
000440	016716	000000G		MOV	UNIT,(SP)	:	1819
000444	016746	000000G		MOV	RC25.ADDR,-(SP)		
000450	016746	000000G		MOV	LOG.UNIT,-(SP)		
000454	012746	000000G		MOV	#DBM1,-(SP)		
000460	012746	0000004		MOV	#4,-(SP)		
000464	010600			MOV	SP,R0	: SP,*	
000466	104414			TRAP	14		
000470	032767	0000001	000000G	BIT	#1,SWP.TOP	:	1821
000476	001403			BEQ	14\$		
000500	005067	000000G		CLR	OFFSET	:	1823
000504	000403			BR	15\$:	1821
000506	012767	001465	000000G	MOV	#1465,OFFSET	:	1825
000514	032767	0000001	000000G	BIT	#1,SWP.LIMIT	:	1827
000522	001011			BNE	16\$		
000524	016767	000000G	000000G	MOV	OFFSET,SWP.START	:	1831
000532	016767	000000G	000000G	MOV	OFFSET,SWP.END	:	1832
000540	062767	001464	000000G	ADD	#1464,SWP.END		
000546	026767	000000G	000000G	CMP	SWP.START,SWP.END	:	1835
000554	103410			BLO	17\$		


```

ZRCFR2      MISCELLANEOUS SECTIONS      27-Mar-1985 15:23:34      VAX-11 Bliss-16 V4.0-579
V03.0      INITIALIZE SECTION           11-Jan-1985 08:19:19      USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1
000556 012716 000000G      MOV      #ERR.03,(SP)      ;
000562 012746 000001      MOV      #1,-(SP)        ;
000566 010600      MOV      SP,R0           ; SP,*
000570 104414      TRAP     14
000572 104444      TRAP     44
000574 005726      TST      (SP)+           ;
000576 062706 000020      17$:    ADD     #20,SP    ;
000602 005726      18$:    TST      (SP)+           ;
000604 000207      RTS      PC

```

```

; Routine Size: 195 words,      Routine Base: AA$CODE * 0320
; Maximum stack depth per invocation: 16 words

```

```

000000 004767 177166      .SBTTL  L$INIT INITIALIZE SECTION
000004 104411      L$INIT::JSR  PC,LINIT    ;
000006 000207      TRAP     11
                                RTS      PC

```

```

; Routine Size: 4 words,      Routine Base: AA$CODE * 1126
; Maximum stack depth per invocation: 2 words

```

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AUTODROP SECTION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

```

: 1844 1 %sbttl 'AUTODROP SECTION'
: 1845 1 !*
: 1846 1 ! THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
: 1847 1 ! THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
: 1848 1 ! SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
: 1849 1 ! DROPPED FROM TESTING.
: 1850 1 !-
: 1851 2 BGNAUTO;
: 1852 2 return;
: 1853 1 ENDAUTO;

```

```

000000 000207          .SBTTL LAUTO AUTODROP SECTION
                    LAUTO: RTS PC ; 1843
: Routine Size: 1 word,      Routine Base: AA$CODE + 1136
: Maximum stack depth per invocation: 0 words

```

```

000000 004767 177772          .SBTTL L$AUTO AUTODROP SECTION
000004 104461          L$AUTO::JSR PC,LAUTO ; 1852
000006 000207          TRAP 61
                    RTS PC
: Routine Size: 4 words,      Routine Base: AA$CODE + 1140
: Maximum stack depth per invocation: 2 words

```

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
CLEANUP CODING SECTION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0112
Page 17
(8)

```

: 1854 1  %sbttl 'CLEANUP CODING SECTION'
: 1855 1  ;
: 1856 1  ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
: 1857 1  ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
: 1858 1  ;
: 1859 2  BGNCLN;
: 1860 2  .CLK_CSR = ZERO;          ! TURN OFF THE CLOCK
: 1861 2  P1 = ZERO;              ! CLEAR ERROR ROUTINE
: 1862 2  P2 = ZERO;              ! PARAMETERS P1 - P6
: 1863 2  P3 = ZERO;
: 1864 2  P4 = ZERO;
: 1865 2  P5 = ZERO;
: 1866 2  P6 = ZERO;
: 1867 2  RET_STATUS = ZERO;      ! CLEAR STATUS AND
: 1868 2  NUM_RETRIES = ZERO;    ! FLAGS
: 1869 2  RETRIES = FALSE;
: 1870 2  IN_BOUND = FALSE;      ! VER:C
: 1871 2  return;
: 1872 1  ENDCLN;

```

```

000000 005077 000000G          .SBTTL  LCLEAN CLEANUP CODING SECTION
000004 005067 000000G          LCLEAN: CLR  @CLK_CSR          ; 1860
000010 005067 000000G          CLR  P1                      ; 1861
000014 005067 000000G          CLR  P2                      ; 1862
000020 005067 000000G          CLR  P3                      ; 1863
000024 005067 000000G          CLR  P4                      ; 1864
000030 005067 000000G          CLR  P5                      ; 1865
000034 005067 000000G          CLR  P6                      ; 1866
000040 005067 000000G          CLR  RET_STATUS          ; 1867
000044 005067 000000G          CLR  NUM_RETRIES        ; 1868
000050 005067 000000G          CLR  RETRIES            ; 1869
000054 000207 000000G          CLR  IN_BOUND          ; 1870
                                RTS   PC                      ; 1853

```

```

; Routine Size: 23 words,      Routine Base: AA$CODE + 1150
; Maximum stack depth per invocation: 0 words

```

```

000000 004767 177716          .SBTTL  L$CLEAN CLEANUP CODING SECTION
000004 104412          L$CLEAN: JSR  PC,LCLEAN          ; 1871
000006 000207          TRAP  12
                                RTS   PC

```

```

; Routine Size: 4 words,      Routine Base: AA$CODE + 1226
; Maximum stack depth per invocation: 2 words

```

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
DROP UNIT SECTION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

```

: 1873 1  #sbttl 'DROP UNIT SECTION'
: 1874 1  !*
: 1875 1  ! THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
: 1876 1  ! TO NO LONGER BE TESTED.
: 1877 1  !-
: 1878 2  BGNDU;
: 1879 2  return;
: 1880 1  ENDDU;

```

```

000000 000207          LDU:  .SBTTL LDU DROP UNIT SECTION          ;          1872
                        RTS    PC

```

```

: Routine Size: 1 word,      Routine Base: AA$CODE + 1236
: Maximum stack depth per invocation: 0 words

```

```

000000 004767 177772  L$DU:: .SBTTL L$DU DROP UNIT SECTION          ;          1879
000004 104453          JSR   PC,LDU
000006 000207          TRAP  53
                        RTS    PC

```

```

: Routine Size: 4 words,      Routine Base: AA$CODE + 1240
: Maximum stack depth per invocation: 2 words

```

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
ADD UNIT SECTION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

```

: 1881 1  #sbttl 'ADD UNIT SECTION'
: 1882 1  !*
: 1883 1  ! THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
: 1884 1  ! TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
: 1885 1  ! TO THE TEST CYCLE.
: 1886 1  !-
: 1887 2  BGNAU;
: 1888 2  return;
: 1889 1  ENDAU;

```

```

000000 000207          LAU:  .SBTTL  LAU ADD UNIT SECTION          ;          1880
                        RTS    PC
: Routine Size: 1 word,      Routine Base: AA$CODE + 1250
: Maximum stack depth per invocation: 0 words

```

```

000000 004767 177772  L$AU:: .SBTTL  L$AU ADD UNIT SECTION      ;          1888
000004 104452          JSR    PC,LAU
000006 000207          TRAP  S2
                        RTS    PC
: Routine Size: 4 words,      Routine Base: AA$CODE + 1252
: Maximum stack depth per invocation: 2 words

```

: 1890 1 !<BLF/PAGE>

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
ADD UNIT SECTION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0115
Page 20
(11)

```

: 1891 1 psect
: 1892 1   code = AB$CODE;
: 1893 1
: 1894 1
: 1895 1   !..
: 1896 1   ! GLOBAL LOCATION "I AM NEX" IS SET TO TRUE WHICH INDICATES
: 1897 1   ! THE INITIALIZATION SEQUENCE INTERRUPT OCCURED.
: 1898 1   !--
: 1899 2   BGNSRV (NXMI);
: 1900 2   I AM NEX = %0'177777';
: 1901 2   CANCEL_TIMER = %0'177777';
: 1902 1   ENDSRV;

```

```

000000          .SBTTL NXMI ADD UNIT SECTION
                .PSECT AB$CODE, RO
000000 012767 177777 000000G      NXMI:: MOV    @-1,I.AM.NEX      ;           1900
000006 012767 177777 000000G      MOV    @-1,CANCEL_TIMER  ;           1901
000014 000002                    RTI                          ;           1899

```

```

: Routine Size: 7 words,      Routine Base: AB$CODE * 0000
: Maximum stack depth per invocation: 0 words

```

```

: 1903 1
: 1904 1   !..
: 1905 1   ! THE CLOCK INTERRUPT SERVICE ROUTINE IS ENTERED AT THE CLOCK RATE
: 1906 1   !--
: 1907 1
: 1908 2   BGNSRV (CLK_INT_SERV);
: 1909 2   TICKS = .TICKS * 1;           ! INCREMENT THE NUMBER OF TICK
: 1910 2
: 1911 2   if .TICKS eqlu .CLK_HERTZ    ! IF TOTAL NUMBER OF TICK = 60
: 1912 2   then                          ! THEN
: 1913 3   begin
: 1914 3   TICKS = 0;                     ! RESET TICK TO ZERO
: 1915 3   SECONDS = .SECONDS * 1;      ! INCREMENT THEN SECOND
: 1916 3
: 1917 3   if .SECONDS eqlu 60          ! IF SECOND = 60
: 1918 3   then                          ! THEN
: 1919 4   begin
: 1920 4   SECONDS = 0;                  ! RESET SECOND TO ZERO
: 1921 4   MINUTES = .MINUTES * 1;     ! INCREMENT THE MINUTES
: 1922 3   end;
: 1923 3
: 1924 2   end;
: 1925 2
: 1926 1   ENDSRV;

```

```

000000 005267 000000G      CLK.INT.SERV:: .SBTTL CLK.INT.SERV ADD UNIT SECTION
000004 026767 000000G 000000G      INC    TICKS                ;           1909
000012 001014                    CMP    TICKS,CLK.HERTZ      ;           1911
000014 005067 000000G      BNE    1$                  ;
                                CLR    TICKS                ;           1914

```

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
ADD UNIT SECTION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 B118-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0116
Page 21
(11)

000020	005267	000000G	INC	SECONDS	:	1915
000024	026727	000000G 000074	CMP	SECONDS,074	:	1917
000032	001004		BNE	18		
000034	005067	000000G	CLR	SECONDS	:	1920
000040	005267	000000G	INC	MINUTES	:	1921
000044	000002		RTI		:	1908

: Routine Size: 19 words, Routine Base: ABSCODE * 0016
 : Maximum stack depth per invocation: 0 words

```

: 1927 1 #sbttl 'FIND CLOCK ROUTINE'
: 1928 1
: 1929 1 global routine FIND_CLOCK : novalue =
: 1930 1 !
: 1931 1 !CHECK TO MAKE SURE THERE IS A CLOCK ON THE SYSTEM. IF NO_CLOCK, ABORT TO
: 1932 1 !SUPERVISOR.
: 1933 1 !OTHERWISE, DETERMINE WHETHER CLOCK IS AN L OR P CLOCK, GET ITS PARAMETERS.
: 1934 1 !
: 1935 2 begin
: 1936 2 CLK_TYPE = NO_CLOCK; !SET FLAG FOR NO CLOCK
: 1937 2
: 1938 3 if CLOCK (P, CLK_ADR) !IS THERE A P_CLOCK?
: 1939 2 then
: 1940 3 begin
: 1941 3 CLK_TYPE = P_CLOCK; !SET THE FLAGE FOR P_CLOCK
: 1942 3 CLK_CSR = ..CLK_ADR; !SAVE THE CSR ADDRESS
: 1943 3 CLK_HERTZ = .(.CLK_ADR * 6); !GET THE CLOCK RATE
: 1944 3 CLK_START = #0'105'; !SAVE THE STARTTING CLOCK VALUE
: 1945 3 end
: 1946 2 else
: 1947 3 begin
: 1948 3
: 1949 4 if CLOCK (L, CLK_ADR) !IS THERE AN L_CLOCK?
: 1950 3 then
: 1951 4 begin
: 1952 4 CLK_TYPE = L_CLOCK; !SET THE FLAG FOR L_CLOCK
: 1953 4 CLK_CSR = ..CLK_ADR; !SAVE THE CSR ADDRESS
: 1954 4 CLK_HERTZ = .(.CLK_ADR * 6); !GET THE CLOCK RATE
: 1955 4 CLK_START = #0'100'; !SAVE THE STARTING CLOCK VALUE
: 1956 3 end;
: 1957 3
: 1958 2 end;
: 1959 2
: 1960 2 if .CLK_TYPE nequ NO_CLOCK !IF CLOCK WAS FOUND THEN
: 1961 2 then
: 1962 3 begin
: 1963 3 VEC_AD = .(.CLK_ADR * 4); !GET CLOCK VECTOR ADDRESS
: 1964 3 SETVEC (.VEC_AD, CLK_INT_SERV, PRI05); !SET VECTOR & SERVICE ADDR.
: 1965 2 end;
: 1966 2
: 1967 1 end;

```

Address	Offset	Hex	Label	Operation	Comment	Line
000000	005067	000000G	.SBTTL	FIND.CLOCK FIND CLOCK ROUTINE		
000004	012700	000120	FIND.CLOCK::	CLR CLK.TYPE		1936
000010	104462			MOV #120,R0		1938
000012	103016			TRAP 62		
000014	010067	000000G		BHIS 1#		
000020	012767	000001 000000G		MOV R0,CLK.ADR	: R0,*	
000026	011067	000000G		MOV #1,CLK.TYPE		1941
000032	016067	000006 000000G		MOV (R0),CLK.CSR	: CLK.ADR,*	1942
000040	012767	000105 000000G		MOV 6(R0),CLK.HERTZ		1943
000046	000421			MOV #105,CLK.START		1944
000050	012700	000114	1#:	BR 2#		1938
000054	104462			MOV #114,R0		1949
				TRAP 62		

B10

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
FIND CLOCK ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0118
Page 23
(12)

000056	103015				BHIS	24			
000060	010067	000000G			MOV	RO,CLK.ADR		; RO,*	
000064	012767	177777	000000G		MOV	#-1,CLK.TYPE		:	
000072	011067	000000G			MOV	(RO),CLK.CSR		; CLK.ADR,*	1952
000076	016067	000006	000000G		MOV	6(RO),CLK.HERTZ		:	1953
000104	012767	000100	000000G		MOV	#100,CLK.START		:	1954
000112	005767	000000G		24:	TST	CLK.TYPE		:	1955
000116	001421				BEQ	34		:	1960
000120	016700	000000G			MOV	CLK.ADR,RO		:	
000124	116067	000004	000000G		MOVB	4(RO),VEC.AD		:	1963
000132	012746	000240			MOV	#240,-(SP)		:	
000136	012746	000016'			MOV	#CLK.INT.SERV,-(SP)		:	1964
000142	005046				CLR	-(SP)		:	
000144	116716	000000G			MOVB	VEC.AD,(SP)		:	
000150	012746	000003			MOV	#3,-(SP)		:	
000154	104437				TRAP	37		:	
000156	062706	000010			ADD	#10,SP		:	1962
000162	000207			34:	RTS	PC		:	1929

; Routine Size: 58 words, Routine Base: AB\$CODE + 0064
; Maximum stack depth per invocation: 6 words

; 1968 1

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
CLOCK INIT ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC2.B16;1

SEQ 0119
Page 24
(13)

```

: 1969 1  %sbttl 'CLOCK INIT ROUTINE'
: 1970 1
: 1971 1  global routine CLOCK_INIT : novalue =
: 1972 1  **
: 1973 1  !INIT CLOCK
: 1974 1  !
: 1975 1  !
: 1976 2    begin
: 1977 2    .CLK_CSR = ZERO;           ! STOP THE CLOCK
: 1978 2    TICKS = 0;                 ! CLEAR THE COUNTER
: 1979 2    SECONDS = 0;
: 1980 2    MINUTES = 0;
: 1981 2    .CLK_CSR = .CLK_START;     ! START THE CLOCK
: 1982 1    end;

```

000000	005077	000000G	.SBTTL	CLOCK.INIT	CLOCK INIT ROUTINE	
			CLOCK.INIT::			
000004	005067	000000G	CLR	@CLK_CSR		1977
000010	005067	000000G	CLR	TICKS		1978
000014	005067	000000G	CLR	SECONDS		1979
000020	016777	000000G 000000G	CLR	MINUTES		1980
000026	000207		MOV	CLK.START,@CLK_CSR		1981
			RTS	PC		1971

```

: Routine Size: 12 words,      Routine Base: AB$CODE + 0250
: Maximum stack depth per invocation: 0 words

```

: 1983 1

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
RC25 CONTROLLER ERROR REPORTING

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0120
Page 25
(14)

```

: 1984 1  %sbttl 'RC25 CONTROLLER ERROR REPORTING'
: 1985 1  BGNMSG (RC25$ERR_RPT);
    
```

```

000000 004767 000000V          .SBTTL RC25$ERR.RPT RC25 CONTROLLER ERROR REPORTING
                                RC25$ERR.RPT::
000004 104423          JSR      PC,M$RC25$ERR.RPT
000006 000207          TRAP    23
                                RTS      PC
    
```

1985

```

: Routine Size: 4 words,      Routine Base: AB$CODE + 0300
: Maximum stack depth per invocation: 2 words
    
```

```

: 1986 2  : **
: 1987 2  : FUNCTIONAL DESCRIPTION:
: 1988 2  :
: 1989 2  :     THIS ROUTINE IS CALLED BY THE DIAGNOSTIC SUPERVISOR VIA
: 1990 2  :     THE "PRLINK" ARGUMENT SPECIFIED IN THE $DS_ERRXXX MACRO
: 1991 2  :     TO REPORT DETAILED RC 25 CONTROLLER ERRORS.
: 1992 2  :
: 1993 2  : FORMAL PARAMETERS:
: 1994 2  :
: 1995 2  :     P1          - POINTER TO FORMATED ERROR MESSAGE.
: 1996 2  :     P2          - FIELD REPLACEABLE UNIT CALL-OUT MASK.
: 1997 2  :     P3          - RC 25 CONTROLLER REGISTER PRINT-OUT MASK.
: 1998 2  :     P4          - DATA.
: 1999 2  :     P5          - DATA.
: 2000 2  :     P6          - DATA.
: 2001 2  :
: 2002 2  : IMPLICIT INPUTS:
: 2003 2  :     RET_STATUS
: 2004 2  :
: 2005 2  : IMPLICIT OUTPUTS:
: 2006 2  :     - NONE -
: 2007 2  :
: 2008 2  : COMPLETION CODES:
: 2009 2  :     - NONE -
: 2010 2  :
: 2011 2  : SIDE EFFECTS:
: 2012 2  :     - NONE -
: 2013 2  :
: 2014 2  : PRINT SUPPLEMENTAL ERROR INFO
: 2015 2  :
: 2016 2  :
: 2017 2  :
: 2018 2  :
: 2019 2  :
: 2020 2  :
: 2021 2  :
: 2022 2  : if .P1 neq 0          ! IF ERROR MESSAGE POINTER
: 2023 2  : then                ! ISN'T 0, THEN PRINT-OUT
: 2024 3  :     begin
: 2025 3  :
: 2026 3  :     if .P_MASK eq 3 then PRINTB (.P1, .P4, .P5, .P6); ! SUPPLEMENTAL ERROR INFO.
: 2027 3  :
: 2028 3  :     if .P_MASK eq 2 then PRINTB (.P1, .P6, .P4, .P5);
    
```

```

: 2029 3
: 2030 3      if .P_MASK equl 1 then PRINTB (.P1, .P4);
: 2031 3
: 2032 2      end;
: 2033 2
: 2034 2      if .P3 nequ 0
: 2035 2      then
: 2036 2          PRINTB (FMT3, .P6, .P3);
: 2037 2
: 2038 2
: 2039 2
: 2040 2      ! PERFORM FIELD REPLACEABLE UNIT CALL-OUT
: 2041 2
: 2042 2      !if .P2 neau 0
: 2043 2      !then
: 2044 2      PRT$FRU_CALLOUT (.P2);
: 2045 2      ! CLEAR ALL PARAMETERS
: 2046 2      P1 = ZERO;
: 2047 2      P2 = ZERO;
: 2048 2      P3 = ZERO;
: 2049 2      P4 = ZERO;
: 2050 2      P5 = ZERO;
: 2051 2      P6 = ZERO;
: 2052 1      ENDMSG;

```

! IF ELIGIBLE REGISTER(S)
! PRESENT, THEN PRINT-OUT
! SELECTED CONTROLLER

! REGISTER(S).

! IF ELIGIBLE FRU CALL-OUT(S)
! PRESENT, THEN PRINT-OUT
! FEILD REPLACEABLE UNITS.

! END OF ROUTINE:

000000	005767	000000G		.SBTTL M\$RC25\$ERR.RPT RC25 CONTROLLER ERROR REPORTING	
			M\$RC25\$ERR.RPT:		
000004	001462		TST	P1	2022
000006	126727	000000G 000003	BEQ	3\$	
000014	001016		CMPB	P.MASK,#3	2026
000016	016746	000000G	BNE	1\$	
000022	016746	000000G	MOV	P6,-(SP)	
000026	016746	000000G	MOV	P5,-(SP)	
000032	016746	000000G	MOV	P4,-(SP)	
000036	012746	000004	MOV	P1,-(SP)	
000042	010600		MOV	#4,-(SP)	
000044	104414		MOV	SP,R0	; SP,*
000046	062706	000012	TRAP	14	
000052	126727	000000G 000002	ADD	#12,SP	
000060	001016		1\$:	CMPB	P.MASK,#2
000062	016746	000000G	BNE	2\$	2028
000066	016746	000000G	MOV	P5,-(SP)	
000072	016746	000000G	MOV	P4,-(SP)	
000076	016746	000000G	MOV	P6,-(SP)	
000102	012746	000004	MOV	P1,-(SP)	
000106	010600		MOV	#4,-(SP)	
000110	104414		MOV	SP,R0	; SP,*
000112	062706	000012	TRAP	14	
000116	126727	000000G 000001	ADD	#12,SP	
000124	001012		2\$:	CMPB	P.MASK,#1
000126	016746	000000G	BNE	3\$	2030
000132	016746	000000G	MOV	P4,-(SP)	
000136	012746	000002	MOV	P1,-(SP)	
000142	010600		MOV	#2,-(SP)	
000144	104414		MOV	SP,R0	; SP,*
			TRAP	14	

F10

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
RC25 CONTROLLER ERROR REPORTING

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0122
Page 27
(14)

000146	062706	000006		ADD	#6,SP		
000152	005767	000000G	3\$:	TST	P3	:	2034
000156	001414			BEQ	4\$		
000160	016746	000000G		MOV	P3,-(SP)	:	2036
000164	016746	000000G		MOV	P6,-(SP)		
000170	012746	000000G		MOV	#FMT3,-(SP)		
000174	012746	000003		MOV	#3,-(SP)		
000200	010600			MOV	SP,R0	: SP,*	
000202	104414			TRAP	14		
000204	062706	000010		ADD	#10,SP		
000210	016746	000000G	4\$:	MOV	P2,-(SP)	:	2044
000214	004767	000000V		JSR	PC,PRT#FRU.CALLOUT		
000220	005067	000000G		CLR	P1	:	2046
000224	005067	000000G		CLR	P2	:	2047
000230	005067	000000G		CLR	P3	:	2048
000234	005067	000000G		CLR	P4	:	2049
000240	005067	000000G		CLR	P5	:	2050
000244	005067	000000G		CLR	P6	:	2051
000250	005726			TST	(SP).	:	1985
000252	000207			RTS	PC	:	

: Routine Size: 86 words, Routine Base: AB#CODE + 0310
: Maximum stack depth per invocation: 7 words

```

: 2053 1 #sbttl 'FIELD REPLACEABLE UNIT REPORTING'
: 2054 1
: 2055 1 global routine PRT$FRU_CALLOUT (FRU$MASK) : novalue =
: 2056 1 **
: 2057 1 : FUNCTIONAL DESCRIPTION:
: 2058 1 :
: 2059 1 :     THIS ROUTINE REPORTS FIELD REPLACEABLE UNITS WHICH ARE
: 2060 1 :     DEEMED ELIGIBLE FOR PRINT-OUT BY THE FAILING TEST.
: 2061 1 :
: 2062 1 : FORMAL PARAMETERS:
: 2063 1 :
: 2064 1 :     FRU$MASK      -   FIELD REPLACEABLE UNIT CALL-OUT MASK.
: 2065 1 :
: 2066 1 : IMPLICIT INPUTS:
: 2067 1 :
: 2068 1 :     - NONE -
: 2069 1 :
: 2070 1 : IMPLICIT OUTPUTS:
: 2071 1 :
: 2072 1 :     - NONE -
: 2073 1 :
: 2074 1 : COMPLETION CODES:
: 2075 1 :
: 2076 1 :     - NONE -
: 2077 1 :
: 2078 1 : SIDE EFFECTS:
: 2079 1 :
: 2080 1 :     - NONE -
: 2081 1 :
: 2082 1 : -- begin
: 2083 1 :
: 2084 1 : local
: 2085 1 :     FRU$MSG;           ! ALLOCATE STORAGE FOR
: 2086 1 :
: 2087 1 :                       ! POINTER TO FRU MESSAGE.
: 2088 1 :
: 2089 1 :     ! PERFORM FIELD REPLACEABLE UNIT CALL-OUT
: 2090 1 :     !
: 2091 1 :
: 2092 1 :     incru FRU_SELECT from 0 to 3 do           ! CHECK EACH FRU FOR
: 2093 1 :                                           ! POSSIBLE CALL-OUT.
: 2094 1 :
: 2095 1 :         if BIT_TST (.FRU$MASK, 1+.FRU_SELECT) ! IF CURRENT FRU ELIGIBLE
: 2096 1 :         then ! FOR PRINT-OUT THEN GET
: 2097 1 :             begin ! POINTER TO FRU MESSAGE.
: 2098 1 :
: 2099 1 :                 selectu 1+.FRU_SELECT of ! SELECT FRU FROM ONE OF
: 2100 1 :                 set ! THE FOLLOWING:
: 2101 1 :
: 2102 1 :                     [ADAPT] :
: 2103 1 :                         FRU$MSG = ADAPTO; ! GET ASYNC FRU MESSAGE.
: 2104 1 :
: 2105 1 :                     [CONTR] :
: 2106 1 :                         FRU$MSG = CONTR0; ! GET SYNC FRU MESSAGE
: 2107 1 :
: 2108 1 :                     [DRIVE] :
: 2109 1 :                         FRU$MSG = DRIVE_; ! GET ARR_DAT FRU MESSAGE

```

H10

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
FIELD REPLACEABLE UNIT REPORTING

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 B118-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0124
Page 29
(15)

: 2110 3
: 2111 3
: 2112 3
: 2113 3
: 2114 3
: 2115 3
: 2116 2
: 2117 2
: 2118 1

```
[MECH] :  
FRU$MSG = MECHAN;  
tes;  
PRINTX (FRU, .FRU$MSG, .UNIT);  
end;  
end;  
! GET MEM ARR FRU MESSAGE  
! PRINT FRU CALL-OUT.  
! END OF ROUTINE;  
! 'PRT$FRU_CALLOUT'.
```

Address	Offset	OpCode	Comment	Label
000000	004167	000000G	SBTTL PRT\$FRU_CALLOUT FIELD REPLACEABLE UNIT REPORTING	
000004	005002	000001	JSR R1,\$SAVE3	2055
000006	012746	000001	CLR R2	2092
000012	010246	000000G	MOV #1,-(SP)	2095
000014	004767	000000G	MOV R2,-(SP)	
000020	010001		JSR PC,BL\$SHF	
000022	005726		MOV R0,R1	
000024	016600	000014	TST (SP)+	
000030	005100		MOV 14(SP),R0	; FRU\$MASK,*
000032	040001		COM R0	
000034	012716	000001	BIC R0,R1	
000040	010246		MOV #1,(SP)	
000042	004767	000000G	MOV R2,-(SP)	; FRU.SELECT,*
000046	022626		JSR PC,BL\$SHF	
000050	020100		CMP (SP)+,(SP)+	
000052	001044		CMP R1,R0	
000054	012746	000001	BNE 6\$	
000060	010246		MOV #1,-(SP)	; FRU.SELECT,*
000062	004767	000000G	MOV R2,-(SP)	
000066	020027	000001	JSR PC,BL\$SHF	
000072	001002		CMP R0,#1	
000074	012703	000000G	BNE 2\$	
000100	020027	000002	MOV #ADAPTO,R3	; *,FRU\$MSG
000104	001002		CMP R0,#2	
000106	012703	000000G	BNE 3\$	
000112	020027	000004	MOV #CONTRO,R3	; *,FRU\$MSG
000116	001002		CMP R0,#4	
000120	012703	000000G	BNE 4\$	
000124	020027	000010	MOV #DRIVE.,R3	; *,FRU\$MSG
000130	001002		CMP R0,#10	
000132	012703	000000G	BNE 5\$	
000136	016716	000000G	MOV #MECHAN,R3	; *,FRU\$MSG
000142	010346		MOV UNIT,(SP)	
000144	012746	000000G	MOV R3,-(SP)	; FRU\$MSG,*
000150	012746	000003	MOV #FRU,-(SP)	
000154	010600		MOV #3,-(SP)	
000156	104415		MOV SP,R0	; SP,*
000160	062706	000012	TRAP 15	
000164	005202		ADD #12,SP	
000166	020227	000003	INC R2	; FRU.SELECT
000172	101705		CMP R2,#3	; FRU.SELECT,*
000174	000207		BLOS 1\$	
			RTS PC	2055

; Routine Size: 63 words, Routine Base: AB\$CODE + 0564

I10

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
FIELD REPLACEABLE UNIT REPORTING

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0125
Page 30
(15)

: Maximum stack depth per invocation: 11 words

: 2119 1

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC INITIALIZATION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

: 2120 1
: 2121 1
: 2122 1
: 2123 1
: 2124 1
: 2125 1
: 2126 1
: 2127 1
: 2128 1
: 2129 1
: 2130 1
: 2131 1
: 2132 1
: 2133 1
: 2134 1
: 2135 1
: 2136 1
: 2137 1
: 2138 1
: 2139 1
: 2140 1
: 2141 1
: 2142 1
: 2143 1
: 2144 1
: 2145 1
: 2146 1
: 2147 1
: 2148 1
: 2149 1
: 2150 1
: 2151 1
: 2152 1
: 2153 1
: 2154 1
: 2155 1
: 2156 1
: 2157 1
: 2158 1
: 2159 1
: 2160 1
: 2161 1
: 2162 1
: 2163 1
: 2164 1
: 2165 1
: 2166 1
: 2167 2
: 2168 2
: 2169 2
: 2170 2
: 2171 2
: 2172 2
: 2173 2
: 2174 2
: 2175 2
: 2176 2

!sbttl 'AZTEC INITIALIZATION'

global routine AZT_INIT =

**
FUNCTIONAL DESCRIPTION:

THIS ROUTINE FIRST STARTS AZTEC INIT BY WRITING TO RCIP.
THEN EXPECTS TO READ STEP 1 BIT IN RCSA INDICATING THAT
THE PORT IS READY TO ACCEPT STEP 1 WRITE DATA. IF THE
STEP READ DATA WAS OK THEN WRITES STEP 1 WRITE DATA TO RCSA
THEN WAITS TO READ STEP 2 BIT IN RCSA. THIS PROCEDURE OF READ
FOLLOWED BY WRITE IS DONE AS GIVEN BY B_MASK.
IF THERE WAS ANY PORT FATAL ERROR IN ANY OF THE STEPS,
THEN THE FAILURE DATA OF RCSA IS PRESERVED AND FURTHER
STEPS ABORTED.
THIS ROUTINE WILL BE USED ONLY IF INTERRUPT WAS ENABLED
IN STEP 1 WRITE.

FORMAL PARAMETERS:

- NONE -

IMPLICIT INPUTS:

DATA1 = STEP 1 WRITE DATA
DATA2 = STEP 2 WRITE DATA
DATA3 = STEP 3 WRITE DATA
DATA4 = STEP 4 WRITE DATA

B_MASK = WHICH STEPS WILL BE DONE
%0 1 = STEP 1
%0 3 = STEP 1,2
%0 7 = STEP 1,2,3
%017 = STEP 1,2,3,4

IMPLICIT OUTPUTS:

IF ERROR OR NO STEP IT WILL RETURN
P1-P6, P_MASK

COMPLETION CODES:

RET_STATUS RETURNS COMPLETION CODE.

SIDE EFFECTS:

- NONE -

--
begin

local

N,
MASK,
COUNT,
DATA;

!STEP NUMBER
!STEP MASK
!TIME OUT COUNT
!WRITE DATA FOR THE STEP

!
! INIT THE AZTEC

```

: 2177 2      !
: 2178 2      ! I_AM_NEX = ALL_ONES;          ! INIT INTERRUPT FLAG
: 2179 2      !
: 2180 2
: 2181 2      ! THE FOLLOWING LOOP WILL DO STEP 1 THRU 4 AS GIVEN BY B_MASK
: 2182 2      ! INPUT SELECTING APPROPRIATE DATA INPUT FOR STEP WRITES. IF
: 2183 2      ! ERROR IN SA REGISTER P1 - P4 AND P_MASK WILL BE SUPPLIED FOR
: 2184 2      ! ERROR REPORT. ONLY SA DATA FOR THE FINAL WRITE STEP IS PRESERVED.
: 2185 2      !
: 2186 2      MASK = %b'0001';          ! STEP MASK
: 2187 2      WRT_RC25 (RCIP, ALL_ONES); ! START INIT
: 2188 2      DELAY (2);                ! WAIT FOR COMPLETION
: 2189 2
: 2190 2      incru N from 0 to 4 do
: 2191 3          begin
: 2192 3
: 2193 4          if (.N eqv 0 or BIT_TST (.B_MASK, .MASK))      ! TEST FOR STEP NUMBER
: 2194 3              then
: 2195 4                  begin
: 2196 4                      !
: 2197 4
: 2198 4                      selectoneu .N of                    ! SELECT CORRECT WRITE
: 2199 4                          set
: 2200 4
: 2201 4                          [0] : DATA = ALL_ONES;          !
: 2202 4
: 2203 4                          [1] : DATA = .DATA1;          ! DATA FOR STEP WRITES
: 2204 4
: 2205 4                          [2] : DATA = .DATA2;
: 2206 4
: 2207 4                          [3] : DATA = .DATA3;
: 2208 4
: 2209 4                          [4] : DATA = .DATA4;
: 2210 4
: 2211 4                          tes;
: 2212 4
: 2213 4                      !
: 2214 4
: 2215 4                      if .N geqv 1 then WRT_RC25 (RCSA, .DATA); ! STEP N WRITE DATA TO SA
: 2216 4
: 2217 4
: 2218 4
: 2219 4
: 2220 4          incru COUNT from 0 to 20 do          ! TIME OUT WAIT LOOP
: 2221 5              begin
: 2222 5                  DELAY (333);                ! DELAY 1 SEC. APPROX.
: 2223 5
: 2224 5                  if .I_AM_NEX eqv ALL_ONES then exitloop;
: 2225 5
: 2226 5                  BREAK;                      ! WATCH FOR CONTROL C.
: 2227 5                  end;
: 2228 4
: 2229 4
: 2230 4          if .I_AM_NEX eqv ALL_ONES          ! IF INTERRUPT OCCURED ?
: 2231 4              then
: 2232 5                  begin
: 2233 5                      DELAY (2);

```

```

: 2234 5      RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];    ! STEP N  READ
: 2235 5
: 2236 5      if .N nequ 0 then MASK = .MASK+1;          ! INCREMENT STEP
: 2237 5
: 2238 5      I_AM_NEX = ZERO;
: 2239 5
: 2240 6      if (.RC25_DATA [RCSA, RCSA_ER] nequ ZERO)          ! IF SA REGISTER CONTAINS
: 2241 5      then
: 2242 6          begin
: 2243 6              RET_STATUS = PFE_CODE;          ! FATAL ERROR
: 2244 6              exitloop;
: 2245 6          end
: 2246 5      else
: 2247 6          begin
: 2248 6
: 2249 7              if (.RC25_DATA [RCSA, RCSA_STEP] nequ .MASK)      ! ERROR OR INCORRECT STEP
: 2250 6              then
: 2251 7                  begin                                  ! SUPPLY P1 THRU P6 AND
: 2252 7                      P_MASK = 2;
: 2253 7                      P1 = FMT3;
: 2254 7                      P2 = ADAPT;
: 2255 7                      P4 = (.RC25_ADDR) * 2;
: 2256 7                      P5 = .RC25_DATA [RCSA, RC_ALL];
: 2257 7                      P6 = .MASK;
: 2258 7                      MSGADR = MSG_14;
: 2259 7                      RET_STATUS = TRUE;
: 2260 7                      return .RET_STATUS;          ! TRUE STATUS.
: 2261 6                  end;
: 2262 6              end;
: 2263 5          end
: 2264 5      else
: 2265 5          begin
: 2266 4              RET_STATUS = CTO_CODE;          ! IF YOU GET HERE PORT
: 2267 5              exitloop;                          ! FAILED TO INTERRUPT
: 2268 5              end;                              ! WITHIN TIME ALLOWED
: 2269 5
: 2270 4          end;
: 2271 4      end;
: 2272 3      end;
: 2273 3
: 2274 2      end;
: 2275 2
: 2276 2      if .RET_STATUS          ! IF STATUS WAS A FAILURE
: 2277 2      then
: 2278 3          begin          ! THEN SUPPLY P1-P6
: 2279 3              P_MASK = 2;          ! FOR TEST MODULE TO
: 2280 3              P1 = FMT3;          ! REPORT ERROR
: 2281 3              P2 = ADAPT;
: 2282 3              P4 = (.RC25_ADDR) * 2;
: 2283 3              P5 = .RC25_DATA [RCSA, RC_ALL];
: 2284 3              P6 = .MASK;
: 2285 3              MSGADR = MSG_14;
: 2286 3              return .RET_STATUS;
: 2287 3          end
: 2288 2      else
: 2289 2          return RET_STATUS = PAS_CODE;          ! OTHERWISE GOOD STATUS
: 2290 2

```

M10

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC INITIALIZATION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0129
Page 34
(16)

: 2291 1 end;

		.SBTTL	AZT.INIT	AZTEC INITIALIZATION	
000000	004167	000000G	AZT.INIT::		
			JSR	R1,\$SAVE5	2122
000004	024646		CMP	-(SP),-(SP)	
000006	012767	177777 000000G	MOV	0-1,I.AM.NEX	2178
000014	012704	000001	MOV	01,R4	2186
000020	012700	177777	MOV	0-1,R0	2187
000024	010077	000000G	MOV	R0,RC25.ADDR	
000030	012701	000002	MOV	02,R1	2188
000034	001411		1\$: BEQ	4\$	
000036	016700	000000G	MOV	L\$DLY,R0	
000042	001404		BEQ	3\$	
000044	005066	000002	2\$: CLR	2(SP)	
000050	005300		DEC	F)	
000052	001374		BNE	2\$	
000054	005301		3\$: DEC	R1	
000056	000766		BR	1\$	
000060	005005		4\$: CLR	R5	2190
000062	005705		5\$: TST	R5	2193
000064	001412		BEQ	6\$	
000066	005000		CLR	R0	
000070	156700	000000G	BISB	B.MASK,R0	
000074	010401		MOV	R4,R1	
000076	005101		COM	R1	
000100	040100		BIC	R1,R0	
000102	020004		CMP	R0,R4	
000104	001402		BEQ	6\$	
000106	000167	000420	JMP	26\$	
000112	010500		6\$: MOV	R5,R0	2198
000114	001003		BNE	7\$	2201
000116	012702	177777	MOV	0-1,R2	2202
000122	000427		BR	11\$	2198
000124	020027	000001	7\$: CMP	R0,01	2204
000130	001003		BNE	8\$	
000132	016702	000000G	MOV	DATA1,R2	2205
000136	000421		BR	11\$	2198
000140	020027	000002	8\$: CMP	R0,02	2207
000144	001003		BNE	9\$	
000146	016702	000000G	MOV	DATA2,R2	2208
000152	000413		BR	11\$	2198
000154	020027	000003	9\$: CMP	R0,03	2210
000160	001003		BNE	10\$	
000162	016702	000000G	MOV	DATA3,R2	2211
000166	000405		BR	11\$	2198
000170	020027	000004	10\$: CMP	R0,04	2213
000174	001002		BNE	11\$	
000176	016702	000000G	MOV	DATA4,R2	2214
000202	005705		11\$: TST	R5	2219
000204	001405		BEQ	12\$	
000206	010201		MOV	R2,R1	
000210	016700	000000G	MOV	RC25.ADDR,R0	
000214	010160	000002	MOV	R1,2(R0)	
000220	005003		12\$: CLR	R3	2221
000222	012701	000515	13\$: MOV	0515,R1	2223

N10

ZRCFR2 V03.0	MISCELLANEOUS SECTIONS AZTEC INITIALIZATION	27-Mar-1985 15:23:34 11-Jan-1985 08:19:19	VAX-11 Bliss-16 V4.0-579 USER#1:(AZTEC.CZRCFC)ZRCFC2.016;1	SEQ 0130 Page 35 (16)	
000226	001411	14:	BEQ 17:		
000230	016700		MOV L#DLY,R0	; *,##TMP1	
000234	001404		BEQ 16:		
000236	005066	000002	15:	CLR 2(SP)	; ##TMP
000242	005300		DEC R0	; ##TMP1	
000244	001374		BNE 15:		
000246	005301		DEC R1	; ##TMP2	
000250	000766		BR 14:		
000252	026727	000000G 177777	17:	CMP I.AM.NEX,0-1	
000260	001405		BEQ 18:		
000262	104422		TRAP 22		
000264	005203		INC R3	; COUNT	
000266	020327	000024	CMP R3,024	; COUNT,*	
000272	101753		BLOS 13:		
000274	026727	000000G 177777	18:	CMP I.AM.NEX,0-1	
000302	001107		BNE 25:		
000304	012703	000002	MOV #2,R3	; *,##TMP2	
000310	001411		19:	BEQ 22:	
000312	016700	000000G	MOV L#DLY,R0	; *,##TMP1	
000316	001404		BEQ 21:		
000320	005066	000002	20:	CLR 2(SP)	; ##TMP
000324	005300		DEC R0	; ##TMP1	
000326	001374		BNE 20:		
000330	005303		21:	DEC R3	; ##TMP2
000332	000766		BR 19:		
000334	016700	000000G	22:	MOV RC25.ADDR,R0	
000340	016016	000002	MOV 2(R0),(SP)	; *,RC.REG	
000344	011667	000002G	MOV (SP),RC25.DATA+2	; RC.REG,*	
000350	005705		TST R5	; N	
000352	001401		BEQ 23:		
000354	006304		ASL R4	; MASK	
000356	005067	000000G	23:	CLR I.AM.NEX	
000362	032767	100000 000002G	BIT #100000,RC25.DATA+2		
000370	001404		BEQ 24:		
000372	012767	000021 000000G	MOV #21,RET.STATUS		
000400	000462		BR 27:		
000402	010401		24:	MOV R4,R1	; MASK,*
000404	016700	000002G	MOV RC25.DATA+2,R0		
000410	006200		ASR R0		
000412	006200		ASR R0		
000414	006200		ASR R0		
000416	000300		SWAB R0		
000420	042700	177760	BIC #177760,R0		
000424	020001		CMP R0,R1		
000426	001441		BEQ 26:		
000430	112767	000002 000000G	MOVB #2,P.MASK		
000436	012767	000000G 000000G	MOV #FMT3,P1		
000444	012767	000001 000000G	MOV #1,P2		
000452	016700	000000G	MOV RC25.ADDR,R0		
000456	062700	000002	ADD #2,R0		
000462	010067	000000G	MOV R0,P4		
000466	016767	000002G 000000G	MOV RC25.DATA+2,P5		
000474	010467	000000G	MOV R4,P6	; MASK,*	
000500	012767	000000G 000000G	MOV #MSG.14,MSGADR		
000506	012767	000001 000000G	MOV #1,RET.STATUS		
000514	016700	000000G	MOV RET.STATUS,R0		
000520	000453		BR 29:		

B11

ZRCFB2	MISCELLANEOUS SECTIONS								
V03.0	AZTEC INITIALIZATION								
				27-Mar-1985 15:23:34	VAX-11 Bliss-16 V4.0-579	SEQ 0131			
				11-Jan-1985 08:19:19	USER#1:(AZTEC.CZRCFC)ZRCFC2.B16;1	Page 36			
						(16)			

000522	012767	000011	000000G		25:	MOV	#11,RET.STATUS	;		
000530	000406					BR	27	;		2268
000532	005205				26:	INC	R5	;	N	2267
000534	020527	000004				CMP	R5,#4	;	N,*	2190
000540	101002					BHI	27			
000542	000167	177314				JMP	5			
000546	032767	000001	000000G		27:	BIT	#1,RET.STATUS	;		2276
000554	001432					BEQ	28			
000556	112767	000002	000000G			MOVB	#2,P.MASK	;		2279
000564	012767	000000G	000000G			MOV	#FMT3,P1	;		2280
000572	012767	000001	000000G			MOV	#1,P2	;		2281
000600	016700	000000G				MOV	RC25.ADDR,R0	;		2282
000604	062700	000002				ADD	#2,R0			
000610	010067	000000G				MOV	R0,P4			
000614	016767	000002G	000000G			MOV	RC25.DATA*2,P5	;		2283
000622	010467	000000G				MOV	R4,P6	;	MASK,*	2284
000626	012767	000000G	000000G			MOV	#MSG.14,MSGADR	;		2285
000634	016700	000000G				MOV	RET.STATUS,R0	;		2289
000640	000403					BR	29			
000642	005067	000000G			28:	CLR	RET.STATUS			
000646	005000					CLR	R0			
000650	022626				29:	CMP	(SP)*,(SP)*	;		2122
000652	000207					RTS	PC			

; Routine Size: 214 words, Routine Base: AB#CODE * 0762
; Maximum stack depth per invocation: 10 words

; 2292 1

```

: 2293 1  #sbttl 'AZTEC INITIALIZATION BY POLING'
: 2294 1
: 2295 1  global routine AZP_INIT =
: 2296 1  **
: 2297 1  ! FUNCTIONAL DESCRIPTION:
: 2298 1
: 2299 1
: 2300 1  !
: 2301 1  ! THIS ROUTINE FIRST STARTS AZTEC INIT BY WRITING TO RCIP.
: 2302 1  ! THEN EXPECTS TO READ STEP 1 BIT IN RCSA INDICATING THAT
: 2303 1  ! THE PORT IS READY TO ACCEPT STEP 1 WRITE DATA. IF THE
: 2304 1  ! STEP READ DATA WAS OK THEN WRITES STEP 1 WRITE DATA TO RCSA
: 2305 1  ! THEN WAITS TO READ STEP 2 BIT IN RCSA. THIS PROCEDURE OF READ
: 2306 1  ! FOLLOWED BY WRITE IS DONE AS GIVEN BY B_MASK.
: 2307 1  ! IF THERE WAS ANY PORT FATAL ERROR IN ANY OF THE STEPS,
: 2308 1  ! THEN THE FAILURE DATA OF RCSA IS PRESERVED ANF FURTHER
: 2309 1  ! STEPS ABORTED.
: 2310 1  ! THIS ROUTINE DOES NOT USE INTERRUPTS AND STEP 1 WRITE DATA
: 2311 1  ! SHOULD NOT ENABLE INTERRUPT.
: 2312 1  !
: 2313 1  ! FORMAL PARAMETERS:
: 2314 1  ! - NONE -
: 2315 1  !
: 2316 1  ! IMPLICIT INPUTS:
: 2317 1  !
: 2318 1  ! DATA1 = STEP 1 WRITE DATA
: 2319 1  ! DATA2 = STEP 2 WRITE DATA
: 2320 1  ! DATA3 = STEP 3 WRITE DATA
: 2321 1  ! DATA4 = STEP 4 WRITE DATA
: 2322 1  !
: 2323 1  ! B_MASK = WITCH STEPS WILL BE DONE
: 2324 1  !     #0 1 = STEP 1
: 2325 1  !     #0 3 = STEP 1,2
: 2326 1  !     #0 7 = STEP 1,2,3
: 2327 1  !     #017 = STEP 1,2,3,4
: 2328 1  !
: 2329 1  ! IMPLICIT OUTPUTS:
: 2330 1  !
: 2331 1  ! IF ERROR OR NO STEP IT WILL RETURN
: 2332 1  ! P1-P6, P_MASK
: 2333 1  ! RET_STATUS
: 2334 1  ! COMPLETION CODES:
: 2335 1  !
: 2336 1  ! RET_STATUS GIVES COMPLETION CODE
: 2337 1  !
: 2338 1  ! SIDE EFFECTS:
: 2339 1  !
: 2340 1  ! - NONE -
: 2341 1  ! --
: 2342 2  ! begin
: 2343 2  !
: 2344 2  ! local
: 2345 2  !     N,
: 2346 2  !     MASK,
: 2347 2  !     COUNT,
: 2348 2  !     DATA;
: 2349 2

```

```

!STEP NUMBER
!STEP MASK
!TIME OUT COUNT
!WRITE DATA FOR THE STEP

```

```

: 2350 2
: 2351 2
: 2352 2
: 2353 2
: 2354 2
: 2355 2
: 2356 2
: 2357 2
: 2358 2
: 2359 2
: 2360 2
: 2361 2
: 2362 2
: 2363 2
: 2364 3
: 2365 3
: 2366 4
: 2367 3
: 2368 4
: 2369 4
: 2370 4
: 2371 4
: 2372 4
: 2373 4
: 2374 4
: 2375 4
: 2376 4
: 2377 4
: 2378 4
: 2379 4
: 2380 4
: 2381 4
: 2382 4
: 2383 4
: 2384 4
: 2385 4
: 2386 4
: 2387 4
: 2388 4
: 2389 4
: 2390 4
: 2391 4
: 2392 4
: 2393 5
: 2394 5
: 2395 5
: 2396 5
: 2397 5
: 2398 5
: 2399 5
: 2400 5
: 2401 4
: 2402 4
: 2403 5
: 2404 4
: 2405 5
: 2406 5

```

```

: INIT THE AZTEC
:
: THE FOLLOWING LOOP WILL DO STEP 1 THRU 4 AS GIVEN BY B_MASK
: INPUT SELECTING APPROPRIATE DATA INPUT FOR STEP WRITES. IF
: ERROR IN SA REGISTER P1 - P4 AND P_MASK WILL BE SUPPLIED FOR
: ERROR REPORT. ONLY SA DATA FOR THE FINAL WRITE STEP IS PRESERVED.
:
: MASK = %b'0001';           ! STEP MASK
: WRT_RC25 (RCIP, ALL_ONES); ! START INIT
: DELAY (2);                 ! WAIT FOR COMPLETION
:
: incru N from 0 to 4 do
:   begin
:     if (.N eqlu 0 or BIT_TST (.B_MASK, .MASK))      ! TEST FOR STEP NUMBER
:     then
:       begin
:         !
:         selectoneu .N of                             ! SELECT CORRECT WRITE
:         set                                           !
:         [0] : DATA = ALL_ONES;                       !
:         [1] : DATA = .DATA1;                         ! DATA FOR STEP WRITES
:         [2] : DATA = .DATA2;
:         [3] : DATA = .DATA3;
:         [4] : DATA = .DATA4;
:         tes;
:       !
:       incru COUNT from 0 to 20 do                     ! TIME OUT WAIT LOOP
:       begin
:         DELAY (333);                                  ! DELAY 1 SEC. APPROX.
:         RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];
:         if .RC25_DATA [RCSA, RCSA_STEP] eqlu .MASK then exitloop;
:         RET_STATUS = CTO_CODE;
:         BREAK;                                         ! WATCH FOR CONTROL C.
:       end;
:       if (.RC25_DATA [RCSA, RCSA_ER] nequ ZERO)      ! IF RCSA ERROR BIT SET
:       then
:         begin
:           RET_STATUS = PFE_CODE;                       ! THEN FATAL ERROR

```



```

: 2407 5      exitloop;
: 2408 5      end
: 2409 4      else
: 2410 5      begin
: 2411 5              ! OTHERWISE, CHECK THE
: 2412 5              ! STEP BITS.
: 2413 6      if (.RC25_DATA [RCSA, RCSA_STEP] nequ .MASK)
: 2414 5      then
: 2415 6              ! IF ERROR THEN SUPPLY
: 2416 6              ! P1- P6
: 2417 6              begin
: 2418 6                  P_MASK = 2;
: 2419 6                  P1 = FMT3;
: 2420 6                  P2 = ADAPT;
: 2421 6                  P4 = (.RC25_ADDR) * 2;
: 2422 6                  P5 = .RC25_DATA [RCSA, RC_ALL];
: 2423 6                  P6 = .MASK;
: 2424 6                  MSGADR = MSG_14;
: 2425 5                  ! AND RETURN
: 2426 6                  return .RET_STATUS;
: 2427 6                  ! TRUE STATUS.
: 2428 6              end
: 2429 5      else
: 2430 6      begin
: 2431 6          RET_STATUS = PAS_CODE;
: 2432 5          ! IF NOT, RETURN GOOD STATUS
: 2433 6      end;
: 2434 4      end;
: 2435 4      if .N nequ ZERO
: 2436 4      then
: 2437 5      begin
: 2438 5          MASK = .MASK+1;
: 2439 5          ! MOVE MASK BIT
: 2440 5          WRT_RC25 (RCSA, .DATA);
: 2441 4          ! STEP N WRITE DATA TO SA
: 2442 4      end;
: 2443 3      end;
: 2444 3      end;
: 2445 2      if .RET_STATUS
: 2446 2      then
: 2447 2          ! IF TRUE STATUS, THEN
: 2448 2          ! SUPPLY P1-P6 FOR TEST
: 2449 2          ! MODULE FOR ERROR PRINTOUT.
: 2450 3      begin
: 2451 3          P_MASK = 2;
: 2452 3          P1 = FMT3;
: 2453 3          P2 = ADAPT;
: 2454 3          P4 = (.RC25_ADDR) * 2;
: 2455 3          P5 = .RC25_DATA [RCSA, RC_ALL];
: 2456 3          P6 = .MASK;
: 2457 3          MSGADR = MSG_14;
: 2458 3          return .RET_STATUS;
: 2459 3      end
: 2460 2      else
: 2461 2          return .RET_STATUS;
: 2462 2          ! OTHERWISE GOOD STATUS.
: 2463 2      end;
: 2464 1      end;

```

F11

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC INITIALIZATION BY POLING

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Blues-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0135
Page 40
(17)

000004	024646			CMP	-(SP), -(SP)		
000006	012705	000001		MOV	#1, R5	; *, MASK	2359
000012	012700	177777		MOV	#-1, R0	; *, RCM.REG	2360
000016	010077	000000G		MOV	R0, @RC25.ADDR	; RCM.REG, *	
000022	012701	000002		MOV	#2, R1	; *, \$\$TMP2	2361
000026	001411		1\$:	BEQ	4\$		
000030	016700	000000G		MOV	L\$DLY, R0	; *, \$\$TMP1	
000034	001404			BEQ	3\$		
000036	005066	000002	2\$:	CLR	2(SP)	; \$\$TMP	
000042	005300			DEC	R0	; \$\$TMP1	
000044	001374			BNE	2\$		
000046	005301		3\$:	DEC	R1	; \$\$TMP2	
000050	000766			BR	1\$		
000052	005003		4\$:	CLR	R3	; N	2363
000054	005703		5\$:	TST	R3	; N	2366
000056	001410			BEQ	6\$		
000060	005000			CLR	R0		
000062	156700	000000G		BISB	B.MASK, R0		
000066	010501			MOV	R5, R1	; MASK, *	
000070	005101			COM	R1		
000072	040100			BIC	R1, R0		
000074	020005			CMP	R0, R5	; *, MASK	
000076	001170			BNE	20\$		
000100	010300		6\$:	MOV	R3, R0	; N, *	2371
000102	001003			BNE	7\$		2374
000104	012702	177777		MOV	#-1, R2	; *, DATA	2375
000110	000427			BR	11\$		2371
000112	020027	000001	7\$:	CMP	R0, #1		2377
000116	001003			BNE	8\$		
000120	016702	000000G		MOV	DATA1, R2	; *, DATA	2378
000124	000421			BR	11\$		2371
000126	020027	000002	8\$:	CMP	R0, #2		2380
000132	001003			BNE	9\$		
000134	016702	000000G		MOV	DATA2, R2	; *, DATA	2381
000140	000413			BR	11\$		2371
000142	020027	000003	9\$:	CMP	R0, #3		2383
000146	001003			BNE	10\$		
000150	016702	000000G		MOV	DATA3, R2	; *, DATA	2384
000154	000405			BR	11\$		2371
000156	020027	000004	10\$:	CMP	R0, #4		2386
000162	001002			BNE	11\$		
000164	016702	000000G		MOV	DATA4, R2	; *, DATA	2387
000170	005004		11\$:	CLR	R4	; COUNT	2392
000172	012701	000515	12\$:	MOV	#515, R1	; *, \$\$TMP2	2394
000176	001411		13\$:	BEQ	16\$		
000200	016700	000000G		MOV	L\$DLY, R0	; *, \$\$TMP1	
000204	001404			BEQ	15\$		
000206	005066	000002	14\$:	CLR	2(SP)	; \$\$TMP	
000212	005300			DEC	R0	; \$\$TMP1	
000214	001374			BNE	14\$		
000216	005301		15\$:	DEC	R1	; \$\$TMP2	
000220	000766			BR	13\$		
000222	016700	000000G	16\$:	MOV	RC25.ADDR, R0		2395
000226	016016	000002		MOV	2(R0), (SP)	; *, RC.REG	
000232	011667	000002G		MOV	(SP), RC25.DATA+2	; RC.REG, *	
000236	010501			MOV	R5, R1	; MASK, *	2397
000240	011600			MOV	(SP), R0	; RC25.DATA+2, *	

G11

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC INITIALIZATION BY POLING

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC2.B16;1

SEQ 0136
Page 41
(17)

000242	006200			ASR	R0		
000244	006200			ASR	R0		
000246	006200			ASR	R0		
000250	000300			SWAB	R0		
000252	042700	177760		BIC	#177760,R0		
000256	020001			CMP	R0,R1		
000260	001410			BEQ	17\$		
000262	012767	000011	000000G	MOV	#11,RET.STATUS	:	2399
000270	104422			TRAP	22		
000272	005204			INC	R4	: COUNT	2392
000274	020427	000024		CMP	R4,#24	: COUNT,*	
000300	101734			BLOS	12\$		
000302	032767	100000	000002G	BIT	#100000,RC25.DATA+2	:	2403
000310	001404			BEQ	18\$		
000312	012767	000021	000000G	MOV	#21,RET.STATUS	:	2406
000320	000465			BR	21\$:	2405
000322	010501			MOV	R5,R1	: MASK,*	2413
000324	016700	000002G		MOV	RC25.DATA+2,R0		
000330	006200			ASR	R0		
000332	006200			ASR	R0		
000334	006200			ASR	R0		
000336	000300			SWAB	R0		
000340	042700	177760		BIC	#177760,R0		
000344	020001			CMP	R0,R1		
000346	001432			BEQ	19\$		
000350	112767	000002	000000G	MOVB	#2,P.MASK	:	2416
000356	012767	000000G	000000G	MOV	#FMT3,P1	:	2417
000364	012767	000001	000000G	MOV	#1,P2	:	2418
000372	016700	000000G		MOV	RC25.ADDR,R0	:	2419
000376	062700	000002		ADD	#2,R0		
000402	010067	000000G		MOV	R0,P4		
000406	016767	000002G	000000G	MOV	RC25.DATA+2,P5	: MASK,*	2420
000414	010567	000000G		MOV	R5,P6	: MASK,*	2421
000420	012767	000000G	000000G	MOV	#MSG.14,MSGADR	:	2422
000426	016700	000000G		MOV	RET.STATUS,R0	:	2415
000432	000460			BR	23\$		
000434	005067	000000G		CLR	RET.STATUS	:	2427
000440	005703			TST	R3	: N	2432
000442	001406			BEQ	20\$		
000444	006305			ASL	R5	: MASK	2435
000446	010204			MOV	R2,R4	: DATA,RCM.REG	2436
000450	016700	000000G		MOV	RC25.ADDR,R0		
000454	010460	000002		MOV	R4,2(R0)	: RCM.REG,*	
000460	005203			INC	R3	: N	2363
000462	020327	000004		CMP	R3,#4	: N,*	
000466	101002			BHI	21\$		
000470	000167	177360		JMP	5\$		
000474	032767	000001	000000G	BIT	#1,RET.STATUS	:	2443
000502	001432			BEQ	22\$		
000504	112767	000002	000000G	MOVB	#2,P.MASK	:	2446
000512	012767	000000G	000000G	MOV	#FMT3,P1	:	2447
000520	012767	000001	000000G	MOV	#1,P2	:	2448
000526	016700	000000G		MOV	RC25.ADDR,R0	:	2449
000532	062700	000002		ADD	#2,R0		
000536	010067	000000G		MOV	R0,P4		
000542	016767	000002G	000000G	MOV	RC25.DATA+2,P5	: MASK,*	2450
000550	010567	000000G		MOV	R5,P6	: MASK,*	2451

H11

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC INITIALIZATION BY POLING

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0137
Page 42
(17)

000554	012767	000000G	000000G	MOV	#MSG.14,MSGADR	;	2452
000562	016700	000000G		MOV	RET.STATUS,RO	;	2456
000566	000402			BR	23#		
000570	016700	000000G		22#:	MOV	RET.STATUS,RO	
000574	022626			23#:	CMP	(SP)*,(SP)*	
000576	000207			RTS	PC		2295

: Routine Size: 192 words, Routine Base: AB\$CODE + 1636
: Maximum stack depth per invocation: 10 words

: 2459 1


```

: 2517 2          end;
: 2518 2
: 2519 2
: 2520 2          !
: 2521 2          ! DEFINE THE ADDRESS LOCATIONS OF THE HEAD_AREA, RECEIVE_RING
: 2522 2          ! AND SEND_RING.
: 2523 2          !
: 2524 2          HEAD AREA = COM_AREA;          ! DEFINE THE HEADER AREA
: 2525 2          RECEIVE_RING = COM_AREA [REC_BASE]; ! DEFINE THE RESPONSE RING AREA
: 2526 2          SEND_RING = COM_AREA [SND_BASE];  ! DEFINE THE COMMAND RING AREA
: 2527 2
: 2528 2          incru I from 0 to 3 do          ! CLEAR THE HEADER AREA
: 2529 2          HEAD_AREA [.I, WORD_REF] = ZERO; !
: 2530 2
: 2531 2          !
: 2532 2          ! LOAD UP THE COMMAND RING DESCRIPTORS WITH AN ENVELOPE ADDRESS,
: 2533 2          ! DEFINE THE "FLAG BIT" TO = 1 (INTERRUPT REQUESTED), DEFINE THE
: 2534 2          ! "OWNERSHIP BIT" TO ZERO (OWNED BY HOST) AND LOAD THE RESERVED
: 2535 2          ! FIELD WITH ZERO.
: 2536 2          !
: 2537 2          !
: 2538 2          !
: 2539 2          !
: 2540 2          !
: 2541 2          !
: 2542 2          !
: 2543 2          !
: 2544 2          !
: 2545 2          !
: 2546 2          !
: 2547 2          !
: 2548 2          !
: 2549 2          !
: 2550 2          !
: 2551 2          !
: 2552 2          !
: 2553 2          !
: 2554 2          !
: 2555 2          !
: 2556 2          !
: 2557 2          !
: 2558 2          !
: 2559 2          !
: 2560 2          !
: 2561 2          !
: 2562 2          !
: 2563 2          !
: 2564 2          !
: 2565 2          !
: 2566 2          !
: 2567 2          !
: 2568 2          !
: 2569 2          !
: 2570 2          !
: 2571 2          !
: 2572 2          !
: 2573 1

```

				.SBTTL	INIT.COM.AREA COMMUNICATION RING INITIALIZATION	
000000	004167	000000G		INIT.COM.AREA::		
				JSR	R1,\$SAVE2	
000004	005002			CLR	R2	; I 2463
000006	005001		1\$:	CLR	R1	; J 2508
000010	010200		2\$:	MOV	R2,R0	; I,* 2510
000012	006300			ASL	R0	
000014	060100			ADD	R1,R0	; J,* 2512
000016	006300			ASL	R0	
000020	005760	000000G		TST	COM.AREA(R0)	
000024	001406			BEQ	3\$	
000026	012767	000000G 000000G		MOV	#FAL.CODE,RET.STATUS	; 2515
000034	016700	000000G		MOV	RET.STATUS,R0	; 2514
000040	000207			RTS	PC	
000042	005201		3\$:	INC	R1	; J 2510
000044	020127	000001		CMP	R1,#1	; J,*
000050	101757			BLOS	2\$	
000052	005202			INC	R2	; I 2508
000054	020227	000037		CMP	R2,#37	; I,*
000060	101752			BLOS	1\$	
000062	012767	000000G 000000G		MOV	#COM.AREA,HEAD.AREA	; 2523
000070	012767	000010G 000000G		MOV	#COM.AREA*10,RECEIVE.RING	; 2524
000076	012767	000110G 000000G		MOV	#COM.AREA*110,SEND.RING	; 2525
000104	005000			CLR	R0	; I 2527
000106	010001		4\$:	MOV	R0,R1	; I,* 2528
000110	006301			ASL	R1	
000112	066701	000000G		ADD	HEAD.AREA,R1	
000116	005011			CLR	(R1)	
000120	005200			INC	R0	; I 2527
000122	020027	000003		CMP	R0,#3	; I,*
000126	101767			BLOS	4\$	
000130	005001			CLR	R1	; I 2537
000132	010102		5\$:	MOV	R1,R2	; I,* 2539
000134	006302			ASL	R2	
000136	006302			ASL	R2	
000140	066702	000000G		ADD	SEND.RING,R2	
000144	010146			MOV	R1,-(SP)	; I,*
000146	012746	000054		MOV	#54,-(SP)	
000152	004767	000000G		JSR	PC,BL\$MUL	
000156	062700	000004G		ADD	#SND.ENVELOPE*4,R0	
000162	010012			MOV	R0,(R2)	
000164	010100			MOV	R1,R0	; I,* 2540
000166	006300			ASL	R0	
000170	006300			ASL	R0	
000172	066700	000000G		ADD	SEND.RING,R0	
000176	142760	000003 000002		BICB	#3,2(R0)	
000204	010100			MOV	R1,R0	; I,* 2541
000206	006300			ASL	R0	
000210	006300			ASL	R0	
000212	066700	000000G		ADD	SEND.RING,R0	
000216	142760	000074 000002		BICB	#74,2(R0)	
000224	010100			MOV	R1,R0	; I,* 2542
000226	006300			ASL	R0	
000230	006300			ASL	R0	
000232	066700	000000G		ADD	SEND.RING,R0	

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
COMMUNICATION RING INITIALIZATION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

000236	042760	037700	000002	BIC	#37700,2(R0)		
000244	010100			MOV	R1,R0	; I,*	2543
000246	006300			ASL	R0		
000250	006300			ASL	R0		
000252	066700	000000G		ADD	SEND.RING,R0		
000256	042760	040000	000002	BIC	#40000,2(R0)		
000264	010100			MOV	R1,R0	; I,*	2544
000266	006300			ASL	R0		
000270	006300			ASL	R0		
000272	066700	000000G		ADD	SEND.RING,R0		
000276	042760	100000	000002	BIC	#100000,2(R0)		
000304	022626			CMP	(SP),,(SP),		2538
000306	005201			INC	R1	; I	2537
000310	020127	000017		CMP	R1,#17	; I,*	
000314	101706			BLOS	5\$		
000316	005002			CLR	R2	; I	2554
000320	010201			MOV	R2,R1	; I,*	2556
000322	006301			ASL	R1		
000324	006301			ASL	R1		
000326	066701	000000G		ADD	RECEIVE.RING,R1		
000332	010200			MOV	R2,R0	; I,*	
000334	000300			SWAB	R0		
000336	106000			RORB	R0		
000340	006000			ROR	R0		
000342	006000			ROR	R0		
000344	142700	000077		BICB	#77,R0		
000350	062700	000004G		ADD	#REC.ENVELOPE+4,R0		
000354	010011			MOV	R0,(R1)		
000356	010200			MOV	R2,R0	; I,*	2557
000360	006300			ASL	R0		
000362	006300			ASL	R0		
000364	066700	000000G		ADD	RECEIVE.RING,R0		
000370	142760	000003	000002	BICB	#3,2(R0)		
000376	010200			MOV	R2,R0	; I,*	2558
000400	006300			ASL	R0		
000402	006300			ASL	R0		
000404	066700	000000G		ADD	RECEIVE.RING,R0		
000410	142760	000074	000002	BICB	#74,2(R0)		
000416	010200			MOV	R2,R0	; I,*	2559
000420	006300			ASL	R0		
000422	006300			ASL	R0		
000424	066700	000000G		ADD	RECEIVE.RING,R0		
000430	042760	037700	000002	BIC	#37700,2(R0)		
000436	010200			MOV	R2,R0	; I,*	2560
000440	006300			ASL	R0		
000442	006300			ASL	R0		
000444	066700	000000G		ADD	RECEIVE.RING,R0		
000450	042760	040000	000002	BIC	#40000,2(R0)		
000456	010200			MOV	R2,R0	; I,*	2561
000460	006300			ASL	R0		
000462	006300			ASL	R0		
000464	066700	000000G		ADD	RECEIVE.RING,R0		
000470	052760	100000	000002	BIS	#100000,2(R0)		
000476	005202			INC	R2	; I	2554
000500	020227	000017		CMP	R2,#17	; I,*	
000504	101705			BLOS	6\$		
000506	005001			CLR	R1	; I	2568

6\$:

M11

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
COMMUNICATION RING INITIALIZATION

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC2.B16;1

SEQ 0142
Page 47
(18)

000510	010100		74:	MOV	R1,R0				
000512	000300			SWAB	R0				
000514	106000			RORB	R0				
000516	006000			ROR	R0				
000520	006000			ROR	R0				
000522	142700	000077		BICB	#77,R0				
000526	012760	000074	000000G	MOV	#74,REC.ENVELOPE(R0)				
000534	005201			INC	R1				
000536	020127	000017		CMP	R1,#17				
000542	101762			BLOS	74				
000544	005067	000000G		CLR	RET.STATUS				
000550	016700	000000G		MOV	RET.STATUS,R0				
000554	000207			RTS	PC				

: Routine Size: 183 words, Routine Base: AB\$CODE * 2436
: Maximum stack depth per invocation: 6 words

: 2574 1

```

: 2575 1  *subttl 'AZTEC GLOBAL ROUTINE'
: 2576 1  !
: 2577 1
: 2578 1  global routine EX_SUP_PRG =
: 2579 1
: 2580 1  !..
: 2581 1  ! FUNCTIONAL DESCRIPTION :
: 2582 1  !
: 2583 1  !     THIS COMMAND CAUSES THE SERVER TO TRANSFER THE PROGRAM FROM HOST
: 2584 1  !     MEMORY TO AN AREA IN THE CONTROLLER AND START ITS EXECUTION.
: 2585 1  !     THE HOST SUPPLIES THE ADDRESS AND LENGTH IN BYTES OF A BUFFER
: 2586 1  !     CONTAINING THE PROGRAM WHICH WAS MADE INTO ONE CONTIGUOUS
: 2587 1  !     VECTOR OF DATA AS FAR AS HOST IS CONCERNED. THIS COMMAND IS
: 2588 1  !     ONLY LEGAL WHEN THE SERVER IS IN THE IDLE STATE AND RETURN
: 2589 1  !     OF A SUCCESSFUL END PACKET PUTS THE SERVER INTO ACTIVE STATE.
: 2590 1  !
: 2591 1  ! FORMAL PARAMETERS :
: 2592 1  !
: 2593 1  ! IMPLICIT INPUTS :  BUF_DESCRPTR, BYTE_COUNT
: 2594 1  !
: 2595 1  ! IMPLICIT OUTPUTS : RET_STATUS AS RECEIVED FROM REC_STATUS ROUTINE
: 2596 1  !                   IS RETURNED TO CALLING ROUTINE
: 2597 1  ! SIDE EFFECTS :
: 2598 1  !
: 2599 1  ! ..
: 2600 1
: 2601 2  begin
: 2602 2
: 2603 2  local
: 2604 2  TEMP;
: 2605 2
: 2606 2
: 2607 2  ! CLEAR THE FLAG HERE TO INSURE THE DETECTION OF THE INTERRUPT.
: 2608 2
: 2609 2  I_AM_NEX = ZERO;
: 2610 2
: 2611 2
: 2612 2  ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
: 2613 2
: 2614 2  SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_ESP;
: 2615 2  SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;
: 2616 2  SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0;
: 2617 2  SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 2;
: 2618 2
: 2619 2  ! DUP COMMAND ENVELOPE FIELD DEFINITION
: 2620 2
: 2621 2  SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;
: 2622 2  SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;
: 2623 2  SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = ZERO;
: 2624 2  SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;
: 2625 2  SND_ENVELOPE [.CMD_SLOT, OP_CODE] = OP_ESP;
: 2626 2  SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;
: 2627 2  SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;
: 2628 2  SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT;      ! BYTE COUNT LOW WORD
: 2629 2  SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO;          ! BYTE COUNT HIGH WORD
: 2630 2  SND_ENVELOPE [.CMD_SLOT, BD_0] = .BUF_DESCRPTR;    ! BUFFER DESCRIPTOR WORD 0
: 2631 2  SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO;             ! BUFFER DESCRIPTOR WORD 1

```

```

: 2632 2      SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO;      ! BUFFER DESCRIPTOR WORD 2
: 2633 2      SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO;      ! BUFFER DESCRIPTOR WORD 3
: 2634 2      SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO;      ! BUFFER DESCRIPTOR WORD 4
: 2635 2      SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO;      ! BUFFER DESCRIPTOR WORD 5
: 2636 2      SND_ENVELOPE [.CMD_SLOT, OBD_0] = ZERO;     ! BUFFER DESCRIPTOR WORD 0
: 2637 2      SND_ENVELOPE [.CMD_SLOT, OBD_1] = ZERO;     ! BUFFER DESCRIPTOR WORD 1
: 2638 2      SND_ENVELOPE [.CMD_SLOT, OBD_2] = ZERO;     ! BUFFER DESCRIPTOR WORD 2
: 2639 2      SND_ENVELOPE [.CMD_SLOT, OBD_3] = ZERO;     ! BUFFER DESCRIPTOR WORD 3
: 2640 2      SND_ENVELOPE [.CMD_SLOT, OBD_4] = ZERO;     ! BUFFER DESCRIPTOR WORD 4
: 2641 2      SND_ENVELOPE [.CMD_SLOT, OBD_5] = ZERO;     ! BUFFER DESCRIPTOR WORD 5
: 2642 2
: 2643 2      ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
: 2644 2
: 2645 2      SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
: 2646 2
: 2647 2      ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
: 2648 2
: 2649 2      TEMP = .RC25_ADDR [RCIP, RC_ALL];
: 2650 2
: 2651 2      ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
: 2652 2
: 2653 2      GET_CMD_SLOT ();
: 2654 2
: 2655 2
: 2656 2      ! CHECK THE END PACKET FOR GOOD STATUS
: 2657 2
: 2658 2      return REC_STATUS ();      !RETURN THE STATUS
: 2659 1      end;

```

```

000000 005746      .SBTTL EX.SUP.PRG AZTEC GLOBAL ROUTINE
EX.SUP.PRG::
000002 005067 000000G      TST      -(SP)      ;      2578
000006 016746 000000G      CLR      I.AM.NEX      ;      2609
000012 012746 000054      MOV      CMD_SLOT, -(SP)      ;      2614
000016 004767 000000G      MOV      #54, -(SP)
000022 012760 000050 000000G      JSR      PC, BL#MUL
000030 016716 000000G      MOV      #50, SND.ENVELOPE(R0)
000034 012746 000054      MOV      CMD_SLOT, (SP)      ;      2615
000040 004767 000000G      MOV      #54, -(SP)
000044 142760 000017 000002G      JSR      PC, BL#MUL
000052 152760 000001 000002G      BICB     #17, SND.ENVELOPE+2(R0)
000060 016716 000000G      BISB     #1, SND.ENVELOPE+2(R0)
000064 012746 000054      MOV      CMD_SLOT, (SP)      ;      2616
000070 004767 000000G      MOV      #54, -(SP)
000074 142760 000360 000002G      JSR      PC, BL#MUL
000102 016716 000000G      BICB     #360, SND.ENVELOPE+2(R0)
000106 012746 000054      MOV      CMD_SLOT, (SP)      ;      2617
000112 004767 000000G      MOV      #54, -(SP)
000116 112760 000002 000003G      JSR      PC, BL#MUL
000124 016716 000000G      MOVVB    #2, SND.ENVELOPE+3(R0)
000130 012746 000054      MOV      CMD_SLOT, (SP)      ;      2621
000134 004767 000000G      MOV      #54, -(SP)
000140 016760 000000G 000004G      JSR      PC, BL#MUL
000146 016716 000000G      MOV      CMD_REF, SND.ENVELOPE+4(R0)
000152 012746 000054      MOV      CMD_SLOT, (SP)      ;      2622
MOV      #54, -(SP)

```

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0145
Page 50
(19)

000156	004767	000000G		JSR	PC, BL \$MUL		
000162	005060	000006G		CLR	SND. ENVELOPE + 6(R0)		
000166	016716	000000G		MOV	CMD. SLOT, (SP)	:	2623
000172	012746	000054		MOV	#54, -(SP)		
000176	004767	000000G		JSR	PC, BL \$MUL		
000202	005060	000010G		CLR	SND. ENVELOPE + 10(R0)		
000206	016716	000000G		MOV	CMD. SLOT, (SP)	:	2624
000212	012746	000054		MOV	#54, -(SP)		
000216	004767	000000G		JSR	PC, BL \$MUL		
000222	005060	000012G		CLR	SND. ENVELOPE + 12(R0)		
000226	016716	000000G		MOV	CMD. SLOT, (SP)	:	2625
000232	012746	000054		MOV	#54, -(SP)		
000236	004767	000000G		JSR	PC, BL \$MUL		
000242	112760	000002	000014G	MOVB	#2, SND. ENVELOPE + 14(R0)		
000250	016716	000000G		MOV	CMD. SLOT, (SP)	:	2626
000254	012746	000054		MOV	#54, -(SP)		
000260	004767	000000G		JSR	PC, BL \$MUL		
000264	105060	000015G		CLRB	SND. ENVELOPE + 15(R0)		
000270	016716	000000G		MOV	CMD. SLOT, (SP)	:	2627
000274	012746	000054		MOV	#54, -(SP)		
000300	004767	000000G		JSR	PC, BL \$MUL		
000304	005060	000016G		CLR	SND. ENVELOPE + 16(R0)		
000310	016716	000000G		MOV	CMD. SLOT, (SP)	:	2628
000314	012746	000054		MOV	#54, -(SP)		
000320	004767	000000G		JSR	PC, BL \$MUL		
000324	016760	000000G	000020G	MOV	BYTE. COUNT, SND. ENVELOPE + 20(R0)		
000332	016716	000000G		MOV	CMD. SLOT, (SP)	:	2629
000336	012746	000054		MOV	#54, -(SP)		
000342	004767	000000G		JSR	PC, BL \$MUL		
000346	005060	000022G		CLR	SND. ENVELOPE + 22(R0)		
000352	016716	000000G		MOV	CMD. SLOT, (SP)	:	2630
000356	012746	000054		MOV	#54, -(SP)		
000362	004767	000000G		JSR	PC, BL \$MUL		
000366	016760	000000G	000024G	MOV	BUF. DESCRPTR, SND. ENVELOPE + 24(R0)		
000374	016716	000000G		MOV	CMD. SLOT, (SP)	:	2631
000400	012746	000054		MOV	#54, -(SP)		
000404	004767	000000G		JSR	PC, BL \$MUL		
000410	005060	000026G		CLR	SND. ENVELOPE + 26(R0)		
000414	016716	000000G		MOV	CMD. SLOT, (SP)	:	2632
000420	012746	000054		MOV	#54, -(SP)		
000424	004767	000000G		JSR	PC, BL \$MUL		
000430	005060	000030G		CLR	SND. ENVELOPE + 30(R0)		
000434	016716	000000G		MOV	CMD. SLOT, (SP)	:	2633
000440	012746	000054		MOV	#54, -(SP)		
000444	004767	000000G		JSR	PC, BL \$MUL		
000450	005060	000032G		CLR	SND. ENVELOPE + 32(R0)		
000454	016716	000000G		MOV	CMD. SLOT, (SP)	:	2634
000460	012746	000054		MOV	#54, -(SP)		
000464	004767	000000G		JSR	PC, BL \$MUL		
000470	005060	000034G		CLR	SND. ENVELOPE + 34(R0)		
000474	016716	000000G		MOV	CMD. SLOT, (SP)	:	2635
000500	012746	000054		MOV	#54, -(SP)		
000504	004767	000000G		JSR	PC, BL \$MUL		
000510	005060	000036G		CLR	SND. ENVELOPE + 36(R0)		
000514	016716	000000G		MOV	CMD. SLOT, (SP)	:	2636
000520	012746	000054		MOV	#54, -(SP)		
000524	004767	000000G		JSR	PC, BL \$MUL		

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0146
Page 51
(19)

000530	005060	000040G	CLR	SND.ENVELOPE+40(R0)		
000534	016716	000000G	MOV	CMD.SLOT,(SP)	:	2637
000540	012746	000054	MOV	#54,-(SP)		
000544	004767	000000G	JSR	PC,BL#MUL		
000550	005060	000042G	CLR	SND.ENVELOPE+42(R0)		
000554	016716	000000G	MOV	CMD.SLOT,(SP)	:	2638
000560	012746	000054	MOV	#54,-(SP)		
000564	004767	000000G	JSR	PC,BL#MUL		
000570	005060	000044G	CLR	SND.ENVELOPE+44(R0)		
000574	016716	000000G	MOV	CMD.SLOT,(SP)	:	2639
000600	012746	000054	MOV	#54,-(SP)		
000604	004767	000000G	JSR	PC,BL#MUL		
000610	005060	000046G	CLR	SND.ENVELOPE+46(R0)		
000614	016716	000000G	MOV	CMD.SLOT,(SP)	:	2640
000620	012746	000054	MOV	#54,-(SP)		
000624	004767	000000G	JSR	PC,BL#MUL		
000630	005060	000050G	CLR	SND.ENVELOPE+50(R0)		
000634	016716	000000G	MOV	CMD.SLOT,(SP)	:	2641
000640	012746	000054	MOV	#54,-(SP)		
000644	004767	000000G	JSR	PC,BL#MUL		
000650	005060	000052G	CLR	SND.ENVELOPE+52(R0)		
000654	016700	000000G	MOV	CMD.SLOT,R0	:	2645
000660	006300		ASL	R0		
000662	006300		ASL	R0		
000664	066700	000000G	ADD	SEND.RING,R0		
000670	052760	100000 000002	BIS	#100000,2(R0)		
000676	017766	000000G 000064	MOV	@RC25.ADDR,64(SP)	:	2649
000704	016600	000064	MOV	64(SP),R0	:	
000710	004767	000000V	JSR	PC,GET.CMD.SLOT	:	2653
000714	004767	000000V	JSR	PC,REC.STATUS	:	2658
000720	062706	000066	ADD	#66,SP	:	2578
000724	000207		RTS	PC	:	

: Routine Size: 235 words, Routine Base: AB#CODE + 3214
: Maximum stack depth per invocation: 28 words

: 2660 1
: 2661 1 !
: 2662 1

: 2663 1
: 2664 1
: 2665 1
: 2666 1
: 2667 1
: 2668 1
: 2669 1
: 2670 1
: 2671 1
: 2672 1
: 2673 1
: 2674 1
: 2675 1
: 2676 1
: 2677 1
: 2678 1
: 2679 1
: 2680 1
: 2681 1
: 2682 1
: 2683 1
: 2684 1
: 2685 1
: 2686 1
: 2687 1
: 2688 1
: 2689 1
: 2690 1
: 2691 2
: 2692 2
: 2693 2
: 2694 2
: 2695 2
: 2696 2
: 2697 2
: 2698 2
: 2699 2
: 2700 2
: 2701 2
: 2702 2
: 2703 2
: 2704 2
: 2705 2
: 2706 2
: 2707 2
: 2708 2
: 2709 2
: 2710 2
: 2711 2
: 2712 2
: 2713 2
: 2714 2
: 2715 2
: 2716 2
: 2717 2
: 2718 2
: 2719 2

global routine SEND_DATA =

! **
FUNCTIONAL DESCRIPTION:

THIS IS ONE OF THE DUP COMMANDS TO COMMUNICATE BETWEEN THE INITIATING HOST PROGRAM AND THE REMOTE PROGRAM. THIS COMMAND SPECIFIES HOST BUFFER DESCRIPTOR (START ADDRESS OF BUFFER) AND BYTE COUNT. THE INFORMATION IN THE BUFFER IS READ BY THE REMOTE PROGRAM AND A SEND DATA RESPONSE SENT BACK TO THE HOST TO ACKNOWLEDGE RECEIPT. THIS COMMAND IS ONLY LEGAL WHEN THE SERVER IS IN THE ACTIVE STATE. IF THE REMOTE PROGRAM TERMINATES ABNORMALLY PUTTING THE SERVER BACK IN THE IDLE STATE, OUTSTANDING COMMAND MAY BE LOST.

FORMAL PARAMETERS :

IMPLICIT INPUTS : BUF_DESCRPTR, BYTE COUNT
 H_SADD, E_SADD, BUF_LENGTH WILL BE MADE AVAILABLE TO REMOTE PROGRAM BY THE POINTER IN BUF_DESCRPTR.

IMPLICIT OUTPUTS : RET_STATUS AS RECEIVED FROM REC_STATUS
 WILL BE RETURNED TO CALLING ROUTINE.

SIDE EFFECTS :

--

begin

local
 TEMP;

! CLEAR THE FLAG HERE TO INSURE THE DETECTION OF THE INTERRUPT.

I_AM_NEX = ZERO;

! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION

SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_SED;
SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;
SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0;
SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 2;

! DUP COMMAND ENVELOPE FIELD DEFINITION

SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;
SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;
SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = ZERO;
SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;
SND_ENVELOPE [.CMD_SLOT, OP_CODE] = OP_SED;
SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;
SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;
SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT; ! BYTE COUNT LOW WORD
SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO; ! BYTE COUNT HIGH WORD

ZRCFB2 V03.0	MISCELLANEOUS SECTIONS AZTEC GLOBAL ROUTINE	27-Mar-1985 15:23:34 11-Jan-1985 08:19:19	VAX-11 Bliss-16 V4.0-579 USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1
000202	005060 000010G	CLR	SND.ENVELOPE+10(R0)
000206	016716 000000G	MOV	CMD.SLOT,(SP)
000212	012746 000054	MOV	#54,-(SP)
000216	004767 000000G	JSR	PC,BL#MUL
000222	005060 000012G	CLR	SND.ENVELOPE+12(R0)
000226	016716 000000G	MOV	CMD.SLOT,(SP)
000232	012746 000054	MOV	#54,-(SP)
000236	004767 000000G	JSR	PC,BL#MUL
000242	112760 000004 000014G	MOV#B	#4,SND.ENVELOPE+14(R0)
000250	016716 000000G	MOV	CMD.SLOT,(SP)
000254	012746 000054	MOV	#54,-(SP)
000260	004767 000000G	JSR	PC,BL#MUL
000264	105060 000015G	CLRB	SND.ENVELOPE+15(R0)
000270	016716 000000G	MOV	CMD.SLOT,(SP)
000274	012746 000054	MOV	#54,-(SP)
000300	004767 000000G	JSR	PC,BL#MUL
000304	005060 000016G	CLR	SND.ENVELOPE+16(R0)
000310	016716 000000G	MOV	CMD.SLOT,(SP)
000314	012746 000054	MOV	#54,-(SP)
000320	004767 000000G	JSR	PC,BL#MUL
000324	016760 000000G 000020G	MOV	BYTE.COUNT,SND.ENVELOPE+20(R0)
000332	016716 000000G	MOV	CMD.SLOT,(SP)
000336	012746 000054	MOV	#54,-(SP)
000342	004767 000000G	JSR	PC,BL#MUL
000346	005060 000022G	CLR	SND.ENVELOPE+22(R0)
000352	016716 000000G	MOV	CMD.SLOT,(SP)
000356	012746 000054	MOV	#54,-(SP)
000362	004767 000000G	JSR	PC,BL#MUL
000366	016760 000000G 000024G	MOV	BUF.DESCRPTR,SND.ENVELOPE+24(R0)
000374	016716 000000G	MOV	CMD.SLOT,(SP)
000400	012746 000054	MOV	#54,-(SP)
000404	004767 000000G	JSR	PC,BL#MUL
000410	005060 000026G	CLR	SND.ENVELOPE+26(R0)
000414	016716 000000G	MOV	CMD.SLOT,(SP)
000420	012746 000054	MOV	#54,-(SP)
000424	004767 000000G	JSR	PC,BL#MUL
000430	005060 000030G	CLR	SND.ENVELOPE+30(R0)
000434	016716 000000G	MOV	CMD.SLOT,(SP)
000440	012746 000054	MOV	#54,-(SP)
000444	004767 000000G	JSR	PC,BL#MUL
000450	005060 000032G	CLR	SND.ENVELOPE+32(R0)
000454	016716 000000G	MOV	CMD.SLOT,(SP)
000460	012746 000054	MOV	#54,-(SP)
000464	004767 000000G	JSR	PC,BL#MUL
000470	005060 000034G	CLR	SND.ENVELOPE+34(R0)
000474	016716 000000G	MOV	CMD.SLOT,(SP)
000500	012746 000054	MOV	#54,-(SP)
000504	004767 000000G	JSR	PC,BL#MUL
000510	005060 000036G	CLR	SND.ENVELOPE+36(R0)
000514	016700 000000G	MOV	CMD.SLOT,R0
000520	006300	ASL	R0
000522	006300	ASL	R0
000524	066700 000000G	ADD	SEND.RING,R0
000530	052760 100000 000002	BIS	#100000,2(R0)
000536	017766 000000G 000050	MOV	@RC25.ADDR,50(SP)
000544	016600 000050	MOV	50(SP),R0
000550	004767 000000V	JSR	PC,GET.CMD.SLOT

H12

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0150
Page 55
(20)

000554 004767 000000V
000560 062706 000052
000564 000207

JSR PC,REC.STATUS
ADD #52,SP
RTS PC

;
;

2741
2663

: Routine Size: 187 words, Routine Base: AB#CODE + 4142
: Maximum stack depth per invocation: 22 words

: 2743 1
: 2744 1 !
: 2745 1

ZRCFB2
V03.0MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE27-Mar-1985 15:23:34
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1SEQ 0151
Page 56
(21)

```

: 2746 1 global routine REC_DATA =
: 2747 1
: 2748 1
: 2749 1
: 2750 1
: 2751 1
: 2752 1
: 2753 1
: 2754 1
: 2755 1
: 2756 1
: 2757 1
: 2758 1
: 2759 1
: 2760 1
: 2761 1
: 2762 1
: 2763 1
: 2764 1
: 2765 1
: 2766 1
: 2767 1
: 2768 1
: 2769 1
: 2770 1
: 2771 1
: 2772 2
: 2773 2
: 2774 2
: 2775 2
: 2776 2
: 2777 2
: 2778 2
: 2779 2
: 2780 2
: 2781 2
: 2782 2
: 2783 2
: 2784 2
: 2785 2
: 2786 2
: 2787 2
: 2788 2
: 2789 2
: 2790 2
: 2791 2
: 2792 2
: 2793 2
: 2794 2
: 2795 2
: 2796 2
: 2797 2
: 2798 2
: 2799 2
: 2800 2
: 2801 2
: 2802 2

```

global routine REC_DATA =

```

: 2749 1
: 2750 1
: 2751 1
: 2752 1
: 2753 1
: 2754 1
: 2755 1
: 2756 1
: 2757 1
: 2758 1
: 2759 1
: 2760 1
: 2761 1
: 2762 1
: 2763 1
: 2764 1
: 2765 1
: 2766 1
: 2767 1
: 2768 1
: 2769 1
: 2770 1
: 2771 1
: 2772 2
: 2773 2
: 2774 2
: 2775 2
: 2776 2
: 2777 2
: 2778 2
: 2779 2
: 2780 2
: 2781 2
: 2782 2
: 2783 2
: 2784 2
: 2785 2
: 2786 2
: 2787 2
: 2788 2
: 2789 2
: 2790 2
: 2791 2
: 2792 2
: 2793 2
: 2794 2
: 2795 2
: 2796 2
: 2797 2
: 2798 2
: 2799 2
: 2800 2
: 2801 2
: 2802 2

```

FUNCTION DESCRIPTION :

THIS IS ONE OF THE DUP COMMANDS TO COMMUNICATE BETWEEN THE INITIATING HOST PROGRAM AND THE REMOTE PROGRAM. THIS COMMAND SPECIFIES HOST BUFFER DSCRIPTOR (START ADDRESS OF BUFFER) AND BYTE COUNT. THE REMOTE PROGRAM WRITES TO THE BUFFER UPTO THE AMOUNT SPECIFIED BY THE BYTE COUNT AND THEN SENDS A RECEIVE DATA RESPONSE TO THE HOST. THIS COMMAND IS ONLY LEGAL WHEN THE SERVER IS IN THE ACTIVE STATE. IF THE REMOTE PROGRAM TERMINATES ABNORMALLY PUTTING THE SERVER BACK IN THE IDLE STATE, OUTSTANDING COMMANDS MAY BE LOST.

FORMAL PARAMETERS :

IMPLICIT INPUTS : BUF_DESCRPTR, BUF_LENGTH

IMPLICIT OUTPUTS : RET_STATUS IS RETURNED AS RECEIVED FROM REC_STATUS ROUTINE.

SIDE EFFECTS :

```

: 2771 1
: 2772 2
: 2773 2
: 2774 2
: 2775 2
: 2776 2
: 2777 2
: 2778 2
: 2779 2
: 2780 2
: 2781 2
: 2782 2
: 2783 2
: 2784 2
: 2785 2
: 2786 2
: 2787 2
: 2788 2
: 2789 2
: 2790 2
: 2791 2
: 2792 2
: 2793 2
: 2794 2
: 2795 2
: 2796 2
: 2797 2
: 2798 2
: 2799 2
: 2800 2
: 2801 2
: 2802 2

```

```

--
begin
local
TEMP;

: CLEAR THE FLAG HERE TO INSURE THE DETECTION OF THE INTERRUPT.
I_AM_NEX = ZERO;

: UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_RED;
SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;
SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0;
SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 2;

: DUP COMMAND ENVELOPE FIELD DEFINITION
SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;
SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;
SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = ZERO;
SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;
SND_ENVELOPE [.CMD_SLOT, OP_CODE] = OP_RED;
SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;
SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;
SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT; ! BYTE COUNT LOW WORD
SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO; ! BYTE COUNT HIGH WORD
SND_ENVELOPE [.CMD_SLOT, BD_0] = .BUF_DESCRPTR; ! BUFFER DESCRIPTOR WORD 0
SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO; ! BUFFER DESCRIPTOR WORD 1

```

```

: 2803 2 SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO; ! BUFFER DESCRIPTOR WORD 2
: 2804 2 SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO; ! BUFFER DESCRIPTOR WORD 3
: 2805 2 SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO; ! BUFFER DESCRIPTOR WORD 4
: 2806 2 SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO; ! BUFFER DESCRIPTOR WORD 5
: 2807 2
: 2808 2 SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
: 2809 2
: 2810 2 SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
: 2811 2
: 2812 2 READ THE IP REGISTER TO STIMULATE PORT POLLING.
: 2813 2
: 2814 2 TEMP = .RC25_ADDR [RCIP, RC_ALL];
: 2815 2
: 2816 2 GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
: 2817 2
: 2818 2 GET_CMD_SLOT ();
: 2819 2
: 2820 2 CHECK THE END PACKET FOR GOOD STATUS
: 2821 2
: 2822 2 return REC_STATUS (); ! RETURN THE STATUS
: 2823 1 end;
    
```

000000	005746		.SBTTL REC.DATA AZTEC GLOBAL ROUTINE	
			REC.DATA::	
			TST -(SP)	2746
000002	005067	000000G	CLR I.AM.NEX	2780
000006	016746	000000G	MOV CMD_SLOT, -(SP)	2785
000012	012746	000054	MOV #54, -(SP)	
000016	004767	000000G	JSR PC, BL \$MUL	
000022	012760	000034 000000G	MOV #34, SND.ENVELOPE(R0)	
000030	016716	000000G	MOV CMD_SLOT, (SP)	2786
000034	012746	000054	MOV #54, -(SP)	
000040	004767	000000G	JSR PC, BL \$MUL	
000044	142760	000017 000002G	BICB #17, SND.ENVELOPE+2(R0)	
000052	152760	000001 000002G	BISB #1, SND.ENVELOPE+2(R0)	
000060	016716	000000G	MOV CMD_SLOT, (SP)	2787
000064	012746	000054	MOV #54, -(SP)	
000070	004767	000000G	JSR PC, BL \$MUL	
000074	142760	000360 000002G	BICB #360, SND.ENVELOPE+2(R0)	
000102	016716	000000G	MOV CMD_SLOT, (SP)	2788
000106	012746	000054	MOV #54, -(SP)	
000112	004767	000000G	JSR PC, BL \$MUL	
000116	112760	000002 000003G	MOVB #2, SND.ENVELOPE+3(R0)	
000124	016716	000000G	MOV CMD_SLOT, (SP)	2792
000130	012746	000054	MOV #54, -(SP)	
000134	004767	000000G	JSR PC, BL \$MUL	
000140	016760	000000G 000004G	MOV CMD_REF, SND.ENVELOPE+4(R0)	
000146	016716	000000G	MOV CMD_SLOT, (SP)	2793
000152	012746	000054	MOV #54, -(SP)	
000156	004767	000000G	JSR PC, BL \$MUL	
000162	005060	000006G	CLR SND.ENVELOPE+6(R0)	
000166	016716	000000G	MOV CMD_SLOT, (SP)	2794
000172	012746	000054	MOV #54, -(SP)	
000176	004767	000000G	JSR PC, BL \$MUL	
000202	005060	000010G	CLR SND.ENVELOPE+10(R0)	
000206	016716	000000G	MOV CMD_SLOT, (SP)	2795

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0153
Page 58
(21)

000212	012746	000054		MOV	#54,-(SP)		
000216	004767	000000G		JSR	PC,BL#MUL		
000222	005060	000012G		CLR	SND.ENVELOPE+12(R0)		
000226	016716	000000G		MOV	CMD.SLOT,(SP)	:	2796
000232	012746	000054		MOV	#54,-(SP)		
000236	004767	000000G		JSR	PC,BL#MUL		
000242	112760	000005	000014G	MOV	#5,SND.ENVELOPE+14(R0)		
000250	016716	000000G		MOV	CMD.SLOT,(SP)	:	2797
000254	012746	000054		MOV	#54,-(SP)		
000260	004767	000000G		JSR	PC,BL#MUL		
000264	105060	000015G		CLRB	SND.ENVELOPE+15(R0)		
000270	016716	000000G		MOV	CMD.SLOT,(SP)	:	2798
000274	012746	000054		MOV	#54,-(SP)		
000300	004767	000000G		JSR	PC,BL#MUL		
000304	005060	000016G		CLR	SND.ENVELOPE+16(R0)		
000310	016716	000000G		MOV	CMD.SLOT,(SP)	:	2799
000314	012746	000054		MOV	#54,-(SP)		
000320	004767	000000G		JSR	PC,BL#MUL		
000324	016760	000000G	000020G	MOV	BYTE.COUNT,SND.ENVELOPE+20(R0)		
000332	016716	000000G		MOV	CMD.SLOT,(SP)	:	2800
000336	012746	000054		MOV	#54,-(SP)		
000342	004767	000000G		JSR	PC,BL#MUL		
000346	005060	000022G		CLR	SND.ENVELOPE+22(R0)		
000352	016716	000000G		MOV	CMD.SLOT,(SP)	:	2801
000356	012746	000054		MOV	#54,-(SP)		
000362	004767	000000G		JSR	PC,BL#MUL		
000366	016760	000000G	000024G	MOV	BUF.DESCRPTR,SND.ENVELOPE+24(R0)		
000374	016716	000000G		MOV	CMD.SLOT,(SP)	:	2802
000400	012746	000054		MOV	#54,-(SP)		
000404	004767	000000G		JSR	PC,BL#MUL		
000410	005060	000026G		CLR	SND.ENVELOPE+26(R0)		
000414	016716	000000G		MOV	CMD.SLOT,(SP)	:	2803
000420	012746	000054		MOV	#54,-(SP)		
000424	004767	000000G		JSR	PC,BL#MUL		
000430	005060	000030G		CLR	SND.ENVELOPE+30(R0)		
000434	016716	000000G		MOV	CMD.SLOT,(SP)	:	2804
000440	012746	000054		MOV	#54,-(SP)		
000444	004767	000000G		JSR	PC,BL#MUL		
000450	005060	000032G		CLR	SND.ENVELOPE+32(R0)		
000454	016716	000000G		MOV	CMD.SLOT,(SP)	:	2805
000460	012746	000054		MOV	#54,-(SP)		
000464	004767	000000G		JSR	PC,BL#MUL		
000470	005060	000034G		CLR	SND.ENVELOPE+34(R0)		
000474	016716	000000G		MOV	CMD.SLOT,(SP)	:	2806
000500	012746	000054		MOV	#54,-(SP)		
000504	004767	000000G		JSR	PC,BL#MUL		
000510	005060	000036G		CLR	SND.ENVELOPE+36(R0)		
000514	016700	000000G		MOV	CMD.SLOT,R0	:	2810
000520	006300			ASL	R0		
000522	006300			ASL	R0		
000524	066700	000000G		ADD	SEND.RING,R0		
000530	052760	100000	000002	BIS	#100000,2(R0)		
000536	017766	000000G	000050	MOV	@RC25.ADDR,50(SP)	:	2814
000544	016600	000050		MOV	50(SP),R0	:	
000550	004767	000000V		JSR	PC,GET.CMD.SLOT	:	2818
000554	004767	000000V		JSR	PC,REC.STATUS	:	2822
000560	062706	000052		ADD	#52,SP	:	2746

L12

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0154
Page 59
(21)

000564 000207

RTS PC

: Routine Size: 187 words, Routine Base: AB\$CODE + 4730
: Maximum stack depth per invocation: 22 words

: 2824 1
: 2825 1 !
: 2826 1

```

: 2827 1 global routine SET_CNTL_R_CHAR =
: 2828 1
: 2829 1
: 2830 1 : **
: 2831 1 : FUNCTION DESCRIPTION :
: 2832 1 : THE SET CONTROLLER CHARACTER COMMAND IS USED TO SET MOST SETTABLE
: 2833 1 : UNIT CHARACTERISTICS AND OBTAIN THOSE UNIT CHARACTERISTICS THAT
: 2834 1 : ARE ESSENTIAL FOR PROPER CLASS DRIVER OPERATION. THIS COMMAND
: 2835 1 : NEVER ALTERS THE UNIT'S STATE ("UNIT-ONLINE", "UNIT-AVAILABLE",
: 2836 1 : "UNIT-OFFLINE").
: 2837 1 : FORMAL PARAMETERS :
: 2838 1 : - NONE -
: 2839 1
: 2840 1 : IMPLICIT INPUTS :
: 2841 1
: 2842 1 : IMPLICIT OUTPUTS :
: 2843 1 : - NONE -
: 2844 1
: 2845 1 : COMPLETION CODES :
: 2846 1 : RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
: 2847 1
: 2848 1
: 2849 1 : SIDE EFFECTS :
: 2850 1 : ANY PREVIOUSLY DEFINED CONTROLLER CHARACTERISTICS WILL POSSIBLY
: 2851 1 : BE ALTERED AFTER EXECUTION OF THIS COMMAND.
: 2852 1 : --
: 2853 1
: 2854 2 begin
: 2855 2
: 2856 2 local
: 2857 2 TEMP;
: 2858 2
: 2859 2
: 2860 2 : UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
: 2861 2 :
: 2862 2 SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_SCC; ! LOAD MESSAGE LENGTH
: 2863 2 SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE; ! LOAD CREDIT SIZE
: 2864 2 SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE 'SEQUENTIAL'
: 2865 2 SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0; ! DEFINE CONNECTION ID 'DUP'
: 2866 2
: 2867 2 : MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
: 2868 2 :
: 2869 2 SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF; ! LOAD COMMAND REFERENCE #
: 2870 2 SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO; ! ZERO HI ORDER CMD REF #
: 2871 2 SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = ZERO; ! NOT USED IN DUP IMPLIMENT.
: 2872 2 SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO; ! NOT USED IN DUP IMPLIMENT.
: 2873 2 SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_SCC; ! DEFINE COMMAND OPCODE
: 2874 2 SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO; ! NOT USED
: 2875 2 SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO; ! DEFINE CMD MODIFIERS
: 2876 2
: 2877 2 : COMMAND SPECIFIC COMMAND ENVELOPE FIELD DEFINITION
: 2878 2 :
: 2879 2 SND_ENVELOPE [.CMD_SLOT, MSCP_VER] = ZERO; ! MSCP VERSION
: 2880 2 SND_ENVELOPE [.CMD_SLOT, CTL_FLAGS] = ZERO; ! CONTROLLER GLAGS
: 2881 2 SND_ENVELOPE [.CMD_SLOT, HOST_TOU] = ZERO; ! HOST TIMEOUT VALUE
: 2882 2 SND_ENVELOPE [.CMD_SLOT, RSVD] = ZERO; ! RESERVED
: 2883 2 SND_ENVELOPE [.CMD_SLOT, T&D_0] = ZERO; ! TIME AND DATE WORD 0

```

```

: 2884 2      SND_ENVELOPE [.CMD_SLOT, T#D_1] = ZERO;      ! TIME AND DATE WORD 1
: 2885 2      SND_ENVELOPE [.CMD_SLOT, T#D_2] = ZERO;      ! TIME AND DATE WORD 2
: 2886 2      SND_ENVELOPE [.CMD_SLOT, T#D_3] = ZERO;      ! TIME AND DATE WORD 3
: 2887 2      SND_ENVELOPE [.CMD_SLOT, CDP_LO] = ZERO;      ! CNTL DEP PARAMETER LO WORD
: 2888 2      SND_ENVELOPE [.CMD_SLOT, CDP_HI] = ZERO;      ! CNTL DEP PARAMETER HI WORD
: 2889 2
: 2890 2      ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
: 2891 2
: 2892 2      SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
: 2893 2
: 2894 2      ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
: 2895 2
: 2896 2      TEMP = .RC25_ADDR [RCIP, RC_ALL];
: 2897 2
: 2898 2      ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
: 2899 2
: 2900 2      GET_CMD_SLOT ();
: 2901 2
: 2902 2      ! CHECK THE END PACKET FOR GOOD STATUS
: 2903 2
: 2904 2      return REC_STATUS ();      ! RETURN THE STATUS
: 2905 1      end;
    
```

```

000000 005746      .SBTTL SET.CNTRL.CHAR AZTEC GLOBAL ROUTINE
                   SET.CNTRL.CHAR::
000002 016746 000000G      TST      -(SP)      ;      2827
000006 012746 000054      MOV      CMD_SLOT, -(SP)      ;      2862
000012 004767 000000G      JSR      PC, BL #MUL
000016 012760 000040 000000G      MOV      #40, SND.ENVELOPE(R0)
000024 016716 000000G      MOV      CMD_SLOT, (SP)      ;
000030 012746 000054      MOV      #54, -(SP)      ;      2863
000034 004767 000000G      JSR      PC, BL #MUL
000040 142760 000017 000002G      BICB     #17, SND.ENVELOPE+2(R0)
000046 152760 000001 000002G      BISB     #1, SND.ENVELOPE+2(R0)
000054 016716 000000G      MOV      CMD_SLOT, (SP)      ;
000060 012746 000054      MOV      #54, -(SP)      ;      2864
000064 004767 000000G      JSR      PC, BL #MUL
000070 142760 000360 000002G      BICB     #360, SND.ENVELOPE+2(R0)
000076 016716 000000G      MOV      CMD_SLOT, (SP)      ;
000102 012746 000054      MOV      #54, -(SP)      ;      2865
000106 004767 000000G      JSR      PC, BL #MUL
000112 105060 000003G      CLRB     SND.ENVELOPE+3(R0)
000116 016716 000000G      MOV      CMD_SLOT, (SP)      ;
000122 012746 000054      MOV      #54, -(SP)      ;      2869
000126 004767 000000G      JSR      PC, BL #MUL
000132 016760 000000G 000004G      MOV      CMD_REF, SND.ENVELOPE+4(R0)
000140 016716 000000G      MOV      CMD_SLOT, (SP)      ;
000144 012746 000054      MOV      #54, -(SP)      ;      2870
000150 004767 000000G      JSR      PC, BL #MUL
000154 005060 000006G      CLR      SND.ENVELOPE+6(R0)
000160 016716 000000G      MOV      CMD_SLOT, (SP)      ;
000164 012746 000054      MOV      #54, -(SP)      ;      2871
000170 004767 000000G      JSR      PC, BL #MUL
000174 005060 000010G      CLR      SND.ENVELOPE+10(R0)
000200 016716 000000G      MOV      CMD_SLOT, (SP)      ;      2872
    
```

000204	012746	000054	MOV	#54,-(SP)		
000210	004767	000000G	JSR	PC,BL#MUL		
000214	005060	000012G	CLR	SND.ENVELOPE+12(R0)		
000220	016716	000000G	MOV	CMD.SLOT,(SP)	:	2873
000224	012746	000054	MOV	#54,-(SP)		
000230	004767	000000G	JSR	PC,BL#MUL		
000234	112760	000004	MOV	#4,SND.ENVELOPE+14(R0)		
000242	016716	000000G	MOV	CMD.SLOT,(SP)	:	2874
000246	012746	000054	MOV	#54,-(SP)		
000252	004767	000000G	JSR	PC,BL#MUL		
000256	105060	000015G	CLRB	SND.ENVELOPE+15(R0)		
000262	016716	000000G	MOV	CMD.SLOT,(SP)	:	2875
000266	012746	000054	MOV	#54,-(SP)		
000272	004767	000000G	JSR	PC,BL#MUL		
000276	005060	000016G	CLR	SND.ENVELOPE+16(R0)		
000302	016716	000000G	MOV	CMD.SLOT,(SP)	:	2879
000306	012746	000054	MOV	#54,-(SP)		
000312	004767	000000G	JSR	PC,BL#MUL		
000316	005060	000020G	CLR	SND.ENVELOPE+20(R0)		
000322	016716	000000G	MOV	CMD.SLOT,(SP)	:	2880
000326	012746	000054	MOV	#54,-(SP)		
000332	004767	000000G	JSR	PC,BL#MUL		
000336	005060	000022G	CLR	SND.ENVELOPE+22(R0)		
000342	016716	000000G	MOV	CMD.SLOT,(SP)	:	2881
000346	012746	000054	MOV	#54,-(SP)		
000352	004767	000000G	JSR	PC,BL#MUL		
000356	005060	000024G	CLR	SND.ENVELOPE+24(R0)		
000362	016716	000000G	MOV	CMD.SLOT,(SP)	:	2882
000366	012746	000054	MOV	#54,-(SP)		
000372	004767	000000G	JSR	PC,BL#MUL		
000376	005060	000026G	CLR	SND.ENVELOPE+26(R0)		
000402	016716	000000G	MOV	CMD.SLOT,(SP)	:	2883
000406	012746	000054	MOV	#54,-(SP)		
000412	004767	000000G	JSR	PC,BL#MUL		
000416	005060	000030G	CLR	SND.ENVELOPE+30(R0)		
000422	016716	000000G	MOV	CMD.SLOT,(SP)	:	2884
000426	012746	000054	MOV	#54,-(SP)		
000432	004767	000000G	JSR	PC,BL#MUL		
000436	005060	000032G	CLR	SND.ENVELOPE+32(R0)		
000442	016716	000000G	MOV	CMD.SLOT,(SP)	:	2885
000446	012746	000054	MOV	#54,-(SP)		
000452	004767	000000G	JSR	PC,BL#MUL		
000456	005060	000034G	CLR	SND.ENVELOPE+34(R0)		
000462	016716	000000G	MOV	CMD.SLOT,(SP)	:	2886
000466	012746	000054	MOV	#54,-(SP)		
000472	004767	000000G	JSR	PC,BL#MUL		
000476	005060	000036G	CLR	SND.ENVELOPE+36(R0)		
000502	016716	000000G	MOV	CMD.SLOT,(SP)	:	2887
000506	012746	000054	MOV	#54,-(SP)		
000512	004767	000000G	JSR	PC,BL#MUL		
000516	005060	000040G	CLR	SND.ENVELOPE+40(R0)		
000522	016716	000000G	MOV	CMD.SLOT,(SP)	:	2888
000526	012746	000054	MOV	#54,-(SP)		
000532	004767	000000G	JSR	PC,BL#MUL		
000536	005060	000042G	CLR	SND.ENVELOPE+42(R0)		
000542	016700	000000G	MOV	CMD.SLOT,R0	:	2892
000546	006300		ASL	R0		

ZRCFB2
V03.0MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE27-Mar-1985 15:23:34
11-Jan-1985 08:19:19VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1SEQ 0158
Page 63
(22)

000550	006300		ASL	RO		
000552	066700	000000G	ADD	SEND.RING,RO		
000556	052760	100000 000002	BIS	#100000,2(RO)		
000564	017766	000000G 000054	MOV	@RC25.ADDR,54(SP)	; *,RC.REG	2896
000572	016600	000054	MOV	54(SP),RO	; RC.REG,TEMP	
000576	004767	000000V	JSR	PC.GET.CMD.SLOT		2900
000602	004767	000000V	JSR	PC.REC.STATUS		2904
000606	062706	000056	ADD	#56,SP		2827
000612	000207		RTS	PC		

; Routine Size: 198 words, Routine Base: AB#CODE * 5516
 ; Maximum stack depth per invocation: 24 words

; 2906 1
 ; 2907 1 !
 ; 2908 1

```

: 2909 1 global routine AVAILABLE =
: 2910 1
: 2911 1
: 2912 1 **
: 2913 1 FUNCTIONAL DESCRIPTION :
: 2914 1 THE AVAILABLE COMMAND IS USED TO SET THE UNIT-ABAILABLE WHEN
: 2915 1 ALL OUTSTANDING COMMANDS FOR THE SPECIFIED UNIT ARE COMPLETED.
: 2916 1 IF THE "SPIN-DOWN" MODIFIER IS SPECIFIED, THE DISK SPINS DOWN
: 2917 1 AND ITS HEADS ARE UNLOADED.
: 2918 1 FORMAL PARAMETERS :
: 2919 1 IMPLICIT INPUTS :
: 2920 1 PLATTER NUMBER (UNIT)
: 2921 1 IMPLICIT OUTPUTS :
: 2922 1 RET_STATUS
: 2923 1 SIDE EFFECTS :
: 2924 1
: 2925 1 --
: 2926 1
: 2927 2 begin
: 2928 2
: 2929 2 local
: 2930 2 TEMP;
: 2931 2
: 2932 2
: 2933 2 UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
: 2934 2
: 2935 2 SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_AVL; ! LOAD MESSAGE LENGTH
: 2936 2 SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE; ! LOAD CREDIT SIZE
: 2937 2 SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE 'SEQUENTIAL'
: 2938 2 SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0; ! DEFINE CONNECTION ID 'DUP'
: 2939 2
: 2940 2 MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
: 2941 2
: 2942 2 SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF; !LOAD COMMAND REFERENCE #
: 2943 2 SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO; ! ZERO HI ORDER CMD REF #
: 2944 2 SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = .UNIT; ! SELECTED UNIT
: 2945 2 SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO; ! NOT USED IN DUP IMPLIMENT.
: 2946 2 SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_AVL; ! DEFINE COMMAND OPCODE
: 2947 2 SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO; ! NOT USED
: 2948 2 SND_ENVELOPE [.CMD_SLOT, MODIFIER] = MD_SPD; ! DEFINE CMD MODIFIERS
: 2949 2
: 2950 2 SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
: 2951 2
: 2952 2 SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
: 2953 2
: 2954 2 READ THE IP REGISTER TO STIMULATE PORT POLLING.
: 2955 2
: 2956 2 TEMP = .RC25_ADDR [RCIP, RC_ALL];
: 2957 2
: 2958 2 GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
: 2959 2
: 2960 2 GET_CMD_SLOT ();
: 2961 2
: 2962 2
: 2963 2 CHECK THE END PACKET FOR GOOD STATUS
: 2964 2
: 2965 2 return REC_STATUS (); ! RETURN THE STATUS

```

; 2966 1 end;

		.SBTTL AVAILABLE AZTEC GLOBAL ROUTINE		
000000	005746		AVAILABLE::	
			TST	-(SP)
000002	016746	000000G	MOV	CMD.SLOT, -(SP)
000006	012746	000054	MOV	#54, -(SP)
000012	004767	000000G	JSR	PC, BL \$MUL
000016	012760	000014 000000G	MOV	#14, SND.ENVELOPE(R0)
000024	016716	000000G	MOV	CMD.SLOT, (SP)
000030	012746	000054	MOV	#54, -(SP)
000034	004767	000000G	JSR	PC, BL \$MUL
000040	142760	000017 000002G	BICB	#17, SND.ENVELOPE+2(R0)
000046	152760	000001 000002G	BISB	#1, SND.ENVELOPE+2(R0)
000054	016716	000000G	MOV	CMD.SLOT, (SP)
000060	012746	000054	MOV	#54, -(SP)
000064	004767	000000G	JSR	PC, BL \$MUL
000070	142760	000360 000002G	BICB	#360, SND.ENVELOPE+2(R0)
000076	016716	000000G	MOV	CMD.SLOT, (SP)
000102	012746	000054	MOV	#54, -(SP)
000106	004767	000000G	JSR	PC, BL \$MUL
000112	105060	000003G	CLRB	SND.ENVELOPE+3(R0)
000116	016716	000000G	MOV	CMD.SLOT, (SP)
000122	012746	000054	MOV	#54, -(SP)
000126	004767	000000G	JSR	PC, BL \$MUL
000132	016760	000000G 000004G	MOV	CMD.REF, SND.ENVELOPE+4(R0)
000140	016716	000000G	MOV	CMD.SLOT, (SP)
000144	012746	000054	MOV	#54, -(SP)
000150	004767	000000G	JSR	PC, BL \$MUL
000154	005060	000006G	CLR	SND.ENVELOPE+6(R0)
000160	016716	000000G	MOV	CMD.SLOT, (SP)
000164	012746	000054	MOV	#54, -(SP)
000170	004767	000000G	JSR	PC, BL \$MUL
000174	016760	000000G 000010G	MOV	UNIT, SND.ENVELOPE+10(R0)
000202	016716	000000G	MOV	CMD.SLOT, (SP)
000206	012746	000054	MOV	#54, -(SP)
000212	004767	000000G	JSR	PC, BL \$MUL
000216	005060	000012G	CLR	SND.ENVELOPE+12(R0)
000222	016716	000000G	MOV	CMD.SLOT, (SP)
000226	012746	000054	MOV	#54, -(SP)
000232	004767	000000G	JSR	PC, BL \$MUL
000236	112760	000010 000014G	MOVB	#10, SND.ENVELOPE+14(R0)
000244	016716	000000G	MOV	CMD.SLOT, (SP)
000250	012746	000054	MOV	#54, -(SP)
000254	004767	000000G	JSR	PC, BL \$MUL
000260	105060	000015G	CLRB	SND.ENVELOPE+15(R0)
000264	016716	000000G	MOV	CMD.SLOT, (SP)
000270	012746	000054	MOV	#54, -(SP)
000274	004767	000000G	JSR	PC, BL \$MUL
000300	012760	000001 000016G	MOV	#1, SND.ENVELOPE+16(R0)
000306	016700	000000G	MOV	CMD.SLOT, R0
000312	006300		ASL	R0
000314	006300		ASL	R0
000316	066700	000000G	ADD	SEND.RING, R0
000322	052760	100000 000002	BIS	#100000, 2(R0)
000330	017766	000000G 000030	MOV	@RC25, ADDR, 30(SP)
				; *,RC.REG

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0161
Page 66
(23)

000336	016600	000030	MOV	30(SP),R0	; RC.REG,TEMP	
000342	004767	000000V	JSR	PC,GET.CMD.SLOT	;	2960
000346	004767	000000V	JSR	PC,REC.STATUS	;	2965
000352	062706	000032	ADD	#32,SP	;	2909
000356	000207		RTS	PC		

; Routine Size: 120 words, Routine Base: AB\$CODE + 6332
; Maximum stack depth per invocation: 14 words

: 2967 1
: 2968 1 !
: 2969 1

```

: 2970 1 global routine ON_LINE =
: 2971 1
: 2972 1
: 2973 1 : **
: 2974 1 : FUNCTIONAL DESCRIPTION :
: 2975 1 : THE ONLINE COMMAND IS USED TO BRING A UNIT "UNIT-ONLINE, SET
: 2976 1 : HOST SETTABLE UNIT CHARACTERISTICS AND OBTAIN THOSE UNIT
: 2977 1 : CHARACTERISTICS THAT ARE ESSENTIAL FOR PROPER CLASS DRIVER
: 2978 1 : OPERATION. THE UNIT IS SPUN-UP, IF NECESSARY, AND IS HEADS
: 2979 1 : ARE LOADED PRIOR TO RETURNING THE ONLINE COMMAND'S END MESSAGE.
: 2980 1 : HOST SETTABLE CHARACTERISTICS COMMAND WERE ISSUED. HOST
: 2981 1 : SETTABLE CHARACTERISTICS ARE SET AFTER THE UNIT HAS BEEN
: 2982 1 : SUCCESSFULLY SPUN-UP AND ANY OTHER VALIDITY CHECKS HAVE SUCCEEDED.
: 2983 1 : FORMAL PARAMETERS :
: 2984 1 : - NONE -
: 2985 1
: 2986 1 : IMPLICIT INPUTS :
: 2987 1
: 2988 1 : INPLICIT OUTPUTS :
: 2989 1 : RET_STATUS
: 2990 1
: 2991 1 : COMPLETION CODES :
: 2992 1 : RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
: 2993 1
: 2994 1
: 2995 1 : SIDE EFFECTS :
: 2996 1 : ANY PREVIOUSLY DEFINED CONTROLLER CHARACTERISTICS WILL POSSIBLY
: 2997 1 : BE ALTERED AFTER EXECUTION OF THIS COMMAND.
: 2998 1 : --
: 2999 1
: 3000 2 begin
: 3001 2
: 3002 2 local
: 3003 2 TEMP;
: 3004 2
: 3005 2
: 3006 2 : UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
: 3007 2 :
: 3008 2 : SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_ONL; ! LOAD MESSAGE LENGTH
: 3009 2 : SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE; ! LOAD CREDIT SIZE
: 3010 2 : SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE 'SEQUENTIAL'
: 3011 2 : SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0; ! DEFINE CONNECTION ID 'DUP'
: 3012 2
: 3013 2 : MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
: 3014 2 :
: 3015 2 : SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF; ! LOAD COMMAND REFERENCE #
: 3016 2 : SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO; ! ZERO HI ORDER CMD REF #
: 3017 2 : SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = .UNIT; ! SELECTED UNIT
: 3018 2 : SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO; ! NOT USED IN DUP IMPLIMENT.
: 3019 2 : SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_ONL; ! DEFINE COMMAND OPCODE
: 3020 2 : SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO; ! NOT USED
: 3021 2 : SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO; ! DEFINE CMD MODIFIERS
: 3022 2
: 3023 2 : COMMAND SPECIFIC COMMAND ENVELOPE FIELD DEFINITION
: 3024 2 :
: 3025 2 : SND_ENVELOPE [.CMD_SLOT, RSV$D] = ZERO; ! RESERVED
: 3026 2 : SND_ENVELOPE [.CMD_SLOT, UNT_FLAGS] = ZERO; ! UNIT FLAG FIELD

```

```

: 3027 2      SND_ENVELOPE [.CMD_SLOT, RSVD#0] = ZERO;      ! RESERVED FIELD
: 3028 2      SND_ENVELOPE [.CMD_SLOT, RSVD#1] = ZERO;      ! RESERVED FIELD
: 3029 2      SND_ENVELOPE [.CMD_SLOT, RSVD#2] = ZERO;      ! RESERVED FIELD
: 3030 2      SND_ENVELOPE [.CMD_SLOT, RSVD#3] = ZERO;      ! RESERVED FIELD
: 3031 2      SND_ENVELOPE [.CMD_SLOT, RSVD#4] = ZERO;      ! RESERVED FIELD
: 3032 2      SND_ENVELOPE [.CMD_SLOT, RSVD#5] = ZERO;      ! RESERVED FIELD
: 3033 2      SND_ENVELOPE [.CMD_SLOT, DDP_LO] = ZERO;      ! DEVICE DEPENDENT PARAMETER
: 3034 2      SND_ENVELOPE [.CMD_SLOT, DDP_HI] = ZERO;      ! DEVICE DEPENDENT PARAMETER
: 3035 2      SND_ENVELOPE [.CMD_SLOT, SHADOW_UNIT] = ZERO;  ! SHADOW UNIT
: 3036 2      SND_ENVELOPE [.CMD_SLOT, COPY_SPEED] = ZERO;  ! COPY SPEED
: 3037 2
: 3038 2      ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
: 3039 2
: 3040 2      SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
: 3041 2
: 3042 2      ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
: 3043 2
: 3044 2      TEMP = .RC25_ADDR [RCIP, RC_ALL];
: 3045 2
: 3046 2      ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
: 3047 2
: 3048 2      GET_CMD_SLOT ();
: 3049 2
: 3050 2      ! CHECK THE END PACKET FOR GOOD STATUS
: 3051 2
: 3052 2      return REC_STATUS ();      ! RETURN THE STATUS
: 3053 1      end;

```

```

000000 005746      .SBTTL  ON.LINE AZTEC GLOBAL ROUTINE
ON.LINE::
000002 016746 000000G      TST      -(SP)      ;      2970
000006 012746 000054      MOV      CMD_SLOT, -(SP)      ;      3008
000012 004767 000000G      MOV      #54, -(SP)
000016 012760 000044 000000G      JSR      PC, BL#MUL
000024 016716 000000G      MOV      #44, SND_ENVELOPE(R0)
000030 012746 000054      MOV      CMD_SLOT, (SP)      ;      3009
000034 004767 000000G      MOV      #54, -(SP)
000040 142760 000017 000002G      JSR      PC, BL#MUL
000046 152760 000001 000002G      BICB    #17, SND_ENVELOPE+2(R0)
000054 016716 000000G      BISB    #1, SND_ENVELOPE+2(R0)
000060 012746 000054      MOV      CMD_SLOT, (SP)      ;      3010
000064 004767 000000G      MOV      #54, -(SP)
000070 142760 000360 000002G      JSR      PC, BL#MUL
000076 016716 000000G      BICB    #360, SND_ENVELOPE+2(R0)
000102 012746 000054      MOV      CMD_SLOT, (SP)      ;      3011
000106 004767 000000G      MOV      #54, -(SP)
000112 105060 000003G      JSR      PC, BL#MUL
000116 016716 000000G      CLRB    SND_ENVELOPE+3(R0)
000122 012746 000054      MOV      CMD_SLOT, (SP)      ;      3015
000126 004767 000000G      MOV      #54, -(SP)
000132 016760 000000G 000004G      JSR      PC, BL#MUL
000140 016716 000000G      MOV      CMD_REF, SND_ENVELOPE+4(R0)
000144 012746 000054      MOV      CMD_SLOT, (SP)      ;      3016
000150 004767 000000G      MOV      #54, -(SP)
000154 005060 000006G      JSR      PC, BL#MUL
      CLR      SND_ENVELOPE+6(R0)

```

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0164
Page 69
(24)

000160	016716	000000G	MOV	CMD.SLOT,(SP)	:	3017
000164	012746	000054	MOV	#54,-(SP)		
000170	004767	000000G	JSR	PC,BL#MUL		
000174	016760	000000G	MOV	UNIT,SND.ENVELOPE+10(R0)		
000202	016716	000000G	MOV	CMD.SLOT,(SP)	:	3018
000206	012746	000054	MOV	#54,-(SP)		
000212	004767	000000G	JSR	PC,BL#MUL		
000216	005060	000012G	CLR	SND.ENVELOPE+12(R0)		
000222	016716	000000G	MOV	CMD.SLOT,(SP)	:	3019
000226	012746	000054	MOV	#54,-(SP)		
000232	004767	000000G	JSR	PC,BL#MUL		
000236	112760	000011	MOV	#11,SND.ENVELOPE+14(R0)		
000244	016716	000000G	MOV	CMD.SLOT,(SP)	:	3020
000250	012746	000054	MOV	#54,-(SP)		
000254	004767	000000G	JSR	PC,BL#MUL		
000260	105060	000015G	CLRB	SND.ENVELOPE+15(R0)		
000264	016716	000000G	MOV	CMD.SLOT,(SP)	:	3021
000270	012746	000054	MOV	#54,-(SP)		
000274	004767	000000G	JSR	PC,BL#MUL		
000300	005060	000016G	CLR	SND.ENVELOPE+16(R0)		
000304	016716	000000G	MOV	CMD.SLOT,(SP)	:	3025
000310	012746	000054	MOV	#54,-(SP)		
000314	004767	000000G	JSR	PC,BL#MUL		
000320	005060	000020G	CLR	SND.ENVELOPE+20(R0)		
000324	016716	000000G	MOV	CMD.SLOT,(SP)	:	3026
000330	012746	000054	MOV	#54,-(SP)		
000334	004767	000000G	JSR	PC,BL#MUL		
000340	005060	000022G	CLR	SND.ENVELOPE+22(R0)		
000344	016716	000000G	MOV	CMD.SLOT,(SP)	:	3027
000350	012746	000054	MOV	#54,-(SP)		
000354	004767	000000G	JSR	PC,BL#MUL		
000360	005060	000024G	CLR	SND.ENVELOPE+24(R0)		
000364	016716	000000G	MOV	CMD.SLOT,(SP)	:	3028
000370	012746	000054	MOV	#54,-(SP)		
000374	004767	000000G	JSR	PC,BL#MUL		
000400	005060	000026G	CLR	SND.ENVELOPE+26(R0)		
000404	016716	000000G	MOV	CMD.SLOT,(SP)	:	3029
000410	012746	000054	MOV	#54,-(SP)		
000414	004767	000000G	JSR	PC,BL#MUL		
000420	005060	000030G	CLR	SND.ENVELOPE+30(R0)		
000424	016716	000000G	MOV	CMD.SLOT,(SP)	:	3030
000430	012746	000054	MOV	#54,-(SP)		
000434	004767	000000G	JSR	PC,BL#MUL		
000440	005060	000032G	CLR	SND.ENVELOPE+32(R0)		
000444	016716	000000G	MOV	CMD.SLOT,(SP)	:	3031
000450	012746	000054	MOV	#54,-(SP)		
000454	004767	000000G	JSR	PC,BL#MUL		
000460	005060	000034G	CLR	SND.ENVELOPE+34(R0)		
000464	016716	000000G	MOV	CMD.SLOT,(SP)	:	3032
000470	012746	000054	MOV	#54,-(SP)		
000474	004767	000000G	JSR	PC,BL#MUL		
000500	005060	000036G	CLR	SND.ENVELOPE+36(R0)		
000504	016716	000000G	MOV	CMD.SLOT,(SP)	:	3033
000510	012746	000054	MOV	#54,-(SP)		
000514	004767	000000G	JSR	PC,BL#MUL		
000520	005060	000040G	CLR	SND.ENVELOPE+40(R0)		
000524	016716	000000G	MOV	CMD.SLOT,(SP)	:	3034

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0165
Page 70
(24)

000530	012746	000054	MOV	#54,-(SP)		
000534	004767	000000G	JSR	PC,BL#MUL		
000540	005060	000042G	CLR	SND.ENVELOPE+42(R0)		
000544	016716	000000G	MOV	CMD.SLOT,(SP)		3035
000550	012746	000054	MOV	#54,-(SP)		
000554	004767	000000G	JSR	PC,BL#MUL		
000560	005060	000044G	CLR	SND.ENVELOPE+44(R0)		
000564	016716	000000G	MOV	CMD.SLOT,(SP)		3036
000570	012746	000054	MOV	#54,-(SP)		
000574	004767	000000G	JSR	PC,BL#MUL		
000600	005060	000046G	CLR	SND.ENVELOPE+46(R0)		
000604	016700	000000G	MOV	CMD.SLOT,R0		3040
000610	006300		ASL	R0		
000612	006300		ASL	R0		
000614	066700	000000G	ADD	SEND.RING,R0		
000620	052760	100000 000002	BIS	#100000,2(R0)		
000626	017766	000000G 000060	MOV	@RC25.ADDR,60(SP)		3044
000634	016600	000060	MOV	60(SP),R0	; *,RC.REG	
000640	004767	000000V	JSR	PC,GET.CMD.SLOT	; RC.REG,TEMP	3048
000644	004767	000000V	JSR	PC,REC.STATUS		3052
000650	062706	000062	ADD	#62,SP		2970
000654	000207		RTS	PC		

; Routine Size: 215 words, Routine Base: AB#CODE + 6712
; Maximum stack depth per invocation: 26 words

; 3054 1
; 3055 1
; 3056 1


```

3057 1 global routine READ_CMD =
3058 1 **
3059 1 : FUNCTIONAL DESCRIPTION :
3060 1 : THE READ COMMAND IS USED TO READ FROM THE UNIT AND TRANSFER
3061 1 : TO THE HOST BUFFER. READ CMD MAY BE CONSIDERED AS SEEK COMMAND
3062 1 : IF THE BYTE_COUNT WAS ZERO.
3063 1 : IF THE FLAG TIP CONTAINS ALL ONES, THEN THIS ROUTINE WILL NOT
3064 1 : CALL REC_STATUS AND WAIT FOR END PACKET STATUS INFO. THIS IS
3065 1 : DONE TO QUEUE SEEK COMMANDS FOR THE CONTROLLER IN SOME TESTS.
3066 1 : IF CMOD CONTAINS MD_EXP BIT AS THE COMMAND MODIFIER, THEN
3067 1 : SEEKS WILL BE DONE BY THE CONTROLLER IN THE ORDER RECEIVED
3068 1 : AND WILL NOT BE OPTIMIZED.
3069 1 :
3070 1 : FORMAL PARAMETERS :
3071 1 : - NONE -
3072 1 :
3073 1 : IMPLICIT INPUTS :
3074 1 :           BUF_DESCRPTR, BYTE_COUNT, UNIT, LBN_ST, TIP,
3075 1 :           CMOD
3076 1 : IMPLICIT OUTPUTS :
3077 1 :           RET_STATUS
3078 1 :
3079 1 : COMPLETION CODES :
3080 1 :   RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
3081 1 :
3082 1 :
3083 1 : SIDE EFFECTS :
3084 1 : - NONE -
3085 1 : --
3086 1 :
3087 2   begin
3088 2
3089 2   local
3090 2     TEMP;
3091 2
3092 2 :
3093 2 : UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
3094 2 :
3095 2 :   SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_RD;           ! LOAD MESSAGE LENGTH
3096 2 :   SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;               ! LOAD CREDIT SIZE
3097 2 :   SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0;                ! MESSAGE TYPE
3098 2 :   SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0;                 ! DEFINE CONNECTION ID
3099 2 :
3100 2 : MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
3101 2 :
3102 2 :   SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;         ! LOAD COMMAND REFERENCE #
3103 2 :   SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;             ! ZERO HI ORDER CMD REF #
3104 2 :   SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = .UNIT;           ! SELECTED UNIT
3105 2 :   SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;            ! NOT USED IN DUP IMPLIMENT.
3106 2 :   SND_ENVELOPE [.CMD_SLOT, OP_CODE] = OP_RD;            ! DEFINE COMMAND OP CODE
3107 2 :   SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;              ! NOT USED
3108 2 :   SND_ENVELOPE [.CMD_SLOT, MODIFIER] = .CMOD;           ! DEFINE CMD MODIFIERS
3109 2 :
3110 2 : COMMAND SPECIFIC COMMAND ENVELOPE FIELD DEFINITION
3111 2 :
3112 2 :   SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT;      ! BYTE COUNT LOW WORD
3113 2 :   SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO;             ! BYTE COUNT HIGH WORD

```

```

: 3114 2 SND_ENVELOPE [.CMD_SLOT, BD_0] = .BUF_DESCRPT; ! BUFFER DESCRIPTOR FIELD
: 3115 2 SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO; ! BUFFER DESCRIPTOR FIELD
: 3116 2 SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO; ! BUFFER DESCRIPTOR FIELD
: 3117 2 SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO; ! BUFFER DESCRIPTOR FIELD
: 3118 2 SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO; ! BUFFER DESCRIPTOR FIELD
: 3119 2 SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO; ! BUFFER DESCRIPTOR FIELD
: 3120 2 SND_ENVELOPE [.CMD_SLOT, LBN_LO] = .LBN_ST; ! LOGICAL BLOCK NUMBER
: 3121 2 SND_ENVELOPE [.CMD_SLOT, LBN_HI] = ZERO; ! LOGICAL BLOCK NUMBER
: 3122 2
: 3123 2 ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
: 3124 2
: 3125 2 SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
: 3126 2
: 3127 2
: 3128 2 if .TIP equl #o'177777' then return RET_STATUS = PAS_CODE; ! IF TIP CONTAINS
: 3129 2
: 3130 2 ! ALL_ONES EXIT HERE
: 3131 2 ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
: 3132 2
: 3133 2 TEMP = .RC25_ADDR [RCIP, RC_ALL];
: 3134 2
: 3135 2 ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
: 3136 2
: 3137 2 GET_CMD_SLOT ();
: 3138 2
: 3139 2 ! CHECK THE END PACKET FOR GOOD STATUS
: 3140 2
: 3141 2 return REC_STATUS (); ! RETURN THE STATUS
: 3142 1 end;

```

```

000000 005746 .SBTTL READ.CMD AZTEC GLOBAL ROUTINE
READ.CMD:
000002 016746 000000G TST -(SP) ; 3057
000006 012746 000054 MOV CMD_SLOT, -(SP) ; 3095
000012 004767 000000G JSR PC, BL $MUL
000016 012760 000040 000000G MOV #40, SND.ENVELOPE(R0)
000024 016716 000000G MOV CMD_SLOT, (SP) ; 3096
000030 012746 000054 MOV #54, -(SP)
000034 004767 000000G JSR PC, BL $MUL
000040 142760 000017 000002G BICB #17, SND.ENVELOPE*2(R0)
000046 152760 000001 000002G BISB #1, SND.ENVELOPE*2(R0)
000054 016716 000000G MOV CMD_SLOT, (SP) ; 3097
000060 012746 000054 MOV #54, -(SP)
000064 004767 000000G JSR PC, BL $MUL
000070 142760 000360 000002G BICB #360, SND.ENVELOPE*2(R0)
000076 016716 000000G MOV CMD_SLOT, (SP) ; 3098
000102 012746 000054 MOV #54, -(SP)
000106 004767 000000G JSR PC, BL $MUL
000112 105060 000003G CLRB SND.ENVELOPE*3(R0)
000116 016716 000000G MOV CMD_SLOT, (SP) ; 3102
000122 012746 000054 MOV #54, -(SP)
000126 004767 000000G JSR PC, BL $MUL
000132 016760 000000G 000004G MOV CMD_REF, SND.ENVELOPE*4(R0)
000140 016716 000000G MOV CMD_SLOT, (SP) ; 3103
000144 012746 000054 MOV #54, -(SP)

```

M13

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

000150	004767	000000G	JSR	PC,BL#MUL		
000154	005060	000006G	CLR	SND.ENVELOPE+6(R0)		
000160	016716	000000G	MOV	CMD.SLOT,(SP)	:	3104
000164	012746	000054	MOV	#54,-(SP)		
000170	004767	000000G	JSR	PC,BL#MUL		
000174	016760	000000G 000010G	MOV	UNIT,SND.ENVELOPE+10(R0)		
000202	016716	000000G	MOV	CMD.SLOT,(SP)	:	3105
000206	012746	000054	MOV	#54,-(SP)		
000212	004767	000000G	JSR	PC,BL#MUL		
000216	005060	000012G	CLR	SND.ENVELOPE+12(R0)		
000222	016716	000000G	MOV	CMD.SLOT,(SP)	:	3106
000226	012746	000054	MOV	#54,-(SP)		
000232	004767	000000G	JSR	PC,BL#MUL		
000236	112760	000041 000014G	MOVB	#41,SND.ENVELOPE+14(R0)		
000244	016716	000000G	MOV	CMD.SLOT,(SP)	:	3107
000250	012746	000054	MOV	#54,-(SP)		
000254	004767	000000G	JSR	PC,BL#MUL		
000260	105060	000015G	CLRB	SND.ENVELOPE+15(R0)		
000264	016716	000000G	MOV	CMD.SLOT,(SP)	:	3108
000270	012746	000054	MOV	#54,-(SP)		
000274	004767	000000G	JSR	PC,BL#MUL		
000300	016760	000000G 000016G	MOV	CMD,SND.ENVELOPE+16(R0)		
000306	016716	000000G	MOV	CMD.SLOT,(SP)	:	3112
000312	012746	000054	MOV	#54,-(SP)		
000316	004767	000000G	JSR	PC,BL#MUL		
000322	016760	000000G 000020G	MOV	BYTE.COUNT,SND.ENVELOPE+20(R0)		
000330	016716	000000G	MOV	CMD.SLOT,(SP)	:	3113
000334	012746	000054	MOV	#54,-(SP)		
000340	004767	000000G	JSR	PC,BL#MUL		
000344	005060	000022G	CLR	SND.ENVELOPE+22(R0)		
000350	016716	000000G	MOV	CMD.SLOT,(SP)	:	3114
000354	012746	000054	MOV	#54,-(SP)		
000360	004767	000000G	JSR	PC,BL#MUL		
000364	016760	000000G 000024G	MOV	BUF.DESCRPTR,SND.ENVELOPE+24(R0)		
000372	016716	000000G	MOV	CMD.SLOT,(SP)	:	3115
000376	012746	000054	MOV	#54,-(SP)		
000402	004767	000000G	JSR	PC,BL#MUL		
000406	005060	000026G	CLR	SND.ENVELOPE+26(R0)		
000412	016716	000000G	MOV	CMD.SLOT,(SP)	:	3116
000416	012746	000054	MOV	#54,-(SP)		
000422	004767	000000G	JSR	PC,BL#MUL		
000426	005060	000030G	CLR	SND.ENVELOPE+30(R0)		
000432	016716	000000G	MOV	CMD.SLOT,(SP)	:	3117
000436	012746	000054	MOV	#54,-(SP)		
000442	004767	000000G	JSR	PC,BL#MUL		
000446	005060	000032G	CLR	SND.ENVELOPE+32(R0)		
000452	016716	000000G	MOV	CMD.SLOT,(SP)	:	3118
000456	012746	000054	MOV	#54,-(SP)		
000462	004767	000000G	JSR	PC,BL#MUL		
000466	005060	000034G	CLR	SND.ENVELOPE+34(R0)		
000472	016716	000000G	MOV	CMD.SLOT,(SP)	:	3119
000476	012746	000054	MOV	#54,-(SP)		
000502	004767	000000G	JSR	PC,BL#MUL		
000506	005060	000036G	CLR	SND.ENVELOPE+36(R0)		
000512	016716	000000G	MOV	CMD.SLOT,(SP)	:	3120
000516	012746	000054	MOV	#54,-(SP)		
000522	004767	000000G	JSR	PC,BL#MUL		

N13

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0169
Page 74
(25)

000526	016760	000000G	000040G		MOV	LBN.ST,SND.ENVELOPE+40(R0)		
000534	016716	000000G			MOV	CMD.SLOT,(SP)		
000540	012746	000054			MOV	#54,-(SP)	:	3121
000544	004767	000000G			JSR	PC,BL#MUL		
000550	005060	000042G			CLR	SND.ENVELOPE+42(R0)		
000554	016700	000000G			MOV	CMD.SLOT,R0	:	3125
000560	006300				ASL	R0		
000562	006300				ASL	R0		
000564	066700	000000G			ADD	SEND.RING,R0		
000570	052760	100000	000002		BIS	#100000,2(R0)		
000576	026727	000000G	177777		CMP	TIP,#-1	:	3128
000604	001006				BNE	1#		
000606	005067	000000G			CLR	RET.STATUS		
000612	062706	000054			ADD	#54,SP		
000616	005000				CLR	R0		
000620	000413				BR	2#		
000622	017766	000000G	000054	1#:	MOV	#RC25.ADDR,54(SP)	:	3133
000630	016600	000054			MOV	54(SP),R0	:	3137
000634	004767	000000V			JSR	PC,GET.CMD.SLOT	:	3141
000640	004767	000000V			JSR	PC,REC.STATUS	:	3057
000644	062706	000054			ADD	#54,SP		
000650	005726			2#:	TST	(SP).	:	
000652	000207				RTS	PC		

: Routine Size: 214 words, Routine Base: AB#CODE * 7570
: Maximum stack depth per invocation: 24 words

: 3143 1
: 3144 1 :
: 3145 1

```

: 3146 1 global routine GET_UNIT_STATUS =
: 3147 1 **
: 3148 1 : FUNCTIONAL DESCRIPTION :
: 3149 1 : THE GET UNIT STATUS COMMAND IS USED TO READ THE CURRENT
: 3150 1 : STATE OF THE UNIT, PLUS CERTAIN UNIT CHARACTERISTIACS.
: 3151 1 :
: 3152 1 : FORMAL PARAMETERS :
: 3153 1 : - NONE -
: 3154 1 :
: 3155 1 : IMPLICIT INPUTS :
: 3156 1 : UNIT
: 3157 1 : INPLICIT OUTPUTS :
: 3158 1 : RET_STATUS
: 3159 1 :
: 3160 1 : COMPLETION CODES :
: 3161 1 : RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
: 3162 1 :
: 3163 1 :
: 3164 1 : SIDE EFFECTS :
: 3165 1 : - NONE -
: 3166 1 : --
: 3167 1 :
: 3168 2 begin
: 3169 2
: 3170 2 local
: 3171 2 TEMP;
: 3172 2
: 3173 2 :
: 3174 2 : UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
: 3175 2 :
: 3176 2 SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_GUS; ! LOAD MESSAGE LENGTH
: 3177 2 SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE; ! LOAD CREDIT SIZE
: 3178 2 SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE
: 3179 2 SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0; ! DEFINE CONNECTION ID
: 3180 2 :
: 3181 2 : MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
: 3182 2 :
: 3183 2 SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF; !LOAD COMMAND REFERENCE #
: 3184 2 SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO; ! ZERO HI ORDER CMD REF #
: 3185 2 SND_ENVELOPE [.CMD_SLOT, UN_LUSED] = .UNIT; ! SELECTED UNIT
: 3186 2 SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO; ! NOT USED IN DUP IMPLIMENT.
: 3187 2 SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_GUS; ! DEFINE COMMAND OPCODE
: 3188 2 SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO; ! NOT USED
: 3189 2 SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO; ! DEFINE CMD MODIFIERS
: 3190 2 :
: 3191 2 : SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
: 3192 2 :
: 3193 2 SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
: 3194 2 :
: 3195 2 : READ THE IP REGISTER TO STIMULATE PORT POLLING.
: 3196 2 :
: 3197 2 TEMP = .RC25_ADDR [RCIP, RC_ALL];
: 3198 2 :
: 3199 2 : GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
: 3200 2 :
: 3201 2 GET_CMD_SLOT ();
: 3202 2 :

```

C14

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0171
Page 76
(26)

```

: 3203 2      ! CHECK THE END PACKET FOR GOOD STATUS
: 3204 2      !
: 3205 2
: 3206 2      if REC_STATUS ( )          ! READ THE STATUS
: 3207 2      then
: 3208 3          begin
: 3209 3          return .RET_STATUS;    ! RETURN WITH A STATUS ERR
: 3210 3          end
: 3211 2      else
: 3212 2          RES_SLOT = .RES_SLOT - 1; ! GET THE CURRENT RES. SLOT
: 3213 2
: 3214 2      RET_UNIT_FLAG = .REC_ENVELOPE [.RES_SLOT, UNIT_FLAG]; ! READ UNIT FLAG
: 3215 2      GET_RES_SLOT ( );        ! GET NEXT RES. SLOT
: 3216 2      return .RET_STATUS;      ! RETURN WITH A PASS CODE
: 3217 1      end;
    
```

```

000000 005746      .SBTTL GET.UNIT.STATUS AZTEC GLOBAL ROUTINE
GET.UNIT.STATUS::
000002 016746 000000G      TST      -(SP)          ;          3146
000006 012746 000054      MOV      CMD.SLOT, -(SP)      ;          3176
000012 004767 000000G      JSR      PC, BL $MUL
000016 012760 000014 000000G      MOV      #14, SND.ENVELOPE(R0)
000024 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3177
000030 012746 000054      MOV      #54, -(SP)
000034 004767 000000G      JSR      PC, BL $MUL
000040 142760 000017 000002G      BICB     #17, SND.ENVELOPE+2(R0)
000046 152760 000001 000002G      BISB     #1, SND.ENVELOPE+2(R0)
000054 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3178
000060 012746 000054      MOV      #54, -(SP)
000064 004767 000000G      JSR      PC, BL $MUL
000070 142760 000360 000002G      BICB     #360, SND.ENVELOPE+2(R0)
000076 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3179
000102 012746 000054      MOV      #54, -(SP)
000106 004767 000000G      JSR      PC, BL $MUL
000112 105060 000003G      CLRB     SND.ENVELOPE+3(R0)
000116 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3183
000122 012746 000054      MOV      #54, -(SP)
000126 004767 000000G      JSR      PC, BL $MUL
000132 016760 000000G 000004G      MOV      CMD.REF, SND.ENVELOPE+4(R0)
000140 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3184
000144 012746 000054      MOV      #54, -(SP)
000150 004767 000000G      JSR      PC, BL $MUL
000154 005060 000006G      CLR      SND.ENVELOPE+6(R0)
000160 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3185
000164 012746 000054      MOV      #54, -(SP)
000170 004767 000000G      JSR      PC, BL $MUL
000174 016760 000000G 000010G      MOV      UNIT, SND.ENVELOPE+10(R0)
000202 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3186
000206 012746 000054      MOV      #54, -(SP)
000212 004767 000000G      JSR      PC, BL $MUL
000216 005060 000012G      CLR      SND.ENVELOPE+12(R0)
000222 016716 000000G      MOV      CMD.SLOT, (SP)      ;          3187
000226 012746 000054      MOV      #54, -(SP)
000232 004767 000000G      JSR      PC, BL $MUL
000236 112760 000003 000014G      MOVB     #3, SND.ENVELOPE+14(R0)
    
```

ZRCFR2	MISCELLANEOUS SECTIONS					
V03.0	AZTEC GLOBAL ROUTINE					
				27-Mar-1985 15:23:34	VAX-11 Bliss-16 V4.0-579	SEQ 0172
				11-Jan-1985 08:19:19	USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1	Page 77
						(26)
000244	016716	000000G		MOV	CMD.SLOT,(SP)	3188
000250	012746	000054		MOV	#54,-(SP)	
000254	004767	000000G		JSR	PC,BL#MUL	
000260	105060	000015G		CLRB	SND.ENVELOPE+15(RO)	
000264	016716	000000G		MOV	CMD.SLOT,(SP)	3189
000270	012746	000054		MOV	#54,-(SP)	
000274	004767	000000G		JSR	PC,BL#MUL	
000300	005060	000016G		CLR	SND.ENVELOPE+16(RO)	
000304	016700	000000G		MOV	CMD.SLOT,RO	3193
000310	006300			ASL	RO	
000312	006300			ASL	RO	
000314	066700	000000G		ADD	SEND.RING,RO	
000320	052760	100000	000002	BIS	#100000,2(RO)	
000326	017766	000000G	000030	MOV	@RC25.ADDR,30(SP)	3197
000334	016600	000030		MOV	30(SP),RO	
000340	004767	000000V		JSR	PC,GET.CMD.SLOT	3201
000344	004767	000000V		JSR	PC,REC.STATUS	3206
000350	006000			ROR	RO	
000352	103005			BCC	1#	
000354	062706	000030		ADD	#30,SP	3209
000360	016700	000000G		MOV	RET.STATUS,RO	3208
000364	000423			BR	2#	
000366	005367	000000G	1#:	DEC	RES.SLOT	3212
000372	016700	000000G		MOV	RES.SLOT,RO	3214
000376	000300			SWAB	RO	
000400	106000			RORB	RO	
000402	006000			ROR	RO	
000404	006000			ROR	RO	
000406	142700	000077		BICB	#77,RO	
000412	016067	000022G	000000G	MOV	REC.ENVELOPE+22(RO),RET.UNIT.FLAG ;	
000420	004767	000000V		JSR	PC,GET.RES.SLOT	3215
000424	062706	000030		ADD	#30,SP	3216
000430	016700	000000G		MOV	RET.STATUS,RO	3168
000434	005726		2#:	TST	(SP)	3146
000436	000207			RTS	PC	

; Routine Size: 144 words, Routine Base: AB#CODE + 10444
 ; Maximum stack depth per invocation: 14 words

; 3218 1
 ; 3219 1
 ; 3220 1

```

: 3221 1 global routine GET_CMD_SLOT =
: 3222 1 :
: 3223 1 : **
: 3224 1 : FUNCTIONAL DESCRIPTION:
: 3225 1 : THIS ROUTINE ASSIGNS A COMMAND SLOT NUMBER FOR THE COMMUNICATION
: 3226 1 : RING, IT WILL WRAP AROUND, AS THE SLOT NUMBER REACHED TO THE BOTTOM.
: 3227 1 : --
: 3228 1 : begin
: 3229 2 :
: 3230 2 if .CMD_SLOT equl SND_ALLOCATE - 1 : IS SLOT # REACHED TO THE END
: 3231 2 then : YES
: 3232 2 CMD_SLOT = ZERO : WRAP AROUND THE COMMAND RING
: 3233 2 else : ELSE
: 3234 2 .. CMD_SLOT = .CMD_SLOT + 1; : INCREMENT THE CMD SLOT NUMBER
: 3235 2 :
: 3236 2 :
: 3237 2 if .SEND_RING [.CMD_SLOT, OWN_BIT] equl PORT_OWNED
: 3238 2 then
: 3239 2 return TRUE : THIS SLOT IS NOT OWNED BY
: 3240 2 : HOST YET. SO WAIT.
: 3241 2 else
: 3242 2 SEND_RING [.CMD_SLOT, FLAG_BIT] = ZERO; : CLEAR CMD_RING FLAG BIT
: 3243 2 :
: 3244 2 return FALSE;
: 3245 1 end;

```

Address	Hex	OpCode	Comment	Label	Address
000000	026727	000000G 000017	GET.CMD.SLOT::	GET.CMD.SLOT AZTEC GLOBAL ROUTINE	
			CMP	CMD.SLOT,#17	3230
000006	001003		BNE	1\$	
000010	005067	000000G	CLR	CMD.SLOT	3232
000014	000402		BR	2\$	3230
000016	005267	000000G	1\$: INC	CMD.SLOT	3234
000022	016700	000000G	2\$: MOV	CMD.SLOT,RO	3237
000026	006300		ASL	RO	
000030	006300		ASL	RO	
000032	066700	000000G	ADD	SEND.RING,RO	
000036	005760	000002	TST	2(RO)	
000042	100003		BPL	3\$	
000044	012700	000001	MOV	#1,RO	3239
000050	000207		RTS	PC	
000052	016700	000000G	3\$: MOV	CMD.SLOT,RO	3242
000056	006300		ASL	RO	
000060	006300		ASL	RO	
000062	066700	000000G	ADD	SEND.RING,RO	
000066	042760	040000 000002	BIC	#40000,2(RO)	
000074	005000		CLR	RO	3228
000076	000207		RTS	PC	3221

; Routine Size: 32 words, Routine Base: AB\$CODE + 11104
; Maximum stack depth per invocation: 0 words

```

: 3246 1
: 3247 1 !
: 3248 1

```


ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0174
Page 79
(28)

```

: 3249 1 global routine GET_RES_SLOT : novalue =
: 3250 1 **
: 3251 1 ! FUNCTIONAL DESCRIPTION:
: 3252 1 !
: 3253 1 ! THIS ROUTINE ASSIGNS A RESPONSE SLOT NUMBER FOR THE COMMUNICATION
: 3254 1 ! RING, IT WILL WRAP AROUND, AS THE SLOT NUMBER REACHED TO THE BOTTOM.
: 3255 1 !
: 3256 2 --
: 3257 2 begin
: 3258 2 RECEIVE_RING [.RES_SLOT, FLAG_BIT] = ZERO; ! CLEAR RECEIVE RING FLAG BIT
: 3259 2
: 3260 2 if .RES_SLOT eqlo REC_ALLOCATE - 1 ! IS SLOT # REACHED TO THE END?
: 3261 2 then ! YES, THEN
: 3262 2 RES_SLOT = ZERO ! WRAP AROUND THE RESPONSE RING
: 3263 2 else ! ELSE
: 3264 2 RES_SLOT = .RES_SLOT + 1; ! INCREMENT THE RES SLOT NUMBER
: 3265 2
: 3266 1 end;

```

000000	016700	000000G	.SBTTL	GET.RES.SLOT AZTEC GLOBAL ROUTINE	
			GET.RES.SLOT::		
000004	006300		MOV	RES.SLOT,RO	3258
000006	006300		ASL	RO	
000010	066700	000000G	ASL	RO	
000014	042760	040000 000002	ADD	RECEIVE_RING,RO	
000022	026727	000000G 000017	BIC	#40000,2(RO)	
000030	001003		CMP	RES.SLOT,#17	3260
000032	005067	000000G	BNE	1\$	
000036	000207		CLR	RES.SLOT	3262
000040	005267	000000G	RTS	PC	3260
000044	000207		1\$: INC	RES.SLOT	3264
			RTS	PC	3249

; Routine Size: 19 words, Routine Base: AB\$CODE + 11204
; Maximum stack depth per invocation: 0 words

```

: 3267 1
: 3268 1 !
: 3269 1

```

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC2.816;1

SEQ 0175
Page 80
(29)

```

: 3270 1 global routine READ_FILL_CMD : novalue =
: 3271 1 **
: 3272 1 : FUNCTIONAL DESCRIPTION:
: 3273 1 :
: 3274 1 : THIS ROUTINE IS USED TO FILL SND ENVELOPE WITH CMD LREF AND LBN ST
: 3275 1 : AND ALSO GIVE THE CMD SLOT TO PORT IMMEDIATELY AFTER RECEIVING IT
: 3276 1 : FROM PORT. THIS ROUTINE WILL BE CALLED WHEN READ_CMD WAS ORIGINALLY
: 3277 1 : ISSUED AND THE SND ENVELOPE IS SUPPOSED TO BE IN TACT EXCEPT FOR
: 3278 1 : THE ONES THAT ARE TOUCHED HERE. THIS ROUTINE IS CALLED IN THE
: 3279 1 : TIMING TESTS ONLY.
: 3280 1 :
: 3281 1 : --
: 3282 1 : FORMAL PARAMETERS :
: 3283 1 :
: 3284 1 : IMPLICIT INPUTS : LBN_ST, CMD_SLOT
: 3285 1 :
: 3286 1 : IMPLICIT OUTPUTS :
: 3287 1 :
: 3288 1 : COMPLETION CODES :
: 3289 1 :
: 3290 1 : SIDE EFFECTS :
: 3291 1 :
: 3292 2 begin
: 3293 2 SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .LBN_ST; ! GIVE LBN IN COMMAND REF SO THAT
: 3294 2 ! LBN RECEIVED IN THE RECEIVE ENVELOPE
: 3295 2 ! CAN BE TAKEN AS FAILING LBN, IF THERE
: 3296 2 ! WAS ANY ERROR.
: 3297 2 SND_ENVELOPE [.CMD_SLOT, LBN_LO] = .LBN_ST; ! LBN TO SEEK
: 3298 2 SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED; ! GIVE THE SLOT TO PORT
: 3299 2 return;
: 3300 1 end;

```

```

.SBTTL READ.FILL.CMD AZTEC GLOBAL ROUTINE
000000 016746 000000G READ.FILL.CMD::
000004 012746 000054 MOV CMD_SLOT, -(SP) ; 3293
000010 004767 000000G MOV #54, -(SP)
000014 016760 000000G 000004G JSR PC, BL $MUL
000022 016716 000000G MOV LBN_ST, SND.ENVELOPE+4(R0)
000026 012746 000054 MOV CMD_SLOT, (SP) ; 3297
000032 004767 000000G MOV #54, -(SP)
000036 016760 000000G 000040G JSR PC, BL $MUL
000044 016700 000000G MOV LBN_ST, SND.ENVELOPE+40(R0)
000050 006300 000000G MOV CMD_SLOT, R0 ; 3298
000052 006300 ASL R0
000054 066700 000000G ADD SEND_RING, R0
000060 052760 100000 000002 BIS #100000, 2(R0)
000066 062706 000006 ADD #6, SP ; 3299
000072 000207 RTS PC ; 3270

```

; Routine Size: 30 words, Routine Base: AB\$CODE + 11252
; Maximum stack depth per invocation: 4 words

; 3301 1

```

: 3302 1  global routine REC_STATUS =
: 3303 1
: 3304 1  !**
: 3305 1  ! FUNCTIONAL DESCRIPTION :
: 3306 1  !
: 3307 1  ! THIS ROUTINE READS THE STATUS OF END MESSAGE PACKET AND RETURN
: 3308 1  ! THE SLOT TO THE CONTROLLER. IF ERROR, A STATUS FLAG AND ERROR CODE
: 3309 1  ! ARE SENT TO THE CALLER.
: 3310 1  !
: 3311 1  ! IF STATUS BIT INDICATES UNSUCCESS, THEN AN ERROR MESSAGE WILL
: 3312 1  ! BE REPORTED BY THE TEST MODULE.
: 3313 1  !
: 3314 1  ! FORMAL PARAMETERS :
: 3315 1  ! IMPLICIT INPUTS :
: 3316 1  !           IN_BOUND
: 3317 1  ! IMPLICIT OUTPUTS :
: 3318 1  !           RET_STATUS, ER_STATUS
: 3319 1  !
: 3320 1  ! COMPLETION CODES :
: 3321 1  !           RET_STATUS
: 3322 1  ! SIDE EFFECTS :
: 3323 1  !
: 3324 1  ! --
: 3325 2  begin
: 3326 2  !
: 3327 2  ! WAITING FOR THE CONTROLLER TO FILL THE DESCRIPTOR AND RELEASING
: 3328 2  ! IT TO THE HOST, IF WAITING TIME EXPIRED THEN AN ERROR WILL BE REPORTED.
: 3329 2  !
: 3330 2  !
: 3331 2  local
: 3332 2  I;
: 3333 2  !
: 3334 2  I = .IN_BOUND;           ! SAVE RECEIVE COUNT
: 3335 2  !
: 3336 2  incru COUNT from 0 to 60000 do           ! SET TIME OUT RANGE
: 3337 3  begin                               ! VER:C
: 3338 3  !
: 3339 3  if .RECEIVE_RING [.RES_SLOT, OWN_BIT] equ 0 ! IF HOST OWNS THE SLOT
: 3340 3  then                               ! THEN
: 3341 4  begin
: 3342 4  !
: 3343 5  if (.REC_ENVELOPE [.RES_SLOT, STA_CODE] ! READ THE STATUS BITS
: 3344 5  nequ ZERO)
: 3345 4  then                               ! IF ERROR
: 3346 5  begin                               ! THEN FLAG THE ERROR
: 3347 5  LBN_ST = .REC_ENVELOPE [.RES_SLOT, CMD_LREF]; !GET CMD REF FAILING
: 3348 5  ER_STATUS = .REC_ENVELOPE [.RES_SLOT, STATUS]; ! SAVE ERROR CODE
: 3349 5  IN_BOUND = .IN_BOUND + 1;           ! RECEIVE COUNT TOTAL
: 3350 5  RET_STATUS = RSE_CODE;             ! REPORT THE ERROR & SET STATUS
: 3351 5  return .RET_STATUS;               ! SET ERROR FLAG
: 3352 5  end
: 3353 4  else
: 3354 5  begin
: 3355 5  IN_BOUND = .IN_BOUND + 1;           ! RECEIVE COUNT TOTAL
: 3356 5  RECEIVE_RING [.RES_SLOT, OWN_BIT] = ONE; ! PORT OWN THE RING
: 3357 5  GET_RES_SLOT ();                   ! GET NEXT RESPONSE SLOT #
: 3358 5  RET_STATUS = PAS_CODE;             ! CLEAR STATUS

```

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0177
Page 82
(30)

```

: 3359 4          end;
: 3360 4
: 3361 4          end
: 3362 3          else
: 3363 4          begin
: 3364 4
: 3365 4          if (.IN_BOUND nequ .I) then return .RET_STATUS;      !IF YOU RECIVE AT
: 3366 4          ! VER:C
: 3367 4          ! LEAST A SLOT THIS TIME THEN RETURN.
: 3368 4          ! IF NOT, WAIT AND TRY BACK.
: 3369 4          DELAY (30);
: 3370 3          end;
: 3371 3          BREAK;          ! WATCH FOR CONTROL C.
: 3372 2          end;
: 3373 2
: 3374 2          ! IF THE MAX TIME IN THE LOOP WAS ELAPSED, THEN READ RCSA
: 3375 2          ! FOR POSSIBLE ERROR INFO.
: 3376 2          RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];      ! GET RCSA DATA
: 3377 2
: 3378 2          if .RC25_DATA [RCSA, RCSA_ER]          ! CHECK SA REG.ERROR BIT
: 3379 2          then
: 3380 3          begin
: 3381 3          RET_STATUS = PFE_CODE;          ! SAVE THE PORT/CTLER FAILURE
: 3382 3          return .RET_STATUS;
: 3383 3          end
: 3384 2          else
: 3385 3          begin
: 3386 3          RET_STATUS = CTO_CODE;          ! SET TIME EXPIRED IN STATUS BUF
: 3387 3          return .RET_STATUS;          ! RETURN WITH A TIME EXPIRED FLAG
: 3388 2          end;
: 3389 2
: 3390 1          end;

```

Address	Offset	Label	Instruction	Comment	Line No.
000000	004167	000000G	REC.STATUS::		
000004	024646		JSR	R1,\$SAVE3	3302
000006	016703	000000G	CMP	-(SP),-(SP)	
000012	005002		MOV	IN.BOUND,R3	3334
000014	016700	000000G	CLR	R2	3336
000020	006300		MOV	RES.SLOT,R0	3339
000022	006300		ASL	R0	
000024	066700	000000G	ASL	R0	
000030	032760	100000 000002	ADD	RECEIVE.RING,R0	
000036	001072		BIT	#100000,2(R0)	
000040	016700	000000G	BNE	3\$	
000044	000300		MOV	RES.SLOT,R0	3343
000046	106000		SWAB	R0	
000050	006000		RORB	R0	
000052	006000		ROR	R0	
000054	142700	000077	ROR	R0	
000060	132760	000037 000016G	BICB	#77,R0	
000066	001436		BITB	#37,REC.ENVELOPE+16(R0)	
000070	016700	000000G	BEQ	2\$	
000074	000300		MOV	RES.SLOT,R0	3347
000076	106000		SWAB	R0	
			RORB	R0	

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 B118-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0178
Page 83
(30)

000100	006000			ROR	RO		
000102	006000			ROR	RO		
000104	142700	000077		BICB	#77,RO		
000110	016067	000004G	000000G	MOV	REC.ENVELOPE+4(RO),LBN.ST		
000116	016700	000000G		MOV	RES.SLOT,RO		3348
000122	000300			SWAB	RO		
000124	106000			RORB	RO		
000126	006000			ROR	RO		
000130	006000			ROR	RO		
000132	142700	000077		BICB	#77,RO		
000136	016067	000016G	000000G	MOV	REC.ENVELOPE+16(RO),ER.STATUS		
000144	005267	000000G		INC	IN.BOUND		3349
000150	012767	000031	000000G	MOV	#31,RET.STATUS		3350
000156	016700	000000G		MOV	RET.STATUS,RO		3346
000162	000471			BR	10\$		
000164	005267	000000G	2\$:	INC	IN.BOUND		3355
000170	016700	000000G		MOV	RES.SLOT,RO		3356
000174	006300			ASL	RO		
000176	006300			ASL	RO		
000200	066700	000000G		ADD	RECEIVE.RING,RO		
000204	052760	100000	000002	BIS	#100000,2(RO)		
000212	004767	177420		JSR	PC,GET.RES.SLOT		3357
000216	005067	000000G		CLR	RET.STATUS		3358
000222	000422			BR	8\$		3339
000224	026703	000000G	3\$:	CMP	IN.BOUND,R3		3365
000230	001403			BEQ	4\$		
000232	016700	000000G		MOV	RET.STATUS,RO		
000236	000443			BR	10\$		
000240	012701	000036	4\$:	MOV	#36,R1		3368
000244	001411		5\$:	BEQ	8\$		
000246	016700	000000G		MOV	L\$DLY,RO		
000252	001404			BEQ	7\$		
000254	005066	000002	6\$:	CLR	2(SP)		
000260	005300			DEC	RO		
000262	001374			BNE	6\$		
000264	005301		7\$:	DEC	R1		
000266	000766			BR	5\$		
000270	104422		8\$:	TRAP	22		3369
000272	005202			INC	R2		3336
000274	020227	165140		CMP	R2,#165140		
000300	101645			BLOS	1\$		
000302	016700	000000G		MOV	RC25.ADDR,RO		3376
000306	016016	000002		MOV	2(RO),(SP)		
000312	011667	000002G		MOV	(SP),RC25.DATA+2		
000316	100006			BPL	9\$		3378
000320	012767	000021	000000G	MOV	#21,RET.STATUS		3381
000326	016700	000000G		MOV	RET.STATUS,RO		3385
000332	000405			BR	10\$		
000334	012767	000011	000000G	MOV	#11,RET.STATUS		3386
000342	016700	000000G		MOV	RET.STATUS,RO		3385
000346	022626		10\$:	CMP	(SP)+,(SP)+		3302
000350	000207			RTS	PC		

; Routine Size: 117 words, Routine Base: AB\$CODE + 11346
; Maximum stack depth per invocation: 8 words

K14

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0179
Page 84
(30)

; 3391 1

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0180
Page 85
(31)

```

: 3392 1 global routine RANDOM_NUM : novalue =
: 3393 1 **
: 3394 1 FUNCTIONAL DESCRIPTION:
: 3395 1
: 3396 1 THIS ROUTINE SUPPLIES AN LBN NUMBER AT RANDOM.
: 3397 1 P2 IS THE SEED. P3 IS LBN NUMBER SUPPLIED.
: 3398 1
: 3399 1 --
: 3400 2 begin
: 3401 2 P2 = (.P2*377 + 6925) mod 32767;          ! RANDOM SEED
: 3402 2
: 3403 2 if .P2 gtru .END_LBN
: 3404 2 then
: 3405 2 P3 = .P2 and .END_LBN                ! P3 IS LBN
: 3406 2 else
: 3407 2 P3 = .P2;
: 3408 2
: 3409 2 P2 = not .P2;                          ! UPDATED SEED
: 3410 1 return;
:                                     end;

```

```

000000 010146 .SBTTL RANDOM.NUM AZTEC GLOBAL ROUTINE
000002 016746 000000G RANDOM.NUM::
000006 012746 000571 MOV R1,-(SP) ; 3392
000012 004767 000000G MOV P2,-(SP) ; 3400
000016 010016 JSR #571,-(SP)
000020 062716 015415 MOV PC,BL#MUL
000024 012746 077777 MOV R0,(SP)
000030 004767 000000G ADD #15415,(SP)
000034 010067 000000G MOV #77777,-(SP)
000040 026767 000000G 000000G JSR PC,BL#MOD
000046 101411 CMP R0,P2 ; 3402
000050 016700 000000G BLOS 1# ;
000054 016701 000000G MOV P2,R0 ; 3404
000060 005101 MOV END_LBN,R1
000062 040100 COM R1
000064 010067 000000G BIC R1,R0
000070 000403 MOV R0,P3
000072 016767 000000G 000000G BR 2# ; 3402
000100 005167 000000G 1#: MOV P2,P3 ; 3406
000104 062706 000006 2#: COM P2 ; 3408
000110 012601 ADD #6,SP ; 3409
000112 000207 MOV (SP),R1 ; 3392
RTS PC ;

```

; Routine Size: 38 words, Routine Base: AB\$CODE + 11720
; Maximum stack depth per invocation: 5 words

; 3411 1

```

: 3412 1 global routine EXAM_DATA : =
: 3413 1
: 3414 1 FUNCTIONAL DESCRIPTION:
: 3415 1
: 3416 1 THE FUNCTION OF THIS ROUTINE IS TO EXAMINE THE
: 3417 1 FREE MEMORY FOR EXPECTED DATA.
: 3418 1
: 3419 1 IMPLICIT INPUTS:
: 3420 1 H_SADD
: 3421 1 BUF_LENGTH
: 3422 1 TIP
: 3423 1 IMPLICIT OUTPUTS:
: 3424 1 RETURN STATUS
: 3425 1
: 3426 1 SIDE EFFECTS:
: 3427 1 - NONE -
: 3428 1
: 3429 1 -- begin
: 3430 1
: 3431 1 local
: 3432 1 PATTERN,
: 3433 1 FLAG;
: 3434 1
: 3435 1 FLAG = ZERO; : INIT ERROR FLAG
: 3436 1 TEMP = .H_SADD; : SAVE ADDR. IN TEMP. BUFFER
: 3437 1 H_EADD = .H_SADD - 2 * (.BUF_LENGTH+2); : END OF FREE HOST MEMORY
: 3438 1 PATTERN = .TIP; : PUT PATTERN FOR COMPARE
: 3439 1
: 3440 1 incru COUNT from .H_SADD to .H_EADD by 2 do : EXAMINE CONTENTS OF MEMORY
: 3441 1 begin
: 3442 1
: 3443 1 if .TIP ealu 1 then PATTERN = ( not .TEMP); : BASED ON THE VALUE RECEIVED
: 3444 1
: 3445 1 if .TIP ealu 2 then PATTERN = .TEMP; : IN TIP, SET UP PATTERN FOR
: 3446 1
: 3447 1 if ..TEMP nequ .PATTERN : COMPARISION.
: 3448 1 then
: 3449 1 begin
: 3450 1 FLAG = TRUE;
: 3451 1 TIP = .PATTERN;
: 3452 1 exitloop;
: 3453 1 end;
: 3454 1
: 3455 1 TEMP = .TEMP + 2;
: 3456 1 end;
: 3457 1
: 3458 1 if .FLAG : IF ERROR WAS FOUND THEN
: 3459 1 then
: 3460 1 begin : GET ERROR DATA
: 3461 1 P_MASK = 2; : FOR TEST MODULE
: 3462 1 P1 = FMT2;
: 3463 1 P2 = ZERO;
: 3464 1 P3 = ZERO;
: 3465 1 P4 = .TIP;
: 3466 1 P5 = ..TEMP;
: 3467 1 P6 = .TEMP;
: 3468 1 return RET_STATUS = TRUE;

```


N14

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 B11ss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC2.B16;1

SEQ 0182
Page 87
(32)

```

: 3469 3
: 3470 2
: 3471 2
: 3472 2
: 3473 1
end
else
return RET_STATUS = FALSE;
end;

```

! GOOD STATUS

000000	004167	000000G	.SBTTL	EXAM.DATA	AZTEC GLOBAL ROUTINE	
000004	005003			EXAM.DATA::	JSR R1, \$SAVE3	3412
000006	016767	000000G			CLR R3	3435
000014	016700	000000G			MOV H.SADD, TEMP	3436
000020	006300				MOV BUF.LENGTH, R0	3437
000022	066700	000000G			ASL R0	
000026	010067	000000G			ADD H.SADD, R0	
000032	162767	000002	000000G		MOV R0, H.EADD	
000040	016700	000000G			SUB #2, H.EADD	
000044	016702	000000G			MOV TIP, R0	3438
000050	016701	000000G			MOV H.EADD, R2	3440
000054	000432				MOV H.SADD, R1	3443
000056	026727	000000G	000001	1\$:	BR 5\$	
000064	001003				CMP TIP, #1	3443
000066	016700	000000G			BNE 2\$	
000072	005100				MOV TEMP, R0	3445
000074	026727	000000G	000002	2\$:	COM R0	
000102	001002				CMP TIP, #2	3445
000104	016700	000000G			BNE 3\$	
000110	027700	000000G			MOV TEMP, R0	3447
000114	001405				CMP @TEMP, R0	3447
000116	012703	000001			BEQ 4\$	
000122	010067	000000G			MOV #1, R3	3450
000126	000407				MOV R0, TIP	3451
000130	062767	000002	000000G	4\$:	BR 6\$	3449
000136	062701	000002			ADD #2, TEMP	3455
000142	020102				ADD #2, R1	3440
000144	101744				CMP R1, R2	3440
000146	006003				BLOS 1\$	
000150	103030				ROR R3	3458
000152	112767	000002	000000G		BCC 7\$	
000160	012767	000000G	000000G		MOVB #2, P.MASK	3461
000166	005067	000000G			MOV #FMT2, P1	3462
000172	005067	000000G			CLR P2	3463
000176	016767	000000G	000000G		CLR P3	3464
000204	017767	000000G	000000G		MOV TIP, P4	3465
000212	016767	000000G	000000G		MOV @TEMP, P5	3466
000220	012700	000001			MOV TEMP, P6	3467
000224	010067	000000G			MOV #1, R0	3468
000230	000207				MOV R0, RET.STATUS	
000232	005067	000000G			RTS PC	3471
000236	005000				CLR RET.STATUS	
000240	000207				CLR R0	3412
					RTS PC	

; Routine Size: 81 words, Routine Base: AB\$CODE + 12034
; Maximum stack depth per invocation: 5 words

B15

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0183
Page 88
(32)

: 3474 1
: 3475 1

!<BLF/PAGE>

C15

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0184
Page 89
(33)

: 3476 1

```

: 3477 1  global routine AZTEC_READY =
: 3478 1
: 3479 1
: 3480 1  **
: 3481 1  FUNCTIONAL DESCRIPTIONS:
: 3482 1  THIS ROUTINE CALLS OTHER ROUTINES TO GET THE AZTEC READY
: 3483 1  PERFORMS THE FOLLOWING:
: 3484 1
: 3485 1  1. DEFINE INITIALIZATION CONSTANTS AND INITIALIZE VARIABLES.
: 3486 1
: 3487 1  2. DO STEP 1 THROUGH STEP 4 CHECK FOR ANY ERRORS
: 3488 1  IN EACH STEP.
: 3489 1
: 3490 1  3. SET UP COMMUNICATION AREA'S.
: 3491 1
: 3492 1  4. SET MOST SETTABLE UNIT CHARACTERISTICS AND OBTAIN THOSE
: 3493 1  UNIT CHARACTERISTICS THAT ARE ESSENTIAL FOR PROPER CLASS
: 3494 1  DRIVER OPERATION.
: 3495 1
: 3496 1  5. BRING A UNIT "UNIT-ONLINE. THE UNIT IS SPUN-UP, IF NECESSARY,
: 3497 1  AND ITS HEADS ARE LOADED PRIOR TO RETURNING THE ONLINE COMMANDS'S
: 3498 1  END MESSAGE.
: 3499 1
: 3500 1  6. IF TIP CONTAINS TEST NUMBER OF 14 THEN THIS ROUTINE SKIPS
: 3501 1  ON_LINE CALL, SO THAT THIS COULD BE DONE IN THE TEST MODULE.
: 3502 1
: 3503 1  FORMAL PARAMETERS:
: 3504 1  -NONE -
: 3505 1
: 3506 1  IMPLICIT INPUTS:
: 3507 1
: 3508 1  IMPLICIT OUTPUTS:
: 3509 1  AS A RESULT OF THIS ROUTINE THE COMMUNICATION AREA WILL
: 3510 1  BE INITIALIZED AND UNIT IS SPUN-UP.
: 3511 1
: 3512 1  COMPLETION CODES:
: 3513 1
: 3514 1  SIDE EFFECTS:
: 3515 1  - NONE -
: 3516 2  begin
: 3517 2  B_MASK = %0'17';           ! SET MASK BIT FOR COMPLETE INIT.
: 3518 2  DATA1<15, 1> = TRUE;      ! SET BIT 15 FOR STEP-1 WRITE
: 3519 2  DATA1<14, 1> = 0;        ! NO DIAGNOSTIC WRAP MODE
: 3520 2  DATA1<11, 3> = SND_SIZ;   ! SET UP COMMAND RINGS LENGTH
: 3521 2  DATA1<8, 3> = REC_SIZ;   ! SET RESPONSE RING LENGTH
: 3522 2  DATA1<7, 1> = 0;        ! DISABLE INTERRUPT
: 3523 2  DATA1<0, 7> = 0;        ! LOAD NO VECTOR ADDRESS
: 3524 2  DATA2 = RINGBASE;       ! LOAD COMMUNICATIONS AREA ADDRESS
: 3525 2  DATA3 = ZERO;         ! HI-ORDER ADDR = ZERO
: 3526 2  DATA4 = %0'177403';     ! "LAST FAIL" PACKET RESPONSE BIT SET
: 3527 2  CMD_SLOT = 0;         ! CLEAR COMMAND RING SLOT POINTER
: 3528 2  RES_SLOT = 0;         ! CLEAR RESPONSE RING SLOT POINTER
: 3529 2  CMOD = MD_EXP;        ! SET EXPRESS BIT FOR READ
: 3530 2  ! COMMAND MODIFIER.
: 3531 2  IN_BOUND = 0;         ! VER:C
: 3532 2
: 3533 2  if AZP_INIT ( )         ! DO STEP INIT AND CHECK FOR ERROR

```

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0186
Page 91
(34)

```

: 3534 2      then
: 3535 2      return .RET_STATUS;
: 3536 2
: 3537 2      if INIT_COM_AREA ( )      ! INIT THE COMMUNICATION AREA
: 3538 2      then                      ! ERROR ?
: 3539 2      return .RET_STATUS;
: 3540 2
: 3541 2      CMD_REF = .CMD_SLOT;      ! SET COMMAND REFERENCE TO 1
: 3542 2
: 3543 2      if SET_CNTRLR_CHAR ( )      ! ISSUE SET CONTROLLER CHAR CMD
: 3544 2      then
: 3545 2      return .RET_STATUS;      ! IF COMMAND FAILED
: 3546 2
: 3547 2      CMD_REF = .CMD_SLOT;      ! SET COMMAND REFERENCE TO 2
: 3548 2
: 3549 2      if .TIP eq 14 then return RET_STATUS = FALSE;      ! TEST 14 WILL DO ONLINE COMMAND
: 3550 2
: 3551 2      ! IN THE MAIN LINE OF CODE
: 3552 2
: 3553 2      if ON_LINE ( )              ! ISSUE ON LINE COMMAND
: 3554 2      then
: 3555 2      return .RET_STATUS;
: 3556 2
: 3557 2      return RET_STATUS = FALSE;
: 3558 1      end;
    
```

Address	Offset	Hex	OpCode	Comment	Line
000000	112767	000017	000000G	.SBTTL AZTEC.READY AZTEC GLOBAL ROUTINE	
000006	012767	122000	000000G	AZTEC.READY::	
000014	012767	000000G	000000G	MOV #17,B.MASK ;	3517
000022	005067	000000G		MOV #122000,DATA1 ;	3523
000026	012767	177403	000000G	MOV #RINGBASE,DATA2 ;	3524
000034	005067	000000G		CLR DATA3 ;	3525
000040	005067	000000G		MOV #-375,DATA4 ;	3526
000044	012767	100000	000000G	CLR CMD.SLOT ;	3527
000052	005067	000000G		CLR RES.SLOT ;	3528
000056	004767	167256		MOV #-100000,CMOD ;	3529
000062	006000			CLR IN.BOUND ;	3531
000064	103003			JSR PC,AZP.INIT ;	3533
000066	016700	000000G		ROR RO ;	
000072	000207			BCC 1\$;	
000074	004767	170040		MOV RET.STATUS,RO ;	3535
000100	006000			RTS PC ;	
000102	103003			JSR PC,INIT.COM.AREA ;	3537
000104	016700	000000G		ROR RO ;	
000110	000207			BCC 2\$;	
000112	016767	000000G	000000G	MOV RET.STATUS,RO ;	3539
000120	004767	173074		RTS PC ;	
000124	006000			MOV CMD.SLOT,CMD.REF ;	3541
000126	103003			JSR PC,SET.CNTRLR.CHAR ;	3543
000130	016700	000000G		ROR RO ;	
000134	000207			BCC 3\$;	
000136	016767	000000G	000000G	MOV RET.STATUS,RO ;	3545
000144	026727	000000G	000016	RTS PC ;	
000152	001004			MOV CMD.SLOT,CMD.REF ;	3547
				CMP TIP,#16 ;	3549
				BNE 4\$;	

F15

Address	Label	Value	Op	Op	Op	Op	Op	Op
000154	005067	000000G	CLR	RET.STATUS				
000160	005000		CLR	RO				
000162	000207		RTS	PC				
000164	004767	174224	4\$: JSR	PC,ON.LINE				3553
000170	006000		ROR	RO				
000172	103003		BCC	5\$				
000174	016700	000000G	MOV	RET.STATUS,RO				3555
000200	000207		RTS	PC				
000202	005067	000000G	5\$: CLR	RET.STATUS				3557
000206	005000		CLR	RO				3516
000210	000207		RTS	PC				3477

: Routine Size: 69 words, Routine Base: AB\$CODE + 12276
: Maximum stack depth per invocation: 1 word

: 3559 1

ZRCFB2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0188
Page 93
(35)

```

: 3560 1 global routine DO_RETRIES : novalue =
: 3561 1
: 3562 1 !! COME HERE ON AN ERROR AND KEEP TRACK OF RETRIES.
: 3563 1 !! IF NECESSARY DROP UNIT UNDER TEST.
: 3564 1 !!
: 3565 2 begin
: 3566 2 NUM_RETRIES = .NUM_RETRIES + 1;
: 3567 2
: 3568 3 if (.NUM_RETRIES lequ .SWP_RETRIES)
: 3569 3 then
: 3570 3 begin
: 3571 3 PRINTB (FMT%A, .NUM_RETRIES);
: 3572 3 end
: 3573 2 else
: 3574 3 begin
: 3575 3 RETRIES = FALSE;
: 3576 3
: 3577 3 if not .SWP_CONTINUE
: 3578 3 then
: 3579 4 begin
: 3580 4 DODU (.LOG_UNIT);
: 3581 4 DOCLN;
: 3582 3 end;
: 3583 3
: 3584 2 end;
: 3585 2
: 3586 1 end;

```

			SBTTL	DO.RETRIES AZTEC GLOBAL ROUTINE	
000000	005267	000000G	DO.RETRIES::	INC	NUM.RETRIES ; 3566
000004	026767	000000G 000000G		CMP	NUM.RETRIES,SWP.RETRIES ; 3568
000012	101013			BHI	1\$;
000014	016746	000000G		MOV	NUM.RETRIES,-(SP) ; 3571
000020	012746	000000G		MOV	#FMT%A,-(SP) ;
000024	012746	000002		MOV	#2,-(SP) ;
000030	010600			MOV	SP,RO ; SP,*
000032	104414			TRAP	14 ;
000034	062706	000006		ADD	#6,SP ; 3570
000040	000207			RTS	PC ; 3568
000042	005067	000000G	1\$:	CLR	RETRIES ; 3575
000046	032767	000001 000000G		BIT	#1,SWP.CONTINUE ; 3577
000054	001004			BNE	2\$;
000056	016700	000000G		MOV	LOG.UNIT,RO ; 3580
000062	104451			TRAP	51 ;
000064	104444			TRAP	44 ;
000066	000207		2\$:	RTS	PC ; 3560

; Routine Size: 28 words, Routine Base: AB\$CODE + 12510
; Maximum stack depth per invocation: 5 words

; 3587 1

```

: 3588 1 global routine DECODE : novalue = !Decodes failing SA reg data
: 3589 1
: 3590 1
: 3591 1
: 3592 1
: 3593 1
: 3594 1
: 3595 1
: 3596 1
: 3597 1
: 3598 1
: 3599 1
: 3600 1
: 3601 1
: 3602 1
: 3603 1
: 3604 1
: 3605 1
: 3606 1
: 3607 1
: 3608 1
: 3609 1
: 3610 1
: 3611 1
: 3612 1
: 3613 1
: 3614 1
: 3615 1
: 3616 1
: 3617 1
: 3618 1
: 3619 1
: 3620 1
: 3621 1
: 3622 1
: 3623 1
: 3624 1
: 3625 1
: 3626 1
: 3627 1
: 3628 1
: 3629 1
: 3630 2
: 3631 2
: 3632 2
: 3633 2
: 3634 2
: 3635 2
: 3636 2
: 3637 2
: 3638 2
: 3639 2
: 3640 2
: 3641 2
: 3642 2
: 3643 2
: 3644 2

```

global routine DECODE : novalue = !Decodes failing SA reg data

```

!
! **
! Functional Description :
! Due to the implimentation of the DUP and UQ Port protocol there
! are two levels at which an issued command to a port/controller
! can fail and they are:
!
! 1. The issued command can time out.
!
! 2. An error can be posted in SA register bit 15 by the port to
! report an error.
!
! 3. The issued command to the port/controller can be executed
! correctly without any errors but the response packet status
! field could have an error or status other than success posted.
!
! This routine will then be called when the return from a queued
! command comes back with an error code or non successfull status
! code. This is by definition when bit 0 in the returned status
! is equal to 1.
!
! Formal Parameters :
! none
!
! Implicit Inputs :
! RET_STATUS: Stored in this global storage is the returned error
! code or non-successful status code from a queued
! command.
!
! Implicit Outputs :
! none
!
! Completion Codes :
! none
!
! Side Effects :
! after execution of this routine the RC25 controller
! is initialized aborting any DM code running in the controller.
!
! --
!
! begin
!
! !
! ! Use the contents of "RET_STATUS" to select what
! ! type error or non-successful status code is to
! ! be processed.
! !
! ! --
!
! if .RET_STATUS eqlu ONE then return RET_STATUS = ZERO; ! NO ACTION IF RET_STATUS IS ONE
!
! selectoneu .RET_STATUS of
! set
! !
! ! "Port/Controller time out" error code
! !
!

```



```

: 3645 2      ! Port/Controller timed out after the specified
: 3646 2      ! time out interval.
: 3647 2      !
: 3648 2      !
: 3649 2      [CTO_CODE] :      !Code equals #o'11'
: 3650 3      begin
: 3651 3      PRINTF (.EMSG_STRUCT [MSG2]);
: 3652 2      end;
: 3653 2      !
: 3654 2      ! "Port fatal error" code
: 3655 2      !
: 3656 2      ! The error bit in the SA Register was set when
: 3657 2      ! examined. This error indicates a Port fatal error code.
: 3658 2      !
: 3659 2      !
: 3660 2      [PFE_CODE] :      !Code equals #o'21'
: 3661 3      begin
: 3662 3      TEMP = .RC25_DATA [RCSA, RCSA_ERC];
: 3663 3      PRINTB (FMT13, .TEMP);      ! PRINT RCSA ERROR CODE
: 3664 3
: 3665 3      if .TEMP gequ 200
: 3666 3      then
: 3667 4      begin
: 3668 4      PRINTF (.RC_STRUCTURE [.TEMP - 200]);      !print RCSA error code
: 3669 4      end
: 3670 3      else
: 3671 4      begin
: 3672 4      PRINTF (.PFE_STRUCT [.TEMP]);
: 3673 3      end;
: 3674 3
: 3675 3      if AZTEC_READY ()      ! Init and bring
: 3676 3      then      ! Aztec ready if
: 3677 4      begin      ! possible
: 3678 4      DODU (.LOG_UNIT);      ! otherwise, drop unit
: 3679 4      DOCLN;      ! and clean up.
: 3680 3      end;
: 3681 3
: 3682 3      return RET_STATUS = FALSE;      ! Return to caller
: 3683 2      end;
: 3684 2      !
: 3685 2      ! "Return status error" code
: 3686 2      !
: 3687 2      ! This indicates that a non-successful return status
: 3688 2      ! code was returned from an issued command.
: 3689 2      !
: 3690 2      !
: 3691 2      [RSE_CODE] :      !Code equals #o'31'
: 3692 3      begin
: 3693 3      TEMP = .ER_STATUS<0, 5>;      ! SAVE MAJORE ERROR CODE
: 3694 3      ! Look at uqport message type to determine if this is a response to
: 3695 3      ! one of the commands given or other unsolicited log packet.
: 3696 3
: 3697 3      if .REC_ENVELOPE [.RES_SLOT, MSG_TYPE] eqlu ZERO
: 3698 3      then
: 3699 4      begin
: 3700 4      PRINTF (.EMSG_STRUCT [MSG0]);
: 3701 4      !

```

```

: 3702 4      ! Look at UQPORT connection ID field to determine the type
: 3703 4      ! of response
: 3704 4      !
: 3705 4
: 3706 4      if .REC_ENVELOPE [.RES_SLOT, CONN_ID] eqv 2      ! CONN_ID = DUP
: 3707 4      then
: 3708 5          begin
: 3709 5              PRINTB (.SDUP_STRUCT [.ER_STATUS]);
: 3710 5          end
: 3711 4      else
: 3712 5          begin
: 3713 5              PRINTB (.SMSCP_STRUCT [.TEMP]);
: 3714 5      ! TRY MODULE CALL OUT BASED ON ERROR CODE FROM THE END PACKET.
: 3715 5
: 3716 5          selectoneu .TEMP of
: 3717 5              set
: 3718 5
: 3719 5              [8] :
: 3720 5                  P2 = %b'1100';          ! MAJOR ERROR CODE 10
: 3721 5                  ! MEANS DRIVE CARD AND MECHANICS
: 3722 5
: 3723 5              [9] :
: 3724 5                  P2 = %b'0011';          ! MAJOR ERROR CODE 11
: 3725 5                  ! MEANS ADAPTER CARD AND CONTROLLER
: 3726 5
: 3727 5              [10] :
: 3728 5                  P2 = %b'0010';          ! MAJOR ERROR CODE 12
: 3729 5                  ! MEANS CONTROLLER CARD
: 3730 5
: 3731 5              [11] :
: 3732 5                  P2 = %b'1100';          ! MAJOR ERROR CODE 13
: 3733 5                  ! MEANS DRIVE CARD AND MECHANICS
: 3734 5
: 3735 5              [otherwise] :
: 3736 5                  P2 = ZERO;          ! NO SELECTION
: 3737 5              tes;
: 3738 5
: 3739 5              if .P2 nequ ZERO then PRT$FRU_CALLOUT (.P2);          ! CALL OUT MODULES
: 3740 5
: 3741 4          end;
: 3742 4
: 3743 4          PRINTB (FMT14, .ER_STATUS);          ! ALSO PRINT ERROR CODE
: 3744 4          end
: 3745 3      else
: 3746 4          begin
: 3747 4              PRINTB (FMT15, .ER_STATUS);          ! GIVE LOG PACKET ERROR
: 3748 3          end;
: 3749 3
: 3750 2      end;
: 3751 2      !
: 3752 2      ! "SUPERVISOR CALL" error code
: 3753 2      !
: 3754 2
: 3755 2      [SEX_CODE] :          !Code equals %o'601'
: 3756 3          begin
: 3757 3              PRINTF (.EMSG_STRUCT [MSG1]);
: 3758 2          end;

```

```

: 3759 2
: 3760 2
: 3761 2
: 3762 2
: 3763 2
: 3764 2
: 3765 3
: 3766 3
: 3767 2
: 3768 2
: 3769 2
: 3770 2
: 3771 2
: 3772 2
: 3773 2
: 3774 2
: 3775 2
: 3776 2
: 3777 2
: 3778 1

```

```

: This is here to trap any unknown return status codes
: sent to this routine.
:
[otherwise] : !Code equals non of the above
  begin
  PRINTF (.EMSG_STRUCT [MSG3]);
  end;
tes;

: Return receive slot to the port and next slot
: if you have to continue.
RECEIVE_RING [.RES_SLOT, OWN_BIT] = ONE; ! RETURN RECEIVE SLOT TO PORT
GET_RES_SLOT (); ! GET NEXT RECEIVE SLOT
RET_STATUS = ZERO;
return;
end;

```

		.SBTTL DECODE AZTEC GLOBAL ROUTINE		
000000	010146		DECODE::MOV R1,-(SP)	3588
000002	026727	000000G 000001	CMP RET.STATUS,#1	3638
000010	001004		BNE 1\$	
000012	005067	000000G	CLR RET.STATUS	
000016	000167	000662	JMP 19\$	
000022	016701	000000G	1\$: MOV RET.STATUS,R1	3640
000026	020127	000011	CMP R1,#11	3649
000032	001011		BNE 2\$	
000034	016746	000004G	MOV EMSG.STRUCT+4,-(SP)	3651
000040	012746	000001	MOV #1,-(SP)	
000044	010600		MOV SP,R0	; SP,*
000046	104417		TRAP 17	
000050	022626		CMP (SP)*,(SP)*	3650
000052	000167	000574	JMP 18\$	3640
000056	020127	000021	2\$: CMP R1,#21	3660
000062	001063		BNE 6\$	
000064	016767	000002G 000000G	MOV RC25.DATA+2,TEMP	3662
000072	042767	174000 000000G	BIC #174000,TEMP	
000100	016746	000000G	MOV TEMP,-(SP)	3663
000104	012746	000000G	MOV #FMT13,-(SP)	
000110	012746	000002	MOV #2,-(SP)	
000114	010600		MOV SP,R0	; SP,*
000116	104414		TRAP 14	
000120	026727	000000G 000310	CMP TEMP,#310	3665
000126	103412		BLO 3\$	
000130	016700	000000G	MOV TEMP,R0	3668
000134	006300		ASL R0	
000136	016016	177160G	MOV RC.STRUCTURE-620(R0),(SP)	
000142	012746	000001	MOV #1,-(SP)	
000146	010600		MOV SP,R0	; SP,*
000150	104417		TRAP 17	
000152	000411		BR 4\$	3665
000154	016700	000000G	3\$: MOV TEMP,R0	3672
000160	006300		ASL R0	

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.316;1

SEQ 0193
Page 98
(36)

000162	016016	000000G		MOV	PFE.STRUCT(RO),(SP)		
000166	012746	000001		MOV	#1,-(SP)		
000172	010600			MOV	SP,RO	; SP,*	
000174	104417			TRAP	17		
000176	004767	177274	4\$:	JSR	PC,AZTEC.READY		3675
000202	006000			ROR	RO		
000204	103004			BCC	5\$		
000206	016700	000000G		MOV	LOG.UNIT,RO		3678
000212	104451			TRAP	51		
000214	104444			TRAP	44		
000216	005067	000000G	5\$:	CLR	RET.STATUS		3682
000222	062706	000010		ADD	#10,SP		
000226	000167	000452		JMP	19\$		3661
000232	020127	000031	6\$:	CMP	R1,#31		3691
000236	001163			BNE	16\$		
000240	016767	000000G 000000G		MOV	ER.STATUS,TEMP		3693
000246	042767	177740 000000G		BIC	#177740,TEMP		
000254	016700	000000G		MOV	RES.SLOT,RO		3697
000260	000300			SWAB	RO		
000262	106000			RORB	RO		
000264	006000			ROR	RO		
000266	006000			ROR	RO		
000270	142700	000077		BICB	#77,RO		
000274	132760	000360 000002G		BITB	#360,REC.ENVELOPE+2(RO)		
000302	001126			BNE	14\$		
000304	016746	000000G		MOV	EMSG.STRUCT,-(SP)		3700
000310	012746	000001		MOV	#1,-(SP)		
000314	010600			MOV	SP,RO	; SP,*	
000316	104417			TRAP	17		
000320	016700	000000G		MOV	RES.SLOT,RO		3706
000324	000300			SWAB	RO		
000326	106000			RORB	RO		
000330	006000			ROR	RO		
000332	006000			ROR	RO		
000334	142700	000077		BICB	#77,RO		
000340	126027	000003G 000002		CMPB	REC.ENVELOPE+3(RO),#2		
000346	001012			BNE	7\$		
000350	016700	000000G		MOV	ER.STATUS,RO		3709
000354	006300			ASL	RO		
000356	016016	000000G		MOV	SDUP.STRUCT(RO),(SP)		
000362	012746	000001		MOV	#1,-(SP)		
000366	010600			MOV	SP,RO	; SP,*	
000370	104414			TRAP	14		
000372	000460			BR	13\$		3706
000374	016700	000000G	7\$:	MOV	TEMP,RO		3713
000400	006300			ASL	RO		
000402	016016	000000G		MOV	SMSCP.STRUCT(RO),(SP)		
000406	012746	000001		MOV	#1,-(SP)		
000412	010600			MOV	SP,RO	; SP,*	
000414	104414			TRAP	14		
000416	016700	000000G		MOV	TEMP,RO		3716
000422	020027	000010		CMP	RO,#10		3719
000426	001004			BNE	8\$		
000430	012767	000014 000000G		MOV	#14,P2		3720
000436	000427			BR	12\$		3716
000440	020027	000011	8\$:	CMP	RO,#11		3723
000444	001004			BNE	9\$		

M15

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 B1188-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC2.B16;1

SEQ 0194
Page 99
(36)

000446	012767	000003	000000G		MOV	#3,P2	:	3724
000454	000420				BR	12#	:	3716
000456	020027	000012		9#:	CMP	RO,#12	:	3727
000462	001004				BNE	10#	:	
000464	012767	000002	000000G		MOV	#2,P2	:	3728
000472	000411				BR	12#	:	3716
000474	020027	000013		10#:	CMP	RO,#13	:	3731
000500	001004				BNE	11#	:	
000502	012767	000014	000000G		MOV	#14,P2	:	3732
000510	000402				BR	12#	:	3716
000512	005067	000000G		11#:	CLR	P2	:	3736
000516	005767	000000G		12#:	TST	P2	:	3739
000522	001404				BEQ	13#	:	
000524	016716	000000G			MOV	P2,(SP)	:	
000530	004767	165230			JSR	PC,PRT#FRU,CALLOUT	:	
000534	016716	000000G		13#:	MOV	ER.STATUS,(SP)	:	3743
000540	012746	000000G			MOV	#FMT14,-(SP)	:	
000544	012746	000002			MOV	#2,-(SP)	:	
000550	010600				MOV	SP,RO	: SP,*	
000552	104414				TRAP	14	:	
000554	022626				CMP	(SP)*,(SP)*	:	3699
000556	000410				BR	15#	:	3697
000560	016746	000000G		14#:	MOV	ER.STATUS,-(SP)	:	3747
000564	012746	000000G			MOV	#FMT15,-(SP)	:	
000570	012746	000002			MOV	#2,-(SP)	:	
000574	010600				MOV	SP,RO	: SP,*	
000576	104414				TRAP	14	:	
000600	062706	000006		15#:	ADD	#6,SP	:	3692
000604	000422				BR	18#	:	3640
000606	020127	000601		16#:	CMP	R1,#601	:	3755
000612	001010				BNE	17#	:	
000614	016746	000002G			MOV	EMSG.STRUCT*2,-(SP)	:	3757
000620	012746	000001			MOV	#1,-(SP)	:	
000624	010600				MOV	SP,RO	: SP,*	
000626	104417				TRAP	17	:	
000630	022626				CMP	(SP)*,(SP)*	:	3756
000632	000407				BR	18#	:	3640
000634	016746	000006G		17#:	MOV	EMSG.STRUCT*6,-(SP)	:	3766
000640	012746	000001			MOV	#1,-(SP)	:	
000644	010600				MOV	SP,RO	: SP,*	
000646	104417				TRAP	17	:	
000650	022626				CMP	(SP)*,(SP)*	:	3765
000652	016700	000000G		18#:	MOV	RES.SLOT,RO	:	3774
000656	006300				ASL	RO	:	
000660	006300				ASL	RO	:	
000662	066700	000000G			ADD	RECEIVE.RING,RO	:	
000666	052760	100000	000002		BIS	#100000,2(RO)	:	
000674	004767	175504			JSR	PC,GET.RES.SLOT	:	3775
000700	005067	000000G			CLR	RET.STATUS	:	3776
000704	012601			19#:	MOV	(SP)*,R1	:	3588
000706	000207				RTS	PC	:	

: Routine Size: 228 words, Routine Base: AB#CODE * 12600
: Maximum stack depth per invocation: 8 words

: 3779 1

ZRCFR2
V03.0

MISCELLANEOUS SECTIONS
AZTEC GLOBAL ROUTINE

27-Mar-1985 15:23:34
11-Jan-1985 08:19:19

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC2.B16;1

SEQ 0195
Page 100
(36)

: 3780 1 end
: 3781 1
: 3782 0 eludom

OTS external references

.GLOBL \$SAVE5, \$SAVE3, \$SAVE2, BL\$SHF
.GLOBL BL\$MOD, BL\$MUL

PSECT SUMMARY

Psect Name	Words	Attributes
AA\$CODE	345	RO : I ; LCL, REL, CON
AB\$CODE	2980	RO : I ; LCL, REL, CON

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
USER#1:[AZTEC.CZRCFC]AZTECO.L16;2	485	209	43	24	00:00.2

COMMAND QUALIFIERS

BLISS/PDP11/LIST ZRCFC2.B16/EN:NOEIS

: Size: 3226 code + 99 data words
: Run Time: 03:38.2
: Elapsed Time: 03:48.4
: Lines/CPU Min: 1040
: Lexemes/CPU Min: 9145
: Memory Used: 299 pages
: Compilation Complete

B16

ZRCFB3

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0196
Page 1
(1)

: 0001 0 MODULE ZRCFB3 (

```

: 0002 0 #TITLE 'CZRCFC0 RC25 FR END TEST'
: 0003 0 IDENT = 'V03.0',
: 0004 0 OPTLEVEL = 0,
: 0005 0 ADDRESSING_MODE (RELATIVE)
: 0006 0 ) =
: 0007 1 BEGIN
: 0008 1 !<BLF/LOWERCASE_KEY>
: 0009 1 !
: 0010 1 #sbttl 'TEST SECTION'
: 0011 1
: 0012 1 library 'AZTECO'; ! AZTEC LIBRARY
: 0013 1
: 0014 1 require 'BLSMAC.REQ'; ! DIAGNOSTIC SUPERVISOR LIBRARY
: 1503 1
: 1504 1 structure ! DEFINE ACCESS ALGORITHM TO
: 1505 1 RC25 [0, P, S, E] = ! ALLOW FIELD REFERANCES TO
: 1506 2 begin ! THE RC25
: 1507 2
: 1508 2
: 1509 2 local
: 1510 2 RC_REG;
: 1511 2
: 1512 2 RC_REG = .(RC25 * #upval*0)<0, #bpval, 0>;
: 1513 2 RC_REG
: 1514 2 end
: 1515 1 <P, S, E>;
: 1516 1
: 1517 1 !<BLF/PAGE>

```



```

: 1575 1      BUF_DESCRPT : word volatile,
: 1576 1      CMD_REF : word volatile,
: 1577 1      CMD_SLOT : word volatile,
: 1578 1      RES_SLOT : word volatile,
: 1579 1      DM_09 : vector [93, word],
: 1580 1      DM_10 : vector [58, word],
: 1581 1      DM_11 : vector [100, word],
: 1582 1      DM_12 : vector [202, word],
: 1583 1      DM_13 : vector [105, word],
: 1584 1      DM_19 : vector [156, word],
: 1585 1      DM_21 : vector [213, word],
: 1586 1      DM_26 : vector [413, word],
: 1587 1      DM_27 : vector [307, word],
: 1588 1      BYTE_COUNT : word volatile,
: 1589 1      MSGADR : word volatile,
: 1590 1      MEM_SIZ : word,
: 1591 1      P_MASK : byte volatile,
: 1592 1      B_MASK : byte volatile,
: 1593 1      DATA1 : word,
: 1594 1      DATA2 : word volatile,
: 1595 1      DATA3 : word volatile,
: 1596 1      DATA4 : word volatile,
: 1597 1      END_LBN : word volatile,
: 1598 1      SWP_CONTINUE : word volatile,
: 1599 1      SWP_MANUAL : word volatile,
: 1600 1      MANU_SW : word volatile,
: 1601 1      SWITCH2 : word volatile,
: 1602 1      RET_UNIT_FLAG : word volatile,
: 1603 1      P1 : word volatile,
: 1604 1      P2 : word volatile,
: 1605 1      P3 : word volatile,
: 1606 1      P4 : word volatile,
: 1607 1      P5 : word volatile,
: 1608 1      P6 : word volatile,
: 1609 1      LBN : word volatile,
: 1610 1      LBN_ST : word volatile,
: 1611 1      LBN_ED : word volatile,
: 1612 1      LBN_SZ : word volatile,
: 1613 1      OFFSET : word,
: 1614 1
: 1615 1      SIZ_LBN : word,
: 1616 1      CLK_ADR : word,
: 1617 1      CLK_CSR : word,
: 1618 1      CLK_START : word,
: 1619 1      CLK_HERTZ : word,
: 1620 1      TICKS : word,
: 1621 1      SECONDS : word,
: 1622 1      MINUTES : word,
: 1623 1      CMOD : word,
: 1624 1      IN_BOUND : word,
: 1625 1      OUT_BOUND : word,
: 1626 1      SWP_START,
: 1627 1      SWP_END,
: 1628 1      SWP_TOP,
: 1629 1      BUF_LENGTH,
: 1630 1      TEMP,
: 1631 1      FREE_MEM_ADDR,
:
: BUFFER DESCRIPTOR AREA
: COMMAND REFERENCE BUFFER
: COMMAND RING SLOT
: RECEIVE RING SLOT
: DM PROGRAM 09
: DM PROGRAM 10
: DM PROGRAM 11
: DM PROGRAM 12
: DM PROGRAM 13
: DM PROGRAM 19
: DM PROGRAM 21
: DM PROGRAM 26
: DM PROGRAM 27
: BYTE COUNT BUFFER
: ERROR MESSAGE ADDRESS
: FREE MEMORY SIZE
: PRINT MASK FOR NUMBER OF ARGUMENTS
: INIT MASK FOR WHAT STEP TO DO
: DATA FOR STEP 1 WRITE
: DATA FOR STEP 2 WRITE
: DATA FOR STEP 3 WRITE
: DATA FOR STEP 4 WRITE
: ENDING LOGICAL BLOCK #
:
: SOFTWARE P-TAB MANUAL SWITCH
: MANUAL INTERVENTION SWITCH1
: MANUAL INTERVENTION SWITCH2
: RETURN UNIT STATUS BUFFER
: FORMAT ADDRESS FOR ERROR REPORT
: FAILING FRU
: FAILING REGISTER
: DATA FOR ERROR REPORT
: DATA FOR ERROR REPORT
: DATA FOR ERROR REPORT
: DATA FOR LBN
: DATA FOR STARTING LBN
: DATA FOR ENDING LBN
: LBN INCREMENTING SIZE
: OFFSET TO LBN TO GET TO
: BOTTOM SURFACE
: LBN SIZE TO GET TO NEXT TRACK
: LOC. TO RETURN CLOCK ADDR.
: STORE CLOCK STARTING ADDR.
: THE CLOCK STARTING VALUE
:
: THE # OF CLOCK INT. BUFFER
: THE NUMBERS OF SECONDS BUFFER
: THE NUMBERS OF MINUTES BUFFER
: COMMAND MODIFIER
:
: STARTING TRACK BUF
: ENDING TRACK BUF
: TOP SURFACE
: BUFFER LENGTH
: TEMP. BUFFER
: STARTING FREE MEMORY ADDR.

```

```

: 1632 1 MEM_SIZE,
: 1633 1 H_SADD,
: 1634 1 H_EADD,
: 1635 1 INI_MSG,
: 1636 1 P_VECTOR,
: 1637 1 P_IP_ADDRESS,
: 1638 1 RET_STATUS,
: 1639 1 ADAPTO,
: 1640 1 TIME,
: 1641 1 MSG_1,
: 1642 1 MSG_2,
: 1643 1 MSG_7,
: 1644 1 MSG_8,
: 1645 1 MSG_9,
: 1646 1 MSG_10,
: 1647 1 MSG_11,
: 1648 1 MSG_13,
: 1649 1 MSG_14,
: 1650 1 QST14,
: 1651 1 QST15,
: 1652 1 END_MSG,
: 1653 1 FMT1,
: 1654 1 FMT2,
: 1655 1 FMT3,
: 1656 1 FMT4,
: 1657 1 FMT5,
: 1658 1 FMT6,
: 1659 1 FMT7,
: 1660 1 FMT7A,
: 1661 1 FMT8,
: 1662 1 FMT9,
: 1663 1 FMT10,
: 1664 1 FMT11,
: 1665 1 FMT12,
: 1666 1 FMT16,
: 1667 1 FMT17,
: 1668 1 FMT18,
: 1669 1 FMT19,
: 1670 1 FMT20,
: 1671 1 FRU,
: 1672 1 FMT4A,
: 1673 1 DBM7,
: 1674 1 DBM8,
: 1675 1 DBM9,
: 1676 1 DBM10,
: 1677 1 DBM11,
: 1678 1 DBM12,
: 1679 1 DBM13,
: 1680 1 DBM14,
: 1681 1 DBM15,
: 1682 1 DBM16,
: 1683 1 DBM17,
: 1684 1 DBM18,
: 1685 1 DBM19,
: 1686 1 DBM20,
: 1687 1 DBM21,
: 1688 1 DBM22,

```

```

! FREE MEMORY SIZE
! FREE HOST MEMORY START AD.
! FREE HOST MEMORY END AD.
! INIT ERROR MESSAGE
! VECTOR BUFFER
! RC25 ADDRESS
! COMMAND STATUS BUFFER
! ADAPTOR FRU MESSAGE
! PLIT LOCATION TO STORE DATA
! ERROR MESSAGE 1 IN MOD 1
! ERROR MESSAGE IN TEST
! ERROR MESSAGE IN TEST
! ERROR MESSAGE IN TEST
! ERROR MESSAGE IN TEST
! ERROR MESSAGE IN TEST
! ERROR MESSAGE IN TEST
! ERROR MESSAGE IN TEST
! ERROR MESSAGE IN TEST
! MESSAGE
! MESSAGE
! ERROR MESSAGE IN TEST
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! FORMATTED MESSAGE
! VER:C
! FRU = MESSAGE

! TEST HEADER MESSAGES

```

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0201
Page 6
(3)

```

: 1689 1 DBM23,
: 1690 1 DBM24,
: 1691 1 DBM25,
: 1692 1 DBM26,
: 1693 1 DBM27,
: 1694 1 DBM28,
: 1695 1 DBM29,
: 1696 1 DBM30,
: 1697 1 DBM31,
: 1698 1 DBM32,
: 1699 1 DBM36,
: 1700 1 DBM37,
: 1701 1 DBM38,
: 1702 1 DBM39,
: 1703 1 MSG_HSWICH_ERR,
: 1704 1 MSG_READ_ERR,
: 1705 1 MSG_SAC_ERR,
: 1706 1 MSG_AVE_TIME,
: 1707 1 MSG_PRO_TIME,
: 1708 1 MSG_SK_TIME,
: 1709 1 MES_SKO_TIME,
: 1710 1 MG_SKF_TIME,
: 1711 1 MSG_ROT_TIME,
: 1712 1 MSG_COM_WPT,
: 1713 1 !
: 1714 1 SK_FOR_ERR,
: 1715 1 SK_REV_ERR,
: 1716 1 SK_RAN_ERR,
: 1717 1 SK_TOG_ERR,
: 1718 1 MSG_WRITE_ERR,
: 1719 1 MSG_GUS_ERR,
: 1720 1 AVAIL_ERR,
: 1721 1 MSG_SEEK_ERR,
: 1722 1 !
: 1723 1 AZT_READY_ERR,
: 1724 1 EXE_SUP_ERR,
: 1725 1 SND_DATA_ERR,
: 1726 1 RE_DATA_ERR,
: 1727 1 BUFF_ERR,
: 1728 1 DMC_ERR,
: 1729 1 BRERR,
: 1730 1 TIP;
: 1731 1
: 1732 1 external routine
: 1733 1 NXMI : novalue,
: 1734 1 AZT_INIT,
: 1735 1 AZP_INIT,
: 1736 1 FIND_CLOCK : novalue,
: 1737 1 CLOCK_INIT : novalue,
: 1738 1 RC25$ERR_RPT : novalue,
: 1739 1 INIT_COM_AREA,
: 1740 1 REC_STATUS,
: 1741 1 EX_SUP_PRG,
: 1742 1 RANDOM_NUM,
: 1743 1 REC_DATA,
: 1744 1 SEND_DATA,
: 1745 1 SET_CNTRLR_CHAR,

```

!TEST IN PROGRESS

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

```

: 1746 1 AVAILABLE,
: 1747 1 READ_CMD,
: 1748 1 READ_FILL_CMD,
: 1749 1 ON_LINE,
: 1750 1 GET_UNIT_STATUS,
: 1751 1 GET_CMD_SLOT,
: 1752 1 DECODE,
: 1753 1 EXAM_DATA,
: 1754 1 AZTEC_READY,
: 1755 1 DO_RETRIES : novalue;
: 1756 1
: 1757 1 !<BLF/PAGE>

```

! EXAMINE THE FREE MEMORY DATA
! GET AZTEC READY

```

: 1758 1  !
: 1759 3  !BGNTST;
: 1760 3  !
: 1761 3  !**
: 1762 3  ! TEST 1: REGISTER EXISTENCE TEST
: 1763 3  ! DESCRIPTION:
: 1764 3  ! THIS TEST WILL FIRST CHECK FOR THE EXISTENCE OF THE ADDRESS OF THE IP
: 1765 3  ! AND SA REGISTERS FOR THE DEVICE UNDER TEST.
: 1766 3  ! IF THESE MEMORY ADDRESSES ARE NON-EXISTENT, THE ERROR WILL BE
: 1767 3  ! REPORTED.
: 1768 3  ! IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM THE
: 1769 3  ! BEGINNING OF SUB TEST.
: 1770 3  !--
: 1771 3  !
: 1772 3  local
: 1773 3  DUMMY;
: 1774 3  !
: 1775 3  if .SWP_TRACE then PRINTF (DBM7);           ! TEST 1
: 1776 3  !
: 1777 5  BGNSUB;
: 1778 5  NUM_RETRIES = ZERO;                       ! CLEAR RETRY COUNTER
: 1779 5  !
: 1780 5  while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 1781 6  begin
: 1782 6  I_AM_NEX = FALSE;                          ! CLEAR OUT NEX FLAG
: 1783 6  SETVEC (4, NXMI, PRI07);                  ! SET UP FOR AN NEX TRAP
: 1784 6  !
: 1785 7  if (.RT_TABLE [RT_IP_ADDRESS] + 2)
: 1786 6  then
: 1787 7  begin
: 1788 7  DUMMY = 1;
: 1789 6  end;
: 1790 6  ! READ THE SA REGISTER
: 1791 6  ! THIS IS SO THAT IF THERE
: 1792 6  ! IS AN NEX THERE WILL BE
: 1793 6  ! A SINGLE OPPERAND INST.
: 1794 6  ! SO THAT IT WILL TRAP
: 1795 7  ! ADDRESS NOT THERE
: 1796 7  !
: 1797 7  CLRVEC (4);                               ! CORRECTLY.
: 1798 7  !
: 1799 7  if .I_AM_NEX eqlu ALL_ONES
: 1800 7  then
: 1801 7  begin
: 1802 7  P_MASK = 1;
: 1803 7  P1 = FMT1;
: 1804 7  P2 = ADAPT;
: 1805 7  P3 = 0;
: 1806 7  P4 = (.RC25_ADDR) + 2;
: 1807 7  ERRDF (1, MSG_1, RC25$ERR_RPT);          ! PRINT ERROR MESSAGE
: 1808 7  CKLOOP;
: 1809 7  DO_RETRIES ();
: 1810 6  end;
: 1811 6  !
: 1812 6  if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 1813 6  !
: 1814 5  end;
: 1815 5  NUM_RETRIES = ZERO;                       ! CLEAR RETRY COUNTER
: 1816 3  ENDSUB;
: 1817 5  BGNSUB;
: 1818 5  !
: 1819 5  while (.NUM_RETRIES lequ .SWP_RETRIES) do

```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

```

: 1815 6      begin
: 1816 6      I_AM_NEX = FALSE;
: 1817 6      SETVEC (4, NXMI, PRI07);
: 1818 6
: 1819 7      if (.RT_TABLE [RT_IP_ADDRESS])
: 1820 6      then
: 1821 7          begin
: 1822 7              DUMMY = 1;
: 1823 6          end;
: 1824 6
: 1825 6      CLRVEC (4);
: 1826 6
: 1827 6      if .I_AM_NEX eqlu ALL_ONES
: 1828 6      then
: 1829 7          begin
: 1830 7              P_MASK = 1;
: 1831 7              P1 = FMT1;
: 1832 7              P2 = ADAPT;
: 1833 7              P4 = .RC25_ADDR;
: 1834 7              ERRDF (2, MSG_2, RC25$ERR_RPT);
: 1835 7              CKLOOP;
: 1836 7              DO_P$TRIES ();
: 1837 6          end;
: 1838 6
: 1839 6      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 1840 6
: 1841 5      end;
: 1842 5
: 1843 5      if .I_AM_NEX eqlu ALL_ONES
: 1844 5      then
: 1845 6          begin
: 1846 6              DODU (.LOG_UNIT);
: 1847 6              DOCLN;
: 1848 5          end;
: 1849 5
: 1850 3      ENDSUB;
: 1851 1      ENDTST;

```

```

! CLEAR TRAP FLAG
! SET UP TRAP VECTOR IF NEX

! READ IP REGISTER

! CLEAR THE VRCTOR

! CHECK FOR TRAPS

! PRINT OUT ERRO MESSAGE

! IF REGISTERS ARE NON-EXISTENT
! THEN DROP THE UNIT FROM TESTING

```

.TITLE ZRCFB3 CZRCFC0 RC25 FR END TEST
.IDENT /V03.0/

000000		.PSECT	\$OWN\$,	D
000000		CMDBF1:	.BLKW	20
000040		ENDBF1:	.BLKW	20
000100		RING.B:	.BLKW	40
000200	111111	DATA.PAT1:		
		.WORD	-66667	
000202	044444	.WORD	44444	
000204	022222	.WORD	22222	
000206	177400	DATA.PAT2:		
		.WORD	-400	
000210	007760	.WORD	7760	
000212	000377	.WORD	377	
000214	155555	DATA.PAT3:		
		.WORD	-22223	

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.816;4

000216 133333
000220 066666
000222 000377

000224 170017
000226 177400
000230

000240

.WORD -44445
.WORD 66666
DATA.PAT4:
.WORD 377
.WORD -7761
.WORD -400
SEND.PKT:
.BLKW 4
PATTERN.ADDR:
.BLKW 1

.GLOBL RT.TABLE, RC25.ADDR, RC25.DATA
.GLOBL UNIT, LOG.UNIT, RETRIES, PASSO
.GLOBL NUM.RETRIES, SWP.TRACE, SWP.RETRIES
.GLOBL I.AM.NEX, CANCEL.TIMER, COM.AREA
.GLOBL HEAD.AREA, RECEIVE.RING, SEND.RING
.GLOBL REC.ENVELOPE, SND.ENVELOPE, XMT.DATA.BUF
.GLOBL RCV.DATA.BUF, RINGBASE, BUF.DESCRPTR
.GLOBL CMD.REF, CMD.SLOT, RES.SLOT, DM.09
.GLOBL DM.10, DM.11, DM.12, DM.13, DM.19
.GLOBL DM.21, DM.26, DM.27, BYTE.COUNT
.GLOBL MSGADR, MEM.SIZ, P.MASK, B.MASK
.GLOBL DATA1, DATA2, DATA3, DATA4, END.LBN
.GLOBL SWP.CONTINUE, SWP.MANUAL, MANU.SW
.GLOBL SWITCH2, RET.UNIT.FLAG, P1, P2
.GLOBL P3, P4, P5, P6, LBN, LBN.ST, LBN.ED
.GLOBL LBN.SZ, OFFSET, SIZ.LBN, CLK.ADR
.GLOBL CLK.CSR, CLK.START, CLK.HERTZ
.GLOBL TICKS, SECONDS, MINUTES, CMOD
.GLOBL IN.BOUND, OUT.BOUND, SWP.START
.GLOBL SWP.END, SWP.TOP, BUF.LENGTH, TEMP
.GLOBL FREE.MEM.ADDR, MEM.SIZE, H.SADD
.GLOBL H.EADD, INI.MSG, P.VECTOR, P.IP.ADDRESS
.GLOBL RET.STATUS, ADAPTO, TIME, MSG.1
.GLOBL MSG.2, MSG.7, MSG.8, MSG.9, MSG.10
.GLOBL MSG.11, MSG.13, MSG.14, QST14
.GLOBL QST15, END.MSG, FMT1, FMT2, FMT3
.GLOBL FMT4, FMT5, FMT6, FMT7, FMT7A
.GLOBL FMT8, FMT9, FMT10, FMT11, FMT12
.GLOBL FMT16, FMT17, FMT18, FMT19, FMT20
.GLOBL FRU, FMT#A, DBM7, DBM8, DBM9, DBM10
.GLOBL DBM11, DBM12, DBM13, DBM14, DBM15
.GLOBL DBM16, DBM17, DBM18, DBM19, DBM20
.GLOBL DBM21, DBM22, DBM23, DBM24, DBM25
.GLOBL DBM26, DBM27, DBM28, DBM29, DBM30
.GLOBL DBM31, DBM32, DBM36, DBM37, DBM38
.GLOBL DBM39, MSG.HSWICH.ERR, MSG.READ.ERR
.GLOBL MSG.SAC.ERR, MSG.AVE.TIME, MSG.PRO.TIME
.GLOBL MSG.SK.TIME, MES.SKO.TIME, MG.SKF.TIME
.GLOBL MSG.ROT.TIME, MSG.COM.WPT, SK.FOR.ERR
.GLOBL SK.REV.ERR, SK.RAN.ERR, SK.TOG.ERR
.GLOBL MSG.WRITE.ERR, MSG.GUS.ERR, AVAIL.ERR
.GLOBL MSG.SEEK.ERR, AZT.READY.ERR, EXE.SUP.ERR
.GLOBL SND.DATA.ERR, RE.DATA.ERR, BUFF.ERR
.GLOBL DMC.ERR, BRERR, TIP, NXMI, AZT.INIT
.GLOBL AZP.INIT, FIND.CLOCK, CLOCK.INIT

.GLOBL RC25\$ERR.RPT, INIT.COM.AREA, REC.STATUS
.GLOBL EX.SUP.PRG, RANDOM.NUM, REC.DATA
.GLOBL SEND.DATA, SET.CNTRL.CHAR, AVAILABLE
.GLOBL READ.CMD, READ.FILL.CMD, ON.LINE
.GLOBL GET.UNIT.STATUS, GET.CMD.SLOT
.GLOBL DECODE, EXAM.DATA, AZTEC.READY
.GLOBL DO.RETRIES

000000

.SBTTL \$T1 TEST SECTION
.PSECT AC\$CODE, RO

000000	010146			\$T1:	MOV	R1, -(SP)	:	1755
000002	032767	000001	000000G		BIT	#1, SWP.TRACE	:	1775
000010	001407				BEQ	1\$		
000012	012746	000000G			MOV	#DBM7, -(SP)		
000016	012746	000001			MOV	#1, -(SP)		
000022	010600				MOV	SP, RO	: SP, *	
000024	104417				TRAP	17		
000026	022626				CMP	(SP), (SP)		
000030	104402			1\$:	TRAP	2		
000032	005067	000000G			CLR	NUM.RETRIES	:	1778
000036	026767	000000G	000000G	2\$:	CMP	NUM.RETRIES, SWP.RETRIES	:	1780
000044	101100				BHI	7\$		
000046	005067	000000G			CLR	I.AM.NEX	:	1782
000052	012746	000340			MOV	#340, -(SP)	:	1783
000056	012746	000000G			MOV	#NXMI, -(SP)		
000062	012746	000004			MOV	#4, -(SP)		
000066	012746	000003			MOV	#3, -(SP)		
000072	104437				TRAP	37		
000074	017700	000000G			MOV	#RT.TABLE, RO	:	1785
000100	032760	000001	000002		BIT	#1, 2(RO)		
000106	001402				BEQ	3\$		
000110	012701	000001			MOV	#1, R1	: *, DUMMY	1788
000114	012700	000004		3\$:	MOV	#4, RO	:	1791
000120	104436				TRAP	36		
000122	026727	000000G	177777		CMP	I.AM.NEX, #-1	:	1793
000130	001035				BNE	5\$		
000132	112767	000001	000000G		MOVB	#1, P.MASK	:	1796
000140	012767	000000G	000000G		MOV	#FMT1, P1	:	1797
000146	012767	000001	000000G		MOV	#1, P2	:	1798
000154	005067	000000G			CLR	P3	:	1799
000160	016700	000000G			MOV	RC25.ADDR, RO	:	1800
000164	062700	000002			ADD	#2, RO		
000170	010067	000000G			MOV	RO, P4		
000174	104455				TRAP	55	:	1801
000176	000001				.WORD	1		
000200	000000G				.WORD	MSG.1		
000202	000000G				.WORD	RC25\$ERR.RPT		
000204	104465				TRAP	65		
000206	006000				ROR	RO		
000210	103003				BCC	4\$		
000212	062706	000010			ADD	#10, SP		
000216	000415				BR	8\$		
000220	004767	000000G		4\$:	JSR	PC, DO.RETRIES	:	1803
000224	005767	000000G		5\$:	TST	NUM.RETRIES	:	1806
000230	001003				BNE	6\$		

M16

ZRCFB3 V03.0	CZRCFC0 RC25 FR END TEST TEST SECTION	27-Mar-1985 15:27:28 27-Mar-1985 13:28:18	VAX-11 Bliss-16 V4.0-579 USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4	SEQ 0207 Page 12 (4)
000232	062706	000010	ADD #10,SP	
000236	000403		BR 7#	
000240	062706	000010	6# : ADD #10,SP	1781
000244	000674		BR 2#	1780
000246	005067	000000G	7# : CLR NUM.RETRIES	1810
000252	104467		8# : TRAP 67	
000254	006000		ROR R0	
000256	103664		BLO 1#	
000260	104402		9# : TRAP 2	1811
000262	026767	000000G 000000G	10# : CMP NUM.RETRIES,SWP.RETRIES	1814
000270	101072		BHI 15#	
000272	005067	000000G	CLR I.AM.NEX	1816
000276	012746	000340	MOV #340,-(SP)	1817
000302	012746	000000G	MOV #NXMI,-(SP)	
000306	012746	000004	MOV #4,-(SP)	
000312	012746	000003	MOV #3,-(SP)	
000316	104437		TRAP 37	
000320	017700	000000G	MOV @RT.TABLE,R0	1819
000324	032710	000001	BIT #1,(R0)	
000330	001402		BEQ 11#	
000332	012701	000001	MOV #1,R1	1822
000336	012700	000004	11# : MOV #4,R0 ; *.DUMMY	1825
000342	104436		TRAP 36	
000344	026727	000000G 177777	CMP I.AM.NEX,#-1	1827
000352	001030		BNE 13#	
000354	112767	000001 000000G	MOVB #1,P.MASK	1830
000362	012767	000000G 000000G	MOV #FMT1,P1	1831
000370	012767	000001 000000G	MOV #1,P2	1832
000376	016767	000000G 000000G	MOV RC25.ADDR,P4	1833
000404	104455		TRAP 55	1834
000406	000002		.WORD 2	
000410	000000G		.WORD MSG.2	
000412	000000G		.WORD RC25#ERR.RP?	
000414	104465		TRAP 65	
000416	006000		ROR R0	
000420	103003		BCC 12#	
000422	062706	000010	ADD #10,SP	
000426	000423		BR 16#	
000430	004767	000000G	12# : JSR PC,DO.RETRIES	1836
000434	005767	000000G	13# : TST NUM.RETRIES	1839
000440	001003		BNE 14#	
000442	062706	000010	ADD #10,SP	
000446	000403		BR 15#	
000450	062706	000010	14# : ADD #10,SP	1815
000454	000702		BR 10#	1814
000456	026727	000000G 177777	15# : CMP I.AM.NEX,#-1	1843
000464	001004		BNE 16#	
000466	016700	000000G	MOV LOG.UNIT,R0	1846
000472	104451		TRAP 51	
000474	104444		TRAP 44	
000476	104467		16# : TRAP 67	1848
000500	006000		ROR R0	
000502	103666		BLO 9#	
000504	012601		MOV (SP)+,R1	1755
000506	000207		RTS PC	

; Routine Size: 164 words, Routine Base: AC#CODE + 0000

B1

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B1100-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0208
Page 13
(4)

: Maximum stack depth per invocation: 7 words

000000	004767	177264				
000000			T1::	.SBTTL	T1 TEST SECTION	
000004	104466		1\$:	JSR	PC,\$T1	
000006	006000			TRAP	66	
000010	103773			ROR	RO	
000012	000207			BLO	1\$	
				RTS	PC	

1850

: Routine Size: 6 words, Routine Base: AC\$CODE + 0510
: Maximum stack depth per invocation: 2 words

: 1852 1 !<BLF/PAGE>

```

: 1853 1  !
: 1854 3  !BGNTST;
: 1855 3
: 1856 3
: 1857 3  !**
: 1858 3  ! TEST 2: INITALZATION TEST (POWER UP DIAGNOSTICS)
: 1859 3  ! DESCRIPTION:
: 1860 3  ! THIS TEST INIT' THE AZTEC AND RUNS THE POWER UP DIAGNOSTICS BY
: 1861 3  ! WRITING WITH STEP1 DATA. THEN IT WILL CHECK FOR ERRORS AND
: 1862 3  ! REPORT IF AZTEC DOES NOT COME UPTO STEP2 READ
: 1863 3  !
: 1864 3  NUM_RETRIES = ZERO;                ! CLEAR RETRY COUNTER
: 1865 3
: 1866 3  if .SWP_TRACE then PRINTF (DBM8);    ! TEST 2
: 1867 3
: 1868 3  while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 1869 4      begin
: 1870 4      ! STEP 1 WRITE WITH STEP 2 READ
: 1871 4      B_MASK = 1;                    ! SELECT B_MASK FOR STEP 1 WRITE
: 1872 4      DATA1 = %0'137600' * .RT_TABLE [RT_VECTOR]/4; ! SELECT STEP1 WRITE DATA WITH
: 1873 4      ! MAX RING SIZES ,IE AND VECTOR
: 1874 4      ! ADDRESS
: 1875 4
: 1876 4      if AZT_INIT ()                  ! PORT SHOULD NOW GET TO STEP2
: 1877 4      ! AFTER FINISHING INTEGRITY CHECK
: 1878 4      ! DIAG. IF NOT REPORT ERROR
: 1879 4      then
: 1880 5          begin
: 1881 5          ERROF (3, MSG_14, RC25$ERR_RPT);
: 1882 5
: 1883 5          if .RET_STATUS then DECODE (); ! DECODE STATUS
: 1884 5
: 1885 5          CKLOOP;
: 1886 5          RETRIES = TRUE;
: 1887 4          end;
: 1888 4
: 1889 4          if (.RETRIES) then DO_RETRIES (); ! RETRY IF ERROR
: 1890 4
: 1891 4          if (.NUM_RETRIES equ ZERO) then exitloop;
: 1892 4
: 1893 3      end;
: 1894 3
: 1895 3  return;
: 1896 1  ENDTST;

```

000000	005067	000000G		.SBTTL	\$T2 TEST SECTION		
000004	032767	000001 000000G	\$T2:	CLR	NUM.RETRIES	:	1864
000012	001407			BIT	#1,SWP.TRACE	:	1866
000014	012746	000000G		BEQ	1\$		
000020	012746	000001		MOV	#DBM8,-(SP)		
000024	010600			MOV	#1,-(SP)		
000026	104417			MOV	SP,RO	:	SP,*
000030	022626			TRAP	17		
000032	026767	000000G 000000G	1\$:	CMP	(SP)*,(SP)*		
000040	101060			CMP	NUM.RETRIES,SWP.RETRIES	:	1868
				BHI	6\$		

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

000042	112767	000001	000000G	MOVB	#1,B.MASK	:	1871
000050	016700	000000G		MOV	RT, TABLE, RO	:	1872
000054	016046	000002		MOV	2(RO), -(SP)		
000060	012746	000004		MOV	#4, -(SP)		
000064	004767	000000G		JSR	PC, BL\$DIV		
000070	010067	000000G		MOV	RO, DATA1		
000074	162767	040200	000000G	SUB	#40200, DATA1		
000102	004767	000000G		JSR	PC, AZT.INIT	:	1876
000106	006000			ROR	RO		
000110	103022			BCC	4\$		
000112	104455			TRAP	55	:	1881
000114	000003			.WORD	3		
000116	000000G			.WORD	MSG.14		
000120	000000G			.WORD	RC25\$ERR.RPT		
000122	032767	000001	000000G	BIT	#1, RET.STATUS	:	1883
000130	001402			BEQ	2\$		
000132	004767	000000G		JSR	PC, DECODE		
000136	104465		2\$:	TRAP	65		
000140	006000			ROR	RO		
000142	103002			BCC	3\$		
000144	022626			CMP	(SP)*, (SP)*		
000146	000207			RTS	PC		
000150	012767	000001	000000G	MOV	#1, RETRIES	:	1886
000156	032767	000001	000000G	BIT	#1, RETRIES	:	1889
000164	001402			BEQ	5\$		
000166	004767	000000G		JSR	PC, DO.RETRIES		
000172	005767	000000G		TST	NUM.RETRIES	:	1891
000176	001002			BNE	7\$		
000200	022626			CMP	(SP)*, (SP)*		
000202	000207		6\$:	RTS	PC		
000204	022626		7\$:	CMP	(SP)*, (SP)*	:	1869
000206	000711			BR	1\$:	1868

: Routine Size: 68 words, Routine Base: AC\$CODE + 0524
: Maximum stack depth per invocation: 4 words

000000	004767	177564	T2::	.SBTTL	T2 TEST SECTION		
000000			1\$:	JSR	PC, \$T2	:	1895
000004	104466			TRAP	66		
000006	006000			ROR	RO		
000010	103773			BLO	1\$		
000012	000207			RTS	PC		

: Routine Size: 6 words, Routine Base: AC\$CODE + 0734
: Maximum stack depth per invocation: 2 words

: 1897 1 !<BLF/PAGE>

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0211
Page 16
(6)

```

: 1898 3  BGNTST;
: 1899 3
: 1900 3
: 1901 3  !**
: 1902 3  ! TEST #3 - DIAGNOSTIC WRAP TEST
: 1903 3
: 1904 3  ! DESCRIPTION:
: 1905 3  !
: 1906 3  ! THE AZTEC WILL BE INITIALIZED IN DIAGNOSTIC WRAP MODE AND A ONE BIT
: 1907 3  ! AND ALSO ZERO BIT FLOATED THROUGH THE SA REGISTER TO SEE THAT IT
: 1908 3  ! ECHOES PROPERLY.
: 1909 3  !
: 1910 3  ! A FAILURE TO ECHO WHAT WAS WRITTEN WILL RESULT IN A CALLOUT TO THE
: 1911 3  ! ADAPTER CARD FRU.
: 1912 3  !
: 1913 3  ! IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, THE PROGRAM WILL LOOP ON
: 1914 3  ! THE FAILING WRITE AND READ.
: 1915 3  ! --
: 1916 3
: 1917 3  local
: 1918 3  TST_PAT;
: 1919 3
: 1920 3  if .SWP_TRACE then PRINTF (DBM10);           ! TEST 3
: 1921 3
: 1922 3  NUM_RETRIES = ZERO;
: 1923 3
: 1924 3  while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 1925 4  begin
: 1926 4  TIP = 4;
: 1927 4  !
: 1928 4  ! STEP1 WRITE
: 1929 4  !
: 1930 4  B_MASK = 0;           ! MASK FOR STEP1 READ
: 1931 4  DATA1 = %o'140000';   ! STEP1 WRITE WITH WRAP MODE BIT SET
: 1932 4  DATA2 = %o'10';     ! TIME OUT COUNTER
: 1933 4  DATA3 = ZERO;      ! TEMP STORAGE FOR RCSA DATA
: 1934 4
: 1935 4  if AZT_INIT ()       ! CALL STEP 1 ROUTINE
: 1936 4  then
: 1937 5  begin
: 1938 5  ERRDF (4, MSG_14, RC25$ERR_RPT);   ! PRINT OUT ERROR REPORT
: 1939 5  CKLOOP;
: 1940 5  RETRIES = TRUE;
: 1941 5  end
: 1942 4  else
: 1943 5  begin
: 1944 5  WRT_RC25 (RCSA, .DATA1);         ! DO STEP1 WRITE WITH DWM.
: 1945 5
: 1946 5  while ((.DATA3 nequ .DATA1) and (.DATA2 nequ ZERO)) do
: 1947 6  begin
: 1948 6  DELAY (333);
: 1949 6  DATA2 = .DATA2 - 1;
: 1950 6  DATA3 = .RC25_ADDR [RCSA, RC_ALL]; !
: 1951 5  end;
: 1952 5
: 1953 5  TST_PAT = %o'000001';           ! START TEST PATTERN TO
: 1954 5

```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

```

: 1955 5      incru FLOAT from 0 to 15 do      ! NOW FLOAT TEST PAT
: 1956 6      begin
: 1957 6
: 1958 6      incru COUNT from 0 to 1 do      ! FLOAT ZEROES AND THEN ONES
: 1959 7      begin
: 1960 7
: 1961 7      if .COUNT equ 1 then TST_PAT = not .TST_PAT;
: 1962 7
: 1963 9      BGNSUB;
: 1964 9      WRT RC25 (RCSA, .TST_PAT);      ! WRITE TEST PATTERN TO SA
: 1965 9      DELAY (10);                    ! WAIT FOR IT TO ECHO.
: 1966 9      RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL]; ! GET RCSA DATA
: 1967 9
: 1968 9      if .RC25_DATA [RCSA, RC_ALL] nequ .TST_PAT      ! TEST SA FOR TEST PATTERN
: 1969 9      then                                          ! IF NOT EQU THEN
: 1970 10     begin                                          ! PRINT OUT ERROR REPORT
: 1971 10     P MASK = 2;
: 1972 10     P1 = FMT2;                                     ! MESSAGE ADDRESS
: 1973 10     P2 = ADAPT;                                    ! FAILING FRU
: 1974 10     P6 = (.RC25_ADDR) + 2;                        ! FAILING ADDRESS
: 1975 10     P4 = .TST_PAT;                                ! GOOD DATA
: 1976 10     P5 = .RC25_DATA [RCSA, RC_ALL];              ! BAD DATA
: 1977 10     ERRDF (5, MSG_7, RC25$ERR_RPT);              !
: 1978 10     CKLOOP;
: 1979 10     RETRIES = TRUE;
: 1980 9     end;
: 1981 9
: 1982 7     ENDSUB;
: 1983 6     end;
: 1984 6
: 1985 6     TST_PAT = not .TST_PAT;
: 1986 6     TST_PAT = .TST_PAT#1;                        ! SHIFT THE BIT DOWN 1
: 1987 5     end;
: 1988 5
: 1989 4     end;
: 1990 4
: 1991 4     if (.RETRIES) then DO_RETRIES ();             ! DO RETRIES IF IN ERROR
: 1992 4
: 1993 4     if (.NUM_RETRIES equ ZERO) then exitloop;
: 1994 4
: 1995 3     end;
: 1996 3
: 1997 3     WRT_RC25 (RCIP, ALL_ONES);                    !REINITIALIZE THE PORT
: 1998 1     ENDTST;

```

.GLOBL L\$DLY

```

000000 004167 000000G      $T3:  .SBTTL  $T3 TEST SECTION
000004 162706 000006      JSR   R1,$SAVE4      ; 1896
000010 032767 000001 000000G  SUB   #6,SP
000016 001407 000000G      BIT   #1,SWP.TRACE  ; 1920
000020 012746 000000G      BEQ   1$
000024 012746 000001      MOV   #DBM10,-(SP)
000030 010600 000000G      MOV   #1,-(SP)
                                ; SP,*

```

G1

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0213
Page 18
(6)

000032	104417				TRAP	17			
000034	022626				CMP	(SP), (SP)			
000036	005067	000000G		1\$:	CLR	NUM.RETRIES	:		1922
000042	026767	000000G	000000G	2\$:	CMP	NUM.RETRIES, SWP.RETRIES	:		1924
000050	101402				BLOS	3\$			
000052	000167	000470			JMP	22\$			
000056	012767	000004	000000G	3\$:	MOV	#4, TIP	:		1926
000064	105067	000000G			CLRB	B.MASK	:		1930
000070	012767	140000	000000G		MOV	#-40000, DATA1	:		1931
000076	012767	000010	000000G		MOV	#10, DATA2	:		1932
000104	005067	000000G			CLR	DATA3	:		1933
000110	004767	000000G			JSR	PC, AZT.INIT	:		1935
000114	006000				ROR	R0			
000116	103015				BCC	5\$			
000120	104455				TRAP	55	:		1938
000122	000004				.WORD	4			
000124	000000G				.WORD	MSG.14			
000126	000000G				.WORD	RC25\$ERR.RPT			
000130	104465				TRAP	65			
000132	006000				ROR	R0			
000134	103002				BHIS	4\$			
000136	000167	000414			JMP	23\$			
000142	012767	000001	000000G	4\$:	MOV	#1, RETRIES	:		1940
000150	000563				BR	20\$:		1935
000152	016701	000000G		5\$:	MOV	DATA1, R1	:	* , RCM.REG	1944
000156	016700	000000G			MOV	RC25.ADDR, R0			
000162	010160	000002			MOV	R1, 2(R0)	:	RCM.REG, *	
000166	026767	000000G	000000G	6\$:	CMP	DATA3, DATA1	:		1946
000174	001430				BEQ	11\$			
000176	005767	000000G			TST	DATA2			
000202	001425				BEQ	11\$			
000204	012701	000515			MOV	#515, R1	:	* , \$\$TMP2	1948
000210	001411			7\$:	BEQ	10\$:	* , \$\$TMP1	
000212	016700	000000G			MOV	L\$DLY, R0	:	* , \$\$TMP1	
000216	001404				BEQ	9\$			
000220	005066	000004		8\$:	CLR	4(SP)	:	\$\$TMP	
000224	005300				DEC	R0	:	\$\$TMP1	
000226	001374				BNE	8\$			
000230	005301			9\$:	DEC	R1	:	\$\$TMP2	
000232	000766				BR	7\$			
000234	005367	000000G		10\$:	DEC	DATA2	:		1949
000240	016700	000000G			MOV	RC25.ADDR, R0	:		1950
000244	016016	000002			MOV	2(R0), (SP)	:	* , RC.REG	
000250	011667	000000G			MOV	(SP), DATA3	:	RC.REG, *	
000254	000744				BR	6\$:		1946
000256	012702	000001		11\$:	MOV	#1, R2	:	* , TST.PAT	1953
000262	005004				CLR	R4	:	FLOAT	1955
000264	005003			12\$:	CLR	R3	:	COUNT	1958
000266	022727	000000	000001		CMP	#0, #1	:		1961
000274	001001			13\$:	BNE	14\$			
000276	005102				COM	R2	:	TST.PAT	
000300	104402			14\$:	TRAP	2			
000302	010201				MOV	R2, R1	:	TST.PAT, RCM.REG	1964
000304	016700	000000G			MOV	RC25.ADDR, R0			
000310	010160	000002			MOV	R1, 2(R0)	:	RCM.REG, *	
000314	012701	000012			MOV	#12, R1	:	* , \$\$TMP2	1965
000320	001411			15\$:	BEQ	18\$			

H1

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0214
Page 19
(6)

000322	016700	000000G		MOV	L\$DLY,R0	; *,\$\$TMP1	
000326	001404			BEQ	17\$		
000330	005066	000004	16\$:	CLR	4(SP)	; \$\$TMP	
000334	005300			DEC	R0	; \$\$TMP1	
000336	001374			BNE	16\$		
000340	005301		17\$:	DEC	R1	; \$\$TMP2	
000342	000766			BR	15\$		
000344	016700	000000G	18\$:	MOV	RC25.ADDR,R0		1966
000350	016066	000002 000002		MOV	2(R0),2(SP)	; *,RC.REG	
000356	016667	000002 000002G		MOV	2(SP),RC25.DATA+2	; RC.REG,*	
000364	026602	000002		CMP	2(SP),R2	; RC25.DATA+2,TST.PAT	1968
000370	001436			BEQ	19\$		
000372	112767	000002 000000G		MOVB	#2,P.MASK		1971
000400	012767	000000G 000000G		MOV	#FMT2,P1		1972
000406	012767	000001 000000G		MOV	#1,P2		1973
000414	016700	000000G		MOV	RC25.ADDR,R0		1974
000420	062700	000002		ADD	#2,R0		
000424	010067	000000G		MOV	R0,P6		
000430	010267	000000G		MOV	R2,P4	; TST.PAT,*	1975
000434	016767	000002G 000000G		MOV	RC25.DATA+2,P5		1976
000442	104455			TRAP	55		1977
000444	000005			.WORD	5		
000446	000000G			.WORD	MSG.7		
000450	000000G			.WORD	RC25\$ERR.RPT		
000452	104465			TRAP	65		
000454	006000			ROR	R0		
000456	103403			BLO	19\$		
000460	012767	000001 000000G		MOV	#1,RETRIES		1979
000466	104467		19\$:	TRAP	67		1980
000470	006000			ROR	R0		
000472	103702			BLO	14\$		
000474	005203			INC	R3	; COUNT	1958
000476	020327	000001		CMP	R3,#1	; COUNT,*	
000502	101674			BLOS	13\$		
000504	005102			COM	R2	; TST.PAT	1985
000506	006302			ASL	R2	; TST.PAT	1986
000510	005204			INC	R4	; FLOAT	1955
000512	020427	000017		CMP	R4,#17	; FLOAT,*	
000516	101662			BLOS	12\$		
000520	032767	000001 000000G	20\$:	BIT	#1,RETRIES		1991
000526	001402			BEQ	21\$		
000530	004767	000000G		JSR	PC,DO.RETRIES		
000534	005767	000000G	21\$:	TST	NUM.RETRIES		1993
000540	001402			BEQ	22\$		
000542	000167	177274		JMP	2\$		
000546	012700	177777	22\$:	MOV	#-1,R0	; *,RCM.REG	1997
000552	010077	000000G		MOV	R0,@RC25.ADDR	; RCM.REG,*	
000556	062706	000006	23\$:	ADD	#6,SP		1896
000562	000207			RTS	PC		

; Routine Size: 186 words, Routine Base: AC\$CODE + 0750
; Maximum stack depth per invocation: 12 words

.SBTTL T3 TEST SECTION

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

000000 004767 177210
000000
000004 104466
000006 006000
000010 103773
000012 000207

T3::
1\$: JSR PC,\$T3
TRAP 66
ROR R0
BLO 1\$
RTS PC

1997

; Routine Size: 6 words, Routine Base: AC\$CODE + 1534
; Maximum stack depth per invocation: 2 words

; 1999 1 !<BLF/PAGE>

```

: 2000 1
: 2001 3
: 2002 3
: 2003 3
: 2004 3
: 2005 3
: 2006 3
: 2007 3
: 2008 3
: 2009 3
: 2010 3
: 2011 3
: 2012 3
: 2013 3
: 2014 3
: 2015 3
: 2016 3
: 2017 3
: 2018 3
: 2019 3
: 2020 3
: 2021 3
: 2022 3
: 2023 3
: 2024 3
: 2025 3
: 2026 3
: 2027 3
: 2028 3
: 2029 3
: 2030 4
: 2031 4
: 2032 4
: 2033 4
: 2034 4
: 2035 4
: 2036 4
: 2037 4
: 2038 4
: 2039 4
: 2040 4
: 2041 5
: 2042 5
: 2043 5
: 2044 5
: 2045 5
: 2046 5
: 2047 5
: 2048 5
: 2049 4
: 2050 5
: 2051 5
: 2052 5
: 2053 5
: 2054 5
: 2055 6
: 2056 6
    
```

```

!
! BGNTST;
!
! **
! TEST #4 - VECTOR AND BR LEVEL TEST
!
! DESCRIPTION:
!
! THE INIT SEQUENCE WILL BE STARTED WITH THE INTERRUPT ENABLE BIT SET TO
! VERIFY THE AZTEC'S VECTOR AND BR LEVEL.
!
! THIS TEST ASSUMES THE VECTOR GIVEN BY THE OPERATOR IS CORRECT.
!
! THE PRIORITY LEVEL OF THE INTERRUPT REQUEST WILL BE VERIFIED.
!
! FAILURE OF THE AZTEC TO VECTOR PROPERLY WILL NECESSITATE THAT THIS
! PROGRAM BE RESTARTED. A COMPLETED INTERRUPT AT THE WRONG BR LEVEL
! WILL BE REPORTED.
!
! LOOP ON ERROR WILL RESTART THIS TEST IF THE ERROR IS RECOVERABLE.
!
! --
!
!
! NUM_RETRIES = ZERO;
!
! if .SWP_TRACE then PRINTF (DBM11);           ! TEST 4
!
! while (.NUM_RETRIES lequ .SWP_RETRIES) do
!   begin
!     TIP = 5;
!     TEMP = PRI07;
!     I_AM_NEX = FALSE;
!     B_MASK = 0;
!     DATA1 = %o'104600' + .RT_TABLE [RT_VECTOR]/4;
!     SETPRI (.TEMP);
!
!     if AZT_INIT ()
!
!     then
!       begin
!         ERRDF (6, MSG_14, RC25$ERR_RPT);
!
!         if .RET_STATUS then DECODE ();
!
!         CKLOOP;
!         RETRIES = TRUE;
!       end
!     else
!       begin
!         WRT_RC25 (RCSA, .DATA1);
!         DELAY (5000);
!
!         while (.TEMP gequ %o'140') do
!           begin
!             ! START WITH HIGHEST PRIORITY
!             ! CLEAR INTERRUPT FLAG
!             ! STEP 1 READ MASK
!             ! INTERRUPT ENABLE BIT SET
!             ! SET MOST PRIORITY
!
!             ! BRING UP TO STEP 1 READ
!             ! AND GET STATUS
!             ! IF ERROR
!             ! THEN
!             ! REPORT IT
!
!             ! DECODE STATUS
!
!             ! WRITE STEP 1 DATA
!             ! WAIT FOR INTERRUPT VER:C
!           end
!         end
!       end
!     end
!   end
! end
    
```

```

: 2057 6      if .I_AM_NEX ealu ALL_ONES then exitloop;  !IF INTERRUPT DID NOT
: 2058 6
: 2059 6      TEMP = .TEMP - %0'40';                      ! NOT OCCUR
: 2060 6      SETPRI (.TEMP);                             ! LOWER CPU PRIORITY
: 2061 6      RETRIES = TRUE;
: 2062 5      end;
: 2063 5
: 2064 4      end;
: 2065 4
: 2066 4      if .I_AM_NEX ealu ALL_ONES                  ! IF INTERRUPT OCCURED
: 2067 4      then
: 2068 5      begin
: 2069 5      TIP = .TEMP+5 * 1;                          ! GET PRIORITY
: 2070 5      SETPRI (PRI00);                             ! SET MOST PRIORITY TO 0
: 2071 5      SETVEC (.RT_TABLE [RT_VECTOR], NXMI, .TIP); ! SET UP SERVICE ROUTINE.
: 2072 5      PRINTF (INI_MSG, .RT_TABLE [RT_VECTOR], .TIP);
: 2073 5
: 2074 5      if .TIP nequ .RT_TABLE [RT_BR_LEVEL] then PRINTF (BRERR);      ! IF RECEIVED BR IS NOT THE
: 2075 5                                          ! SAME AS TYPED REPORT ERROR
: 2076 5
: 2077 5      RETRIES = FALSE;
: 2078 5      end
: 2079 4      else
: 2080 5      begin
: 2081 5      RETRIES = TRUE;
: 2082 5      ERRDF (7, END_MSG, 0);                      ! ERROR
: 2083 5      CKLOOP;
: 2084 4      end;
: 2085 4
: 2086 4      if .RETRIES then DO_RETRIES ();
: 2087 4
: 2088 4      if (.NUM_RETRIES ealu ZERO) then exitloop;
: 2089 4
: 2090 3      end;
: 2091 3
: 2092 1      ENDTST;

```

Address	Hex	Dec	Label	Instruction	Comment	Line
000000	010146		\$T4:	MOV R1, -(SP)		1998
000002	005746			TST -(SP)		
000004	005067	000000G		CLR NUM.RETRIES		2025
000010	032767	000001 000000G		BIT #1, SWP.TRACE		2027
000016	001407			BEQ 1\$		
000020	012746	000000G		MOV #DBM11, -(SP)		
000024	012746	000001		MOV #1, -(SP)		
000030	010600			MOV SP, R0	: SP,*	
000032	104417			TRAP 17		
000034	022626			CMP (SP), (SP)		
000036	026767	000000G 000000G	1\$:	CMP NUM.RETRIES, SWP.RETRIES		2029
000044	101402			BLOS 2\$		
000046	000167	000520		JMP 16\$		
000052	012767	000005 000000G	2\$:	MOV #5, TIP		2031
000060	012767	000340 000000G		MOV #340, TEMP		2032
000066	005067	000000G		CLR I.AM.NEX		2033
000072	105067	000000G		CLRB B.MASK		2034
000076	016700	000000G		MOV RT.TABLE, R0		2035

000102	016046	000002			MOV	2(RO),-(SP)			
000106	012746	000004			MOV	#4,-(SP)			
000112	004767	000000G			JSR	PC,BL\$DIV			
000116	010067	000000G			MOV	RO,DATA1			
000122	162767	073200	000000G		SUB	#73200,DATA1			
000130	016700	000000G			MOV	TEMP,RO	:		2036
000134	104441				TRAP	41			
000136	004767	000000G			JSR	PC,AZT.INIT	:		2038
000142	006000				ROR	RO			
000144	103023				BCC	5\$			
000146	104455				TRAP	55	:		2042
000150	000006				.WORD	6			
000152	000000G				.WORD	MSG.14			
000154	000000G				.WORD	RC25\$ERR.RPT			
000156	032767	000001	000000G		BIT	#1,RET.STATUS	:		2044
000164	001402				BEQ	3\$			
000166	004767	000000G			JSR	PC,DECODE			
000172	104465			3\$:	TRAP	65			
000174	006000				ROR	RO			
000176	103002				BCC	4\$			
000200	022626				CMP	(SP)*,(SP)*			
000202	000573				BR	16\$			
000204	012767	000001	000000G	4\$:	MOV	#1,RETRIES	:		2047
000212	000444				BR	10\$:		2038
000214	016701	000000G		5\$:	MOV	DATA1,R1	:	*.RCM.REG	2051
000220	016700	000000G			MOV	RC25.ADDR,RO			
000224	010160	000002			MOV	R1,2(RO)	:	RCM.REG,*	
000230	012701	011610			MOV	#11610,R1	:	*. \$\$TMP2	2052
000234	001411			6\$:	BEQ	9\$			
000236	016700	000000G			MOV	L\$DLY,RO	:	*. \$\$TMP1	
000242	001404				BEQ	8\$			
000244	005066	000004		7\$:	CLR	4(SP)	:	\$\$TMP	
000250	005300				DEC	RO	:	\$\$TMP1	
000252	001374				BNE	7\$			
000254	005301			8\$:	DEC	R1	:	\$\$TMP2	
000256	000766				BR	6\$			
000260	026727	000000G	000140	9\$:	CMP	TEMP,#140	:		2054
000266	103416				BLO	10\$			
000270	026727	000000G	177777		CMP	I.AM.NEX,#-1	:		2057
000276	001412				BEQ	10\$			
000300	162767	000040	000000G		SUB	#40,TEMP	:		2059
000306	016700	000000G			MOV	TEMP,RO	:		2060
000312	104441				TRAP	41			
000314	012767	000001	000000G		MOV	#1,RETRIES	:		2061
000322	000756				BR	9\$:		2054
000324	026727	000000G	177777	10\$:	CMP	I.AM.NEX,#-1	:		2066
000332	001065				BNE	12\$			
000334	016716	000000G			MOV	TEMP,(SP)	:		2069
000340	012746	177773			MOV	#-5,-(SP)			
000344	004767	000000G			JSR	PC,BL\$SHF			
000350	010067	000000G			MOV	RO,TIP			
000354	005267	000000G			INC	TIP			
000360	005000				CLR	RO	:		2070
000362	104441				TRAP	41			
000364	016716	000000G			MOV	TIP,(SP)	:		2071
000370	012746	000000G			MOV	#NXMI,-(SP)			
000374	016700	000000G			MOV	RT.TABLE,RO			

M1

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0219
Page 24
(7)

000400	016046	000002		MOV	2(RO),-(SP)		
000404	012746	000003		MOV	#3,-(SP)		
000410	104437			TRAP	37		
000412	016716	000000G		MOV	TIP,(SP)	:	2072
000416	016700	000000G		MOV	RT, TABLE, RO		
000422	016046	000002		MOV	2(RO),-(SP)		
000426	012746	000000G		MOV	#INI,MSG,-(SP)		
000432	012746	000003		MOV	#3,-(SP)		
000436	010600			MOV	SP,RO	: SP,*	
000440	104417			TRAP	17		
000442	016700	000000G		MOV	RT, TABLE, RO	:	2074
000446	026760	000000G	000004	CMP	TIP,4(RO)		
000454	001407			BEQ	11\$		
000456	012716	000000G		MOV	#BRERR,(SP)		
000462	012746	000001		MOV	#1,-(SP)		
000466	010600			MOV	SP,RO	: SP,*	
000470	104417			TRAP	17		
000472	005726			TST	(SP).		
000474	005067	000000G	11\$:	CLR	RETRIES	:	2077
000500	062706	000016		ADD	#16,SP	:	2068
000504	000414			BR	13\$:	2066
000506	012767	000001	000000G	MOV	#1,RETRIES	:	2081
000514	104455		12\$:	TRAP	55	:	2082
000516	000007			.WORD	7		
000520	000000G			.WORD	END.MSG		
000522	000000			.WORD	0		
000524	104465			TRAP	65		
000526	006000			ROR	RO		
000530	103002			BCC	13\$		
000532	022626			CMP	(SP),.(SP).		
000534	000416			BR	16\$		
000536	032767	000001	000000G	BIT	#1,RETRIES	:	2086
000544	001402			BEQ	14\$		
000546	004767	000000G		JSR	PC,DO.RETRIES		
000552	005767	000000G	14\$:	TST	NUM.RETRIES	:	2088
000556	001002			BNE	15\$		
000560	022626			CMP	(SP),.(SP).		
000562	000403			BR	16\$		
000564	022626		15\$:	CMP	(SP),.(SP).	:	2030
000566	000167	177244		JMP	1\$:	2029
000572	005726		16\$:	TST	(SP).	:	1998
000574	012601			MOV	(SP),.R1		
000576	000207			RTS	PC		

: Routine Size: 192 words, Routine Base: AC\$CODE * 1550
: Maximum stack depth per invocation: 14 words

000000	004767	177174	T4::	.SBTTL	T4 TEST SECTION		
000000			1\$:	JSR	PC,\$T4	:	2090
000004	104466			TRAP	66		
000006	006000			ROR	RO		
000010	103773			BLO	1\$		
000012	000207			RTS	PC		

N1

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B1199-16 V1.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0220
Page 25
(7)

: Routine Size: 6 words. Routine Base: AC#CODE * 2350
: Maximum stack depth per invocation: 2 words

: 2093 1 !<BLF/PAGE>

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 B1100-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4SEQ 0221
Page 26
(8)

```

: 2094 3  BGNTS1;
: 2095 3
: 2096 3  !..
: 2097 3
: 2098 3  ! TEST 5: STEP 1 -3 INITIALIZATION TEST
: 2099 3
: 2100 3  ! DESCRIPTION:
: 2101 3
: 2102 3  ! THIS TEST WILL CHECK FOR INFORMATIONS ECHOED FROM PORT AT
: 2103 3  ! EACH STEP READ COMING UPTO THAT STEP FROM SCRATCH. IF THERE WAS
: 2104 3  ! AN ERROR REPORTED OR ECHOED INFORMATIONS WERE INCORRECT
: 2105 3  ! THE SAME WILL BE REPORTED.
: 2106 3  ! LOOP ON ERROR WILL BE FROM THE BEGINNING OF SUB TEST.
: 2107 3  !..
: 2108 3
: 2109 3  NUM_RETRIES = ZERO;                ! CLEAR RETRY COUNTER
: 2110 3
: 2111 3  if .SWP_TRACE then PRINTF (DBM9);    ! TEST 5
: 2112 3
: 2113 3  while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 2114 4      begin
: 2115 4
: 2116 4      ! STEP1 READ
: 2117 4
: 2118 6      BGNSUB;
: 2119 6
: 2120 6
: 2121 6  !check if using Q_bus and flag
: 2122 6      TEMP = READBUS ();
: 2123 6
: 2124 6  ! STEP 1 READ
: 2125 6
: 2126 6      B_MASK = 0;                ! START PORT INIT WITH MASK = 0
: 2127 6
: 2128 6      if AZT_INIT ()                ! BRING UP TO STEP 1 READ
: 2129 6          ! AND GET STATUS
: 2130 6      then                            ! IF ERROR
: 2131 7          begin                            ! THEN
: 2132 7              ERRDF (8, MSG_14, RC25ERR_RPT);    ! REPORT IT
: 2133 7
: 2134 7              if .RET_STATUS then DECODE ();    ! DECODE STATUS
: 2135 7
: 2136 7              CKLOOP;
: 2137 7              RETRIES = TRUE;
: 2138 6          end;
: 2139 6
: 2140 6  ! CHECK FOR CONTROLLER DEPENDENT INFORMATION FROM RCSA AT STEP 1 READ
: 2141 6
: 2142 8      if ((.RC25_DATA [RCSA, RCSA_NV])                ! CHECK THAT THE NV BIT DID
: 2143 8          ! NOT SET.
: 2144 7          or not (.RC25_DATA [RCSA, RCSA_DI]))        ! CHECK IF DI BIT SET
: 2145 7          ! or (.TEMP) and not (.RC25_DATA [RCSA, RCSA_QB]) ! CHECK THE QB BIT
: 2146 7          ! or not (.TEMP) and (.RC25_DATA [RCSA, RCSA_QB]))
: 2147 6      then                            ! IF NOT SET
: 2148 7          begin                            ! THEN
: 2149 7              P_MASK = 2;
: 2150 7              PI = FMT3;

```


ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0222
Page 27
(8)

```

: 2151 7      P2 = ADAPT;
: 2152 7      P4 = (.RC25_ADDR) * 2;
: 2153 7      P5 = .RC25_DATA [RCSA, RC_ALL];
: 2154 7      P6 = %'01';
: 2155 7      ERRDF (9, MSG_14, RC25$ERR_RPT);
: 2156 7      CKLOOP;
: 2157 7      RETRIES = TRUE;
: 2158 6      end;
: 2159 6
: 2160 6      TEMP = .RC25_DATA [RCSA, RC_ALL];
: 2161 6      TEMP = .TEMP<6, 5>;
: 2162 6      PRINTF (FMT5, .TEMP);
: 2163 4      ENDSUB;
: 2164 4
: 2165 4      ! STEP1 WRITE WITH STEP 2 READ
: 2166 4
: 2167 6      BGNSUB;
: 2168 6      B_MASK = 1;
: 2169 6      DATA1 = %'137600' * .RT_TABLE [RT_VECTOR]/4;
: 2170 6      ! STEP1 WRITE DATA FOR MAX
: 2171 6      ! RING LENGTHS, IE AND
: 2172 6      ! VECTOR ADDRESS
: 2173 6      if AZT_INIT ()
: 2174 6      then
: 2175 7      begin
: 2176 7      ERRDF (10, MSG_14, RC25$ERR_RPT);
: 2177 7      ! REPORT ERROR
: 2178 7      if .RET_STATUS then DECODE ();
: 2179 7      ! DECODE STATUS
: 2180 7      CKLOOP;
: 2181 7      RETRIES = TRUE;
: 2182 7      end
: 2183 6      else
: 2184 7      begin
: 2185 7      ! CHECK FOR ECHOED INFORMATIONS AT STEP2 READ
: 2186 7      TEMP = .DATA1<8, 8>;
: 2187 7      ! SAVE EXPECTED DATA
: 2188 8      if (.RC25_DATA [RCSA, RCSA_7_0] nequ .TEMP)
: 2189 7      then
: 2190 8      begin
: 2191 8      ! IF ECHOED INFO DOES NOT
: 2192 8      ! MATCH REPORT ERROR
: 2193 8      P_MASK = 2;
: 2194 8      P1 = FMT2;
: 2195 8      P2 = ADAPT;
: 2196 8      P4 = .TEMP;
: 2197 8      P5 = .RC25_DATA [RCSA, RCSA_7_0];
: 2198 8      P6 = .RT_TABLE [RT_IP_ADDRESS] * 2;
: 2199 8      ERRDF (11, MSG_11, RC25$ERR_RPT);
: 2200 8      CKLOOP;
: 2201 7      RETRIES = TRUE;
: 2202 7      end;
: 2203 6      end;
: 2204 6      PRINTF (FMT4, .RC25_DATA [RCSA, RCSA_PTN]); ! GIVE PORT TYPE NUMBER
: 2205 4      ENDSUB;
: 2206 4
: 2207 4      ! STEP 2 WRITE WITH A STEP 3 READ

```


ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

```

; 2265 4      ENDSUB;
; 2266 4
; 2267 4      if (.RETRIES) then DO_RETRIES ();
; 2268 4
; 2269 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
; 2270 4
; 2271 3      end;
; 2272 3
; 2273 1      ENDTST;

```

000000	005067	000000G		.SBTTL	\$T5 TEST SECTION		
000004	032767	000001	000000G	\$T5:	CLR	NUM.RETRIES	2109
000012	001407				BIT	#1,SWP.TRACE	2111
000014	012746	000000G			BEQ	1\$	
000020	012746	000001			MOV	#DBM9,-(SP)	
000024	010600				MOV	#1,-(SP)	
000026	104417				MOV	SP,R0	; SP,*
000030	022626				TRAP	17	
000032	026767	000000G	000000G	1\$:	CMP	(SP)+,(SP)+	
000040	101401				CMP	NUM.RETRIES,SWP.RETRIES	2113
000042	000207				BLOS	2\$	
000044	104402			2\$:	RTS	PC	
000046	105067	000000G			TRAP	2	2114
000052	004767	000000G			CLRB	B.MASK	2126
000056	006000				JSR	PC,AZT.INIT	2128
000060	103023				ROR	R0	
000062	104455				BCC	5\$	
000064	000010				TRAP	55	2132
000066	000000G				.WORD	10	
000070	000000G				.WORD	MSG.14	
000072	032767	000001	000000G		.WORD	RC25\$ERR.RPT	
000100	001402				BIT	#1,RET.STATUS	2134
000102	004767	000000G			BEQ	3\$	
000106	104465			3\$:	JSR	PC,DECODE	
000110	006000				TRAP	65	
000112	103003				ROR	R0	
000114	162706	000006			BCC	4\$	
000120	000507				SUB	#6,SP	
000122	012767	000001	000000G	4\$:	BR	9\$	2137
000130	032767	002000	000002G	5\$:	MOV	#1,RETRIES	2142
000136	001004				BIT	#2000,RC25.DATA+2	
000140	032767	000400	000002G		BNE	6\$	
000146	001042				BIT	#400,RC25.DATA+2	2144
000150	112767	000002	000000G	6\$:	BNE	8\$	
000156	012767	000000G	000000G		MOVB	#2,P.MASK	2149
000164	012767	000001	000000G		MOV	#FMT3,P1	2150
000172	016700	000000G			MOV	#1,P2	2151
000176	062700	000002			MOV	RC25.ADDR,R0	2152
000202	010067	000000G			ADD	#2,R0	
000206	016767	000002G	000000G		MOV	R0,P4	
000214	012767	000001	000000G		MOV	RC25.DATA+2,P5	2153
000222	104455				MOV	#1,P6	2154
000224	000011				TRAP	55	2155
000226	000000G				.WORD	11	
000230	000000G				.WORD	MSG.14	
					.WORD	RC25\$ERR.RPT	

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0225
Page 30
(8)

000232	104465				TRAP	65			
000234	006000				ROR	R0			
000236	103003				BCC	7\$			
000240	162706	000006			SUB	#6,SP			
000244	000435				BR	9\$			
000246	012767	000001	000000G	7\$:	MOV	#1,RETRIES	:		2157
000254	016767	000002G	000000G	8\$:	MOV	RC25.DATA+2,TEMP	:		2160
000262	006267	000000G			ASR	TEMP	:		2161
000266	006267	000000G			ASR	TEMP			
000272	006267	000000G			ASR	TEMP			
000276	006267	000000G			ASR	TEMP			
000302	006267	000000G			ASR	TEMP			
000306	006267	000000G			ASR	TEMP			
000312	042767	177740	000000G		BIC	#177740,TEMP			
000320	016746	000000G			MOV	TEMP,-(SP)	:		2162
000324	012746	000000G			MOV	#FMT5,-(SP)			
000330	012746	000002			MOV	#2,-(SP)			
000334	010600				MOV	SP,R0	:	SP,*	
000336	104417				TRAP	17			
000340	062706	000006		9\$:	ADD	#6,SP	:		2114
000344	104467				TRAP	67	:		2162
000346	006000				ROR	R0			
000350	103635				BLO	2\$			
000352	104402			10\$:	TRAP	2	:		2163
000354	112767	000001	000000G		MOVB	#1,B.MASK	:		2168
000362	016700	000000G			MOV	RT.TABLE,R0	:		2169
000366	016046	000002			MOV	2(R0),-(SP)			
000372	012746	000004			MOV	#4,-(SP)			
000376	004767	000000G			JSR	PC,BL\$DIV			
000402	010067	000000G			MOV	R0,DATA1			
000406	162767	040200	000000G		SUB	#40200,DATA1			
000414	004767	000000G			JSR	PC,AZT.INIT	:		2173
000420	006000				ROR	R0			
000422	103023				BCC	13\$			
000424	104455				TRAP	55	:		2176
000426	000012				.WORD	12			
000430	000000G				.WORD	MSG.14			
000432	000000G				.WORD	RC25\$ERR.RPT			
000434	032767	000001	000000G		BIT	#1,RET.STATUS	:		2178
000442	001402				BEQ	11\$			
000444	004767	000000G			JSR	PC,DECODE			
000450	104465			11\$:	TRAP	65			
000452	006000				ROR	R0			
000454	103002				BCC	12\$			
000456	024646				CMP	-(SP),-(SP)			
000460	000476				BR	16\$			
000462	012767	000001	000000G	12\$:	MOV	#1,RETRIES	:		2181
000470	000456				BR	15\$:		2173
000472	005067	000000G		13\$:	CLR	TEMP	:		2186
000476	116767	000001G	000000G		MOVB	DATA1+1,TEMP			
000504	005000				CLR	R0	:		2188
000506	156700	000002G			BISB	RC25.DATA+2,R0			
000512	020067	000000G			CMP	R0,TEMP			
000516	001443				BEQ	15\$			
000520	112767	000002	000000G		MOVB	#2,P.MASK	:		2191
000526	012767	000000G	000000G		MOV	#FMT2,P1	:		2192
000534	012767	000001	000000G		MOV	#1,P2	:		2193

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0227
Page 32
(8)

001036	005000			CLR	R0	:		2235
001040	156700	000002G		BISB	RC25.DATA+2,R0	:		
001044	010067	000000G		MOV	R0,P5	:		
001050	017700	000000G		MOV	ERR.TABLE,R0	:		2236
001054	062700	000002		ADD	#2,R0	:		
001060	010067	000000G		MOV	R0,P6	:		
001064	104455			TRAP	55	:		2237
001066	000015			.WORD	15	:		
001070	000000G			.WORD	MSG.11	:		
001072	000000G			.WORD	RC25\$ERR.RPT	:		
001074	104465			TRAP	65	:		
001076	006000			ROR	R0	:		
001100	103403			BLO	20\$:		
001102	012767	000001	000000G	MOV	#1,RETRIES	:		2239
001110	104467		20\$:	TRAP	67	:		2242
001112	006000			ROR	R0	:		
001114	103665			BLO	17\$:		
001116	104402		21\$:	TRAP	2	:		2244
001120	112767	000007	000000G	MOVB	#7,B.MASK	:		2249
001126	005067	000000G		CLR	DATA3	:		2250
001132	004767	000000G		JSR	PC,AZT.INIT	:		2252
001136	006000			ROR	R0	:		
001140	103023			BCC	24\$:		
001142	104455			TRAP	55	:		2255
001144	000016			.WORD	16	:		
001146	000000G			.WORD	MSG.14	:		
001150	000000G			.WORD	RC25\$ERR.RPT	:		
001152	032767	000001	000000G	BIT	#1,RET.STATUS	:		2257
001160	001402			BEQ	22\$:		
001162	004767	000000G		JSR	PC,DECODE	:		
001166	104465		22\$:	TRAP	65	:		
001170	006000			ROR	R0	:		
001172	103003			BCC	23\$:		
001174	162706	000010		SUB	#10,SP	:		
001200	000426			BR	25\$:		
001202	012767	000001	000000G	MOV	#1,RETRIES	:		2260
001210	016746	000002G	24\$:	MOV	RC25.DATA+2,-(SP)	:		2264
001214	042716	177760		BIC	#177760,(SP)	:		
001220	016700	000002G		MOV	RC25.DATA+2,R0	:		
001224	006200			ASR	R0	:		
001226	006200			ASR	R0	:		
001230	006200			ASR	R0	:		
001232	006200			ASR	R0	:		
001234	042700	177760		BIC	#177760,R0	:		
001240	010046			MOV	R0,-(SP)	:		
001242	012746	000000G		MOV	#FMT6,-(SP)	:		
001246	012746	000003		MOV	#3,-(SP)	:		
001252	010600			MOV	SP,R0	:	SP,*	
001254	104417			TRAP	17	:		
001256	062706	000010	25\$:	ADD	#10,SP	:		2244
001262	104467			TRAP	67	:		2264
001264	006000			ROR	R0	:		
001266	103713			BLO	21\$:		
001270	032767	000001	000000G	BIT	#1,RETRIES	:		2267
001276	001402			BEQ	26\$:		
001300	004767	000000G		JSR	PC,DO.RETRIES	:		
001304	005767	000000G	26\$:	TST	NUM.RETRIES	:		2269

ZRCFB3 CZRCFC0 RC25 FR END TEST
V03.0 TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

001310 001402
001312 000167 176514
001316 000207

274: BEQ 274
JMP 14
RTS PC

2092

: Routine Size: 360 words, Routine Base: AC\$CODE + 2364
: Maximum stack depth per invocation: 6 words

000000 004767 176454 TS:: .SBTTL T5 TEST SECTION

000000 14: JSR PC,\$T5
000004 104466 TRAP 66
000006 006000 ROR R0
000010 103773 BLO 14
000012 000207 RTS PC

2271

: Routine Size: 6 words, Routine Base: AC\$CODE + 3704
: Maximum stack depth per invocation: 2 words

: 2274 1 !<BLF/PAGE>

```

: 2275 1  !
: 2276 3  !BGNTST;
: 2277 3  !
: 2278 3  !
: 2279 3  !**
: 2280 3  ! TEST #6 - PURGE AND POLL TEST
: 2281 3  !
: 2282 3  ! DESCRIPTION:
: 2283 3  !
: 2284 3  ! THIS TEST WILL PERFORM THE FIRST THREE STEPS OF THE INIT SEQUENCE.
: 2285 3  ! WHEN THE HOST RESPONDS TO THE STEP 3 TRANSITION IT WILL WRITE A ONE
: 2286 3  ! BIT TO BIT 15 OF THE SA REGISTER, THERBY REQUESTING THE EXECUTION OF
: 2287 3  ! PURGE AND POLL TESTING. THE HOST THEN WAITS FOR THE SA REGISTER TO
: 2288 3  ! TRANSITION TO A ZERO VALUE. THE HOST THEN WRITES ZEROS TO THE SA
: 2289 3  ! REGISTER SIMULATING A "PURGE COMPLETED" HOST ACTION. THE HOST THEN
: 2290 3  ! READS THE IP REGISTER TO SMULATE A "START POLLING" COMMAND FROM THE
: 2291 3  ! HOST TO THE PORT. THE TEST IS COMPLETE WHEN THE CONTROLLER ANNOUNCES
: 2292 3  ! THE TRANSITION TO STEP 4 IN THE SA REGISTER.
: 2293 3  !
: 2294 3  ! FAILURE TO PROPERLY COMPLETE THIS TEST WILL BE REPORTED.
: 2295 3  !
: 2296 3  ! LOOP ON ERROR WILL RESTART THE TEST.
: 2297 3  !
: 2298 3  !
: 2299 3  !
: 2300 3  ! if .SWP_TRACE then PRINTF (DBM12);           ! TEST 6
: 2301 3  !
: 2302 3  ! NUM_RETRIES = ZERO;
: 2303 3  !
: 2304 4  ! while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 2305 4  ! begin
: 2306 4  !   TIP = 6;
: 2307 4  !   B_MASK = 3;
: 2308 4  !   DATA1 = %o'100200' + .RT_TABLE [RT_VECTOR]/4;           ! IE AND VECTOR ADDRESS
: 2309 4  !   DATA2 = RINGBASE;                                       ! RING BASE LOW ADDRESS
: 2310 4  !   DATA3 = %o'100000';                                       ! PURGE AND POLL
: 2311 4  !
: 2312 4  !   if AZT_INIT ()                                           ! DO UPTO STEP 3 READ AND
: 2313 4  !   then                                                    ! CHECK FOR ERRORS
: 2314 4  !     begin                                                  ! IF ERRORS THEN
: 2315 4  !       ERRDF (15, MSG_14, RC25$ERR_RPT);                   ! REPORT THEM
: 2316 4  !     if .RET_STATUS then DECODE ();                          ! DECODE STATUS
: 2317 4  !
: 2318 4  !     CKLOOP;
: 2319 4  !     RETRIES = TRUE;
: 2320 4  !   else
: 2321 4  !     begin
: 2322 4  !       WRT_RC25 (RCSA, .DATA3);                               ! WRITE PURGE AND POLL
: 2323 4  !
: 2324 4  !       while (.RC25_ADDR [RCSA, RC_ALL] nequ ZERO) do
: 2325 4  !         DELAY (10);                                         ! WAIT UNTIL SA=0
: 2326 4  !
: 2327 4  !       WRT_RC25 (RCSA, FALSE);                               ! WRITE ALL ZERO'S TO SA
: 2328 4  !       DATA1 = .RC25_ADDR [RCIP, RC_ALL];                   ! READ THE IP REGISTER
: 2329 4  !       DATA1 = %o'10';                                       ! INIT THE LOOP COUNT
: 2330 4  !
: 2331 4  !       while (.DATA1 nequ ZERO) do

```



```

: 2332 6      begin
: 2333 6      delay (333);
: 2334 6
: 2335 6      if .I_AM_NEX ealu ALL_ONES then exitloop;
: 2336 6
: 2337 6      DATA1 = .DATA1 - 1;
: 2338 5      end;
: 2339 5
: 2340 5      if .I_AM_NEX ealu ALL_ONES
: 2341 5      then
: 2342 6          begin
: 2343 6          RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];
: 2344 6
: 2345 6          if .RC25_DATA [RCSA, RCSA_ER]      ! IF PORT FATAL ERROR
: 2346 6          then
: 2347 7              begin
: 2348 7                  RET_STATUS = PFE_CODE;
: 2349 7                  P1 = FMT3;
: 2350 7                  P2 = ADAPT;
: 2351 7                  P4 = (.RC25_ADDR) * 2;
: 2352 7                  P5 = .RC25_DATA [RCSA, RC_ALL];
: 2353 7                  P6 = %0'04';
: 2354 7                  P MASK = 2;
: 2355 7                  ERRDF (16, MSG_14, RC25$ERR_RPT);
: 2356 7                  DECODE ();
: 2357 7                  CKLOOP;
: 2358 7                  RETRIES = TRUE;
: 2359 6              end;
: 2360 6
: 2361 7          if (.RC25_DATA [RCSA, RCSA_STEP] nequ %b'1000')      ! CHECK FOR STEP 4 COMPLETE
: 2362 6          then
: 2363 7              begin
: 2364 7                  P1 = FMT3;
: 2365 7                  P2 = ADAPT;
: 2366 7                  P4 = (.RC25_ADDR) * 2;
: 2367 7                  P5 = .RC25_DATA [RCSA, RC_ALL];
: 2368 7                  P6 = %0'10';      ! MASK = STEP 4
: 2369 7                  P MASK = 2;
: 2370 7                  ERRDF (17, MSG_14, RC25$ERR_RPT);
: 2371 7                  CKLOOP;
: 2372 7                  RETRIES = TRUE;
: 2373 6              end;
: 2374 6
: 2375 6          end
: 2376 5      else
: 2377 6          begin
: 2378 6          RET_STATUS = CTO_CODE;
: 2379 6          RETRIES = TRUE;
: 2380 6          ERRDF (18, MSG_9, 0);
: 2381 6          DECODE ();
: 2382 5          end;
: 2383 5
: 2384 4      end;
: 2385 4
: 2386 4      if (.RETRIES) then DO_RETRIES ();
: 2387 4
: 2388 4      if (.NUM_RETRIES ealu ZERO) then exitloop;

```

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0231
Page 36
(9)

: 2389 4
: 2390 3
: 2391 3
: 2392 1

end;
ENDTST;

000000	010146		\$T6:	.SBTTL \$T6 TEST SECTION			
000002	162706	000010		MOV R1,-(SP)	:		2273
000006	032767	000001 000000G		SUB #10,SP	:		
000014	001407			BIT #1,SWP.TRACE	:		2298
000016	012746	000000G		BEQ 1\$			
000022	012746	000001		MOV #DBM12,-(SP)			
000026	010600			MOV #1,-(SP)			
000030	104417			MOV SP,RO	: SP,*		
000032	022626			TRAP 17			
000034	005067	000000G	1\$:	CMP (SP),.(SP).			
000040	026767	000000G 000000G	2\$:	CLR NUM.RETRIES	:		2300
000046	101402			CMP NUM.RETRIES,SWP.RETRIES	:		2302
000050	000167	000674		BLOS 3\$			
000054	012767	000006 000000G	3\$:	JMP 25\$			
000062	112767	000003 000000G		MOV #6,TIP	:		2304
000070	016700	000000G		MOVB #3,B.MASK	:		2305
000074	016046	000002		MOV RT.TABLE,RO	:		2306
000100	012746	000004		MOV 2(RO),-(SP)			
000104	004767	000000G		MOV #4,-(SP)			
000110	010067	000000G		JSR PC,BL\$DIV			
000114	162767	077600 000000G		MOV RO,DATA1			
000122	012767	000000G 000000G		SUB #77600,DATA1			
000130	012767	100000 000000G		MOV #RINGBASE,DATA2	:		2307
000136	004767	000000G		MOV #-100000,DATA3	:		2308
000142	006000			JSR PC,AZT.INIT	:		2310
000144	103025			ROR RO			
000146	104455			BCC 6\$			
000150	000017			TRAP 55	:		2313
000152	000000G			.WORD 17			
000154	000000G			.WORD MSG.14			
000156	032767	000001 000000G		.WORD RC25\$ERR.RPT			
000164	001402			BIT #1,RET.STATUS	:		2315
000166	004767	000000G		BEQ 4\$			
000172	104465		4\$:	JSR PC,DECODE			
000174	006000			TRAP 65			
000176	103003			ROR RO			
000200	022626			BCC 5\$			
000202	000167	000542		CMP (SP),.(SP).			
000206	012767	000001 000000G	5\$:	JMP 25\$			
000214	000167	000474		MOV #1,RETRIES	:		2318
000220	016701	000000G	6\$:	JMP 22\$:		2310
000224	016700	000000G		MOV DATA3,R1	: *.RCM.REG		2322
000230	010160	000002		MOV RC25.ADDR,RO			
000234	016700	000000G	7\$:	MOV R1,2(RO)	: RCM.REG,*		
000240	016066	000002 000004		MOV RC25.ADDR,RO	:		2324
000246	001414			MOV 2(RO),4(SP)	: *.RC.REG		
000250	012701	000012		BEQ 11\$			
000254	001767		8\$:	MOV #12,R1	: *,\$\$TMP2		2325
000256	016700	000000G		BEQ 7\$			
000262	001404			MOV L\$DLY,RO	: *,\$\$TMP1		
				BEQ 10\$			

M2

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B1:00-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0232
Page 37
(9)

000264	005066	000012	98:	CLR	12(SP)	:	\$\$TMP	
000270	005300			DEC	R0	:	\$\$TMP1	
000272	001374			BNE	98			
000274	005301		108:	DEC	R1	:	\$\$TMP2	
000276	000766			BR	88			
000300	005001		118:	CLR	R1	:	RCM.REG	2327
000302	016700	000000G		MOV	RC25.ADDR,R0			
000306	005060	000002		CLR	2(R0)			
000312	011066	000006		MOV	(R0),6(SP)	:	RC25.ADDR,RC.REG	2328
000316	012767	000010	000000G	MOV	#10,DATA1	:		2329
000324	001423		128:	BEQ	178	:		2331
000326	012701	000515		MOV	#515,R1	:	*,\$\$TMP2	2333
000332	001411		138:	BEQ	168			
000334	016700	000000G		MOV	L#DLY,R0	:	*,\$\$TMP1	
000340	001404			BEQ	158			
000342	005066	000012	148:	CLR	12(SP)	:	\$\$TMP	
000346	005300			DEC	R0	:	\$\$TMP1	
000350	001374			BNE	148			
000352	005301		158:	DEC	R1	:	\$\$TMP2	
000354	000766			BR	138			
000356	026727	000000G	168:	CMP	I.AM.NEX,#-1	:		2335
000364	001403			BEQ	178			
000366	005367	000000G		DEC	DATA1	:		2337
000372	000754			BR	128	:		2331
000374	026727	000000G	178:	CMP	I.AM.NEX,#-1	:		2340
000402	001130			BNE	218			
000404	016700	000000G		MOV	RC25.ADDR,R0	:		2343
000410	016066	000002	000010	MOV	2(R0),10(SP)	:	*.RC.REG	
000416	016667	000010	000002G	MOV	10(SP),RC25.DATA.2	:	RC.REG,*	
000424	100046			BPL	198	:		2345
000426	012767	000021	000000G	MOV	#21,RET.STATUS	:		2348
000434	012767	000000G	000000G	MOV	#FMT3,P1	:		2349
000442	012767	000001	000000G	MOV	#1,P2	:		2350
000450	016700	000000G		MOV	RC25.ADDR,R0	:		2351
000454	062700	000002		ADD	#2,R0			
000460	010067	000000G		MOV	R0,P4			
000464	016767	000002G	000000G	MOV	RC25.DATA.2,P5	:		2352
000472	012767	000004	000000G	MOV	#4,P6	:		2353
000500	112767	000002	000000G	MOVB	#2,P.MASK	:		2354
000506	104455			TRAP	55	:		2355
000510	000020			.WORD	20			
000512	000000G			.WORD	MSG.14			
000514	000000G			.WORD	RC25\$ERR.RPT			
000516	004767	000000G		JSR	PC,DECODE	:		2356
000522	104465			TRAP	65			
000524	006000			ROR	R0			
000526	103002			BCC	188			
000530	022626			CMP	(SP).,(SP).			
000532	000506			BR	258			
000534	012767	000001	000000G	MOV	#1,RETURNS	:		2358
000542	016700	000002G		MOV	RC25.DATA.2,R0	:		2361
000546	042700	103777		BIC	#103777,R0			
000552	020027	040000		CMP	R0,#40000			
000556	001456			BEQ	228			
000560	012767	000000G	000000G	MOV	#FMT3,P1	:		2364
000566	012767	000001	000000G	MOV	#1,P2	:		2365
000574	016700	000000G		MOV	RC25.ADDR,R0	:		2366

ZRCF03
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0233
Page 38
(9)

```

000600 062700 000002          ADD      #2,R0
000604 010067 000000G        MOV      R0,P4
000610 016767 000002G 000000G  MOV      RC25,DATA+2,P5
000616 012767 000010 000000G  MOV      #10,P6
000624 112767 000002 000000G  MOVB     #2,P.MASK
000632 104455                TRAP     55
000634 000021                .WORD   21
000636 000000G              .WORD   MSG.14
000640 000000G              .WORD   RC25+ERR.RPT
000642 104465                TRAP     65
000644 006000                ROR      R0
000646 103002                BCC     20#
000650 022626                CMP     (SP)+,(SP)+
000652 000436                BR      25#
000654 012767 000001 000000G  20#:    MOV     #1,RETRIES
000662 000414                BR      22#
000664 012767 000011 000000G  21#:    MOV     #11,RET.STATUS
000672 012767 000001 000000G  MOV     #1,RETRIES
000700 104455                TRAP     55
000702 000022                .WORD   22
000704 000000G              .WORD   MSG.9
000706 000000                .WORD   0
000710 004767 000000G        JSR     PC,DECODE
000714 032767 000001 000000G  22#:    BIT     #1,RETRIES
000722 001402                BEQ     23#
000724 004767 000000G        JSR     PC,DO.RETRIES
000730 005767 000000G        TST     NUM.RETRIES
000734 001002                BNE     24#
000736 022626                CMP     (SP)+,(SP)+
000740 000403                BR      25#
000742 022626                24#:    CMP     (SP)+,(SP)+
000744 000167 177070        JMP     2#
000750 062706 000010        25#:    ADD     #10,SP
000754 012601                MOV     (SP)+,R1
000756 000207                RTS     PC

```

: Routine Size: 248 words, Routine Base: AC#CODE * 3720
: Maximum stack depth per invocation: 9 words

```

000000 004767 177014        T6::    .SBTTL  T6 TEST SECTION
000000 1#:    JSR     PC,#T6
000004 104466                TRAP     66
000006 006000                ROR      R0
000010 103773                BLO     1#
000012 000207                RTS     PC

```

: Routine Size: 6 words, Routine Base: AC#CODE * 4700
: Maximum stack depth per invocation: 2 words

: 2393 1 !<BLF/PAGE>

```

: 2394 1
: 2395 3
: 2396 3
: 2397 3
: 2398 3
: 2399 3
: 2400 3
: 2401 3
: 2402 3
: 2403 3
: 2404 3
: 2405 3
: 2406 3
: 2407 3
: 2408 3
: 2409 3
: 2410 3
: 2411 3
: 2412 3
: 2413 3
: 2414 3
: 2415 3
: 2416 3
: 2417 3
: 2418 3
: 2419 4
: 2420 4
: 2421 4
: 2422 4
: 2423 4
: 2424 4
: 2425 4
: 2426 4
: 2427 4
: 2428 4
: 2429 4
: 2430 4
: 2431 5
: 2432 5
: 2433 5
: 2434 5
: 2435 5
: 2436 5
: 2437 5
: 2438 5
: 2439 4
: 2440 5
: 2441 5
: 2442 5
: 2443 5
: 2444 6
: 2445 6
: 2446 6
: 2447 6
: 2448 5
: 2449 5
: 2450 4

!
BGNTST;

!..
! TEST #7 - SMALL RING BUFFER INIT TEST

! DESCRIPTION:

! THE AZTEC WILL BE INITIALIZED WITHOUT INTERRUPTS AND USING THE
! SMALLEST RING BUFFER. THIS WILL BE THE FIRST TIME THAT THE
! INITIALIZATION SEQUENCE IS CARRIED OUT TO COMPLETION. INITIALIZING
! WITH THE SMALLEST RING BUFFER MINIMIZES THE HOST MEMORY AREA WITH
! WHICH THE AZTEC CONTROLLER MUST BE ABLE TO COMMUNICATE.

! FAILURE TO PROPERLY INITIATE THE AZTEC WILL BE REPORTED.

! IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM THE
! START OF THIS TEST.

!
if .SWP_TRACE then PRINTF (DBM13);          ! TEST 7

NUM_RETRIES = ZERO;

while (.NUM_RETRIES lequ .SWP_RETRIES) do
begin
  TIP = 7;
  B_MASK = %o'17';
  DATA1 = %o'100200';
  DATA2 = RING_B [0];
  DATA3 = 0;
  DATA4<0, 1> = 1;
  RING_B [0] = ALL_ONES;
  RING_B [1] = ALL_ONES;

  if AZP_INIT ()
  then
  begin
    ERRDF (19, MSG_14, RC25ERR_RPT);
    if .RET_STATUS then DECODE ();

    CKLOOP;
    RETRIES = TRUE;
  end
else
begin
  if .RING_B [0] nequ 0 and .RING_B [1] nequ 0
  then
  begin
    ERRDF (20, MSG_10, 0);
    CKLOOP;
    RETRIES = TRUE;
  end;
end;

! VER.B0
! SELECT ALL STEPS
! STEP 1 WRITE WITH MIN. RING SIZES
! SET UP RING BASE ADDRESS
! INIT RING_B [0] AND [1]
! WITH ALL ONES (-1)
! DO INIT STEPS
! IF ERROR THEN
! THEN REPORT THE ERROR
! DECODE RETURN STATUS
! TEST THAT THE RC25 CLEARED
! RING BUFFERS
! IF NOT THEN ERROR
! AND REPORT IT

```

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

```

: 2451 4
: 2452 4      if (.RETRIES) then DO_RETRIES ();
: 2453 4
: 2454 4      if (.NUM_RETRIES eqv ZERO) then exitloop;
: 2455 4
: 2456 3      end;
: 2457 3
: 2458 3      return;
: 2459 1      ENDTST;
    
```

```

000000 032767 000001 000000G      $T7:  .SBTTL  $T7 TEST SECTION
000006 001407                      BIT      #1,SWP.TRACE      ;      2414
000010 012746 000000G              BEQ      1$
000014 012746 000001              MOV      #DBM13,-(SP)
000020 010600                      MOV      #1,-(SP)
000022 104417                      MOV      SP,R0      ; SP,*
000024 022626                      TRAP     17
000026 005067 000000G              CMP      (SP),.(SP).
000032 026767 000000G 000000G      1$:  CLR      NUM.RETRIES      ;      2416
000040 101105                      2$:  CMP      NUM.RETRIES,SWP.RETRIES      ;      2418
000042 012767 000007 000000G      MOV      #7,TIP      ;      2420
000050 112767 000017 000000G      MOV      #17,B.MASK      ;      2421
000056 012767 100200 000000G      MOV      #-77600,DATA1      ;      2422
000064 012767 000100' 000000G      MOV      #RING.B,DATA2      ;      2423
000072 005067 000000G      CLR      DATA3      ;      2424
000076 152767 000001 000000G      BISB    #1,DATA4      ;      2425
000104 012767 177777 000100'      MOV      #-1,RING.B      ;      2426
000112 012767 177777 000102'      MOV      #-1,RING.B+2      ;      2427
000120 004767 000000G      JSR     PC,AZP.INIT      ;      2429
000124 006000                      ROR      R0
000126 103021                      BCC     4$
000130 104455                      TRAP     55      ;      2432
000132 000023                      .WORD   23
000134 000000G                      .WORD   MSG.14
000136 000000G                      .WORD   RC25$ERR.RPT
000140 032767 000001 000000G      BIT      #1,RET.STATUS      ;      2434
000146 001402                      BEQ     3$
000150 004767 000000G              JSR     PC,DECODE
000154 104465                      3$:  TRAP     65
000156 006000                      ROR      R0
000160 103435                      BLO     7$
000162 012767 000001 000000G      MOV      #1,RETRIES      ;      2437
000170 000420                      BR      5$      ;      2429
000172 005767 000100'              4$:  TST     RING.B      ;      2442
000176 001415                      BEQ     5$
000200 005767 000102'              TST     RING.B+2
000204 001412                      BEQ     5$
000206 104455                      TRAP     55      ;      2445
000210 000024                      .WORD   24
000212 000000G                      .WORD   MSG.10
000214 000000                      .WORD   0
000216 104465                      TRAP     65
000220 006000                      ROR      R0
000222 103414                      BLO     7$
000224 012767 000001 000000G      MOV      #1,RETRIES      ;      2447
    
```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0236
Page 41
(10)

000232	032767	000001	000000G	5#:	BIT	#1,RETRIES	:	2452
000240	001402				BEQ	6#		
000242	004767	000000G			JSR	PC,DO,RETRIES		
000246	005767	000000G		6#:	TST	NUM.RETRIES	:	2454
000252	001267				BNE	2#		
000254	000207			7#:	RTS	PC	:	2392

: Routine Size: 87 words, Routine Base: AC#CODE + 4714
: Maximum stack depth per invocation: 4 words

000000	004767	177516		T7::	.SBTTL	T7 TEST SECTION		
000000				1#:	JSR	PC,\$T7	:	2458
000004	104466				TRAP	66		
000006	006000				ROR	RO		
000010	103773				BLO	1#		
000012	000207				RTS	PC		

: Routine Size: 6 words, Routine Base: AC#CODE + 5172
: Maximum stack depth per invocation: 2 words

: 2460 1 !<BLF/PAGE>

```

: 2461 1  !
: 2462 3  !BGNTST;
: 2463 3  !
: 2464 3  !
: 2465 3  !**
: 2466 3  ! TEST #8 - LARGE RING BUFFER INIT TEST
: 2467 3  !
: 2468 3  ! DESCRIPTION:
: 2469 3  !
: 2470 3  ! THE INIT SEQUENCE IS EXECUTED WITHOUT INTERRUPTS WITH A RING BUFFER
: 2471 3  ! LARGE ENOUGH TO COVER THE NORMAL HOST COMMUNICATIONS AREA PACKET AND
: 2472 3  ! BUFFER SPACE ( A 5 IN MESSAGE LENGTH AND A 5 IN COMMAND LENGTH).
: 2473 3  !
: 2474 3  ! A FAILURE TO COMPLETE THE INITIALIZATION SEQUENCE WITHOUT ERROR WILL BE
: 2475 3  ! REPORTED.
: 2476 3  !
: 2477 3  ! IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE TO THE
: 2478 3  ! BEGINNING OF THIS TEST.
: 2479 3  !
: 2480 3  !--
: 2481 3  !
: 2482 3  ! if .SWP_TRACE then PRINTF (DBM14);           ! TEST 8
: 2483 3  !
: 2484 3  ! NUM_RETRIES = ZERO;
: 2485 3  !
: 2486 4  ! while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 2487 4  ! begin
: 2488 4  !   TIP = 8;
: 2489 4  !   B MASK = %o'17';           ! SET MASK BIT FOR COMPLETE INIT.
: 2490 4  !   DATA1<15, 1> = TRUE;     ! SET BIT 15 FOR STEP-1 WRITE
: 2491 4  !   DATA1<14, 1> = 0;       ! NO DIAGNOSTIC WRAP MODE
: 2492 4  !   DATA1<11, 3> = SND_SIZ; ! SET UP 16 COMMAND RINGS LENGTH
: 2493 4  !   DATA1<8, 3> = REC_SIZ;  ! SET UP 16 RESPONSE RINGS LENGTH
: 2494 4  !   DATA1<7, 1> = 0;        ! DISABLE INTERRUPT
: 2495 4  !   DATA1<0, 7> = 0;        ! LOAD INTERRUPT VECTOR ADDRESS
: 2496 4  !   DATA2 = COM_AREA;       ! LOAD COMMUNICATIONS AREA ADDRESS
: 2497 4  !   DATA3 = ZERO;          ! HI-ORDER ADDR = ZERO
: 2498 4  !   DATA4 = %o'177403';     ! "LAST FAIL" PACKET RESPONSE BIT SET
: 2499 4  ! !INITIALIZE COM_AREA WITH ALL_ONES PRIOR TO INIT
: 2500 4  !   incru I from 0 to RING_SIZE - 1 do
: 2501 4  !     incru J from 0 to 1 do
: 2502 4  !       COM_AREA [.I, .J, WORD_REF] = ALL_ONES;
: 2503 4  !
: 2504 4  !   if AZP_INIT ()           ! DO STEP INIT AND CHECK FOR ERROR
: 2505 4  !   then
: 2506 5  !     begin
: 2507 5  !       ERRDF (21, MSG_14, RC25$ERR_RPT); ! IF ERRORS THEN
: 2508 5  !       ! REPORT ERROR
: 2509 5  !       if .RET_STATUS then DECODE ();    ! DECODE STATUS
: 2510 5  !
: 2511 5  !       CKLOOP;
: 2512 5  !       RETRIES = TRUE;
: 2513 5  !     end
: 2514 4  !   else
: 2515 5  !     begin
: 2516 5  !       ! VER.80
: 2517 5  !       incru I from 0 to RING_SIZE - 1 do ! TEST RING AREA FOR ZEROES

```


ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0238
Page 43
(11)

```

: 2518 5
: 2519 5      incru J from 0 to 1 do
: 2520 5
: 2521 5      if .COM_AREA [.I, .J, WORD_REF] nequ 0 ! IF RING AREA IS NOT CLEAR
: 2522 5      then
: 2523 6      begin
: 2524 6      ERRDF (22, MSG_10, 0); ! THEN
: 2525 6      CKLOOP; ! REPORT ERROR
: 2526 6      RETRIES = TRUE;
: 2527 5      end;
: 2528 5
: 2529 4      end;
: 2530 4
: 2531 4      if (.RETRIES) then DO_RETRIES ();
: 2532 4
: 2533 4      if (.NUM_RETRIES eqv ZERO) then exitloop;
: 2534 4
: 2535 3      end;
: 2536 3
: 2537 3      return;
: 2538 1      ENDTST;

```

000000	004167	000000G	\$T8:	.SBTTL	\$T8 TEST SECTION			2459
000004	032767	000001 000000G		JSR	R1,\$SAVE2	:		2480
000012	001407			BIT	#1,SWP.TRACE	:		
000014	012746	000000G		BEQ	1\$			
000020	012746	000001		MOV	#DBM14,-(SP)			
000024	010600			MOV	#1,-(SP)			
000026	104417			MOV	SP,R0	:	SP,*	
000030	022626			TRAP	17			
000032	005067	000000G	1\$:	CMP	(SP)*,(SP)*			
000036	026767	000000G 000000G	2\$:	CLR	NUM.RETRIES	:		2482
000044	101133			CMP	NUM.RETRIES,SWP.RETRIES	:		2484
000046	012767	000010 000000G		BHI	12\$			
000054	112767	000017 000000G		MOV	#10,TIP	:		2486
000062	012767	122000 000000G		MOVB	#17,B.MASK	:		2487
000070	012767	000000G 000000G		MOV	#122000,DATA1	:		2493
000076	005067	000000G		MOV	#COM.AREA,DATA2	:		2494
000102	012767	177403 000000G		CLR	DATA3	:		2495
000110	005001			MOV	#-375,DATA4	:		2496
000112	005002		3\$:	CLR	R1	:	I	2499
000114	010100		4\$:	CLR	R2	:	J	2501
000116	006300			MOV	R1,R0	:	I,*	2502
000120	060200			ASL	R0			
000122	006300			ADD	R2,R0	:	J,*	
000124	012760	177777 000000G		ASL	R0			
000132	005202			MOV	#-1,COM.AREA(R0)			
000134	020227	000001		INC	R2	:	J	2501
000140	101765			CMP	R2,#1	:	J,*	
000142	005201			BLOS	4\$			
000144	020127	000037		INC	R1	:	I	2499
000150	101760			CMP	R1,#37	:	I,*	
000152	004767	000000G		BLOS	3\$			
000156	006000			JSR	PC,AZP.INIT	:		2504
000160	103021			ROR	R0			
				BCC	6\$			

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

000162	104455			TRAP	55			2507
000164	000025			.WORD	25			
000166	000000G			.WORD	MSG.14			
000170	000000G			.WORD	RC25ERR.RPT			
000172	032767	000001	000000G	BIT	#1,RET.STATUS			2509
000200	001402			BEQ	5#			
000202	004767	000000G		JSR	PC,DECODE			
000206	104465		5#:	TRAP	65			
000210	006000			ROR	R0			
000212	103450			BLO	12#			
000214	012767	000001	000000G	MOV	#1,RETRIES			2512
000222	000433			BR	10#			2504
000224	005001		6#:	CLR	R1		I	2517
000226	005002		7#:	CLR	R2		J	2519
000230	010100		8#:	MOV	R1,R0		I,*	2521
000232	006300			ASL	R0			
000234	060200			ADD	R2,R0		J,*	
000236	006300			ASL	R0			
000240	005760	000000G		TST	COM.AREA(R0)			
000244	001412			BEQ	9#			
000246	104455			TRAP	55			2524
000250	000026			.WORD	26			
000252	000000G			.WORD	MSG.10			
000254	000000			.WORD	0			
000256	104465			TRAP	65			
000260	006000			ROR	R0			
000262	103424			BLO	12#			
000264	012767	000001	000000G	MOV	#1,RETRIES			2526
000272	005202		9#:	INC	R2		J	2519
000274	020227	000001		CMP	R2,#1		J,*	
000300	101753			BLOS	8#			
000302	005201			INC	R1		I	2517
000304	020127	000037		CMP	R1,#37		I,*	
000310	101746			BLOS	7#			
000312	032767	000001	000000G	BIT	#1,RETRIES			2531
000320	001402			BEQ	11#			
000322	004767	000000G		JSR	PC,DO.RETRIES			
000326	005767	000000G	11#:	TST	NUM.RETRIES			2533
000332	001241			BNE	2#			
000334	000207		12#:	RTS	PC			2459

; Routine Size: 111 words, Routine Base: AC\$CODE + 5206
; Maximum stack depth per invocation: 7 words

000000	004767	177436	T8::	.SBTTL	T8 TEST SECTION			
000000			1#:	JSR	PC,\$T8			2537
000004	104466			TRAP	66			
000006	006000			ROR	R0			
000010	103773			BLO	1#			
000012	000207			RTS	PC			

; Routine Size: 6 words, Routine Base: AC\$CODE + 5544
; Maximum stack depth per invocation: 2 words

H3

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0240
Page 45
(11)

; 2539 1 !<BLF/PAGE>

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0241
Page 46
(12)

```

: 2540 1      !
: 2541 3      ! BGNTST;
: 2542 3
: 2543 3
: 2544 3      ! **
: 2545 3      ! TEST #9 - "DIAGNOSTIC MACHINE" CODE DOWN LINE LOAD TEST
: 2546 3      !
: 2547 3      ! DESCRIPTION:
: 2548 3      !     THIS "DIAGNOSTIC MACHINE" PROGRAM WILL ATTEMPT TO TRANSFER A BLOCK
: 2549 3      !     OF DATA FROM HOST MEMORY TO AN AREA IN THE CONTROLLER AND THEN
: 2550 3      !     EXAMINE THE TRANSFERED DATA.
: 2551 3      !
: 2552 3      !     IF THE TRANSFERED DATA NOT COMPARE CORRECTLY, THEN THE ERROR WILL
: 2553 3      !     BE REPORTED. THIS TEST ALSO REPORTS ERRORS IF ANY OF THE ROUTINES
: 2554 3      !     USED RETURNED FAILURE CODE.
: 2555 3      !
: 2556 3      !     IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM
: 2557 3      !     THE START OF THIS TEST.
: 2558 3      !
: 2559 3      ! --
: 2560 3      label
: 2561 3      BLOCK1;
: 2562 3
: 2563 3      if .SWP_TRACE then PRINTF (DBM15);           ! TEST 9
: 2564 3
: 2565 3      NUM_RETRIES = ZERO;
: 2566 3
: 2567 3      while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 2568 4          begin
: 2569 4
: 2570 4              if AZTEC_READY ()                   ! GET AZTEC READY
: 2571 4              then
: 2572 5                  begin
: 2573 5                      ERRDF (23, AZT_READY_ERR, 0); ! IF ERROR REPORT ERROR
: 2574 5
: 2575 5                      if .RET_STATUS then DECODE ();
: 2576 5
: 2577 5                      CKLOOP;
: 2578 5                      RETRIES = TRUE;
: 2579 5                  end
: 2580 4              else
: 2581 4      BLOCK1 :
: 2582 5                  begin
: 2583 5                      TEMP = .FREE_MEM_ADDR;       ! SAVE FREE MEMORY STARTING ADDR.
: 2584 5
: 2585 5                      incru COUNT from 0 to 1023 do ! FILL NEXT 1024 LOC. WITH DATAS
: 2586 5                          ! VER.80 LIMIT CHANGED TO 1023
: 2587 6                          begin
: 2588 6                              .TEMP = #o'125252'; ! WRITE DATA 0'125252' INTO MEMORY
: 2589 6                              TEMP = .TEMP + 2; ! INCREMENT THE POINTER BY 2
: 2590 6                          end;
: 2591 5
: 2592 5                      CMD_REF = 3;                ! SET COMMAND REFERENCE #3
: 2593 5                      BUF_DESCRPTR = DM_09;      ! DM-PROGRAM STARTING ADDRESS
: 2594 5                      BYTE_COUNT = 93*2;        ! TOTAL DM PROGRAM LENGTH BYTE COUNTS
: 2595 5
: 2596 5                      if EX_SUP_PRG ()           ! ISSUE AN "EXECUTE SUPPLIED PRG" CMD

```

```

: 2597 5      then
: 2598 6      begin
: 2599 6      ERRDF (24, EXE_SUP_ERR, 0);
: 2600 6
: 2601 6      if .RET_STATUS then DECODE ();
: 2602 6
: 2603 6      CKLOOP;
: 2604 6      RETRIES = TRUE;
: 2605 6      leave BLOCK1;
: 2606 5      end;
: 2607 5
: 2608 5      H_SADD = .FREE_MEM_ADDR;
: 2609 5      H_EADD = 0;
: 2610 5      BUF_LENGTH = 1024;
: 2611 5      CMD_REF = 4;
: 2612 5      BUF_DESCRPTR = H_SADD;
: 2613 5      BYTE_COUNT = 06;
: 2614 5
: 2615 5      if SEND_DATA ()
: 2616 5      then
: 2617 6      begin
: 2618 6      ERRDF (25, SND_DATA_ERR, 0);
: 2619 6
: 2620 6      if .RET_STATUS then DECODE ();
: 2621 6
: 2622 6      CKLOOP;
: 2623 6      RETRIES = TRUE;
: 2624 6      leave BLOCK1;
: 2625 5      end;
: 2626 5
: 2627 5      CMD_REF = 5;
: 2628 5      BUF_DESCRPTR = TIP;
: 2629 5      BYTE_COUNT = 02;
: 2630 5
: 2631 5      if REC_DATA ()
: 2632 5      then
: 2633 6      begin
: 2634 6      ERRDF (26, RE_DATA_ERR, 0);
: 2635 6
: 2636 6      if .RET_STATUS then DECODE ();
: 2637 6
: 2638 6      CKLOOP;
: 2639 6      RETRIES = TRUE;
: 2640 6      leave BLOCK1;
: 2641 5      end;
: 2642 5
: 2643 5      if .TIP nequ %o'104'
: 2644 5      then
: 2645 6      begin
: 2646 6      ERRDF (27, DMC_ERR, 0);
: 2647 6      CKLOOP;
: 2648 6      RETRIES = TRUE;
: 2649 5      end;
: 2650 5
: 2651 4      end;
: 2652 4
: 2653 4      if (.RETRIES) then DO_RETRIES ();

```

```

! STATUS BIT INDICATES ERROR
! THEN
!

```

```

! VER.B0

```

```

! LO BYTE FREE HOST MEMORY ADDRESS
! HIGH BYTE FREE MEMORY ADDRESS
! TOTAL FREE HOST MEMORY SIZE
! COMMAND REFERENCE 04
! DESCRIPTOR ADDRESS
! TOTAL BYTES TO BE TRANSFER

```

```

! ISSUE SEND DATA COMMAND
! STATUS BIT INDICATES ERROR
! THEN
!

```

```

! VER.B0

```

```

! CLEAN THE BUFFER
! SET BYTE COUNTS = 2

```

```

! SENT A RECEIVE DATA COMMAND
! STATUS BIT INDICATES ERROR
! THEN
!

```

```

! VER.B0

```

```

! IS REMOTE PROGRAM SENT DONE FLAG -
! TO THE HOST
! NO. THEN
! REPORT ERROR

```

```

: 2654 4
: 2655 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 2656 4
: 2657 3      end;
: 2658 3
: 2659 3      return;
: 2660 1      ENDTST;

```

```

000000 032767 000001 000000G      $T9:  .SBTTL  $T9 TEST SECTION
000006 001407                      BIT      #1,SWP.TRACE      ;
000010 012746 000000G              BEQ      1$
000014 012746 000001              MOV      #DBM15,-(SP)
000020 010600                      MOV      #1,-(SP)
000022 104417                      MOV      SP,R0      ; SP,*
000024 022626                      TRAP     17
000026 005067 000000G              CMP      (SP), (SP).
000032 026767 000000G 000000G    1$:    CLR      NUM.RETRIES      ;
000040 101401                      2$:    CMP      NUM.RETRIES,SWP.RETRIES ;
000042 000207                      BLOS    3$
000044 004767 000000G              RTS      PC
000050 006000                      3$:    JSR      PC,AZTEC.READY      ;
000052 103022                      ROR      R0
000054 104455                      BCC     6$
000056 000027                      TRAP     55      ;
000060 000000G                    .WORD   27
000062 000000                    .WORD   AZT.READY.ERR
000064 032767 000001 000000G      BIT      #1,RET.STATUS      ;
000072 001402                      BEQ      4$
000074 004767 000000G              JSR      PC,DECODE
000100 104465                      4$:    TRAP     65
000102 006000                      ROR      R0
000104 103001                      BHIS    5$
000106 000207                      RTS      PC
000110 012767 000001 000000G      5$:    MOV      #1,RETRIES      ;
000116 000576                      BR       14$
000120 016767 000000G 000000G    6$:    MOV      FREE.MEM.ADDR,TEMP      ;
000126 005000                      CLR      R0      ; COUNT
000130 012777 125252 000000G      7$:    MOV      #-52526,@TEMP      ;
000136 062767 000002 000000G      ADD     #2,TEMP      ;
000144 005200                      INC     R0      ; COUNT
000146 020027 001777              CMP     R0,#1777      ; COUNT,*
000152 101766                      BLOS    7$
000154 012767 000003 000000G      MOV     #3,CMD.REF      ;
000162 012767 000000G 000000G      MOV     #DM.09,BUF.DESCRPTR ;
000170 012767 000272 000000G      MOV     #272,BYTE.COUNT    ;
000176 004767 000000G              JSR     PC,EX.SUP.PRG      ;
000202 006000                      ROR     R0
000204 103021                      BCC     9$
000206 104455                      TRAP     55      ;
000210 000030                      .WORD   30
000212 000000G                    .WORD   EXE.SUP.ERR
000214 000000                    .WORD   0
000216 032767 000001 000000G      BIT     #1,RET.STATUS      ;
000224 001402                      BEQ     8$
000226 004767 000000G              JSR     PC,DECODE

```

000232	104465		8%:	TRAP	65			
000234	006000			ROR	RO			
000236	103541			BLO	16%			
000240	012767	000001	000000G	MOV	#1,RETRIES	:		2604
000246	000522			BR	14%	:		2598
000250	016767	000000G	000000G	MOV	FREE.MEM.ADDR,H.SADD	:		2608
000256	005067	000000G		CLR	H.EADD	:		2609
000262	012767	002000	000000G	MOV	#2000,BUF.LENGTH	:		2610
000270	012767	000004	000000G	MOV	#4,CMD.REF	:		2611
000276	012767	000000G	000000G	MOV	#H.SADD,BUF.DESCRPTR	:		2612
000304	012767	000006	000000G	MOV	#6,BYTE.COUNT	:		2613
000312	004767	000000G		JSR	PC.SEND.DATA	:		2615
000316	006000			ROR	RO			
000320	103021			BCC	11%			
000322	104455			TRAP	55	:		2618
000324	000031			.WORD	31			
000326	000000G			.WORD	SND.DATA.ERR			
000330	000000			.WORD	0			
000332	032767	000001	000000G	BIT	#1,RET.STATUS	:		2620
000340	001402			BEQ	10%			
000342	004767	000000G		JSR	PC.DECODE			
000346	104465			TRAP	65			
000350	006000			ROR	RO			
000352	103473			BLO	16%			
000354	012767	000001	000000G	MOV	#1,RETRIES	:		2623
000362	000454			BR	14%	:		2617
000364	012767	000005	000000G	MOV	#5,CMD.REF	:		2627
000372	012767	000000G	000000G	MOV	#TIP,BUF.DESCRPTR	:		2628
000400	012767	000002	000000G	MOV	#2,BYTE.COUNT	:		2629
000406	004767	000000G		JSR	PC.REC.DATA	:		2631
000412	006000			ROR	RO			
000414	103021			BCC	13%			
000416	104455			TRAP	55	:		2634
000420	000032			.WORD	32			
000422	000000G			.WORD	RE.DATA.ERR			
000424	000000			.WORD	0			
000426	032767	000001	000000G	BIT	#1,RET.STATUS	:		2636
000434	001402			BEQ	12%			
000436	004767	000000G		JSR	PC.DECODE			
000442	104465			TRAP	65			
000444	006000			ROR	RO			
000446	103435			BLO	16%			
000450	012767	000001	000000G	MOV	#1,RETRIES	:		2639
000456	000416			BR	14%	:		2633
000460	026727	000000G	000104	13%:	CMP	TIP,#104	:	2643
000466	001412			BEQ	14%	:		
000470	104455			TRAP	55	:		2646
000472	000033			.WORD	33			
000474	000000G			.WORD	DMC.ERR			
000476	000000			.WORD	0			
000500	104465			TRAP	65			
000502	006000			ROR	RO			
000504	103416			BLO	16%			
000506	012767	000001	000000G	MOV	#1,RETRIES	:		2648
000514	032767	000001	000000G	14%:	BIT	#1,RETRIES	:	2653
000522	001402			BEQ	15%			
000524	004767	000000G		JSR	PC.DO.RETRIES			

M3

ZRCFR3 CZRCFC0 RC25 FR END TEST 27-Mar-1985 15:27:28 VAX-11 Bliss-16 V4.0-579
 V03.0 TEST SECTION 27-Mar-1985 13:28:18 USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

```
000530 005767 000000G      15:  TST  NUM.RETRIES      ;
000534 001402              BEQ  16:
000536 000167 177270      JMP  2:
000542 000207              RTS  PC                ;
```

: Routine Size: 178 words, Routine Base: AC#CODE * 5560
 : Maximum stack depth per invocation: 4 words

```
000000 004767 177230      19::  .SBTTL  T9 TEST SECTION
000000              1:  JSR  PC,19
000004 104466              TRAP 66
000006 006000              ROR  R0
000010 103773              BLO  1:
000012 000207              RTS  PC
```

: Routine Size: 6 words, Routine Base: AC#CODE * 6324
 : Maximum stack depth per invocation: 2 words

```
: 2661 1  :
: 2662 3  : BGNTST;
: 2663 3  :
: 2664 3  :
: 2665 3  : TEST #10 - NONEXISTENT MEMORY TEST
: 2666 3  :
: 2667 3  : DESCRIPTION:
: 2668 3  :
: 2669 3  : THIS "DIAGNOSTIC MACHINE" PROGRAM WILL ATTEMPT TO READ THE FIRST
: 2670 3  : ADDRESS OF THE I/O PAGE OF THE HOST CPU. THIS LOCATION IS RESERVED
: 2671 3  : FOR DIAGNOSTICS AND A NXM SHOULD OCCUR.
: 2672 3  :
: 2673 3  : IF THE CONTROLLER DOES NOT SEE THE NXM, THERE WILL BE A FRU CALLOUT
: 2674 3  : OF THE ADAPTER CARD.
: 2675 3  :
: 2676 3  : IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM
: 2677 3  : THE START OF THIS TEST.
: 2678 3  :
: 2679 3  :
: 2680 3  : label
: 2681 3  :   BLOCK1;                ! VER.80
: 2682 3  :
: 2683 3  : if .SWP_TRACE then PRINTF (DBM16);    ! TEST 10
: 2684 3  :
: 2685 3  : NUM_RETRIES = ZERO;
: 2686 3  :
: 2687 3  : while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 2688 4  :   begin
: 2689 4  :     TIP = 0;                ! INIT TIP
: 2690 4  :
: 2691 4  :     if AZTEC_READY ()      ! GET AZTEC READY FOR OPERATION
: 2692 4  :     then
: 2693 5  :       begin
: 2694 5  :         ERRDF (28, AZT_READY_ERR, 0);
```



```

: 2695 5
: 2696 5
: 2697 5
: 2698 5
: 2699 5
: 2700 5
: 2701 4
: 2702 4
: 2703 5
: 2704 5
: 2705 5
: 2706 5
: 2707 5
: 2708 5
: 2709 5
: 2710 5
: 2711 5
: 2712 5
: 2713 5
: 2714 6
: 2715 6
: 2716 6
: 2717 6
: 2718 6
: 2719 6
: 2720 6
: 2721 6
: 2722 5
: 2723 5
: 2724 5
: 2725 5
: 2726 5
: 2727 5
: 2728 5
: 2729 5
: 2730 5
: 2731 5
: 2732 5
: 2733 6
: 2734 6
: 2735 6
: 2736 6
: 2737 6
: 2738 6
: 2739 6
: 2740 6
: 2741 5
: 2742 5
: 2743 5
: 2744 5
: 2745 6
: 2746 6
: 2747 6
: 2748 6
: 2749 5
: 2750 5
: 2751 4

      if .RET_STATUS then DECODE ();

      CKLOOP;
      RETRIES = TRUE;
      end
    else
BLOCK1 :
      begin
!VER.80 VEC_AD = 04;
!VER.80 SETVEC (.VEC_AD, NXMI, PRI04);
!VER.80 SET_INT_VECTOR ();
!VER.80 WRT_RC25 (RCSA, ONE);
      CMD_REF = 3;
      BUF_DESCPTR = DM_10;
      BYTE_COUNT = 58*2;

      if EX_SUP_PRG ()
      then
        begin
          ERRDF (29, EXE_SUP_ERR, 0);

          if .RET_STATUS then DECODE ();

          CKLOOP;
          RETRIES = TRUE;
          leave BLOCK1;
        end;
      ! VER.80

      : SET INT. VECTOR ADDR. TO 4
      : SET THE VECTOR ADDR., SERVICE
      : ROUTINE ADDR. AND INT. PRIORITY
      : COMMAND REFERENCE #
      : DMCODE STARTING ADDRESS
      : BYTE COUNTS

      : ISSUE AN EXECUTE SUPPLIED CMD
      : IF ERROR
      : THEN

      : COMMAND REFERENCE #
      : CLEAN THE BUFFER
      : SET BYTE COUNTS = 2

      : SENT A RECEIVE DATA COMMAND
      : STATUS BIT INDICATES ERROR
      : THEN

      : DID YOU GET SUCCESS FROM DM CODE?
      : NO
      : THEN
      : REPORT ERROR

      : WAIT FOR "DONE" SIGNAL FROM DM

      CMD_REF = 4;
      BUF_DESCPTR = TIP;
      BYTE_COUNT = 02;

      if REC_DATA ()
      then
        begin
          ERRDF (30, RE_DATA_ERR, 0);

          if .RET_STATUS then DECODE ();

          CKLOOP;
          RETRIES = TRUE;
          leave BLOCK1;
        end;
      ! VER.80

      if .TIP ealu ZERO
      then
        begin
          ERRDF (31, DMC_ERR, 0);
          CKLOOP;
          RETRIES = TRUE;
          end;
        end;
      end;

```

```

: 2752 4
: 2753 4      if (.RETRIES) then DO_RETRIES ();
: 2754 4
: 2755 4      if (.NUM_RETRIES eqv ZERO) then exitloop;
: 2756 4
: 2757 3      end;
: 2758 3
: 2759 3      return;
: 2760 1      ENDTST;
    
```

000000	032767	000001	000000G	\$T10:	.SBTTL \$T10 TEST SECTION		
000006	001407				BIT #1,SWP.TRACE	:	2683
000010	012746	000000G			BEQ 1\$		
000014	012746	000001			MOV #DBM16,-(SP)		
000020	010600				MOV #1,-(SP)		
000022	104417				MOV SP,R0	: SP,*	
000024	022626				TRAP 17		
000026	005067	000000G		1\$:	CMP (SP),(SP)		
000032	026767	000000G	000000G	2\$:	CLR NUM.RETRIES	:	2685
000040	101151				CMP NUM.RETRIES,SWP.RETRIES	:	2687
000042	005067	000000G			BHI 11\$		
000046	004767	000000G			CLR TIP	:	2689
000052	006000				JSR PC,AZTEC.READY	:	2691
000054	103021				ROR R0		
000056	104455				BCC 4\$		
000060	000034				TRAP 55	:	2694
000062	000000G				.WORD 34		
000064	000000				.WORD AZT.READY.ERR		
000066	032767	000001	000000G		.WORD 0		
000074	001402				BIT #1,RET.STATUS	:	2696
000076	004767	000000G			BEQ 3\$		
000102	104465			3\$:	JSR PC,DECODE		
000104	006000				TRAP 65		
000106	103526				ROR R0		
000110	012767	000001	000000G		BLO 11\$		
000116	000511				MOV #1,RETRIES	:	2699
000120	012767	000003	000000G	4\$:	BR 9\$:	2691
000126	012767	000000G	000000G		MOV #3,CMD.REF	:	2708
000134	012767	000164	000000G		MOV #DM.10,BUF.DESCRPTR	:	2709
000142	004767	000000G			MOV #164,BYTE.COUNT	:	2710
000146	006000				JSR PC,EX.SUP.PRG	:	2712
000150	103021				ROR R0		
000152	104455				BCC 6\$		
000154	000035				TRAP 55	:	2715
000156	000000G				.WORD 35		
000160	000000				.WORD EXE.SUP.ERR		
000162	032767	000001	000000G		.WORD 0		
000170	001402				BIT #1,RET.STATUS	:	2717
000172	004767	000000G			BEQ 5\$		
000176	104465			5\$:	JSR PC,DECODE		
000200	006000				TRAP 65		
000202	103470				ROR R0		
000204	012767	000001	000000G		BLO 11\$		
000212	000453				MOV #1,RETRIES	:	2720
000214	012767	000004	000000G	6\$:	BR 9\$:	2714
					MOV #4,CMD.REF	:	2727

000222	012767	000000G	000000G		MOV	#TIP, BUF.DESCRPTR	:	2728
000230	012767	000002	000000G		MOV	#2, BYTE.COUNT	:	2729
000236	004767	000000G			JSR	PC, REC.DATA	:	2731
000242	006000				ROR	R0		
000244	103021				BCC	8#		
000246	104455				TRAP	55	:	2734
000250	000036				.WORD	36		
000252	000000G				.WORD	RE.DATA.ERR		
000254	000000				.WORD	0		
000256	032767	000001	000000G		BIT	#1, RET.STATUS	:	2736
000264	001402				BEQ	7#		
000266	004767	000000G			JSR	PC, DECODE		
000272	104465			7#:	TRAP	65		
000274	006000				ROR	R0		
000276	103432				BLO	11#		
000300	012767	000001	000000G		MOV	#1, RETRIES	:	2739
000306	000415				BR	9#	:	2733
000310	005767	000000G		8#:	TST	TIP	:	2743
000314	001012				BNF	9#		
000316	104455				TRAP	55	:	2746
000320	000037				.WORD	37		
000322	000000G				.WORD	DMC.ERR		
000324	000000				.WORD	0		
000326	104465				TRAP	65		
000330	006000				ROR	R0		
000332	103414				BLO	11#		
000334	012767	000001	000000G		MOV	#1, RETRIES	:	2748
000342	032767	000001	000000G	9#:	BIT	#1, RETRIES	:	2753
000350	001402				BEQ	10#		
000352	004767	000000G			JSR	PC, DO.RETRIES		
000356	005767	000000G		10#:	TST	NUM.RETRIES	:	2755
000362	001223				BNE	2#		
000364	000207			11#:	RTS	PC	:	2660

: Routine Size: 123 words, Routine Base: AC#CODE + 6340
: Maximum stack depth per invocation: 4 words

					.SBTTL	T10 TEST SECTION		
000000	004767	177406		T10::				
000000				1#:	JSR	PC, #T10	:	2759
000004	104466				TRAP	66		
000006	006000				ROR	R0		
000010	103773				BLO	1#		
000012	000207				RTS	PC		

: Routine Size: 6 words, Routine Base: AC#CODE + 6726
: Maximum stack depth per invocation: 2 words

: 2761 1 !
: 2762 3 ! BGNTST;
: 2763 3 !
: 2764 3 ! ..
: 2765 3 ! TEST #11 - BUS ADDRESSING/DATA TEST A


```

: 2823 5      H_SADD = .FREE_MEM_ADDR;      ! LO-BYTE FREE HOST MEMORY ADDRESS
: 2824 5      TEMP = .H_SADD;              ! LOAD START ADDRESS FOR INIT
: 2825 5      BUF_LENGTH = .MEM_SIZ;      ! TOTAL FREE HOST MEMORY SIZE
: 2826 5      H_EADD = .H_SADD - 2 * (.BUF_LENGTH*2); ! END OF FREE MEM ADDRESS
: 2827 5      CMD_REF = 4;                 ! COMMAND REFERENCE 04
: 2828 5      BUF_DESCRPTR = H_SADD;      ! DESCRIPTOR ADDRESS
: 2829 5      BYTE_COUNT = 06;            ! TOTAL BYTES TO BE TRANSFER
: 2830 5      ! INITIALIZE MEMORY BUFFER WITH A PATTERN BEFORE
: 2831 5      ! ASKING DM CODE TO WRITE TO THE BUFFER
: 2832 5
: 2833 5      incru COUNT from .H_SADD to .H_EADD by 2 do
: 2834 6          begin
: 2835 6              .TEMP = %o'177777';
: 2836 6              TEMP = .TEMP + 2;
: 2837 6          end;
: 2838 5
: 2839 5      H_EADD = 0;                    ! HIGH BYTE FREE MEMORY ADDRESS
: 2840 5
: 2841 5      if SEND_DATA ()
: 2842 5      then
: 2843 6          begin
: 2844 6              ERRDF (34, SND_DATA_ERR, 0);
: 2845 6
: 2846 6              if .RET_STATUS then DECODE ();
: 2847 6
: 2848 6              CKLOOP;
: 2849 6              RETRIES = TRUE;
: 2850 6              leave BLOCK1;         ! VER.B0
: 2851 6          end;
: 2852 5
: 2853 5      CMD_REF = 5;
: 2854 5      BUF_DESCRPTR = TIP;           ! CLEAN THE BUFFER
: 2855 5      BYTE_COUNT = 02;            ! SET BYTE COUNTS = 2
: 2856 5
: 2857 5      if REC_DATA ()
: 2858 5      then
: 2859 6          begin
: 2860 6              ERRDF (35, RE_DATA_ERR, 0);
: 2861 6
: 2862 6              if .RET_STATUS then DECODE ();
: 2863 6
: 2864 6              CKLOOP;
: 2865 6              RETRIES = TRUE;
: 2866 6              leave BLOCK1;         ! VER.B0
: 2867 6          end;
: 2868 5
: 2869 5      ! EXAMINE THE FREE HOST MEMORY
: 2870 5      !
: 2871 5      TIP = 2;                       ! ADDRESS CONTAIN OWN ADDRESS
: 2872 5
: 2873 5      if EXAM_DATA ()
: 2874 5      then
: 2875 6          begin
: 2876 6              ERRDF (36, BUFF_ERR, RC25$ERR_RPT);
: 2877 6              CKLOOP;
: 2878 6              RETRIES = TRUE;
: 2879 6          end;

```

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0251
Page 56
(12)

```

: 2880 5
: 2881 4      end;
: 2882 4
: 2883 4      if (.RETRIES) then DO_RETRIES ();
: 2884 4
: 2885 4      if (.NUM_RETRIES eqv ZERO) then exitloop;
: 2886 4
: 2887 3      end;
: 2888 3
: 2889 3      return;
: 2890 1      ENDTST;

```

```

000000 010146          .SBTTL  $T11 TEST SECTION
000002 032767 000001 000000G $T11: MOV R1,-(SP) ; 2760
000010 001407          BIT #1,SWP.TRACE ; 2785
000012 012746 000000G    BEQ 1$
000016 012746 000001    MOV #DBM17,-(SP)
000022 010600          MOV #1,-(SP)
000024 104417          MOV SP,R0 ; SP,*
000026 022626          TRAP 17
000030 005067 000000G    CMP (SP),-(SP)
000034 026767 000000G 000000G 1$: CLR NUM.RETRIES ; 2787
000042 101402          2$: CMP NUM.RETRIES,SWP.RETRIES ; 2789
000044 000167 000556    BLOS 3$
000050 012767 000013 000000G 3$: JMP 18$
000056 004767 000000G    MOV #13,TIP ; 2791
000062 006000          JSR PC,AZTEC.READY ; 2793
000064 103024          ROR R0
000066 104455          BCC 6$
000070 000040          TRAP 55 ; 2796
000072 000000G        .WORD 40
000074 000000          .WORD AZT.READY.ERR
000076 032767 000001 000000G .WORD 0
000104 001402          BIT #1,RET.STATUS ; 2798
000106 004767 000000G    BEQ 4$
000112 104465          4$: JSR PC,DECODE
000114 006000          TRAP 65
000116 103002          ROR R0
000120 000167 000502    BHS 5$
000124 012767 000001 000000G 5$: JMP 18$
000132 000167 000442    MOV #1,RETRIES ; 2801
000136 012767 000003 000000G 6$: JMP 16$ ; 2793
000144 012767 000000G 000000G MOV #3,CMD.REF ; 2807
000152 012767 000000G 000000G MOV #DM.11,BUF.DESCRPTR ; 2808
000160 004767 000310 000000G MOV #310,BYTE.COUNT ; 2809
000164 006000          JSR PC,EX.SUP.PRG ; 2811
000166 103023          ROR R0
000170 104455          BCC 9$
000172 000041          TRAP 55 ; 2814
000174 000000G        .WORD 41
000176 000000          .WORD EXE.SUP.ERR
000200 032767 000001 000000G .WORD 0
000206 001402          BIT #1,RET.STATUS ; 2816
000210 004767 000000G    BEQ 7$
000214 104465          7$: JSR PC,DECODE

```

000216	006000			ROR	RO		
000220	103002			BHIS	8\$		
000222	000167	000400		JMP	18\$		
000226	012767	000001	000000G	8\$: MOV	#1,RETURNS	:	2819
000234	000561			BR	16\$:	2813
000236	016767	000000G	000000G	9\$: MOV	FREE.MEM.ADDR,H.SADD	:	2823
000244	016767	000000G	000000G	MOV	H.SADD,TEMP	:	2824
000252	016767	000000G	000000G	MOV	MEM.SIZ,BUF.LENGTH	:	2825
000260	016700	000000G		MOV	BUF.LENGTH,RO	:	2826
000264	006300			ASL	RO		
000266	066700	000000G		ADD	H.SADD,RO		
000272	010067	000000G		MOV	RO,H.EADD		
000276	162767	000002	000000G	SUB	#2,H.EADD		
000304	012767	000004	000000G	MOV	#4,CMD.REF	:	2827
000312	012767	000000G	000000G	MOV	#H.SADD,BUF.DESCRPTR	:	2828
000320	012767	000006	000000G	MOV	#6,BYTE.COUNT	:	2829
000326	016701	000000G		MOV	H.EADD,R1	:	2833
000332	016700	000000G		MOV	H.SADD,RO	: *.COUNT	
000336	000410			BR	11\$		
000340	012777	177777	000000G	10\$: MOV	#-1,@TEMP	:	2835
000346	062767	000002	000000G	ADD	#2,TEMP	:	2836
000354	062700	000002		ADD	#2,RO	: *.COUNT	2833
000360	020001			11\$: CMP	RO,R1	: COUNT, *	
000362	101766			BLOS	10\$		
000364	005067	000000G		CLR	H.EADD	:	2839
000370	004767	000000G		JSR	PC,SEND.DATA	:	2841
000374	006000			ROR	RO		
000376	103021			BCC	13\$		
000400	104455			TRAP	55	:	2844
000402	000042			.WORD	42		
000404	000000G			.WORD	SND.DATA.ERR		
000406	000000			.WORD	0		
000410	032767	000001	000000G	BIT	#1,RET.STATUS	:	2846
000416	001402			BEQ	12\$		
000420	004767	000000G		JSR	PC,DECODE		
000424	104465			12\$: TRAP	65		
000426	006000			ROR	RO		
000430	103476			BLO	18\$		
000432	012767	000001	000000G	MOV	#1,RETURNS	:	2849
000440	000457			BR	16\$:	2843
000442	012767	000005	000000G	13\$: MOV	#5,CMD.REF	:	2853
000450	012767	000000G	000000G	MOV	#TIP,BUF.DESCRPTR	:	2854
000456	012767	000002	000000G	MOV	#2,BYTE.COUNT	:	2855
000464	004767	000000G		JSR	PC,REC.DATA	:	2857
000470	006000			ROR	RO		
000472	103021			BCC	15\$		
000474	104455			TRAP	55	:	2860
000476	000043			.WORD	43		
000500	000000G			.WORD	RE.DATA.ERR		
000502	000000			.WORD	0		
000504	032767	000001	000000G	BIT	#1,RET.STATUS	:	2862
000512	001402			BEQ	14\$		
000514	004767	000000G		JSR	PC,DECODE		
000520	104465			14\$: TRAP	65		
000522	006000			ROR	RO		
000524	103440			BLO	18\$		
000526	012767	000001	000000G	MOV	#1,RETURNS	:	2865

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

```

000534 000421          BR      16$          ;          2859
000536 012767 000002 000000G      15$:  MOV      #2,TIP          ;          2871
000544 004767 000000G          JSR      PC,EXAM.DATA      ;          2873
000550 006000          ROR      R0
000552 103012          BCC     16$
000554 104455          TRAP    55          ;          2876
000556 000044          .WORD  44
000560 000000G          .WORD  BUFF.ERR
000562 000000G          .WORD  RC25$ERR.RPT
000564 104465          TRAP    65
000566 006000          ROR      R0
000570 103416          BLO     18$
000572 012767 000001 000000G          MOV      #1,RETRIES      ;          2878
000600 032767 000001 000000G      16$:  BIT      #1,RETRIES      ;          2883
000606 001402          BEQ     17$
000610 004767 000000G          JSR      PC,DO.RETRIES
000614 005767 000000G      17$:  TST     NUM.RETRIES      ;          2885
000620 001402          BEQ     18$
000622 000167 177206          JMP     2$
000626 012601      18$:  MOV      (SP)+,R1      ;          2760
000630 000207          RTS     PC

```

; Routine Size: 205 words, Routine Base: AC\$CODE + 6742
; Maximum stack depth per invocation: 5 words

```

000000 004767 177142          .SBTTL  T11 TEST SECTION
000000          T11::
000004 104466      1$:  JSR      PC,$T11          ;          2889
000006 006000          TRAP    66
000010 103773          ROR      R0
000012 000207          BLO     1$
          RTS     PC

```

; Routine Size: 6 words, Routine Base: AC\$CODE + 7574
; Maximum stack depth per invocation: 2 words

; 2891 1 !<BLF/PAGE>

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0254
Page 59
(13)

```

: 2892 1
: 2893 3
: 2894 3
: 2895 3
: 2896 3
: 2897 3
: 2898 3
: 2899 3
: 2900 3
: 2901 3
: 2902 3
: 2903 3
: 2904 3
: 2905 3
: 2906 3
: 2907 3
: 2908 3
: 2909 3
: 2910 3
: 2911 3
: 2912 3
: 2913 3
: 2914 3
: 2915 3
: 2916 3
: 2917 3
: 2918 3
: 2919 3
: 2920 3
: 2921 3
: 2922 3
: 2923 3
: 2924 3
: 2925 3
: 2926 3
: 2927 3
: 2928 3
: 2929 4
: 2930 4
: 2931 4
: 2932 4
: 2933 4
: 2934 5
: 2935 5
: 2936 5
: 2937 5
: 2938 5
: 2939 5
: 2940 5
: 2941 5
: 2942 4
: 2943 4
: 2944 5
: 2945 5
: 2946 5
: 2947 5
: 2948 5
!
! BGNTST;
!
! **
! TEST #12 - BUS ADDRESSING/DATA TEST B
!
! DESCRIPTION:
!
! THIS TEST FIRST BRINGS AZTEC DRIVE READY AND ONLINE AND THEN
! LOADS DM_12 PROGRAM VECTOR TO PORT CONTROLLER MEMORY. THEN
! DOES THE FOLLOWING:
!
!     A. GIVE FREE MEMORY ADDRESS AND BUFFER SIZE TO DM CODE
!        AND ASK DM CODE WRITE A PATTERN OF ONE'S COMPLEMENT
!        OF ADDRESS AT THE ADDRESS AND EXPECTS TO RECEIVE
!        SUCCESS OR FAILURE CODE FROM DM PROGRAM. THEN CHECKS
!        MEMORY BUFFER FOR THE EXPECTED PATTERN AND REPORTS
!        ERROR IF ENCOUNTERED.
!
!     B. IF SUCCESS, ASKS DM CODE TO WRITE TO MEMORY A PATTERN
!        OF ALL ONES AND CHECKS FOR THE PATTERN IN MEMORY.
!
!     C. IF SUCCESS, ASKS DM CODE TO WRITE TO MEMORY A PATTERN
!        OF ALL ZEROES AND CHECKS FOR THE PATTERN IN MEMORY.
!
! IF OPERATOR ASKS FOR RETRIES THE WHOLE TEST WILL BE RETRIED
! ONLY IF FAILURE ENCOUNTERED.
!
! --
label
  BLOCK1;                                ! VER.B0
if .SWP_TRACE then PRINTF (DBM18);       ! TEST 12
NUM_RETRIES = ZERO;
while (.NUM_RETRIES lequ .SWP_RETRIES) do
  begin
    TIP = 12;
    if AZTEC_READY ()                     ! GET AZTEC READY FOR OPERATION
    then
      begin
        ERRDF (37, AZT_READY_ERR, 0);      !
        if .RET_STATUS then DECODE ();
      end
    else
      BLOCK1 :
      begin
! SEND DOWN LINE LOAD THE DM CODE AND EXECUTE THE DM PROGRAM WHICH IT WILL
! WRITE THE FREE HOST MEMORY WITH COMPLEMENT THE TESTING ADDRESS

```

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0255
Page 60
(13)

```

: 2949 5      CMD_REF = 3;          ! COMMAND REFERENCE #
: 2950 5      BUF_DESCRPTR = DM_12; ! DMCODE STARTING ADDRESS
: 2951 5      BYTE_COUNT = 202*2;  ! BYTE COUNTS
: 2952 5
: 2953 5      if EX_SUP_PRG ( )      ! ISSUE AN EXECUTE SUPPLIED -
: 2954 5      then                  ! IF STATUS BIT INDICATES ERROR
: 2955 6          begin              ! THEN
: 2956 6          ERRDF (38, EXE_SUP_ERR, 0); !
: 2957 6
: 2958 6          if .RET_STATUS then DECODE ( );
: 2959 6
: 2960 6          CKLOOP;
: 2961 6          RETRIES = TRUE;
: 2962 6          leave BLOCK1;      ! VER.B0
: 2963 6          end;
: 2964 5
: 2965 5      incru COUNT from 0 to 2 do
: 2966 6          begin
: 2967 6          H_SADD = .FREE_MEM_ADDR; ! LO-BYTE FREE HOST MEMORY ADDRESS
: 2968 6          TEMP = .H_SADD;
: 2969 6          BUF_LENGTH = .MEM_SIZ; ! TOTAL FREE HOST MEMORY SIZE
: 2970 6          H_EADD = .FREE_MEM_ADDR - 2 * .BUF_LENGTH*2; ! END ADDRESS OF BUFFER
: 2971 6
: 2972 6      ! SENT FREE HOST MEMORY ADDRESS AND IT LENGTH TO DM PROGRAM
: 2973 6
: 2974 6          CMD_REF = 4;          ! COMMAND REFERENCE 04
: 2975 6          BUF_DESCRPTR = H_SADD; ! DESCRIPTOR ADDRESS
: 2976 6          BYTE_COUNT = 06;      ! TOTAL BYTES TO BE TRANSFER
: 2977 6      ! INITIALIZE MEMORY BUFFER WITH A PATTERN BEFORE
: 2978 6      ! ASKING DM CODE TO WRITE TO THE BUFFER
: 2979 6
: 2980 6          incru LOOP from .H_SADD to .H_EADD by 2 do
: 2981 7              begin
: 2982 7                  .TEMP = %o'125252';
: 2983 7                  TEMP = .TEMP + 2;
: 2984 7              end;
: 2985 6
: 2986 6          H_EADD = 0;          ! HIGH BYTE FREE MEM ADDRESS
: 2987 6
: 2988 6          if SEND_DATA ( )      ! ISSUE SEND DATA COMMAND
: 2989 6          then                  ! STATUS BIT INDICATES ERROR
: 2990 7              begin              ! THEN
: 2991 7              ERRDF (39, SND_DATA_ERR, 0); !
: 2992 7
: 2993 7              if .RET_STATUS then DECODE ( );
: 2994 7
: 2995 7              CKLOOP;
: 2996 7              RETRIES = TRUE;
: 2997 7              leave BLOCK1;      ! VER.B0
: 2998 7              end;
: 2999 6
: 3000 6      !
: 3001 6      ! WAIT FOR "DONE" SIGNAL FROM DM
: 3002 6      !
: 3003 6          CMD_REF = 5;          ! COMMAND REFERENCE #
: 3004 6          BUF_DESCRPTR = TIP;   ! CLEAN THE BUFFER
: 3005 6          BYTE_COUNT = 02;     ! SET BYTE COUNTS = 2

```

```

: 3006 6
: 3007 6      if REC_DATA ()          ! SENT A RECEIVE DATA COMMAND
: 3008 6      then                    ! STATUS BIT INDICATES ERROR
: 3009 7          begin                ! THEN
: 3010 7          ERRDF (40, RE_DATA_ERR, 0); ! REPORT ERROR
: 3011 7
: 3012 7          if .RET_STATUS then DECODE ();
: 3013 7
: 3014 7          CKLOOP;
: 3015 7          RETRIES = TRUE;
: 3016 7          leave BLOCK1;        ! VER.B0
: 3017 6          end;
: 3018 6
: 3019 6      if .TIP nequ #0'104'    ! IF DM RETURNS FAILURE CODE
: 3020 6      then                    ! THEN ABORT DM PROGRAM
: 3021 7          begin
: 3022 7          ERRDF (41, DMC_ERR, 0);
: 3023 7          RETRIES = TRUE;
: 3024 7          CKLOOP;
: 3025 7          exitloop;
: 3026 6          end;
: 3027 6
: 3028 6      !
: 3029 6      ! EXAMINE THE FREE HOST MEMORY
: 3030 6      !
: 3031 6
: 3032 6          if .COUNT eqlu 0 then TIP = 1; ! ADDRESS CONTAINS COMPLEMENT
: 3033 6                                          ! OF ADDRESS
: 3034 6
: 3035 6          if .COUNT eqlu 1 then TIP = ALL_ONES; ! MEMORY PATTERN SECOND TIME
: 3036 6
: 3037 6          if .COUNT eqlu 2 then TIP = ZERO; ! MEMORY PATTERN THIRD TIME
: 3038 6
: 3039 6      if EXAM_DATA ()
: 3040 6      then
: 3041 7          begin
: 3042 7          ERRDF (42, BUFF_ERR, RC25$ERR_RPT);
: 3043 7          CKLOOP;
: 3044 7          RETRIES = TRUE;
: 3045 7          leave BLOCK1;        ! VER.B0
: 3046 7          end;
: 3047 6
: 3048 6
: 3049 6      !
: 3050 6      ! SIGNAL DM TO CONTINUE TO EXECUTE THE PROGRAM
: 3051 6      !
: 3052 5          end;                ! ASK DM CODE TO CONT.
: 3053 5
: 3054 4          end;
: 3055 4
: 3056 4          if (.RETRIES) then DO_RETRIES ();
: 3057 4
: 3058 4          if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 3059 4
: 3060 3          end;
: 3061 3
: 3062 3      return;

```

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0257
Page 62
(13)

: 3063 1 ENDTST;

000000	004167	000000G		\$T12:	.SBTTL \$T12 TEST SECTION			
000004	032767	000001	000000G		JSR R1,\$SAVE2	:		2890
000012	001407				BIT #1,SWP.TRACE	:		2924
000014	012746	000000G			BEQ 1\$			
000020	012746	000001			MOV #DBM18,-(SP)			
000024	010600				MOV #1,-(SP)			
000026	104417				MOV SP,R0	:	SP,*	
000030	022626				TRAP 17			
000032	005067	000000G		1\$:	CMP (SP), (SP).			
000036	026767	000000G	000000G	2\$:	CLR NUM.RETRIES	:		2926
000044	101401				CMP NUM.RETRIES,SWP.RETRIES	:		2928
000046	000207				BLOS 3\$			
000050	012767	000014	000000G	3\$:	RTS PC			
000056	004767	000000G			MOV #14,TIP	:		2930
000062	006000				JSR PC,AZTEC.READY	:		2932
000064	103023				ROR R0			
000066	104455				BCC 6\$			
000070	000045				TRAP 55	:		2935
000072	000000G				.WORD 45			
000074	000000				.WORD AZT.READY.ERR			
000076	032767	000001	000000G		.WORD 0			
000104	001402				BIT #1,RET.STATUS	:		2937
000106	004767	000000G			BEQ 4\$			
000112	104465			4\$:	JSR PC,DECODE			
000114	006000				TRAP 65			
000116	103001				ROR R0			
000120	000207				BHIS 5\$			
000122	012767	000001	000000G	5\$:	RTS PC			
000130	000167	000552			MOV #1,RETRIES	:		2940
000134	012767	000003	000000G	6\$:	JMP 22\$:		2932
000142	012767	000000G	000000G		MOV #3,CMD.REF	:		2949
000150	012767	000624	000000G		MOV #DM.12,BUF.DESCRPTR	:		2950
000156	004767	000000G			MOV #624,BYTE.COUNT	:		2951
000162	006000				JSR PC,EX.SUP.PRG	:		2953
000164	103023				ROR R0			
000166	104455				BCC 9\$			
000170	000046				TRAP 55	:		2956
000172	000000G				.WORD 46			
000174	000000				.WORD EXE.SUP.ERR			
000176	032767	000001	000000G		.WORD 0			
000204	001402				BIT #1,RET.STATUS	:		2958
000206	004767	000000G			BEQ 7\$			
000212	104465			7\$:	JSR PC,DECODE			
000214	006000				TRAP 65			
000216	103001				ROR R0			
000220	000207				BHIS 8\$			
000222	012767	000001	000000G	8\$:	RTS PC			
000230	000167	000452			MOV #1,RETRIES	:		2961
000234	005002			9\$:	JMP 22\$:		2955
000236	016767	000000G	000000G	10\$:	CLR R2	:	COUNT	2965
000244	016767	000000G	000000G		MOV FREE.MEM.ADDR,H.SADD	:		2967
000252	016767	000000G	000000G		MOV H.SADD,TEMP	:		2968
000260	016700	000000G			MOV MEM.SIZ,BUF.LENGTH	:		2969
					MOV BUF.LENGTH,R0	:		2970

000264	006300			ASL	RO			
000266	066700	000000G		ADD	FREE.MEM.ADDR,RO			
000272	010067	000000G		MOV	RO,H.EADD			
000276	162767	000002	000000G	SUB	#2,H.EADD			
000304	012767	000004	000000G	MOV	#4,CMD.REF	:	2974	
000312	012767	000000G	000000G	MOV	#H,JADD,BUF.DESCRPTR	:	2975	
000320	012767	000006	000000G	MOV	#6,BYTE.COUNT	:	2976	
000326	016701	000000G		MOV	H.EADD,R1	:	2980	
000332	016700	000000G		MOV	H.SADD,RO	: *.LOOP		
000336	000410			BR	12#			
000340	012777	125252	000000G	11#:	MOV	#-52526,@TEMP	:	2982
000346	062767	000002	000000G	ADD	#2,TEMP	:	2983	
000354	062700	000002		ADD	#2,RO	: *.LOOP	2980	
000360	020001			12#:	CMP	RO,R1	: LOOP,*	
000362	101766			BLOS	11#			
000364	005067	000000G		CLR	H.EADD	:	2986	
000370	004767	000000G		JSR	PC.SEND.DATA	:	2988	
000374	006000			ROR	RO			
000376	103021			BCC	14#			
000400	104455			TRAP	55	:	2991	
000402	000047			.WORD	47			
000404	000000G			.WORD	SND.DATA.ERR			
000406	000000			.WORD	0			
000410	032767	000001	000000G	BIT	#1,RET.STATUS	:	2993	
000416	001402			BEQ	13#			
000420	004767	000000G		JSR	PC.DECODE			
000424	104465			13#:	TRAP	65		
000426	006000			ROR	RO			
000430	103541			BLO	24#			
000432	012767	000001	000000G	MOV	#1,RETRIES	:	2996	
000440	000522			BR	22#	:	2990	
000442	012767	000005	000000G	14#:	MOV	#5,CMD.REF	:	3003
000450	012767	000000G	000000G	MOV	#TIP,BUF.DESCRPTR	:	3004	
000456	012767	000002	000000G	MOV	#2,BYTE.COUNT	:	3005	
000464	004767	000000G		JSR	PC.REC.DATA	:	3007	
000470	006000			ROR	RO			
000472	103021			BCC	16#			
000474	104455			TRAP	55	:	3010	
000476	000050			.WORD	50			
000500	000000G			.WORD	RE.DATA.ERR			
000502	000000			.WORD	0			
000504	032767	000001	000000G	BIT	#1,RET.STATUS	:	3012	
000512	001402			BEQ	15#			
000514	004767	000000G		JSR	PC.DECODE			
000520	104465			15#:	TRAP	65		
000522	006000			ROR	RO			
000524	103503			BLO	24#			
000526	012767	000001	000000G	MOV	#1,RETRIES	:	3015	
000534	00464			BR	22#	:	3009	
000536	026727	000000G	000104	16#:	CMP	TIP,#104	:	3019
000544	001413			BEQ	17#	:		
000546	104455			TRAP	55	:	3022	
000550	000051			.WORD	51			
000552	000000G			.WORD	DMC.ERR			
000554	000000			.WORD	0			
000556	012767	000001	000000G	MOV	#1,RETRIES	:	3023	
000564	104465			TRAP	65			

N4

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B1100-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0259
Page 64
(13)

000566	006000			ROR	R0		
000570	103046			BCC	22		
000572	000207			RTS	PC		
000574	005702		17:	TST	R2	: COUNT	3032
000576	001003			BNE	18		
000600	012767	000001	000000G	MOV	#1,TIP		
000606	020227	000001		18:	CMP	R2,#1	: COUNT,*
000612	001003			BNE	19		3036
000614	012767	177777	000000G	MOV	#-1,TIP		
000622	020227	000002		19:	CMP	R2,#2	: COUNT,*
000626	001002			BNE	20		3038
000630	005067	000000G		CLR	TIP		
000634	004767	000000G		20:	JSR	PC,EXAM.DATA	:
000640	006000			ROR	R0		3040
000642	103013			BCC	21		
000644	104455			TRAP	55		3043
000646	000052			.WORD	52		
000650	000000G			.WORD	BUFF.ERR		
000652	000000G			.WORD	RC25ERR.RPT		
000654	104465			TRAP	65		
000656	006000			ROR	R0		
000660	1.3425			BLO	24		
000662	012767	000001	000000G	MOV	#1,RETRIES		3045
000670	000406			BR	22		3042
000672	005202		21:	INC	R2	: COUNT	2965
000674	020227	000002		CMP	R2,#2	: COUNT,*	
000700	101002			BHI	22		
000702	000167	177330		JMP	10		
000706	032767	000001	000000G	22:	BIT	#1,RETRIES	:
000714	001402			BEQ	23		3056
000716	004767	000000G		JSR	PC,DO.RETRIES		
000722	005767	000000G		23:	TST	NUM.RETRIES	:
000726	001402			BEQ	24		3058
000730	000167	177102		JMP	2		
000734	000207		24:	RTS	PC	:	2890

: Routine Size: 239 words, Routine Base: AC#CODE * 7610
: Maximum stack depth per invocation: 7 words

000000	004767	177036		.SBTTL	T12 TEST SECTION		
000000			T12::				
000004	104466		1:	JSR	PC,#T12		3062
000006	006000			TRAP	66		
000010	103773			ROR	R0		
000012	000207			BLO	1		
				RTS	PC		

: Routine Size: 6 words, Routine Base: AC#CODE * 10546
: Maximum stack depth per invocation: 2 words

: 3064 1 !<BLF/PAGE>

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0261
Page 66
(14)

```

: 3122 5      begin
: 3123 5      ERRDF (43, AZT_READY_ERR, 0);      !
: 3124 5
: 3125 5      if .RET_STATUS then DECODE ();      ! DECODE THE STATUS, IF ANY
: 3126 5
: 3127 5      RETRIES = TRUE;      ! SET RETRIES FLAG
: 3128 5      end
: 3129 4      else
: 3130 4      ! LOAD DM CODE INTO CONTROLLER'S MEMORY
: 3131 4      BLOCK1 :
: 3132 5      begin
: 3133 5      CMD_REF = .CMD_SLOT;      ! COMMAND REFERENCE #
: 3134 5      BUF_DESCPTR = DM_13;      ! DM CODE STARTING ADDRESS
: 3135 5      BYTE_COUNT = 105*2;      ! BYTE COUNTS
: 3136 5
: 3137 5      if EX_SUP_PRG ()
: 3138 5      then      ! ISSUE AN EXECUTE SUPPLIED PROGRAM
: 3139 6      begin      ! IF STATUS BIT INDICATES ERROR
: 3140 6      ERRDF (44, EXE_SUP_ERR, 0);      ! THEN REPORT ERROR
: 3141 6
: 3142 6      if .RET_STATUS then DECODE ();      ! DECODE STATUS, IF ANY
: 3143 6
: 3144 6      RETRIES = TRUE;
: 3145 6      leave BLOCK1;
: 3146 5      end;
: 3147 5
: 3148 5      incru COUNT from 0 to 3 do
: 3149 6      begin
: 3150 6      !
: 3151 6      ! SELECT ONE OF THE PATTERNS AND INITIALIZE HOST TRANSMIT BUFFER
: 3152 6      ! WITH THE SELECTED PATTERN.
: 3153 6      !
: 3154 6
: 3155 6      selectoneu .COUNT of
: 3156 6      set
: 3157 6
: 3158 6      [0] :
: 3159 6      PATTERN_ADDR = DATA_PAT1;
: 3160 6
: 3161 6      [1] :
: 3162 6      PATTERN_ADDR = DATA_PAT2;
: 3163 6
: 3164 6      [2] :
: 3165 6      PATTERN_ADDR = DATA_PAT3;
: 3166 6
: 3167 6      [3] :
: 3168 6      PATTERN_ADDR = DATA_PAT4;
: 3169 6      tes;
: 3170 6
: 3171 6      incru J from 0 to 254 do      ! INITIALIZE TRANSMIT
: 3172 7      begin      ! BUFFER OF 256 WORDS
: 3173 7
: 3174 7      incru K from 0 to 2 do      ! WITH THE PATTERN
: 3175 8      begin
: 3176 8      XMT_DATA_BUF [.J] = .PATTERN_ADDR [.K];
: 3177 8      J = .J + 1;      ! INCREMENT J WITHIN INNER LOOP
: 3178 7      end;

```



```

: 3179 7
: 3180 7      J = .J - 1;      ! DECREMENT J TO ADJUST POINTER
: 3181 6      end;
: 3182 6
: 3183 6      ! GIVE START ADDRESS OF TRANSMIT AND RECEIVE BUFFER AND SIZE TO DM CODE
: 3184 6      !
: 3185 6      SEND_PKT [WORD0] = XMT_DATA_BUF [0];      ! LOW WORD OF TRANSMIT ADDRESS
: 3186 6      SEND_PKT [WORD1] = 0;      ! HIGH WORD OF TRANSMIT ADDRESS
: 3187 6      SEND_PKT [WORD2] = RCV_DATA_BUF [0];      ! LOW WORD OF RECEIVE ADDRESS
: 3188 6      SEND_PKT [WORD3] = 0;      ! HIGH WORD OF RECEIVE ADDRESS
: 3189 6      CMD_REF = .CMD_SLOT;      ! COMMAND REFERENCE 04
: 3190 6      BUF_DESCRPTR = SEND_PKT;      ! DESCRIPTOR ADDRESS
: 3191 6      BYTE_COUNT = 08;      ! TOTAL BYTES TO BE TRANSFERRED
: 3192 6
: 3193 6      if SEND_DATA ()      ! ISSUE SEND DATA COMMAND
: 3194 6      then      ! IF STATUS BIT INDICATES ERROR
: 3195 7          begin      ! THEN REPORT ERROR
: 3196 7              ERRDF (45, SND_DATA_ERR, 0);
: 3197 7
: 3198 7              if .RET_STATUS then DECODE ();      ! DECODE RETURN STATUS
: 3199 7
: 3200 7              RETRIES = TRUE;
: 3201 7              exitloop;
: 3202 6              end;
: 3203 6
: 3204 6      ! ISSUE A REC_DATA COMMAND AND GET THE RESULT OF DM CODE STATUS
: 3205 6      ! IN 'TIP'
: 3206 6      CMD_REF = .CMD_SLOT;      ! COMMAND REFERENCE #
: 3207 6      BUF_DESCRPTR = TIP;      ! CLEAN THE BUFFER
: 3208 6      BYTE_COUNT = 02;      ! SET BYTE COUNTS = 2
: 3209 6
: 3210 6      if REC_DATA ()      ! SENT A RECEIVE DATA COMMAND
: 3211 6      then      ! IF STATUS BIT INDICATES ERROR
: 3212 7          begin      ! THEN
: 3213 7              ERRDF (46, RE_DATA_ERR, 0);      ! REPORT ERROR
: 3214 7
: 3215 7              if .RET_STATUS then DECODE ();
: 3216 7
: 3217 7              RETRIES = TRUE;
: 3218 7              exitloop;
: 3219 6              end;
: 3220 6
: 3221 6      if .TIP nequ #0'104'      ! IF DM RETURNS FAILURE CODE
: 3222 6      then      ! THEN ABORT DM PROGRAM
: 3223 7          begin
: 3224 7              ERRDF (47, DMC_ERR, 0);
: 3225 7              RETRIES = TRUE;
: 3226 7              exitloop;
: 3227 6              end;
: 3228 6
: 3229 6      ! COMPARE TRANSMIT AND RECEIVE BUFFERS FOR THE PATTERN
: 3230 6
: 3231 6      incru J from 0 to 255 do
: 3232 7          begin
: 3233 7
: 3234 7              if .XMT_DATA_BUF [.J] nequ .RCV_DATA_BUF [.J]
: 3235 7              then

```

E5

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0263
Page 68
(14)

```

: 3236 8      begin
: 3237 8      P3 = XMT_DATA_BUF [.J];      ! TRANSMIT BUF FAILURE ADDRESS
: 3238 8      P4 = RCV_DATA_BUF [.J];      ! RECEIVE BUF FAILURE ADDRESS
: 3239 8      P5 = .XMT_DATA_BUF [.J];      ! TRANSMIT DATA
: 3240 8      P6 = .RCV_DATA_BUF [.J];      ! RECEIVE DATA
: 3241 8      ERRDF (48, BUFF_ERR, 0);      ! PRINT ERROR INFO.
: 3242 8      PRINTB (FMT7, .P3, .P4);
: 3243 8      PRINTB (FMT7A, .P5, .P6);
: 3244 8      RETRIES = TRUE;              ! TURN ON RETRIES
: 3245 8      exitloop;
: 3246 7      end;
: 3247 7
: 3248 6      end;
: 3249 6
: 3250 5      end;
: 3251 5
: 3252 4      end;
: 3253 4
: 3254 4      if (.RETRIES) then DO_RETRIES ();
: 3255 4
: 3256 4      if (.NUM_RETRIES eqv ZERO) then exitloop;
: 3257 4
: 3258 3      end;
: 3259 3
: 3260 3      return;
: 3261 1      ENDTST;

```

000000	004167	000000G		.SBTTL	\$T13 TEST SECTION		
000004	032767	000001	000000G	\$T13:	JSR	R1,\$SAVE4	3063
000012	001407				BIT	#1,SWP.TRACE	3111
000014	012746	000000G			BEQ	1\$	
000020	012746	000001			MOV	#DBM19,-(SP)	
000024	010600				MOV	#1,-(SP)	
000026	104417				MOV	SP,R0	; SP,*
000030	022626				TRAP	17	
000032	005067	000000G			CMP	(SP),-(SP)	
000036	026767	000000G	000000G	1\$:	CLR	NUM.RETRIES	3113
000044	101401			2\$:	CMP	NUM.RETRIES,SWP.RETRIES	3115
000046	000207				BLOS	3\$	
000050	012767	000015	000000G	3\$:	RTS	PC	
000056	004767	000000G			MOV	#15,TIP	3117
000062	006000				JSR	PC,AZTEC.READY	3120
000064	103017				ROR	R0	
000066	104455				BCC	5\$	
000070	000053				TRAP	55	3123
000072	000000G				.WORD	53	
000074	000000				.WORD	AZT.READY.ERR	
000076	032767	000001	000000G		.WORD	0	
000104	001402				BIT	#1,RET.STATUS	3125
000106	004767	000000G			BEQ	4\$	
000112	012767	000001	000000G	4\$:	JSR	PC,DECODE	
000120	000167	000674			MOV	#1,RETRIES	3127
000124	016767	000000G	000000G	5\$:	JMP	23\$	3120
000132	012767	000000G	000000G		MOV	CMD.SLOT,CMD.REF	3133
000140	012767	000322	000000G		MOV	#DM.13,BUF.DESCRPTR	3134
					MOV	#322,BYTE.COUNT	3135

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0264
Page 69
(14)

000146	004767	000000G			JSR	PC,EX.SUP.PRG	:		3137
000152	006000				ROR	R0	:		
000154	103017				BCC	7\$:		
000156	104455				TRAP	55	:		3140
000160	000054				.WORD	54	:		
000162	000000G				.WORD	EXE.SUP.ERR	:		
000164	000000				.WORD	0	:		
000166	032767	000001	000000G		BIT	#1,RET.STATUS	:		3142
000174	001402				BEQ	6\$:		
000176	004767	000000G			JSR	PC,DECODE	:		
000202	012767	000001	000000G	6\$:	MOV	#1,RETRIES	:		3144
000210	000167	000604			JMP	23\$:		3139
000214	005004			7\$:	CLR	R4	:	COUNT	3148
000216	010400			8\$:	MOV	R4,R0	:	COUNT,*	3155
000220	001004				BNE	9\$:		3158
000222	012767	000200'	000240'		MOV	#DATA.PAT1,PATTERN.ADDR	:		3159
000230	000424				BR	12\$:		3155
000232	020027	000001		9\$:	CMP	R0,#1	:		3161
000236	001004				BNE	10\$:		
000240	012767	000206'	000240'		MOV	#DATA.PAT2,PATTERN.ADDR	:		3162
000246	000415				BR	12\$:		3155
000250	020027	000002		10\$:	CMP	R0,#2	:		3164
000254	001004				BNE	11\$:		
000256	012767	000214'	000240'		MOV	#DATA.PAT3,PATTERN.ADDR	:		3165
000264	000406				BR	12\$:		3155
000266	020027	000003		11\$:	CMP	R0,#3	:		3167
000272	001003				BNE	12\$:		
000274	012767	000222'	000240'		MOV	#DATA.PAT4,PATTERN.ADDR	:		3168
000302	005001			12\$:	CLR	R1	:	J	3171
000304	005003			13\$:	CLR	R3	:	K	3174
000306	010100			14\$:	MOV	R1,R0	:	J,*	3176
000310	006300				ASL	R0	:		
000312	010302				MOV	R3,R2	:	K,*	
000314	006302				ASL	R2	:		
000316	066702	000240'			ADD	PATTERN.ADDR,R2	:		
000322	011260	000000G			MOV	(R2),XMT.DATA.BUF(R0)	:		
000326	005201				INC	R1	:	J	3177
000330	005203				INC	R3	:	K	3174
000332	020327	000002			CMP	R3,#2	:	K,*	
000336	101763				BLOS	14\$:		
000340	020127	000376			CMP	R1,#376	:	J,*	3171
000344	101757				BLOS	13\$:		
000346	012767	000000G	000230'		MOV	#XMT.DATA.BUF,SEND.PKT	:		3185
000354	005067	000232'			CLR	SEND.PKT+2	:		3186
000360	012767	000000G	000234'		MOV	#RCV.DATA.BUF,SEND.PKT+4	:		3187
000366	005067	000236'			CLR	SEND.PKT+6	:		3188
000372	016767	000000G	000000G		MOV	CMD.SLOT,CMD.REF	:		3189
000400	012767	000230'	000000G		MOV	#SEND.PKT,BUF.DESCRPTR	:		3190
000406	012767	000010	000000G		MOV	#10,BYTE.COUNT	:		3191
000414	004767	000000G			JSR	PC,SEND.DATA	:		3193
000420	006000				ROR	R0	:		
000422	103016				BCC	16\$:		
000424	104455				TRAP	55	:		3196
000426	000055				.WORD	55	:		
000430	000000G				.WORD	SND.DATA.ERR	:		
000432	000000				.WORD	0	:		
000434	032767	000001	000000G		BIT	#1,RET.STATUS	:		3198

ZRCFR3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0265
Page 70
(14)

000442	001402			BEQ	15#				
000444	004767	000000G		JSR	PC,DECODE				
000450	012767	000001	000000G	15#:	MOV	#1,RETRIES	:		3200
000456	000560			BR	23#		:		3195
000460	016767	000000G	000000G	16#:	MOV	CMD.SLOT,CMD.REF	:		3206
000466	012767	000000G	000000G		MOV	#TIP,BUF.DESCRPTR	:		3207
000474	012767	000002	000000G		MOV	#2,BYTE.COUNT	:		3208
000502	004767	000000G			JSR	PC.REC.DATA	:		3210
000506	006000				ROR	R0			
000510	103016				BCC	18#			
000512	104455				TRAP	55	:		3213
000514	000056				.WORD	56			
000516	000000G				.WORD	RE.DATA.ERR			
000520	000000				.WORD	0			
000522	032767	000001	000000G		BIT	#1,RET.STATUS	:		3215
000530	001402				BEQ	17#			
000532	004767	000000G			JSR	PC,DECODE			
000536	012767	000001	000000G	17#:	MOV	#1,RETRIES	:		3217
000544	000525				BR	23#	:		3212
000546	026727	000000G	000104	18#:	CMP	TIP,#104	:		3221
000554	001410				BEQ	19#			
000556	104455				TRAP	55	:		3224
000560	000057				.WORD	57			
000562	000000G				.WORD	DMC.ERR			
000564	000000				.WORD	0			
000566	012767	000001	000000G		MOV	#1,RETRIES	:		3225
000574	000511				BR	23#	:		3223
000576	005001			19#:	CLR	R1	:	J	3231
000600	010103			20#:	MOV	R1,R3	:	J,*	3234
000602	006303				ASL	R3			
000604	010100				MOV	R1,R0	:	J,*	
000606	006300				ASL	R0			
000610	026360	000000G	000000G		CMP	XMT.DATA.BUF(R3),RCV.DATA.BUF(R0)	:		
000616	001466				BEQ	21#			
000620	010100				MOV	R1,R0	:	J,*	3237
000622	006300				ASL	R0			
000624	062700	000000G			ADD	#XMT.DATA.BUF,R0			
000630	010067	000000G			MOV	R0,P3			
000634	010100				MOV	R1,R0	:	J,*	3238
000636	006300				ASL	R0			
000640	062700	000000G			ADD	#RCV.DATA.BUF,R0			
000644	010067	000000G			MOV	R0,P4			
000650	010100				MOV	R1,R0	:	J,*	3239
000652	006300				ASL	R0			
000654	016067	000000G	000000G		MOV	XMT.DATA.BUF(R0),P5			
000662	010100				MOV	R1,R0	:	J,*	3240
000664	006300				ASL	R0			
000666	016067	000000G	000000G		MOV	RCV.DATA.BUF(R0),P6			
000674	104455				TRAP	55	:		3241
000676	000060				.WORD	60			
000700	000000G				.WORD	BUFF.ERR			
000702	000000				.WORD	0			
000704	016746	000000G			MOV	P4,-(SP)	:		3242
000710	016746	000000G			MOV	P3,-(SP)			
000714	012746	000000G			MOV	#FMT7,-(SP)			
000720	012746	000003			MOV	#3,-(SP)			
000724	010600				MOV	SP,R0	:	SP,*	

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0266
Page 71
(14)

000726	104414			TRAP	14			
000730	062706	000006		ADD	#6,SP			
000734	016716	000000G		MOV	P6,(SP)			3243
000740	016746	000000G		MOV	P5, -(SP)			
000744	012746	000000G		MOV	#FMT7A, -(SP)			
000750	012746	000003		MOV	#3, -(SP)			
000754	010600			MOV	SP,R0		; SP,*	
000756	104414			TRAP	14			
000760	062706	000010		ADD	#10,SP			
000764	012767	000001	000000G	MOV	#1,RETRIES			3244
000772	000404			BR	22#			3236
000774	005201		21#:	INC	R1		; J	3231
000776	020127	000377		CMP	R1,#377		; J,*	
001002	101676			BLOS	20#			
001004	005204		22#:	INC	R4		; COUNT	3148
001006	020427	000003		CMP	R4,#3		; COUNT,*	
001012	101002			BHI	23#			
001014	000167	177176		JMP	8#			
001020	032767	000001	000000G	23#:	BIT	#1,RETRIES		3254
001026	001402			BEQ	24#			
001030	004767	000000G		JSR	PC,DO.RETRIES			
001034	005767	000000G	24#:	TST	NUM.RETRIES			3256
001040	001402			BEQ	25#			
001042	000167	176770		JMP	2#			
001046	000207		25#:	RTS	PC			3063

; Routine Size: 276 words, Routine Base: AC\$CODE + 10562
; Maximum stack depth per invocation: 11 words

				.SBTTL	T13 TEST SECTION			
000000	004767	176724	T13::					
000000			1#:	JSR	PC,\$T13			3260
000004	104466			TRAP	66			
000006	006000			ROR	R0			
000010	103773			BLO	1#			
000012	000207			RTS	PC			

; Routine Size: 6 words, Routine Base: AC\$CODE + 11632
; Maximum stack depth per invocation: 2 words

:	3262	1	:	
:	3263	3	:	BGNTST;
:	3264	3	:	
:	3265	3	:	!..
:	3266	3	:	! TEST #14 - SPIN UP/HEAD LOAD SEQUENCE
:	3267	3	:	! TEST #14 - SPIN UP/HEAD LOAD SEQUENCE
:	3268	3	:	! TEST #14 - SPIN UP/HEAD LOAD SEQUENCE
:	3269	3	:	! DESCRIPTION:
:	3270	3	:	! THIS TEST FIRST INITIALIZES RC25 CONTROLLER, INITIALIZES COM_AREA,
:	3271	3	:	! AND DOES SET CONTROL CHARACTERISTICS.
:	3272	3	:	! ..
:	3273	3	:	! ..
:	3274	3	:	! ..

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0267
Page 72
(14)

```

: 3275 3  !
: 3276 3  !
: 3277 3  !
: 3278 3  !
: 3279 3  !
: 3280 3  !
: 3281 3  !
: 3282 3  !
: 3283 3  !
: 3284 3  !
: 3285 3  !
: 3286 3  !
: 3287 3  !
: 3288 3  !
: 3289 3  !
: 3290 3  !
: 3291 3  !
: 3292 3  !
: 3293 3  !
: 3294 3  !
: 3295 3  !
: 3296 3  !
: 3297 3  !
: 3298 3  !
: 3299 4  !
: 3300 4  !
: 3301 4  !
: 3302 4  !
: 3303 4  !
: 3304 4  !
: 3305 4  !
: 3306 4  !
: 3307 4  !
: 3308 4  !
: 3309 5  !
: 3310 5  !
: 3311 5  !
: 3312 5  !
: 3313 5  !
: 3314 5  !
: 3315 5  !
: 3316 4  !
: 3317 4  !
: 3318 5  !
: 3319 5  !
: 3320 5  !
: 3321 5  !
: 3322 5  !
: 3323 5  !
: 3324 5  !
: 3325 6  !
: 3326 6  !
: 3327 6  !
: 3328 6  !
: 3329 6  !
: 3330 6  !
: 3331 6  !

! THEN, THIS TEST WILL FIRST ISSUE THE MSCP "AVAILABLE" COMMAND WITH THE
! SPIN DOWN MODIFIER SET. IT WILL THEN WAIT FOR 30 SECONDS TO INSURE
! THAT THE DRIVE HAS HAD TIME TO SPIN DOWN. IT WILL THEN ISSUE THE MSCP
! "ONLINE" COMMAND TO SPIN THE DRIVE UP. THIS OPERATION WILL BE TIMED
! AND THE TIME WILL BE REPORTED TO THE OPERATOR SO THAT THIS TIME CAN BE
! VERIFIED TO MAKE SURE IT IS WITH IN LIMITS. THE RUN/START AND HEAD
! LOAD INTERNAL DIAGNOSTICS WILL RUN DURING THIS TIME. IF AN ERROR IS
! ENCOUNTERED THE RETURNED STATUS OF THE "ONLINE" COMMAND WILL BE
! SOMETHING OTHER THAN "SUCCESS" AND THIS STATUS WILL BE DECODED
! AND REPORTED WITH ERROR MESSAGE.

! IF THE OPERATOR HAS SPECIFIED RETRIES ON ERROR, THE TEST WILL BE
! REPEATED.

! VER:C
label
  BLOCK1:

  if .SWP_TRACE then PRINTF (DBM20);          ! TEST 14

  NUM_RETRIES = ZERO;

  while (.NUM_RETRIES lequ .SWP_RETRIES) do
    begin
      TIP = 14;                                ! THIS IS A FLAG TO INDICATE
                                              ! TO AZTEC_READY ROUTINE
                                              ! TO SKIP ONLINE MSCP COMMAND.

      ! GET AZTEC AVAILABLE BY INITIALIZING RC25 CONTROLLER,
      ! AND COM_AREA
      !
      if AZTEC_READY ()                        ! IF FAILURE REPORT ERROR
      then
        begin
          ERRDF (49, AZT_READY_ERR, 0);        !
          if .RET_STATUS then DECODE ();       ! DECODE THE STATUS, IF ANY
          RETRIES = TRUE;                       ! SET RETRIES FLAG
        end
      else
        BLOCK1 :
          begin
            ! ISSUE AVAILABLE COMMAND WITH SPIN DOWN MODIFIER SET
            !
            if AVAILABLE ()                    ! ISSUE AVAILABLE COMMAND
            then
              begin
                ERRDF (50, AVAIL_ERR, 0);       ! DISPLAY ERROR MESSAGE
                if .RET_STATUS then DECODE ();  ! DECODE RETURN STATUS
                RETRIES = TRUE;
              end
            leave BLOCK1;
          end
        end

```

```

: 3332 5      end;
: 3333 5
: 3334 5      ! WAIT AT LEAST 30 SEC. TO MAKE SURE THE SPINDLE IS SPUN DOWN.
: 3335 5
: 3336 5      CLOCK_INIT ();          ! VER:C
: 3337 5
: 3338 5      incru TIME from 1 to 1000 do      ! DELAY 30 SECOND;
: 3339 6      begin
: 3340 6      DELAY (100);          ! DELAY
: 3341 6      IF .SECONDS GEQU 30 THEN EXITLOOP; ! VER:C
: 3342 6      BREAK;              ! LOOK FOR CONTROL C
: 3343 6      end;
: 3344 5
: 3345 5      .CLK_CSR = ZERO;          ! STOP THE CLOCK
: 3346 5
: 3347 5      ! ISSUE A ON LINE COMMAND AND START THE CLOCK.
: 3348 5
: 3349 5      CLOCK_INIT ();          ! INITIALIZE VARIABLES FOR CLOCK
: 3350 5      CMD_REF = .CMD_SLOT;      ! SET COMMAND REFERENCE TO 3
: 3351 5
: 3352 5      if ON_LINE ()          ! SEND ON LINE COMMAND
: 3353 5      then                    ! IF COMMAND FAILED
: 3354 6      begin
: 3355 6      ERRDF (51, AZT_READY_ERR, 0); ! REPORT ERROR
: 3356 6
: 3357 6      if .RET_STATUS then DECODE (); ! DECODE STATUS
: 3358 6
: 3359 6      RETRIES = TRUE;
: 3360 6      leave BLOCK1;
: 3361 5      end;
: 3362 5
: 3363 5      .CLK_CSR = ZERO;          ! TURN OFF THE CLOCK
: 3364 5      ! CALCULATE TIME ELAPSED
: 3365 5
: 3366 5      P4 = .TICKS*100/.CLK_HERTZ; ! TICKS CONVERTED TO 100TH OF A SEC
: 3367 5      P5 = .TICKS*100 mod .CLK_HERTZ; ! REMAINDER
: 3368 5      P4 = .P4 + .P5*2/.CLK_HERTZ; ! ADD 1 TO TICKS IF > .5
: 3369 5      PRINTB (FMT8, .MINUTES, .SECONDS, .P4); ! PRINT TIME TAKEN TO COME ONLINE
: 3370 4      end;
: 3371 4
: 3372 4      if (.RETRIES) then DO_RETRIES ();
: 3373 4
: 3374 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 3375 4
: 3376 4      PASSO = FALSE;          ! RESET FLAG
: 3377 3      end;
: 3378 3
: 3379 3      return;
: 3380 1      ENDTST;

```

000000	004167	000000G	\$T14:	.SBTTL	\$T14 TEST SECTION		
000004	005746			JSR	R1,\$SAVE2	:	3261
000006	032767	000001 000000G		TST	-(SP)	:	
000014	001407			BIT	#1,SWP.TRACE	:	3294
000016	012746	000000G		BEQ	1\$		
				MOV	#DBM20,-(SP)		

000022	012746	000001		MOV	#1,-(SP)			
000026	010600			MOV	SP,R0	; SP,*		
000030	104417			TRAP	17			
000032	022626			CMP	(SP)*,(SP)*			
000034	005067	000000G	1#:	CLR	NUM.RETRIES			3296
000040	026767	000000G 000000G	2#:	CMP	NUM.RETRIES,SWP.RETRIES			3298
000046	101402			BLOS	3#			
000050	000167	000474		JMP	19#			
000054	012767	000016 000000G	3#:	MOV	#16,TIP			3300
000062	004767	000000G		JSR	PC,AZTEC.READY			3307
000066	006000			ROR	R0			
000070	103016			BCC	5#			
000072	104455			TRAP	55			3310
000074	000061			.WORD	61			
000076	000000G			.WORD	AZT.READY.ERR			
000100	000000			.WORD	0			
000102	032767	000001 000000G		BIT	#1,RET.STATUS			3312
000110	001402			BEQ	4#			
000112	004767	000000G		JSR	PC,DECODE			
000116	012767	000001 000000G	4#:	MOV	#1,RETRIES			3314
000124	000574			BR	17#			3307
000126	004767	000000G	5#:	JSR	PC,AVAILABLE			3323
000132	006000			ROR	R0			
000134	103020			BCC	7#			
000136	104455			TRAP	55			3326
000140	000062			.WORD	62			
000142	000000G			.WORD	AVAIL.ERR			
000144	000000			.WORD	0			
000146	032767	000001 000000G		BIT	#1,RET.STATUS			3328
000154	001402			BEQ	6#			
000156	004767	000000G		JSR	PC,DECODE			
000162	012767	000001 000000G	6#:	MOV	#1,RETRIES			3330
000170	162706	000024		SUB	#24,SP			3325
000174	000546			BR	16#			
000176	004767	000000G	7#:	JSR	PC,CLOCK.INIT			3336
000202	012702	000001		MOV	#1,R2	; *.TIME		3338
000206	012701	000144	8#:	MOV	#144,R1	; **TMP2		3340
000212	001410		9#:	BEQ	12#			
000214	016700	000000G		MOV	L#DLY,R0	; **TMP1		
000220	001403			BEQ	11#			
000222	005016		10#:	CLR	(SP)	; **TMP		
000224	005300			DEC	R0	; **TMP1		
000226	001375			BNE	10#			
000230	005301		11#:	DEC	R1	; **TMP2		
000232	000767			BR	9#			
000234	026727	000000G 000036	12#:	CMP	SECONDS,#36			3341
000242	103005			BHIS	13#			
000244	104422			TRAP	22			
000246	005202			INC	R2	; TIME		3338
000250	020227	001750		CMP	R2,#1750	; TIME,*		
000254	101754			BLOS	8#			
000256	005077	000000G	13#:	CLR	@CLK.CSR			3345
000262	004767	000000G		JSR	PC,CLOCK.INIT			3349
000266	016767	000000G 000000G		MOV	CMD.SLOT,CMD.REF			3350
000274	004767	000000G		JSR	PC,ON.LINE			3352
000300	006000			ROR	R0			
000302	103020			BCC	15#			

ZRCFB3	CZRCFC0 RC25 FR END TEST	TEST SECTION	27-Mar-1985 15:27:28	VAX-11 Bliss-16 V4.0-579	SEQ 0270
V03.0			27-Mar-1985 13:28:18	USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4	Page 75 (14)
000304	104455		TRAP	55	
000306	000063		.WORD	63	
000310	000000G		.WORD	AZT.READY.ERR	
000312	000000		.WORD	0	
000314	032767	000001 000000G	BIT	#1,RET.STATUS	
000322	001402		BEQ	14#	
000324	004767	000000G	JSR	PC,DECODE	
000330	012767	000001 000000G	MOV	#1,RETRIES	
000336	162706	000024	SUB	#24,SP	
000342	000463		BR	16#	
000344	005077	000000G	CLR	@CLK.CSR	
000350	016746	000000G	MOV	TICKS,-(SP)	
000354	012746	000144	MOV	#144,-(SP)	
000360	004767	000000G	JSR	PC,BL#MUL	
000364	010016		MOV	RO,(SP)	
000366	016746	000000G	MOV	CLK.HERTZ,-(SP)	
000372	004767	000000G	JSR	PC,BL#DIV	
000376	010067	000000G	MOV	RO,P4	
000402	016716	000000G	MOV	TICKS,(SP)	
000406	012746	000144	MOV	#144,-(SP)	
000412	004767	000000G	JSR	PC,BL#MUL	
000416	010016		MOV	RO,(SP)	
000420	016746	000000G	MOV	CLK.HERTZ,-(SP)	
000424	004767	000000G	JSR	PC,BL#MOD	
000430	010067	000000G	MOV	RO,P5	
000434	016716	000000G	MOV	P5,(SP)	
000440	006316		ASL	(SP)	
000442	016746	000000G	MOV	CLK.HERTZ,-(SP)	
000446	004767	000000G	JSR	PC,BL#DIV	
000452	066700	000000G	ADD	P4,RO	
000456	010067	000000G	MOV	RO,P4	
000462	016716	000000G	MOV	P4,(SP)	
000466	016746	000000G	MOV	SECONDS,-(SP)	
000472	016746	000000G	MOV	MINUTES,-(SP)	
000476	012746	000000G	MOV	#FMT8,-(SP)	
000502	012746	000004	MOV	#4,-(SP)	
000506	010600		MOV	SP,RO	
000510	104414		TRAP	14	
000512	062706	000024	ADD	#24,SP	
000516	032767	000001 000000G	BIT	#1,RETRIES	
000524	001402		BEQ	18#	
000526	004767	000000G	JSR	PC,DO.RETRIES	
000532	005767	000000G	TST	NUM.RETRIES	
000536	001404		BEQ	19#	
000540	005067	000000G	CLR	PASSO	
000544	000167	177270	JMP	2#	
000550	005726		TST	(SP).	
000552	000207		RTS	PC	

; Routine Size: 182 words, Routine Base: AC#CODE - 11646
 ; Maximum stack depth per invocation: 16 words

000000 004767 177220 T14:: .SBTTL T14 TEST SECTION

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0271
Page 76
(14)

000000		18:	JSR	PC,\$T14		
000004	104466		TRAP	66	:	
000006	006000		ROR	RO		
000010	103773		BLO	18		
000012	000207		RTS	PC		

3379

: Routine Size: 6 words. Routine Base: AC\$CODE * 12422
: Maximum stack depth per invocation: 2 words

: 3381 1 !<BLF/PAGE>

```

: 3382 1
: 3383 3
: 3384 3
: 3385 3
: 3386 3
: 3387 3
: 3388 3
: 3389 3
: 3390 3
: 3391 3
: 3392 3
: 3393 3
: 3394 3
: 3395 3
: 3396 3
: 3397 3
: 3398 3
: 3399 3
: 3400 3
: 3401 3
: 3402 3
: 3403 3
: 3404 3
: 3405 3
: 3406 3
: 3407 3
: 3408 3
: 3409 3
: 3410 3
: 3411 3
: 3412 3
: 3413 3
: 3414 3
: 3415 3
: 3416 4
: 3417 4
: 3418 4
: 3419 4
: 3420 4
: 3421 4
: 3422 5
: 3423 5
: 3424 5
: 3425 5
: 3426 5
: 3427 3
: 3428 3
: 3429 4
: 3430 5
: 3431 5
: 3432 5
: 3433 5
: 3434 5
: 3435 5
: 3436 5
: 3437 5
: 3438 5

!
BGNTST;

!..
! TEST #15 - SEQUENTIAL SEEK AND VERIFY TEST

! DESCRIPTION:

! THIS TEST BRINGS RC25 CONTROLLER AND UNIT ONLINE AND READY TO ACCEPT
! MSCP DUP COMMANDS.

! STARTING WITH THE USER SPECIFIED BEGINNING TRACK AND INCREMENTING
! THROUGH EVERY TRACK TO THE USER SPECIFIED ENDING TRACK, THIS TEST WILL
! SEEK FROM TRACK TO TRACK IN A FORWARD DIRECTION, THEN IT WILL REPEAT
! THE OPERATION IN THE REVERSE DIRECTION, FROM THE ENDING TRACK TO THE
! BEGINNING.

! THIS IS A SINGLE SURFACE TEST AND IS DONE ON TOP SURFACE. THE OPERATOR
! CAN SELECT BOTTOM SURFACE ALSO.

! A FAILURE REPORT INCLUDES STARTING TRACK, ENDING TRACK AND DESIRED
! TRACK. AFTER REPORTING THE FAILURE, THE PROGRAM WILL ABORT CURRENT
! SEEK AND WILL JUMP TO REVERSE SEEK.

!..

label
  BLOCK1,
  BLOCK2;

if .SWP_TRACE then PRINTF (DBM21);          ! TEST 15

NUM_RETRIES = ZERO;

while (.NUM_RETRIES lequ .SWP_RETRIES) do
  begin
    TIP = 15;
    ! GET AZTEC READY FOR OPERATION

    if AZTEC_READY ()                       ! IF FAILURE REPORT ERROR
    then
      begin
        ERRDF (52, AZT_READY_ERR, 0);      !
      end

      if .RET_STATUS then DECODE ();        ! DECODE THE STATUS, IF ANY

      RETRIES = TRUE;                       ! SET RETRIES FLAG
    end
  else
    begin
      ! PREPARE FOR READ CMD
      BYTE_COUNT = ZERO;                    ! SET BYTE COUNT TO ZERO
      BUF_DESCRIPTOR = ZERO;                ! CLEAN THE BUFFER
    end

    ! FORWARD DIRECTION SEEK

    ! USE STARTING TRACK OF 0 OR AS SPECIFIED BY THE OPERATOR
    ! USE ENDING TRACK OF 820 OR AS SPECIFIED BY THE OPERATOR

```

```

: 3439 5 ! SIZ_LBN = 31 INCREMENTS THE TRACK NUMBER BY 1
: 3440 5 LBN_ST = (.SWP_START*.SIZ_LBN); ! FIND STARTING LBN
: 3441 5 LBN_ED = (.SWP_END*.SIZ_LBN); ! FIND ENDING LBN NUMBER
: 3442 5 LBN_SZ = .SIZ_LBN; ! TRACK INCREMENTING SIZE
: 3443 5 ! LBN_ST IS THE DESIRED LBN TO SEEK. THIS WILL BE USED BY
: 3444 5 ! READ_CMD.
: 3445 5 BLOCK1 :
: 3446 6 begin
: 3447 6
: 3448 6 while TRUE do ! DO SEEK FROM STARTING TRACK
: 3449 7 begin ! TO ENDING TRACK
: 3450 7 CMD_REF = .LBN_ST; ! PUT LBN IN CMD_LREF
: 3451 7
: 3452 7 if READ_CMD () ! ISSUE A SEEK COMMAND
: 3453 7 then
: 3454 8 begin ! IF ERROR, REPORT
: 3455 8 ERRDF (53, SK_FOR_ERR, 0);
: 3456 8 PRINTB (FMT9, .SWP_START, .SWP_END, .LBN_ST);
: 3457 8
: 3458 8 if .RET_STATUS then DECODE (); ! DECODE STATUS OF READ_CMD
: 3459 8
: 3460 8 RETRIES = TRUE;
: 3461 8 leave BLOCK1;
: 3462 8 end
: 3463 7 else
: 3464 8 begin
: 3465 8
: 3466 8 if .LBN_ST eqv .LBN_ED then exitloop;
: 3467 8
: 3468 8 LBN_ST = .LBN_ST + .LBN_SZ; ! INCREMENT THE TRACK NUMBER BY 1
: 3469 7 end;
: 3470 7
: 3471 6 end;
: 3472 6
: 3473 5 end;
: 3474 5
: 3475 5 ! REVERSE DIRECTION SEEK
: 3476 5
: 3477 5 ! SWAP LBN NUMBERS SUCH THAT LBN_ST CONTAINS THE HIGHEST NUMBER LBN
: 3478 5 ! TO SEEK
: 3479 5 TEMP = .LBN_ED;
: 3480 5 LBN_ED = (.SWP_START*.SIZ_LBN); ! RESTORE ENDING TRACK NUMBER
: 3481 5 LBN_ST = .TEMP;
: 3482 5 BLOCK2 :
: 3483 6 begin
: 3484 6
: 3485 6 while TRUE do ! DO SEEK FROM HIGHEST LBN
: 3486 7 begin ! TO LOWEST LBN
: 3487 7 CMD_REF = .LBN_ST; ! PUT LBN IN CMD_LREF
: 3488 7
: 3489 7 if READ_CMD () ! ISSUE A SEEK COMMAND
: 3490 7 then
: 3491 8 begin
: 3492 8 ERRDF (54, SK_REV_ERR, 0);
: 3493 8 PRINTB (FMT9, .SWP_END, .SWP_START, .LBN_ST);
: 3494 8
: 3495 8 if .RET_STATUS then DECODE (); ! DECODE STATUS OF READ_CMD

```

```

: 3496 8
: 3497 8
: 3498 8
: 3499 8
: 3500 7
: 3501 8
: 3502 8
: 3503 8
: 3504 8
: 3505 8
: 3506 7
: 3507 7
: 3508 6
: 3509 6
: 3510 5
: 3511 4
: 3512 4
: 3513 4
: 3514 4
: 3515 4
: 3516 4
: 3517 3
: 3518 3
: 3519 3
: 3520 1

      RETRIES = TRUE;
      leave BLOCK2;
      end
    else
      begin
        if .LBN_ST eqlu .LBN_ED then exitloop; ! WHEN ALL SEEKS DONE EXIT
        LBN_ST = .LBN_ST - .LBN_SZ; ! DECREMENT TRACK NUMBER BY 1
        end;
      end;
    end;
  end;
end;
if (.RETRIES) then JO_RETRIES ();
if (.NUM_RETRIES eqlu ZERO) then exitloop;
end;
return;
ENDTST;

```

000000	032767	000001	000000G	\$T15:	.SBTTL \$T15 TEST SECTION		
000006	001407				BIT #1,SWP.TRACE	:	3411
000010	012746	000000G			BEQ 1\$		
000014	012746	000001			MOV #DBM21,-(SP)		
000020	010600				MOV #1,-(SP)		
000022	104417				MOV SP,R0	: SP,*	
000024	022626				TRAP 17		
000026	005067	000000G		1\$:	CMP (SP),.(SP).		
000032	026767	000000G	000000G	2\$:	CLR NUM.RETRIES	:	3413
000040	101401				CMP NUM.RETRIES,SWP.RETRIES	:	3415
000042	000207				BLOS 3\$		
000044	012767	000017	000000G	3\$:	RTS PC		
000052	004767	000000G			MOV #17,TIP	:	3417
000056	006000				JSR PC,AZTEC.READY	:	3420
000060	103016				ROR R0		
000062	104455				BCC 5\$		
000064	000064				TRAP 55	:	3423
000066	000000G				.WORD 64		
000070	000000				.WORD AZT.READY.ERR		
000072	032767	000001	000000G		.WORD 0		
000100	001402				BIT #1,RET.STATUS	:	3425
000102	004767	000000G			BEQ 4\$		
000106	012767	000001	000000G	4\$:	JSR PC,DECODE		
000114	000575				MOV #1,RETRIES	:	3427
000116	005067	000000G			BR 14\$:	3420
000122	005067	000000G		5\$:	CLR BYTE.COUNT	:	3432
000126	016746	000000G			CLR BUF.DESCRPTR	:	3433
000132	016746	000000G			MOV SWP.START,-(SP)	:	3440
000136	004767	000000G			MOV SIZ.LBN,-(SP)	:	
					JSR PC,BL\$MUL		

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0275
Page 80
(15)

000142	010067	000000G			MOV	RO,LBN.ST		
000146	016716	000000G			MOV	SWP.END,(SP)	:	3441
000152	016746	000000G			MOV	SIZ.LBN,-(SP)		
000156	004767	000000G			JSR	PC,BL#MUL		
000162	010067	000000G			MOV	RO,LBN.ED		
000166	016767	000000G	000000G		MOV	SIZ.LBN,LBN.SZ	:	3442
000174	016767	000000G	000000G	6\$:	MOV	LBN.ST,CMD.REF	:	3450
000202	004767	000000G			JSR	PC,READ.CMD	:	3452
000206	006000				ROR	RO		
000210	103034				BCC	8\$		
000212	104455				TRAP	55	:	3455
000214	000065				.WORD	65		
000216	000000G				.WORD	SK.FOR.ERR		
000220	000000				.WORD	0		
000222	016716	000000G			MOV	LBN.ST,(SP)	:	3456
000226	016746	000000G			MOV	SWP.END,-(SP)		
000232	016746	000000G			MOV	SWP.START,-(SP)		
000236	012746	000000G			MOV	#FMT9,-(SP)		
000242	012746	000004			MOV	#4,-(SP)		
000246	010600				MOV	SP,RO	: SP,*	
000250	104414				TRAP	14		
000252	062706	000010			ADD	#10,SP		
000256	032767	000001	000000G		BIT	#1,RET.STATUS	:	3458
000264	001402				BEQ	7\$		
000266	004767	000000G			JSR	PC,DECODE		
000272	012767	000001	000000G	7\$:	MOV	#1,RETRIES	:	3460
000300	000410				BR	9\$:	3454
000302	026767	000000G	000000G	8\$:	CMP	LBN.ST,LBN.ED	:	3466
000310	001404				BEQ	9\$		
000312	066767	000000G	000000G		ADD	LBN.SZ,LBN.ST	:	3468
000320	000725				BR	6\$:	3448
000322	016767	000000G	000000G	9\$:	MOV	LBN.ED,TEMP	:	3479
000330	016716	000000G			MOV	SWP.START,(SP)	:	3480
000334	016746	000000G			MOV	SIZ.LBN,-(SP)		
000340	004767	000000G			JSR	PC,BL#MUL		
000344	010067	000000G			MOV	RO,LBN.ED		
000350	016767	000000G	000000G		MOV	TEMP,LBN.ST	:	3481
000356	016767	000000G	000000G	10\$:	MOV	LBN.ST,CMD.REF	:	3487
000364	004767	000000G			JSR	PC,READ.CMD	:	3489
000370	006000				ROR	RO		
000372	103034				BCC	12\$		
000374	104455				TRAP	55	:	3492
000376	000066				.WORD	66		
000400	000000G				.WORD	SK.REV.ERR		
000402	000000				.WORD	0		
000404	016716	000000G			MOV	LBN.ST,(SP)	:	3493
000410	016746	000000G			MOV	SWP.START,-(SP)		
000414	016746	000000G			MOV	SWP.END,-(SP)		
000420	012746	000000G			MOV	#FMT9,-(SP)		
000424	012746	000004			MOV	#4,-(SP)		
000430	010600				MOV	SP,RO	: SP,*	
000432	104414				TRAP	14		
000434	062706	000010			ADD	#10,SP		
000440	032767	000001	000000G		BIT	#1,RET.STATUS	:	3495
000446	001402				BEQ	11\$		
000450	004767	000000G			JSR	PC,DECODE		
000454	012767	000001	000000G	11\$:	MOV	#1,RETRIES	:	3497

E6

ZRCFB3 CZRCFC0 RC25 FR END TEST 27-Mar-1985 15:27:28 VAX-11 Bliss-16 V4.0-579 SEQ 0276
 V03.0 TEST SECTION 27-Mar-1985 13:28:18 USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4 Page 81
 (15)

000462	000410			BR	13‡					
000464	026767	000000G	000000G	12‡:	CMP	LBN.ST,LBN.ED				3491
000472	001404				BEQ	13‡				3503
000474	166767	000000G	000000G		SUB	LBN.SZ,LBN.ST				
000502	000725				BR	10‡				3505
000504	062706	000010		13‡:	ADD	#10,SP				3485
000510	032767	000001	000000G	14‡:	BIT	#1,RETRIES				3430
000516	001402				BEQ	15‡				3513
000520	004767	000000G			JSR	PC,DO.RETRIES				
000524	005767	000000G		15‡:	TST	NUM.RETRIES				3515
000530	001402				BEQ	16‡				
000532	000167	177274			JMP	2‡				
000536	000207			16‡:	RTS	PC				3380

: Routine Size: 176 words, Routine Base: AC‡CODE + 12436
 : Maximum stack depth per invocation: 10 words

000000 004767 177234 T15:: .SBTTL T15 TEST SECTION
 000000 1‡: JSR PC,‡T15 ; 3519
 000004 104466 TRAP 66
 000006 006000 ROR R0
 000010 103773 BLO 1‡
 000012 000207 RTS PC

: Routine Size: 6 words, Routine Base: AC‡CODE + 13176
 : Maximum stack depth per invocation: 2 words

: 3521 1 !<BLF/PAGE>

```

: 3522 1 !
: 3523 3 !BGNTST;
: 3524 3
: 3525 3 !**
: 3526 3 ! TEST #16 - SAWTOOTH SEEK AND VERIFY TEST
: 3527 3
: 3528 3 ! DESCRIPTION:
: 3529 3
: 3530 3 ! THIS TEST BRINGS RC25 CONTROLLER AND THE UNIT ONLINE AND READY
: 3531 3 ! TO TAKE MSCP COMMANDS.
: 3532 3
: 3533 3 ! STARTING WITH THE USER SPECIFIED BEGINNING TRACK AND INCREMENTING
: 3534 3 ! THROUGH EVERY TRACK IN THE SELECTED RANGE, THIS TEST WILL PERFORM A
: 3535 3 ! SEEK TO THE SELECTED TRACK AND THEN A SEEK BACK TO THE BEGINNING
: 3536 3 ! TRACK. WHEN ALL TRACKS HAVE BEEN COVERED, IT WILL DO THE SAME
: 3537 3 ! OPERATION IN THE REVERSE DIRECTION WITH THE ENDING TRACK AS THE
: 3538 3 ! BASE.
: 3539 3
: 3540 3 ! THIS IS A SINGLE SURFACE TEST AND IS DONE ON TOP SURFACE.
: 3541 3 ! THE OPERATOR CAN SELECT BOTTOM SURFACE ALSO.
: 3542 3
: 3543 3 ! ERROR REPORTS WILL STATE STARTING, ENDING AND DESIRED TRACKS.
: 3544 3 ! IF THERE WAS AN ERROR THE TEST WILL BE ABORTED UNLESS THE OPERATOR
: 3545 3 ! HAS SELECTED FOR RETRIES.
: 3546 3
: 3547 3 !--
: 3548 3
: 3549 3 local
: 3550 3 LBN_SA;
: 3551 3
: 3552 3 label
: 3553 3 BLOCK1,
: 3554 3 BLOCK2;
: 3555 3
: 3556 3 if .SWP_TRACE then PRINTF (DBM22); ! TEST 16
: 3557 3
: 3558 3 NUM_RETRIES = ZERO;
: 3559 3
: 3560 3 while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 3561 4 begin
: 3562 4 TIP = 16;
: 3563 4 ! GET AZTEC READY FOR OPERATION
: 3564 4
: 3565 4 if AZTEC_READY () ! IF FAILURE REPORT ERROR
: 3566 4 then
: 3567 5 begin
: 3568 5 ERROF (55, AZT_READY_ERR, 0); !
: 3569 5
: 3570 5 if .RET_STATUS then DECODE (); ! DECODE THE STATUS, IF ANY
: 3571 5
: 3572 5 RETRIES = TRUE; ! SET RETRIES FLAG
: 3573 5 end
: 3574 4 else
: 3575 5 begin
: 3576 5 ! PREPARE FOR READ_CMD
: 3577 5 BYTE_COUNT = ZERO; ! SET BYTE COUNT TO ZERO
: 3578 5 BUF_DESCRPT = ZERO; ! CLEAN THE BUFFER

```


ZRCFR3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.816;4SEQ 0278
Page 83
(16)

```

: 3579 5
: 3580 5
: 3581 5
: 3582 5
: 3583 5
: 3584 5
: 3585 5
: 3586 5
: 3587 5
: 3588 5
: 3589 5
: 3590 5
: 3591 5
: 3592 6
: 3593 6
: 3594 6
: 3595 7
: 3596 7
: 3597 7
: 3598 7
: 3599 8
: 3600 8
: 3601 8
: 3602 8
: 3603 8
: 3604 9
: 3605 9
: 3606 9
: 3607 9
: 3608 9
: 3609 9
: 3610 9
: 3611 9
: 3612 9
: 3613 8
: 3614 8
: 3615 8
: 3616 7
: 3617 7
: 3618 7
: 3619 7
: 3620 7
: 3621 6
: 3622 6
: 3623 5
: 3624 5
: 3625 5
: 3626 5
: 3627 5
: 3628 5
: 3629 5
: 3630 5
: 3631 5
: 3632 6
: 3633 6
: 3634 6
: 3635 7

!
! FORWARD DIRECTION SEEK
! USE STARTING TRACK OF 0 OR AS SPECIFIED BY THE OPERATOR
! USE ENDING TRACK OF 820 OR AS SPECIFIED BY THE OPERATOR
! LBN_SZ = 31 INCREMENTS THE TRACK NUMBER BY 1
! LBN_ST = (.SWP_START*.SIZ_LBN); ! FIND STARTING LBN
! LBN_ED = (.SWP_END*.SIZ_LBN); ! FIND ENDING LBN NUMBER
! LBN_SZ = .SIZ_LBN; ! TRACK INCREMENTING SIZE
! LBN_SA = .LBN_ST; ! LOAD CURRENT LBN
! LBN_ST IS THE DESIRED LBN TO SEEK. THIS WILL BE USED BY
! READ_CMD.
BLOCK1 :
begin
while TRUE do ! DO SEEK FROM STARTING TRACK
begin ! TO ENDING TRACK BY INCREMENTING
LBN_ST = .LBN_SA; ! ONE TRACK AT A TIME.

incru COUNT from 0 to 1 do ! ISSUE TWO SEEK COMMANDS
begin ! ONE TO THE CURRENT TRACK
CMD_REF = .LBN_ST; ! PUT LBN IN CMD_LREF

if READ_CMD () ! AND THE SECOND TO THE STARTING
then ! TRACK EVERY TIME YOU INCREMENT
begin ! TRACK NUMBER. IF FAILURE
ERRDF (56, SK_FOR_ERR, 0); ! WILL BE REPORTED WITH
PRINTB (FMT9, .SWP_START, .SWP_END, .LBN_ST); ! ERROR INFO.

if .RET_STATUS then DECODE (); ! DECODE STATUS OF READ_CMD

RETRIES = TRUE; ! TURN ON RETRIES
leave BLOCK1; ! ABORT IF ERROR OCCURED
end
else
LBN_ST = (.SWP_START*.SIZ_LBN);

end;

if .LBN_SA ealu .LBN_ED then exitloop;

LBN_SA = .LBN_SA + .LBN_SZ; ! INCREMENT TRACK NUMBER BY 1
end;

end;

! REVERSE DIRECTION SEEK
! SWAP LBN NUMBERS SUCH THAT LBN_ST CONTAINS THE HIGHEST NUMBER LBN
! TO SEEK
! LBN_SA = .LBN_ED;
! LBN_ED = (.SWP_START*.SIZ_LBN); ! RESTORE ENDING TRACK NUMBER
BLOCK2 :
begin
while TRUE do ! DO SEEK FROM HIGHEST LBN
begin ! TO LOWEST LBN

```

```

: 3636 7      LBN_ST = .LBN_SA;
: 3637 7
: 3638 7      incru COUNT from 0 to 1 do
: 3639 8      begin
: 3640 8      CMD_REF = .LBN_ST;          ! PUT LBN IN CMD_LREF
: 3641 8
: 3642 8      if READ_CMD ()          ! ISSUE A SEEK COMMAND
: 3643 8      then
: 3644 9      begin
: 3645 9      ERRDF (57, SK_REV_ERR, 0);
: 3646 9      PRINTB (FMT9, .SWP_END, .SWP_START, .LBN_ST);
: 3647 9
: 3648 9      if .RET_STATUS then DECODE ();    ! DECODE STATUS OF READ_CMD
: 3649 9
: 3650 9      RETRIES = TRUE;          ! TURN ON RETRIES
: 3651 9      leave BLOCK2;          ! ABORT REVERSE SEEK
: 3652 9      end
: 3653 8      else
: 3654 8      LBN_ST = (.SWP_END*.SIZ_LBN);    ! STARTING TRACK
: 3655 8
: 3656 7      end;
: 3657 7
: 3658 7      if .LBN_SA eqlu .LBN_ED then exitloop;
: 3659 7
: 3660 7      LBN_SA = .LBN_SA - .LBN_SZ;    ! NEXT TRACK IN SEQUENCE
: 3661 6      end;
: 3662 6
: 3663 5      end;
: 3664 4      end;
: 3665 4
: 3666 4      if (.RETRIES) then DO_RETRIES ();
: 3667 4
: 3668 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 3669 4
: 3670 3      end;
: 3671 3
: 3672 3      return;
: 3673 1      ENDTST;

```

000000	004167	000000G		.SBTTL	\$T16 TEST SECTION		
000004	032767	000001	000000G	\$T16:	JSR	R1,\$SAVE2	3520
000012	001407				BIT	#1,SWP.TRACE	3556
000014	012746	000000G			BEQ	1\$	
000020	012746	000001			MOV	#DBM22,-(SP)	
000024	010600				MOV	#1,-(SP)	
000026	104417				MOV	SP,R0	: SP,*
000030	022626				TRAP	17	
000032	005067	000000G			CMP	(SP)*,(SP)*	
000036	026767	000000G	000000G	1\$:	CLR	NUM.RETRIES	3558
000044	101401			2\$:	CMP	NUM.RETRIES,SWP.RETRIES	3560
000046	000207				BLOS	3\$	
000050	012767	000020	000000G	3\$:	RTS	PC	
000056	004767	000000G			MOV	#20,TIP	3562
000062	006000				JSR	PC,AZTEC.READY	3565
000064	103017				ROR	R0	
					BCC	5\$	

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0280
Page 85
(16)

000066	104455			TRAP	55	:		3568
000070	000067			.WORD	67	:		
000072	000000G			.WORD	AZT.READY.ERR	:		
000074	000000			.WORD	0	:		
000076	032767	000001	000000G	BIT	#1,RET.STATUS	:		3570
000104	001402			BEQ	4\$:		
000106	004767	000000G		JSR	PC,DECODE	:		
000112	012767	000001	000000G	MOV	#1,RETRIES	:		3572
000120	000167	000456		JMP	16\$:		3565
000124	005067	000000G		CLR	BYTE.COUNT	:		3577
000130	005067	000000G		CLR	BUF.DESCRPTR	:		3578
000134	016746	000000G		MOV	SWP.START, -(SP)	:		3585
000140	016746	000000G		MOV	SIZ.LBN, -(SP)	:		
000144	004767	000000G		JSR	PC,BL\$MUL	:		
000150	010067	000000G		MOV	R0,LBN.ST	:		
000154	016716	000000G		MOV	SWP.END, (SP)	:		3586
000160	016746	000000G		MOV	SIZ.LBN, -(SP)	:		
000164	004767	000000G		JSR	PC,BL\$MUL	:		
000170	010067	000000G		MOV	R0,LBN.ED	:		
000174	016767	000000G	000000G	MOV	SIZ.LBN,LBN.SZ	:		3587
000202	016702	000000G		MOV	LBN.ST,R2	:	* ,LBN.SA	3588
000206	010267	000000G		MOV	R2,LBN.ST	:	LBN.SA,*	3596
000212	005001			CLR	R1	:	COUNT	3598
000214	016767	000000G	000000G	MOV	LBN.ST,CMD.REF	:		3600
000222	004767	000000G		JSR	PC,READ.CMD	:		3602
000226	006000			ROR	R0	:		
000230	103034			BCC	9\$:		
000232	104455			TRAP	55	:		3605
000234	000070			.WORD	70	:		
000236	000000G			.WORD	SK.FOR.ERR	:		
000240	000000			.WORD	0	:		
000242	016716	000000G		MOV	LBN.ST, (SP)	:		3606
000246	016746	000000G		MOV	SWP.END, -(SP)	:		
000252	016746	000000G		MOV	SWP.START, -(SP)	:		
000256	012746	000000G		MOV	#FMT9, -(SP)	:		
000262	012746	000004		MOV	#4, -(SP)	:		
000266	010600			MOV	SP,R0	:	SP,*	
000270	104414			TRAP	14	:		
000272	062706	000010		ADD	#10,SP	:		
000276	032767	000001	000000G	BIT	#1,RET.STATUS	:		3608
000304	001402			BEQ	8\$:		
000306	004767	000000G		JSR	PC,DECODE	:		
000312	012767	000001	000000G	MOV	#1,RETRIES	:		3610
000320	000423			BR	10\$:		3604
000322	016716	000000G		MOV	SWP.START, (SP)	:		3614
000326	016746	000000G		MOV	SIZ.LBN, -(SP)	:		
000332	004767	000000G		JSR	PC,BL\$MUL	:		
000336	010067	000000G		MOV	R0,LBN.ST	:		
000342	005726			TST	(SP)+	:		
000344	005201			INC	R1	:	COUNT	3598
000346	020127	000001		CMP	R1,#1	:	COUNT,*	
000352	101720			BLOS	7\$:		
000354	020267	000000G		CMP	R2,LBN.ED	:	LBN.SA,*	3618
000360	001403			BEQ	10\$:		
000362	066702	000000G		ADD	LBN.SZ,R2	:	* ,LBN.SA	3620
000366	000707			BR	6\$:		3594
000370	016702	000000G		MOV	LBN.ED,R2	:	* ,LBN.SA	3629

ZRCFB3	CZRCFC0	RC25	FR	END	TEST	27-Mar-1985	15:27:28	VAX-11	Bliss-16	V4.0-579	SEQ 0281	Page 86
V03.0	TEST SECTION					27-Mar-1985	13:28:18	USER#1:	[AZTEC.CZRCFC]	ZRCFC3.B16;4	(16)	
000374	016716	000000G										3630
000400	016746	000000G										
000404	004767	000000G										
000410	010067	000000G										
000414	010267	000000G			11#:							3636
000420	005001	000000G										3638
000422	016767	000000G	000000G		12#:							3640
000430	004767	000000G										3642
000434	006000											
000436	103034											
000440	104455											
000442	000071											3645
000444	000000G											
000446	000000											
000450	016716	000000G										3646
000454	016746	000000G										
000460	016746	000000G										
000464	012746	000000G										
000470	012746	0000004										
000474	010600											
000476	104414											
000500	062706	000010										
000504	032767	000001	000000G									3648
000512	001402											
000514	004767	000000G										
000520	012767	000001	000000G		13#:							3650
000526	000423											3644
000530	016716	000000G			14#:							3654
000534	016746	000000G										
000540	004767	000000G										
000544	010067	000000G										
000550	005726											
000552	005201											
000554	020127	000001										3638
000560	101720											
000562	020267	000000G										3658
000566	001403											
000570	166702	000000G										3660
000574	000707											3634
000576	062706	000010			15#:							3575
000602	032767	000001	000000G		16#:							3666
000610	001402											
000612	004767	000000G										
000616	005767	000000G			17#:							3668
000622	001402											
000624	000167	177206										
000630	000207				18#:							3520

; Routine Size: 205 words, Routine Base: AC\$CODE * 13212
; Maximum stack depth per invocation: 13 words

000000	004767	177142		T16::	.SBTTL	T16 TEST SECTION						
000000				1#:	JSR	PC,\$T16						3672

K6

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0282
Page 87
(16)

000004 104466
000006 006000
000010 103773
000012 000207

TRAP 66
ROR RO
BLO 1\$
RTS PC

: Routine Size: 6 words. Routine Base: AC\$CODE * 14044
: Maximum stack depth per invocation: 2 words

: 3674 1 !<BLF/PAGE>

ZRCFR3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.016;4SEQ 0283
Page 88
(17)

```

: 3675 1
: 3676 3
: 3677 3
: 3678 3
: 3679 3
: 3680 3
: 3681 3
: 3682 3
: 3683 3
: 3684 3
: 3685 3
: 3686 3
: 3687 3
: 3688 3
: 3689 3
: 3690 3
: 3691 3
: 3692 3
: 3693 3
: 3694 3
: 3695 3
: 3696 3
: 3697 3
: 3698 3
: 3699 3
: 3700 3
: 3701 3
: 3702 3
: 3703 3
: 3704 3
: 3705 3
: 3706 3
: 3707 3
: 3708 3
: 3709 3
: 3710 3
: 3711 3
: 3712 3
: 3713 4
: 3714 4
: 3715 4
: 3716 4
: 3717 4
: 3718 4
: 3719 5
: 3720 5
: 3721 5
: 3722 5
: 3723 5
: 3724 3
: 3725 3
: 3726 4
: 3727 4
: 3728 5
: 3729 5
: 3730 5
: 3731 5

!
! BGNTST;
!
! ..
! TEST #17 - CONVERGING/DIVERGING SEEK AND VERIFY TEST
!
! DESCRIPTION:
!
! THIS TEST FIRST BRINGS RC25 CONTROLLER AND UNIT ONLINE SO THAT
! MSCP COMMANDS CAN BE ISSUED.
!
! THIS TEST PERFORMS SEEKS TO THE BEGINNING TRACK, THEN TO THE ENDING
! TRACK, THEN TO THE BEGINNING TRACK + 1, ENDING TRACK - 1, BEGINNING
! TRACK + 2, ETC. UNTIL THE TRACKS CONVERGE AND THEN DIVERGE AGAIN
! BACK TO THE BEGINNING AND ENDING TRACKS.
!
! THIS IS A SINGLE SURFACE TEST AND IS DONE ON TOP SURFACE. THE
! OPERATOR CAN SELECT BOTTOM SURFACE ALSO.
!
! ERROR REPORTS WILL INCLUDE STARTING, ENDING AND DESIRED TRACKS.
! IF FAILURE IN SEEK THE TEST WILL BE ABORTED UNLESS THE OPERATOR
! SELECTS RETRIES.
!
! ..
local
  LBN_SA,           ! START OF LBN
  LBN_MID;         ! MID POINT FOR LBN
label
  BLOCK1,
  BLOCK2;
NUM_RETRIES = ZERO;
if .SWP_TRACE then PRINTF (DBM23);           ! TEST 17
while (.NUM_RETRIES lequ .SWP_RETRIES) do
  begin
  TIP = 17;
  ! GET AZTEC READY FOR OPERATION
  if AZTEC_READY ()                          ! IF FAILURE REPORT ERROR
  then
    begin
    ERRDF (58, AZT_READY_ERR, 0);           !
    if .RET_STATUS then DECODE ();         ! DECODE THE STATUS, IF ANY
    RETRIES = TRUE;                         ! SET RETRIES FLAG
    end
  else
  BLOCK2 :
    begin
    ! PREPARE FOR READ_CMD
    BYTE_COUNT = ZERO;                     ! SET BYTE COUNT TO ZERO
    BUF_DESCRIPTOR = ZERO;                 ! CLEAN THE BUFFER

```

```

: 3732 5
: 3733 5 : CONVERGING SEEK
: 3734 5
: 3735 5 : USE STARTING TRACK OF 0 OR AS SPECIFIED BY THE OPERATOR
: 3736 5 : USE ENDING TRACK OF 820 OR AS SPECIFIED BY THE OPERATOR
: 3737 5 : SIZ_LBN = 31 INCREMENTS THE TRACK NUMBER BY 1
: 3738 5 : LBN_ST = (.SWP_START*.SIZ_LBN); : FIND STARTING LBN
: 3739 5 : LBN_ED = (.SWP_END*.SIZ_LBN); : FIND ENDING LBN NUMBER
: 3740 5 : LBN_SZ = .SIZ_LBN; : TRACK INCREMENTING SIZE
: 3741 5 : LBN_SA = .LBN_ST; : LOAD CURRENT LBN
: 3742 5 : LBN_MID = (.LBN_ED - .LBN_ST)/2; : MIDPOINT BETWEEN STARTING & ENDING
: 3743 5 : LBN_ST IS THE DESIRED LBN TO SEEK. THIS WILL BE USED BY
: 3744 5 : READ_CMD.
: 3745 5 BLOCK1 :
: 3746 6 : begin
: 3747 6
: 3748 6 : while .LBN_MID gequ .LBN_SA do : DO SEEK FROM STARTING TRACK
: 3749 7 : begin : TO ENDING TRACK BY INCREMENTING
: 3750 7 : LBN_ST = .LBN_SA; : ONE TRACK AT A TIME.
: 3751 7
: 3752 7 : incru COUNT from 0 to 1 do : ISSUE TWO SEEK COMMANDS
: 3753 8 : begin : ONE TO THE CURRENT TRACK
: 3754 8 : CMD_REF = .LBN_ST; : PUT LBN IN CMD_LREF
: 3755 8
: 3756 8 : if READ_CMD () : AND THE SECOND TO THE MIRROR IMAGE
: 3757 8 : then : TRACK EVERY TIME YOU INCREMENT
: 3758 9 : begin : TRACK NUMBER. IF FAILURE
: 3759 9 : ERRDF (59, SK_FOR_ERR, 0); : WILL BE REPORTED WITH
: 3760 9 : PRINTB (FMT9, .SWP_START, .SWP_END, .LBN_ST); : ERROR INFO.
: 3761 9
: 3762 9 : if .RET_STATUS then DECODE (); : DECODE STATUS OF READ_CMD
: 3763 9
: 3764 9 : RETRIES = TRUE;
: 3765 9 : leave BLOCK1; : ABORT IF ERROR OCCURED
: 3766 9 : end
: 3767 8 : else
: 3768 8 : LBN_ST = .LBN_ED;
: 3769 8
: 3770 7 : end;
: 3771 7
: 3772 7 : LBN_SA = .LBN_SA + .LBN_SZ; : INCREMENT TRACK NUMBER BY 1
: 3773 7 : LBN_ED = .LBN_ED - .LBN_SZ; : MIRROR IMAGE OF CURRENT TRACK
: 3774 6 : end;
: 3775 6
: 3776 5 : end;
: 3777 5
: 3778 5 : DIVERGING SEEK
: 3779 5
: 3780 5 : LBN_SA = .LBN_MID; : START FROM MID TRACK LBN
: 3781 5 : LBN_ED = .LBN_MID; : START FROM MID TRACK LBN
: 3782 5
: 3783 5 : while .LBN_SA lequ .LBN_MID do : DO SEEK FROM HIGHEST LBN
: 3784 5
: 3785 6 : begin : TO LOWEST LBN
: 3786 6 : LBN_ST = .LBN_SA;
: 3787 6
: 3788 6 : incru COUNT from 0 to 1 do

```

```

: 3789 7      begin
: 3790 7      CMD_REF = .LBN_ST;          ! PUT LBN IN CMD_LREF
: 3791 7
: 3792 7      if READ_CMD ()           ! ISSUE A SEEK COMMAND
: 3793 7      then
: 3794 8          begin
: 3795 8              ERRDF (60, SK_REV_ERR, 0);
: 3796 8              PRINTB (FMT9, .SWP_START, .SWP_END, .LBN_ST);
: 3797 8
: 3798 8              if .RET_STATUS then DECODE ();      ! DECODE STATUS OF READ_CMD
: 3799 8
: 3800 8              RETRIES = TRUE;
: 3801 8              leave BLOCK2;          ! ABORT REVERSE SEEK
: 3802 8              end
: 3803 7          else
: 3804 7              LBN_ST = .LBN_ED;      ! STARTING TRACK
: 3805 7
: 3806 6          end;
: 3807 6
: 3808 6      LBN_SA = .LBN_SA - .LBN_SZ;    ! NEXT TRACK IN SEQUENCE
: 3809 6      LBN_ED = .LBN_ED + .LBN_SZ;    ! MIRROR IMAGE TRACK
: 3810 5      end;
: 3811 5
: 3812 4      end;
: 3813 4
: 3814 4      !
: 3815 4
: 3816 4      if .RETRIES then DO_RETRIES ();
: 3817 4
: 3818 4      if (.NUM_RETRIES eglu ZERO) then exitloop;
: 3819 4
: 3820 3      end;
: 3821 3
: 3822 3      re:rn;
: 3823 1      ENDIST;

```

Address	Hex	Dec	Label	Instruction	Comment	Address
000000	004167	000000G		.SBTTL	\$T17 TEST SECTION	
000004	005067	000000G	\$T17:	JSR	R1, \$SAVE3	
000010	032767	000001 000000G		CLR	NUM.RETRIES	3673
000016	001407			BIT	@1, SWP.TRACE	3708
000020	012746	000000G		BEQ	1#	3710
000024	012746	000001		MOV	@DBM23, -(SP)	
000030	010600			MOV	@1, -(SP)	
000032	104417			MOV	SP, R0	: SP, *
000034	022626			TRAP	17	
000036	026767	000000G 000000G	1#:	CMP	(SP), (SP)	
000044	101401			CMP	NUM.RETRIES, SWP.RETRIES	3712
000046	000207			BLOS	2#	
000050	012767	000021 000000G	2#:	RTS	PC	
000056	004767	000000G		MOV	@21, TIP	3714
000062	006000			JSR	PC, AZTEC.READY	3717
000064	103017			ROR	R0	
000066	104455			BCC	4#	
000070	000072			TRAP	55	
000072	000000G			.WORD	72	3720
				.WORD	AZT.READY.ERR	

000074	000000				.WORD	0			
000076	032767	000001	000000G		BIT	#1,RET.STATUS	:		3722
000104	001402				BEQ	7#			
000106	004767	000000G			JSR	PC,DECODE			
000112	012767	000001	000000G	3#:	MOV	#1,RETRIES	:		3724
000120	000167	000442			JMP	15#	:		3717
000124	005067	000000G		4#:	CLR	BYTE,COUNT	:		3730
000130	005067	000000G			CLR	BUF.DESCRPTR	:		3731
000134	016746	000000G			MOV	SWP.START,-(SP)	:		3738
000140	016746	000000G			MOV	SIZ.LBN,-(SP)			
000144	004767	000000G			JSR	PC,BL#MUL			
000150	010067	000000G			MOV	RO,LBN.ST			
000154	016716	000000G			MOV	SWP.END,(SP)	:		3739
000160	016746	000000G			MOV	SIZ.LBN,-(SP)			
000164	004767	000000G			JSR	PC,BL#MUL			
000170	010067	000000G			MOV	RO,LBN.ED			
000174	016767	000000G	000000G		MOV	SIZ.LBN,LBN.SZ	:		3740
000202	016702	000000G			MOV	LBN.ST,R2	:	*.LBN.SA	3741
000206	016716	000000G			MOV	LBN.ED,(SP)	:		3742
000212	166716	000000G			SUB	LBN.ST,(SP)			
000216	012746	000002			MOV	#2,-(SP)			
000222	004767	000000G			JSR	PC,BL#DIV			
000226	010003				MOV	RO,R3	:	*.LBN.MID	
000230	020302			5#:	CMP	R3,R2	:	LBN.MID,LBN.SA	3748
000232	103463				BLO	9#			
000234	010267	000000G			MOV	R2,LBN.ST	:	LBN.SA,*	3750
000240	005001				CLR	R1	:	COUNT	3752
000242	016767	000000G	000000G	6#:	MOV	LBN.ST,CMD.REF	:		3754
000250	004767	000000G			JSR	PC,READ.CMD	:		3756
000254	006000				ROR	RO			
000256	103034				BCC	8#			
000260	104455				TRAP	55	:		3759
000262	000073				.WORD	73			
000264	000000G				.WORD	SK.FOR.ERR			
000266	000000				.WORD	0			
000270	016716	000000G			MOV	LBN.ST,(SP)	:		3760
000274	016746	000000G			MOV	SWP.END,-(SP)			
000300	016746	000000G			MOV	SWP.START,-(SP)			
000304	012746	000000G			MOV	#FMT9,-(SP)			
000310	012746	000004			MOV	#4,-(SP)			
000314	010600				MOV	SP,RO	:	SP,*	
000316	104414				TRAP	14			
000320	062706	000010			ADD	#10,SP			
000324	032767	000001	000000G		BIT	#1,RET.STATUS	:		3762
000332	001402				BEQ	7#			
000334	004767	000000G			JSR	PC,DECODE			
000340	012767	000001	000000G	7#:	MOV	#1,RETRIES	:		3764
000346	000415				BR	9#	:		3758
000350	016767	000000G	000000G	8#:	MOV	LBN.ED,LBN.ST	:		3768
000356	005201				INC	R1	:	COUNT	3752
000360	020127	000001			CMP	R1,#1	:	COUNT,*	
000364	101726				BLOS	6#			
000366	066702	000000G			ADD	LBN.SZ,R2	:	*.LBN.SA	3772
000372	166767	000000G	000000G		SUB	LBN.SZ,LBN.ED	:		3773
000400	000713				BR	5#	:		3748
000402	010302			9#:	MOV	R3,R2	:	LBN.MID,LBN.SA	3780
000404	010367	000000G			MOV	R3,LBN.ED	:	LBN.MID,*	3781

ZRCFR3	CZRCFC0 RC25 FR END TEST	27-Mar-1985 15:27:28	VAX-11 Bliss-16 V4.0-579	SEQ 0287
V03.0	TEST SECTION	27-Mar-1985 13:28:18	USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4	Page 92 (17)
000410	020203	10‡:	CMP R2,R3	; LBN.SA,LBN.MID 3783
000412	101063		BHI 14‡	
000414	010267	000000G	MOV R2,LBN.ST	; LBN.SA,* 3786
000420	005001		CLR R1	; COUNT 3788
000422	016767	000000G 000000G	11‡: MOV LBN.ST,CMD.REF	; 3790
000430	004767	000000G	JSR PC,READ.CMD	; 3792
000434	006000		ROR R0	
000436	103034		BCC 13‡	
000440	104455		TRAP 55	; 3795
000442	000074		.WORD 74	
000444	000000G		.WORD SK.REV.ERR	
000446	000000		.WORD 0	
000450	016716	000000G	MOV LBN.ST,(SP)	; 3796
000454	016746	000000G	MOV SWP.END,-(SP)	
000460	016746	000000G	MOV SWP.START,-(SP)	
000464	012746	000000G	MOV #FMT9,-(SP)	
000470	012746	0000004	MOV #4,-(SP)	
000474	010600		MOV SP,R0	; SP,*
000476	104414		TRAP 14	
000500	062706	000010	ADD #10,SP	
000504	032767	000001 000000G	BIT #1,RET.STATUS	; 3798
000512	001402		BEQ 12‡	
000514	004767	000000G	JSR PC,DECODE	
000520	012767	000001 000000G	12‡: MOV #1,RETRIES	; 3800
000526	000415		BR 14‡	; 3794
000530	016767	000000G 000000G	13‡: MOV LBN.ED,LBN.ST	; 3804
000536	005201		INC R1	; COUNT 3788
000540	020127	000001	CMP R1,#1	; COUNT,*
000544	101726		BLOS 11‡	
000546	166702	000000G	SUB LBN.SZ,R2	; *,LBN.SA 3808
000552	066767	000000G 000000G	ADD LBN.SZ,LBN.ED	; 3809
000560	000713		BR 10‡	; 3783
000562	062706	000010	14‡: ADD #10,SP	; 3717
000566	032767	000001 000000G	15‡: BIT #1,RETRIES	; 3816
000574	001402		BEQ 16‡	
000576	004767	000000G	JSR PC,DO.RETRIES	
000602	005767	000000G	16‡: TST NUM.RETRIES	; 3818
000606	001402		BEQ 17‡	
000610	000167	177222	JMP 1‡	
000614	000207	17‡:	RTS PC	; 3673

; Routine Size: 199 words, Routine Base: AC#CODE * 14060
; Maximum stack depth per invocation: 14 words

000000	004767	177156	T17::	.SBTTL T17 TEST SECTION	
000000			1‡:	JSR PC,#T17	; 3822
000004	104466			TRAP 66	
000006	006000			ROR R0	
000010	103773			BLO 1‡	
000012	000207			RTS PC	

; Routine Size: 6 words, Routine Base: AC#CODE * 14676
; Maximum stack depth per invocation: 2 words

D7

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0288
Page 93
(17)

: 3824 1 !<BLF/PAGE>

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 B1199-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.816;4SEQ 0289
Page 94
(18)

```

: 3825 1 !
: 3826 3 !
: 3827 3 !
: 3828 3 !
: 3829 3 !
: 3830 3 !
: 3831 3 !
: 3832 3 !
: 3833 3 !
: 3834 3 !
: 3835 3 !
: 3836 3 !
: 3837 3 !
: 3838 3 !
: 3839 3 !
: 3840 3 !
: 3841 3 !
: 3842 3 !
: 3843 3 !
: 3844 3 !
: 3845 3 !
: 3846 3 !
: 3847 3 !
: 3848 3 !
: 3849 3 !
: 3850 3 !
: 3851 3 !
: 3852 3 !
: 3853 3 !
: 3854 3 !
: 3855 3 !
: 3856 3 !
: 3857 3 !
: 3858 3 !
: 3859 3 !
: 3860 3 !
: 3861 3 !
: 3862 3 !
: 3863 3 !
: 3864 4 !
: 3865 4 !
: 3866 4 !
: 3867 4 !
: 3868 4 !
: 3869 4 !
: 3870 5 !
: 3871 5 !
: 3872 5 !
: 3873 5 !
: 3874 5 !
: 3875 5 !
: 3876 5 !
: 3877 4 !
: 3878 4 !
: 3879 5 !
: 3880 5 !
: 3881 5 !

!
BGNTST;

!..
! TEST #18 - TOGGLE SEEK AND VERIFY TEST
!
! DESCRIPTION:
!
! THIS TEST BRINGS RC25 CONTROLLER AND THE UNIT ON LINE AND READY TO
! ACCEPT MSCP COMMANDS.
!
! ONE THOUSAND SEEK COMMANDS WILL BE ISSUED ONE AT A TIME TO TOGGLE
! BETWEEN THE BEGINNING TRACK OF 0 (LBN = 0) AND THE ENDING TRACK
! OF 820 (LBN = 820 * 31).
!
! THIS IS A SINGLE SURFACE TEST. SEEK IS DONE ONLY ON TOP SURFACE
! UNLESS THE OPERATOR CHOSE TO SEEK ON BOTTOM SURFACE BY ANSWERING
! ONE OF THE SOFTWARE QUESTIONS. THE OPERATOR HAS CONTROL OVER THE
! BEGINNING AND ENDING TRACKS, IF DESIRED BY ANSWERING QUESTIONS.
!
! ERROR REPORTS INCLUDE STARTING, ENDING AND DESIRED TRACKS. AFTER
! REPORTING THE FAILURE THE DIAGNOSTIC WILL ABORT THE TEST, UNLESS
! RETRIES IS ENABLED.
!
!..

local
  LBN_SA;

label
  BLOCK1,
  BLOCK2;

if .SWP_TRACE then PRINTF (DBM24);          ! TEST 18

NUM_RETRIES = ZERO;

while (.NUM_RETRIES lequ .SWP_RETRIES) do
  begin
    TIP = 18;
    ! GET AZTEC READY FOR OPERATION

    if AZTEC_READY ()                      ! IF FAILURE REPORT ERROR
    then
      begin
        ERROF (61, AZT_READY_ERR, 0);      !
      end

      if .RET_STATUS then DECODE ();       ! DECODE THE STATUS, IF ANY

      RETRIES = TRUE;                     ! SET RETRIES FLAG
    end
  else
    BLOCK1 :
    begin
      ! PREPARE FOR READ_CMD
      BYTE_COUNT = ZERO;                  ! SET BYTE COUNT TO ZERO
    end
  end
end

```

```

: 3882 5      BUF_DESCPTR = ZERO;          ! CLEAN THE BUFFER
: 3883 5      !
: 3884 5      ! SEEK BETWEEN BEGINNING TRACK AND ENDING TRACK
: 3885 5      !
: 3886 5      LBN_SA = (.SWP_START*.SIZ_LBN); ! FIND STARTING LBN
: 3887 5      LBN_ED = (.SWP_END*.SIZ_LBN); ! FIND ENDING LBN NUMBER
: 3888 5      TIP = ZERO;                 ! CLEAR COUNTER TO ZERO
: 3889 5      BLOCK2 :
: 3890 6      begin
: 3891 6
: 3892 6      while .TIP lequ 500 do      ! DO SEEK
: 3893 7      begin                       ! TO LOWEST LBN
: 3894 7      LBN_ST = .LBN_SA;          ! BEGINNING TRACK LBN
: 3895 7
: 3896 7      incru COUNT from 0 to 1 do
: 3897 8      begin
: 3898 8      CMD_REF = .LBN_ST;         ! PUT LBN IN CMD_LREF
: 3899 8
: 3900 8      if READ_CMD ( )            ! ISSUE A SEEK COMMAND
: 3901 8      then
: 3902 9      begin
: 3903 9      ERRDF (62, SK_TOG_ERR, 0);
: 3904 9      PRINTB (FMT9, .SWP_START, .SWP_END, .LBN_ST);
: 3905 9
: 3906 9      if .RET_STATUS then DECODE ( ); ! DECODE STATUS OF READ_CMD
: 3907 9
: 3908 9      RETRIES = TRUE;
: 3909 9      leave BLOCK2;              ! ABORT SEEK
: 3910 9      end
: 3911 8      else
: 3912 8      LBN_ST = .LBN_ED;          ! HIGHEST TRACK NUMBER
: 3913 8
: 3914 7      end;
: 3915 7
: 3916 7      TIP = .TIP + 1;            ! INCREMENT COUNTER
: 3917 6      end;                       ! AND REPEAT 500 TIMES
: 3918 6
: 3919 5      end;
: 3920 4      end;
: 3921 4
: 3922 4      if (.RETRIES) then DO_RETRIES ( );
: 3923 4
: 3924 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 3925 4
: 3926 3      end;
: 3927 3
: 3928 3      return;
: 3929 1      ENDTST;

```

```

000000 004167 000000G      .SBTTL $T18 TEST SECTION
000004 032767 000001 000000G      $T18: JSR R1,$SAVE2 ; 3823
000012 001407              BIT #1,SWP.TRACE ; 3859
000014 012746 000000G      BEQ 1$
000020 012746 000001      MOV #DBM24,-(SP)
000024 010600              MOV #1,-(SP)
                                MOV SP,R0 ; SP,*

```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

000026	104417				TRAP	17				
000030	022626				CMP	(SP)*,(SP)*				
000032	005067	000000G		1\$:	CLR	NUM.RETRIES				3861
000036	026767	000000G	000000G	2\$:	CMP	NUM.RETRIES,SWP.RETRIES				3863
000044	101151				BHI	12\$				
000046	012767	000022	000000G		MOV	#22,TIP				3865
000054	004767	000000G			JSR	PC,AZTEC.READY				3868
000060	006000				ROR	RO				
000062	103016				BCC	4\$				
000064	104455				TRAP	55				3871
000066	000075				.WORD	75				
000070	000000G				.WORD	AZT.READY.ERR				
000072	000000				.WORD	0				
000074	032767	000001	000000G		BIT	#1,RET.STATUS				3873
000102	001402				BEQ	3\$				
000104	004767	000000G			JSR	PC,DECODE				
000110	012767	000001	000000G	3\$:	MOV	#1,RETRIES				3875
000116	000513				BR	10\$				3868
000120	005067	000000G		4\$:	CLR	BYTE.COUNT				3881
000124	005067	000000G			CLR	BUF.DESCRPTR				3882
000130	016746	000000G			MOV	SWP.START,-(SP)				3886
000134	016746	000000G			MOV	SIZ.LBN,-(SP)				
000140	004767	000000G			JSR	PC,BL\$MUL				
000144	010002				MOV	RO,R2				
000146	016716	000000G			MOV	SWP.END,(SP)				3887
000152	016746	000000G			MOV	SIZ.LBN,-(SP)				
000156	004767	000000G			JSR	PC,BL\$MUL				
000162	010067	000000G			MOV	RO,LBN.ED				
000166	005067	000000G			CLR	TIP				3888
000172	026727	000000G	000764	5\$:	CMP	TIP,#764				3892
000200	101060				BHI	9\$				
000202	010267	000000G			MOV	R2,LBN.ST				3894
000206	005001				CLR	R1				3896
000210	016767	000000G	000000G	6\$:	MOV	LBN.ST,CMD.REF				3898
000216	004767	000000G			JSR	PC,READ.CMD				3900
000222	006000				ROR	RO				
000224	103034				BCC	8\$				
000226	104455				TRAP	55				3903
000230	000076				.WORD	76				
000232	000000G				.WORD	SK.TOG.ERR				
000234	000000				.WORD	0				
000236	016716	000000G			MOV	LBN.ST,(SP)				3904
000242	016746	000000G			MOV	SWP.END,-(SP)				
000246	016746	000000G			MOV	SWP.START,-(SP)				
000252	012746	000000G			MOV	#FMT9,-(SP)				
000256	012746	000004			MOV	#4,-(SP)				
000262	010600				MOV	SP,RO				
000264	104414				TRAP	14				
000266	062706	000010			ADD	#10,SP				
000272	032767	000001	000000G		BIT	#1,RET.STATUS				3906
000300	001402				BEQ	7\$				
000302	004767	000000G			JSR	PC,DECODE				
000306	012767	000001	000000G	7\$:	MOV	#1,RETRIES				3908
000314	000412				BR	9\$				3902
000316	016767	000000G	000000G	8\$:	MOV	LBN.ED,LBN.ST				3912
000324	005201				INC	R1				3896
000326	020127	000001			CMP	R1,#1				

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

000332	101726			BLOS	6\$		
000334	005267	000000G		INC	TIP	:	3916
000340	000714			BR	5\$:	3892
000342	062706	000006	9\$:	ADD	#6,SP	:	3868
000346	032767	000001 000000G	10\$:	BIT	#1,RETRIES	:	3922
000354	001402			BEQ	11\$		
000356	004767	000000G		JSR	PC,DO.RETRIES		
000362	005767	000000G	11\$:	TST	NUM.RETRIES	:	3924
000366	001223			BNE	2\$		
000370	000207		12\$:	RTS	PC	:	3823

; Routine Size: 125 words, Routine Base: AC\$CODE * 14712
; Maximum stack depth per invocation: 12 words

000000	004767	177402		.SBTTL	T18 TEST SECTION		
000000			T18::				
000004	104466		1\$:	JSR	PC,\$T18	:	3928
000006	006000			TRAP	66		
000010	103773			ROR	RO		
000012	000207			BLO	1\$		
				RTS	PC		

; Routine Size: 6 words, Routine Base: AC\$CODE * 15304
; Maximum stack depth per invocation: 2 words

; 3930 1 !<BLF/PAGE>

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0293
Page 98
(19)

```

: 3931 1 !
: 3932 3 !
: 3933 3 !
: 3934 3 !
: 3935 3 !
: 3936 3 !
: 3937 3 !
: 3938 3 !
: 3939 3 !
: 3940 3 !
: 3941 3 !
: 3942 3 !
: 3943 3 !
: 3944 3 !
: 3945 3 !
: 3946 3 !
: 3947 3 !
: 3948 3 !
: 3949 3 !
: 3950 3 !
: 3951 3 !
: 3952 3 !
: 3953 3 !
: 3954 3 !
: 3955 3 !
: 3956 3 !
: 3957 3 !
: 3958 3 !
: 3959 3 !
: 3960 3 !
: 3961 3 !
: 3962 3 !
: 3963 4 !
: 3964 4 !
: 3965 4 !
: 3966 4 !
: 3967 4 !
: 3968 4 !
: 3969 5 !
: 3970 5 !
: 3971 5 !
: 3972 5 !
: 3973 5 !
: 3974 5 !
: 3975 5 !
: 3976 4 !
: 3977 4 !
: 3978 5 !
: 3979 5 !
: 3980 5 !
: 3981 5 !
: 3982 5 !
: 3983 5 !
: 3984 5 !
: 3985 5 !
: 3986 5 !
: 3987 5 !

!
! BGNTST;
!
! **
! TEST #19 - HEAD SWITCH TEST
!
! DESCRIPTION:
!
! THIS TEST WILL BRING RC25 CONTROLLER AND THE UNIT ONLINE
! AND WILL LOAD DM CODE PROGRAM TO CONTROLLER'S MEMORY USING
! EX_SUP_PROG COMMAND.
!
! DM CODE WILL SEEK TO BOTH SURFACES OF THE UNIT. THE XFC STATUS
! WILL BE USED TO VERIFY THAT THE PROPER TRACK HAS BEEN REACHED.
! BLOCK HEADERS WILL BE READ TO VERIFY THAT THE PROPER HEADS ARE
! SELECTED. DM CODE WILL RETRY IF THERE WAS ANY ERROR IN SEEK.
! DM CODE WILL GIVE SUCCESS OR FAILURE CODE TO THE HOST.
!
! IF FAILURE, THE TRACK, HEAD AND UNIT WILL BE REPORTED AS RECEIVED
! FROM DM CODE.
!
! IF RETRIES ARE TURNED ON THE TEST WILL BE REPEATED.
!
! --
label
  BLOCK1:
  if .SWP_TRACE then PRINTF (DBM25);          ! TEST 19
  NUM_RETRIES = ZERO;
  while (.NUM_RETRIES lequ .SWP_RETRIES) do
    begin
      TIP = 19;
      ! GET AZTEC READY FOR OPERATION
      if AZTEC_READY ()                       ! IF FAILURE REPORT ERROR
      then
        begin
          ERRDF (63, AZT_READY_ERR, 0);      !
          if .RET_STATUS then DECODE ();     ! DECODE THE STATUS, IF ANY
          RETRIES = TRUE;                    ! SET RETRIES FLAG
        end
      else
        BLOCK1 :
          begin
            !
            ! ISSUE AN EX_SUP_PROG COMMAND WITH START ADDRESS OF
            ! DM_19 VECTOR ARRAY AND BYTE COUNT.
            CMD_REF = .CMD_SLOT;             ! COMMAND REFERENCE NUMBER
            BUF_DESCRPT = DM_19;             ! DMCODE STARTING ADDRESS
            BYTE_COUNT = 156*2;              ! BYTE COUNTS
            if EX_SUP_PRG ()                  ! ISSUE AN EXECUTE SUPPLIED COMMAND
            then                               ! REPORT IF FAILED

```



```

: 3988 6      begin
: 3989 6      ERRDF (64, EXE_SUP_ERR, 0);
: 3990 6
: 3991 6      if .RET_STATUS then DECODE ();      ! DECODE STATUS
: 3992 6
: 3993 6      RETRIES = TRUE;
: 3994 6      leave BLOCK1;                          ! ABORT TEST
: 3995 6      end;
: 3996 5
: 3997 5      CMD_REF = .CMD_SLOT;                    ! COMMAND REFERENCE 04
: 3998 5      BUF_DESCRPT = UNIT;                    ! DESCRIPTOR ADDRESS
: 3999 5      BYTE_COUNT = 02;                       ! TOTAL BYTES TO BE TRANSFERRED
: 4000 5
: 4001 5      if SEND_DATA ()                        ! ISSUE SEND DATA COMMAND
: 4002 5      then                                  ! IF STATUS BIT INDICATES ERROR
: 4003 6          begin                              ! THEN REPORT ERROR
: 4004 6          ERRDF (65, SND_DATA_ERR, 0);      !
: 4005 6
: 4006 6          if .RET_STATUS then DECODE ();    ! DECODE RETURN STATUS
: 4007 6
: 4008 6          RETRIES = TRUE;
: 4009 6          leave BLOCK1;
: 4010 6          end;
: 4011 5
: 4012 5      ! ISSUE A REC_DATA COMMAND AND WAIT FOR END PACKET
: 4013 5      ! TO GET THE STATUS SENT BY DM CODE AFTER DOING
: 4014 5      ! HEAD SWITCH TEST.
: 4015 5      CMD_REF = .CMD_SLOT;                    ! COMMAND REFERENCE #
: 4016 5      BUF_DESCRPT = RCV_DATA_BUF [0];        ! SET THE BUFFER AREA TO
: 4017 5      ! RECEIVE 5 WORDS FROM DM CODE
: 4018 5      BYTE_COUNT = 10;                        ! SET BYTE COUNTS = 10
: 4019 5
: 4020 5      if REC_DATA ()                          ! SEND A RECEIVE DATA COMMAND
: 4021 5      then                                  ! IF FAILURE REPORT ERROR
: 4022 6          begin
: 4023 6          ERRDF (66, RE_DATA_ERR, 0);
: 4024 6
: 4025 6          if .RET_STATUS then DECODE ();    ! DECODE STATUS
: 4026 6
: 4027 6          RETRIES = TRUE;
: 4028 6          leave BLOCK1;
: 4029 6          end;
: 4030 5
: 4031 5      ! CHECK DM CODE FLAG FOR SUCCESS. IF FAILURE REPORT ERROR
: 4032 5
: 4033 5      if .RCV_DATA_BUF [0] nequ #o'104'      ! IF NOT SUCCESS, REPORT ERROR
: 4034 5      then
: 4035 6          begin
: 4036 6          ERRDF (67, MSG_HSWICH_ERR, 0);    ! REPORT HEAD SWITCH FAILURE
: 4037 6          PRINTB (FMT10, .RCV_DATA_BUF [1], .RCV_DATA_BUF [2], .RCV_DATA_BUF [3]); ! PRINT UNIT, HEAD AND
: 4038 6          ! TRACK NUMBER
: 4039 6          RETRIES = TRUE;
: 4040 6          end;
: 4041 5
: 4042 4      end;
: 4043 4
: 4044 4      if (.RETRIES) then DO_RETRIES ();

```

```

: 4045 4
: 4046 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 4047 4
: 4048 3      end;
: 4049 3
: 4050 3      return;
: 4051 1      ENDTST;

```

Address	Hex	OpCode	OpCode	Label	Instruction	Comment	Line
000000	032767	000001	000000G	\$T19:	.SBTTL \$T19 TEST SECTION		
000006	001407				BIT #1,SWP.TRACE		3958
000010	012746	000000G			BEQ 1\$		
000014	012746	000001			MOV #DBM25,-(SP)		
000020	010600				MOV #1,-(SP)		
000022	104417				MOV SP,R0	; SP,*	
000024	022626				TRAP 17		
000026	005067	000000G		1\$:	CMP (SP),(SP)		
000032	026767	000000G	000000G	2\$:	CLR NUM.RETRIES		3960
000040	101401				CMP NUM.RETRIES,SWP.RETRIES		3962
000042	000207				BLOS 3\$		
000044	012767	000023	000000G	3\$:	RTS PC		
000052	004767	000000G			MOV #23,TIP		3964
000056	006000				JSR PC,AZTEC.READY		3967
000060	103016				ROR R0		
000062	104455				BCC 5\$		
000064	000077				TRAP 55		3970
000066	000000G				.WORD 77		
000070	000000				.WORD AZT.READY.ERR		
000072	032767	000001	000000G		.WORD 0		
000100	001402				BIT #1,RET.STATUS		3972
000102	004767	000000G			BEQ 4\$		
000106	012767	000001	000000G	4\$:	JSR PC,DECODE		
000114	000552				MOV #1,RETRIES		3974
000116	016767	000000G	000000G	5\$:	BR 12\$		3967
000124	012767	000000G	000000G		MOV CMD.SLOT,CMD.REF		3982
000132	012767	000000G	000000G		MOV #DM.19,BUF.DESCRPTR		3983
000140	012767	000470	000000G		MOV #470,BYTE.COUNT		3984
000144	004767	000000G			JSR PC,EX.SUP.PRG		3986
000146	006000				ROR R0		
000148	103016				BCC 7\$		
000150	104455				TRAP 55		3989
000152	000100				.WORD 100		
000154	000000G				.WORD EXE.SUP.ERR		
000156	000000				.WORD 0		
000160	032767	000001	000000G		BIT #1,RET.STATUS		3991
000166	001402				BEQ 6\$		
000170	004767	000000G			JSR PC,DECODE		
000174	012767	000001	000000G	6\$:	MOV #1,RETRIES		3993
000202	000517				BR 12\$		3988
000204	016767	000000G	000000G	7\$:	MOV CMD.SLOT,CMD.REF		3997
000212	012767	000000G	000000G		MOV #UNIT,BUF.DESCRPTR		3998
000220	012767	000002	000000G		MOV #2,BYTE.COUNT		3999
000226	004767	000000G			JSR PC,SEND.DATA		4001
000232	006000				ROR R0		
000234	103016				BCC 9\$		
000236	104455				TRAP 55		4004
000240	000101				.WORD 101		

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0296
Page 101
(19)

000242	000000G			.WORD	SND.DATA.ERR			
000244	000000			.WORD	0			
000246	032767	000001	000000G	BIT	#1,RET.STATUS	:	4006	
000254	001402			BEQ	8#			
000256	004767	000000G		JSR	PC.DECODE			
000262	012767	000001	000000G	MOV	#1,RETRIES	:	4008	
000270	000464			BR	12#	:	4003	
000272	016767	000000G	000000G	MOV	CMD.SLOT,CMD.REF	:	4015	
000300	012767	000000G	000000G	MOV	#RCV.DATA.BUF,BUF.DESCRPTR	:	4016	
000306	012767	000012	000000G	MOV	#12,BYTE.COUNT	:	4018	
000314	004767	000000G		JSR	PC.REC.DATA	:	4020	
000320	006000			ROR	R0			
000322	103016			BCC	11#			
000324	104455			TRAP	55	:	4023	
000326	000102			.WORD	102			
000330	000000G			.WORD	RE.DATA.ERR			
000332	000000			.WORD	0			
000334	032767	000001	000000G	BIT	#1,RET.STATUS	:	4025	
000342	001402			BEQ	10#			
000344	004767	000000G		JSR	PC.DECODE			
000350	012767	000001	000000G	MOV	#1,RETRIES	:	4027	
000356	000431			BR	12#	:	4022	
000360	026727	000000G	000104	11#:	CMP	RCV.DATA.BUF,#104	:	4033
000366	001425			BEQ	12#			
000370	104455			TRAP	55	:	4036	
000372	000103			.WORD	103			
000374	000000G			.WORD	MSG.HSWICH.ERR			
000376	000000			.WORD	0			
000400	016746	000006G		MOV	RCV.DATA.BUF*6,-(SP)	:	4037	
000404	016746	000004G		MOV	RCV.DATA.BUF*4,-(SP)			
000410	016746	000002G		MOV	RCV.DATA.BUF*2,-(SP)			
000414	012746	000000G		MOV	#FMT10,-(SP)			
000420	012746	000004		MOV	#4,-(SP)			
000424	010600			MOV	SP,R0	:	SP.*	
000426	104414			TRAP	14			
000430	012767	000001	000000G	MOV	#1,RETRIES	:	4039	
000436	062706	000012		ADD	#12,SP	:	4035	
000442	032767	000001	000000G	12#:	BIT	#1,RETRIES	:	4044
000450	001402			BEQ	13#			
000452	004767	000000G		JSR	PC,DO.RETRIES			
000456	005767	000000G		13#:	TST	NUM.RETRIES	:	4046
000462	001402			BEQ	14#			
000464	000167	177342		JMP	2#			
000470	000207			14#:	RTS	PC	:	3929

: Routine Size: 157 words, Routine Base: AC\$CODE * 15320
: Maximum stack depth per invocation: 7 words

000000	004767	177302		.SBTTL	T19 TEST SECTION		
000000			T19::				
000004	104466		1#:	JSR	PC,\$T19	:	4050
000006	006000			TRAP	66		
000010	103773			ROR	R0		
				BLO	1#		

M7

ZRCFR3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B1100-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0297
Page 102
(19)

000012 000207

RTS PC

: Routine Size: 6 words. Routine Base: AC\$CODE * 16012
: Maximum stack depth per invocation: 2 words

: 4052 1 !<BLF/PAGE>

```

: 4053 1
: 4054 1
: 4055 1
: 4056 1
: 4057 1
: 4058 1
: 4059 1
: 4060 1
: 4061 1
: 4062 1
: 4063 1
: 4064 1
: 4065 1
: 4066 1
: 4067 1
: 4068 1
: 4069 1
: 4070 1
: 4071 1
: 4072 1
: 4073 1
: 4074 1
: 4075 1
: 4076 1
: 4077 1
: 4078 1
: 4079 3
: 4080 4
: 4081 4
: 4082 4
: 4083 4
: 4084 4
: 4085 5
: 4086 5
: 4087 5
: 4088 5
: 4089 5
: 4090 5
: 4091 5
: 4092 4
: 4093 4
: 4094 3
: 4095 3
: 4096 3
: 4097 5
: 4098 5
: 4099 5
: 4100 5
: 4101 3
: 4102 6
: 4103 6
: 4104 6
: 4105 6
: 4106 6
: 4107 6
: 4108 6
: 4109 7

```

```

!
BGNTST;
!..
! TEST #20 - RANDOM SEEK AND VERIFY TEST
!
! DESCRIPTION:
!
! THIS TEST BRINGS RC25 CONTROLLER AND THE SELECTED UNIT ONLINE
! AND THEN ISSUES 1000 SEEKS ONE AT A TIME TO RANDOMLY SELECTED
! LBN TRACKS BETWEEN THE RANGE OF 0 - 1641. THIS WILL ENSURE HEAD
! SWITCH AS WELL BECAUSE TRACKS OVER 820 WILL BE IN THE BOTTOM
! SURFACE OF SELECTED UNIT.
!
! ERROR REPORTS INCLUDE SEEK COUNT AND FAILING TRACK NUMBER. IF
! LOOP ON ERROR FLAG IS SET, FAILING TRACK WILL BE RETRIED FOR EVER.
!..
label
  BLOCK1:
  if .SWP_TRACE then PRINTF (DBM26);          ! TEST 20
  NUM_RETRIES = ZERO;
  while (.NUM_RETRIES lequ .SWP_RETRIES) do
    begin
    ! GET AZTEC READY FOR OPERATION
    if AZTEC_READY ()                          ! IF FAILURE REPORT ERROR
    then
      begin
      ERRDF (68, AZT_READY_ERR, 0);           !
      if .RET_STATUS then DECODE ();          ! DECODE THE STATUS, IF ANY
      RETRIES = TRUE;                         ! SET RETRIES FLAG
      end
    else
    BLOCK1 :
      begin
      !
      ! SET BYTE COUNT TO ZERO
      ! CLEAN THE BUFFER
      ! INIT P2 FOR RANDOM NUMBER
      ! INIT COUNTER TO ONE
      BYTE_COUNT = ZERO;
      BUF_DESCRIPTOR = ZERO;
      P2 = .TICKS;
      TIP = ONE;
      ! DO SEEK 1000 TIMES
      while .TIP lequ 1000 do
        begin
        ! GET A RANDOM LBN NUMBER
        ! THIS IS IT
        ! PUT LBN IN CMD_LREF
        ! ISSUE A SEEK COMMAND
        RANDOM_NUM ();
        LBN_ST = .P3;
        CMD_REF = .LBN_ST;
        if READ_CMD ()
        then
          begin

```

```

: 4110 7      ERRDF (69, SK_RAN_ERR, 0);
: 4111 7      PRINTB (FMT11, .TIP, .LBN_ST);
: 4112 7
: 4113 7      if .RET_STATUS then DECODE (); ! DECODE STATUS OF READ_CMD
: 4114 7
: 4115 7      RETRIES = TRUE;
: 4116 7      leave BLOCK1; ! ABORT SEEK
: 4117 7      end
: 4118 6      else
: 4119 6      TIP = .TIP + 1; ! INCREMENT COUNTER
: 4120 6
: 4121 5      end;
: 4122 5
: 4123 4      end;
: 4124 4
: 4125 4      if (.RETRIES) then DO_RETRIES ();
: 4126 4
: 4127 4      if (.NUM_RETRIES eqv ZERO) then exitloop;
: 4128 4
: 4129 3      end;
: 4130 3
: 4131 3      return;
: 4132 1      ENDTST;

```

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comment	Address
000000	032767	000001	000000G	\$T20:	.SBTTL	\$T20 TEST SECTION	
000006	001407				BIT	#1,SWP.TRACE	4075
000010	012746	000000G			BEQ	1\$	
000014	012746	000001			MOV	#DBM26,-(SP)	
000020	010600				MOV	#1,-(SP)	
000022	104417				MOV	SP,R0	; SP,*
000024	022626				TRAP	17	
000026	005067	000000G			CMP	(SP)*,(SP)*	
000032	026767	000000G	000000G	1\$:	CLR	NUM.RETRIES	4077
000040	101122			2\$:	CMP	NUM.RETRIES,SWP.RETRIES	4079
000042	004767	000000G			BHI	10\$	
000046	006000				JSR	PC,AZTEC.READY	4083
000050	103016				ROR	R0	
000052	104455				BCC	4\$	
000054	000104				TRAP	55	4086
000056	000000G				.WORD	104	
000060	000000				.WORD	AZT.READY.ERR	
000062	032767	000001	000000G		.WORD	0	
000070	001402				BIT	#1,RET.STATUS	4088
000072	004767	000000G			BEQ	3\$	
000076	012767	000001	000000G	3\$:	JSR	PC,DECODE	
000104	000467				MOV	#1,RETRIES	4090
000106	005067	000000G			BR	8\$	4083
000112	005067	000000G		4\$:	CLR	BYTE.COUNT	4096
000116	016767	000000G	000000G		CLR	BUF.DESCPTR	4097
000124	012767	000001	000000G		MOV	TICKS,P2	4098
000132	026727	000000G	001750	5\$:	MOV	#1,TIP	4099
000140	101051				CMP	TIP,#1750	4101
000142	004767	000000G			BHI	8\$	
000146	016767	000000G	000000G		JSR	PC,RANDOM.NUM	4103
000154	016767	000000G	000000G		MOV	P3,LBN.ST	4104
					MOV	LBN.ST,CMD.REF	4105

```

ZRCFB3          CZRCFC0 RC25 FR END TEST          27-Mar-1985 15:27:28    VAX-11 Bliss-16 V4.0-579    SEQ 0300
V03.0          TEST SECTION                       27-Mar-1985 13:28:18    USER#1:[AZTEC.CZRCFC]ZRCFC3.816;4    Page 105
                                                    (20)

000162 004767 000000G      JSR    PC,READ.CMD      ;      4107
000166 006000      ROR    RO
000170 103032      BCC    7#
000172 104455      TRAP   55              ;      4110
000174 000105      .WORD 105
000176 000000G      .WORD SK,RAN.ERR
000200 000000      .WORD 0
000202 016746 000000G      MOV    LBN,ST,-(SP)    ;      4111
000206 016746 000000G      MOV    TIP,-(SP)
000212 012746 000000G      MOV    #FMT11,-(SP)
000216 012746 0000003      MOV    #3,-(SP)
000222 010600      MOV    SP,RO          ; SP,*
000224 104414      TRAP   14
000226 062706 000010      ADD    #10,SP
000232 032767 000001 000000G    BIT    #1,RET.STATUS  ;      4113
000240 001402      BEQ    6#
000242 004767 000000G      JSR    PC,DECODE
000246 012767 000001 000000G    6#:   MOV    #1,RETRIES  ;      4115
000254 000403      BR     8#              ;      4109
000256 005267 000000G      7#:   INC    TIP          ;      4119
000262 000723      BR     5#              ;      4101
000264 032767 000001 000000G    8#:   BIT    #1,RETRIES  ;      4125
000272 001402      BEQ    9#
000274 004767 000000G      JSR    PC,DO.RETRIES
000300 005767 000000G      9#:   TST    NUM.RETRIES  ;      4127
000304 001252      BNE    2#
000306 000207      10#:  RTS    PC          ;      4051
    
```

```

; Routine Size: 100 words,      Routine Base: AC#CODE * 16026
; Maximum stack depth per invocation: 6 words
    
```

```

000000 004767 177464      T20:: .SBTTL T20 TEST SECTION
000000 1#:   JSR    PC,#T20      ;      4131
000004 104466      TRAP   66
000006 006000      ROR    RO
000010 103773      BLO    1#
000012 000207      RTS    PC
    
```

```

; Routine Size: 6 words,      Routine Base: AC#CODE * 16336
; Maximum stack depth per invocation: 2 words
    
```

```

; 4133 1 !<BLF/PAGE>
    
```

```

: 4134 1 !
: 4135 3 !
: 4136 3 !
: 4137 3 !
: 4138 3 !
: 4139 3 !
: 4140 3 !
: 4141 3 !
: 4142 3 !
: 4143 3 !
: 4144 3 !
: 4145 3 !
: 4146 3 !
: 4147 3 !
: 4148 3 !
: 4149 3 !
: 4150 3 !
: 4151 3 !
: 4152 3 !
: 4153 3 !
: 4154 3 !
: 4155 3 !
: 4156 3 !
: 4157 3 !
: 4158 3 !
: 4159 3 !
: 4160 3 !
: 4161 3 !
: 4162 3 !
: 4163 3 !
: 4164 3 !
: 4165 3 !
: 4166 4 !
: 4167 4 !
: 4168 4 !
: 4169 4 !
: 4170 4 !
: 4171 4 !
: 4172 5 !
: 4173 5 !
: 4174 5 !
: 4175 5 !
: 4176 5 !
: 4177 5 !
: 4178 5 !
: 4179 4 !
: 4180 4 !
: 4181 5 !
: 4182 5 !
: 4183 5 !
: 4184 5 !
: 4185 5 !
: 4186 5 !
: 4187 5 !
: 4188 5 !
: 4189 5 !
: 4190 5 !

!
! BGNTST;
!
! **
! TEST #21 - SECTOR ACCESS TEST
!
! DESCRIPTION:
!
! THIS TEST BRINGS RC25 CONTROLLER AND SELECTED UNIT ONLINE
! AND THEN LOADS DM 21 VECTOR ARRAY INTO CONTROLLER'S MEMORY
! BY GIVING EX_SUP_PROG COMMAND.
!
! THE DM PROGRAM WILL SEEK TO DIAGNOSTIC TRACK 0 AND READ 32
! BLOCKS AFTER MAKING SURE THAT GOOD HEADER IS FOUND. DM CODE
! WILL RETRY IF ANY ERROR WAS FOUND. DM CODE WILL SEND STATUS BACK
! TO HOST WITH FAILING UNIT, HEAD AND TRACK. ERROR WILL BE REPORTED
! BY HOST CODE.
!
! THIS IS A SINGLE SURFACE TEST. TOP SURFACE WILL BE ACCESSED
! UNLESS THE OPERATOR CHOSE BOTTOM SURFACE BY ANSWERING ONE OF
! THE SOFTWARE QUESTIONS.
!
! ---
!
label
  BLOCK1;
!
if .SWP_TRACE then PRINTF (DBM27);          ! TEST 21
!
NUM_RETRIES = ZERO;
!
while (.NUM_RETRIES lequ .SWP_RETRIES) do
  begin
    TIP = 21;
! GET AZTEC READY FOR OPERATION
!
    if AZTEC_READY ()                        ! IF FAILURE REPORT ERROR
    then
      begin
        ERRDF (70, AZT_READY_ERR, 0);      !
!
        if .RET_STATUS then DECODE ();     ! DECODE THE STATUS, IF ANY
!
        RETRIES = TRUE;                    ! SET RETRIES FLAG
      end
    else
      BLOCK1 :
        begin
!
! ISSUE AN EX_SUP_PROG COMMAND WITH START ADDRESS OF
! DM_21 VECTOR ARRAY AND BYTE COUNT.
!
          CMD_REF = .CMD_SLOT;              ! COMMAND REFERENCE NUMBER
          BUF_DESCRPT = DM_21;              ! DM CODE STARTING ADDRESS
          BYTE_COUNT = 213*2;               ! BYTE COUNTS
!
          if EX_SUP_PRG ()                  ! ISSUE AN EXECUTE SUPPLIED COMMAND
          then                              ! REPORT IF FAILED

```



```

: 4191 6      begin
: 4192 6      ERRDF (71, EXE_SUP_ERR, 0);
: 4193 6
: 4194 6      if .RET_STATUS then DECODE ();      ! DECODE STATUS
: 4195 6
: 4196 6      RETRIES = TRUE;
: 4197 6      leave BLOCK1;                          ! ABORT TEST
: 4198 5      end;
: 4199 5
: 4200 5      SEND_PKT [WORD0] = .UNIT;
: 4201 5      SEND_PKT [WORD1] = .SWP_TOP;
: 4202 5      CMD_REF = .CMD_SLOT;                  ! COMMAND REFERENCE 04
: 4203 5      BUF_DESCRPT = SEND_PKT;              ! DESCRIPTOR ADDRESS
: 4204 5      BYTE_COUNT = 04;                     ! TOTAL BYTES TO BE TRANSFERRED
: 4205 5
: 4206 5      if SEND_DATA ()                       ! ISSUE SEND DATA COMMAND
: 4207 5      then                                  ! IF STATUS BIT INDICATES ERROR
: 4208 6          begin                             ! THEN REPORT ERROR
: 4209 6          ERRDF (72, SND_DATA_ERR, 0);      !
: 4210 6
: 4211 6          if .RET_STATUS then DECODE ();    ! DECODE RETURN STATUS
: 4212 6
: 4213 6          RETRIES = TRUE;
: 4214 6          leave BLOCK1;
: 4215 5          end;
: 4216 5
: 4217 5      ! ISSUE A REC_DATA COMMAND AND WAIT FOR END PACKET
: 4218 5      ! TO GET THE STATUS SENT BY DM CODE AFTER DOING
: 4219 5      ! SECTOR ACCESS TEST.
: 4220 5      CMD_REF = .CMD_SLOT;                  ! COMMAND REFERENCE #
: 4221 5      BUF_DESCRPT = RCV_DATA_BUF [0];      ! SET THE BUFFER AREA TO
: 4222 5                                          ! RECEIVE 5 WORDS FROM DM CODE
: 4223 5      BYTE_COUNT = 10;                     ! SET BYTE COUNTS = 10
: 4224 5
: 4225 5      if REC_DATA ()                       ! SEND A RECEIVE DATA COMMAND
: 4226 5      then                                  ! IF FAILURE REPORT ERROR
: 4227 6          begin
: 4228 6          ERRDF (73, RE_DATA_ERR, 0);
: 4229 6
: 4230 6          if .RET_STATUS then DECODE ();    ! DECODE STATUS
: 4231 6
: 4232 6          RETRIES = TRUE;
: 4233 6          leave BLOCK1;
: 4234 5          end;
: 4235 5
: 4236 5      ! CHECK DM CODE FLAG FOR SUCCESS. IF FAILURE REPORT ERROR
: 4237 5
: 4238 5      if .RCV_DATA_BUF [0] nequ %o'104'    ! IF NOT SUCCESS, REPORT ERROR
: 4239 5      then
: 4240 6          begin
: 4241 6          ERRDF (74, MSG_SAC_ERR, 0);        ! REPORT SECTOR ACCESS FAILURE
: 4242 6          PRINTB (FMT10, .RCV_DATA_BUF [1], .RCV_DATA_BUF [2], .RCV_DATA_BUF [3]); ! PRINT UNIT, HEAD AND
: 4243 6                                          ! TRACK NUMBER
: 4244 6          RETRIES = TRUE;
: 4245 5          end;
: 4246 5
: 4247 4      end;

```

```

: 4248 4
: 4249 4      if (.RETRIES) then DO_RETRIES ();
: 4250 4
: 4251 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 4252 4
: 4253 3      end;
: 4254 3
: 4255 3      return;
: 4256 1      ENDTST;
    
```

```

000000 032767 000001 000000G      $T21: .SBTTL $T21 TEST SECTION
000006 001407                      BIT #1,SWP.TRACE ; 4161
000010 012746 000000G              BEQ 1$ ;
000014 012746 000001              MOV #DBM27,-(SP)
000020 010600                      MOV #1,-(SP)
000022 104417                      MOV SP,R0 ; SP,*
000024 022626                      TRAP 17
000026 005067 000000G              CMP (SP),(SP).
000032 026767 000000G 000000G    1$: CLR NUM.RETRIES ; 4163
000040 101401                      2$: CMP NUM.RETRIES,SWP.RETRIES ; 4165
000042 000207                      BLOS 3$
000044 012767 000025 000000G      3$: RTS PC
000052 004767 000000G              MOV #25,TIP ; 4167
000056 006000                      JSR PC,AZTEC.READY ; 4170
000060 103016                      ROR R0
000062 104455                      BCC 5$
000064 000106                      TRAP 55 ; 4173
000066 000000G                    .WORD 106
000070 000000                    .WORD AZT.READY.ERR
000072 032767 000001 000000G      BIT #1,RET.STATUS ; 4175
000100 001402                      BEQ 4$
000102 004767 000000G              JSR PC,DECODE
000106 012767 000001 000000G      4$: MOV #1,RETRIES ; 4177
000114 000560                      BR 12$ ; 4170
000116 016767 000000G 000000G    5$: MOV CMD.SLOT,CMD.REF ; 4185
000124 012767 000000G 000000G    MOV #DM.21,BUF.DESCRPTR ; 4186
000132 012767 000652 000000G    MOV #652,BYTE.COUNT ; 4187
000140 004767 000000G              JSR PC,EX.SUP.PRG ; 4189
000144 006000                      ROR R0
000146 103016                      BCC 7$
000150 104455                      TRAP 55 ; 4192
000152 000107                    .WORD 107
000154 000000G                    .WORD EXE.SUP.ERR
000156 000000                    .WORD 0
000160 032767 000001 000000G      BIT #1,RET.STATUS ; 4194
000166 001402                      BEQ 6$
000170 004767 000000G              JSR PC,DECODE
000174 012767 000001 000000G      6$: MOV #1,RETRIES ; 4196
000202 000525                      BR 12$ ; 4191
000204 016767 000000G 000230'    7$: MOV UNIT,SEND.PKT ; 4200
000212 016767 000000G 000232'    MOV SWP.TOP,SEND.PKT+2 ; 4201
000220 016767 000000G 000000G    MOV CMD.SLOT,CMD.REF ; 4202
000226 012767 000230' 000000G    MOV #SEND.PKT,BUF.DESCRPTR ; 4203
000234 012767 000004 000000G    MOV #4,BYTE.COUNT ; 4204
000242 004767 000000G              JSR PC,SEND.DATA ; 4206
    
```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0304
Page 109
(21)

000246	006000			ROR	RO				
000250	103016			BCC	9\$				
000252	104455			TRAP	55				
000254	000110			.WORD	110				4209
000256	000000G			.WORD	SND.DATA.ERR				
000260	000000			.WORD	0				
000262	032767	000001	000000G	BIT	#1,RET.STATUS				4211
000270	001402			BEQ	8\$				
000272	004767	000000G		JSR	PC,DECODE				
000276	012767	000001	000000G	MOV	#1,RETRIES				4213
000304	000464			BR	12\$				4208
000306	016767	000000G	000000G	MOV	CMD.SLOT,CMD.REF				4220
000314	012767	000000G	000000G	MOV	#RCV.DATA.BUF,BUF.DESCRPTR				4221
000322	012767	000012	000000G	MOV	#12,BYTE.COUNT				4223
000330	004767	000000G		JSR	PC,REC.DATA				4225
000334	006000			ROR	RO				
000336	103016			BCC	11\$				
000340	104455			TRAP	55				4228
000342	000111			.WORD	111				
000344	000000G			.WORD	RE.DATA.ERR				
000346	000000			.WORD	0				
000350	032767	000001	000000G	BIT	#1,RET.STATUS				4230
000356	001402			BEQ	10\$				
000360	004767	000000G		JSR	PC,DECODE				
000364	012767	000001	000000G	MOV	#1,RETRIES				4232
000372	000431			BR	12\$				4227
000374	026727	000000G	000104	CMP	RCV.DATA.BUF,#104				4238
000402	001425			BEQ	12\$				
000404	104455			TRAP	55				4241
000406	000112			.WORD	112				
000410	000000G			.WORD	MSG.SAC.ERR				
000412	000000			.WORD	0				
000414	016746	000006G		MOV	RCV.DATA.BUF+6,-(SP)				4242
000420	016746	000004G		MOV	RCV.DATA.BUF+4,-(SP)				
000424	016746	000002G		MOV	RCV.DATA.BUF+2,-(SP)				
000430	012746	000000G		MOV	#FMT10,-(SP)				
000434	012746	000004		MOV	#4,-(SP)				
000440	010600			MOV	SP,RO				
000442	104414			TRAP	14				
000444	012767	000001	000000G	MOV	#1,RETRIES				4244
000452	062706	000012		ADD	#12,SP				4240
000456	032767	000001	000000G	BIT	#1,RETRIES				4249
000464	001402			BEQ	13\$				
000466	004767	000000G		JSR	PC,DO.RETRIES				
000472	005767	000000G		TST	NUM.RETRIES				4251
000476	001402			BEQ	14\$				
000500	000167	177326		JMP	2\$				
000504	000207			RTS	PC				4132

; Routine Size: 163 words, Routine Base: AC\$CODE + 16352
; Maximum stack depth per invocation: 7 words

000000 004767 177266 T21:: .SBTTL T21 TEST SECTION

H8

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0305
Page 110
(21)

000000		14:	JSR	PC,\$T21	
000004	104466		TRAP	66	
000006	006000		ROR	RO	
000010	103773		BLO	14	
000012	000207		RTS	PC	

4255

; Routine Size: 6 words, Routine Base: AC\$CODE + 17060
; Maximum stack depth per invocation: 2 words

; 4257 1 !<BLF/PAGE>

```

: 4258 1  !
: 4259 3  !BGNTST;
: 4260 3  !
: 4261 3  !
: 4262 3  !**
: 4263 3  ! TEST #22 - CONTROLLER PROCESSING TIME TEST
: 4264 3  !
: 4265 3  ! DESCRIPTION:
: 4266 3  !
: 4267 3  !     THIS TEST BRINGS RC25 CONTROLLER AND SELECTED UNIT ONLINE.
: 4268 3  !
: 4269 3  !     THE CONTROLLER PROCESSING TIME IS MEASURED BY AVERAGING THE
: 4270 3  !     TIME IT TAKES TO DO 100 ZERO LENGTH SEEKS, THAT IS, SEEKS THAT
: 4271 3  !     ARE ZERO TRACKS LONG.
: 4272 3  !
: 4273 3  !     THIS IS A SINGLE SURFACE TEST. SEEK WILL BE DONE ON TOP SURFACE
: 4274 3  !     UNLESS THE OPERATOR CHOSE TO SEEK ON BOTTOM SURFACE. TRACK 0
: 4275 3  !     WILL BE USED OR THE STARTING TRACK NUMBER AS GIVEN BY THE OPERATOR
: 4276 3  !     WILL BE USED.
: 4277 3  !
: 4278 3  !     IF THERE WAS ANY ERROR IN SEEK, THIS WILL BE REPORTED WITH THE
: 4279 3  !     THE NUMBER OF SEEKS COMPLETED AND DESIRED TRACK. THE TEST WILL
: 4280 3  !     BE ABORTED UNLESS RETRIES ARE ENABLED.
: 4281 3  !
: 4282 3  !     IF SUCCESS, THE AVERAGE TIME WILL BE REPORTED.
: 4283 3  !
: 4284 3  !
: 4285 3  ! label
: 4286 3  !     BLOCK1;
: 4287 3  ! if .SWP_TRACE then PRINTF (DBM28);           ? TEST 22
: 4288 3  !
: 4289 3  ! NUM_RETRIES = ZERO;
: 4290 3  !
: 4291 3  ! while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 4292 4  !     begin
: 4293 4  !     TIP = ALL_ONES;           ! THIS FLAG INHIBITS READ_CMD
: 4294 4  !                               ! WAITING FOR END PACKET.
: 4295 4  ! ! GET AZTEC READY FOR OPERATION
: 4296 4  !
: 4297 4  !     if AZTEC_READY ()           ! IF FAILURE REPORT ERROR
: 4298 4  !     then
: 4299 5  !     begin
: 4300 5  !     ERRDF (75, AZT_READY_ERR, 0);           !
: 4301 5  !
: 4302 5  !     if .RET_STATUS then DECODE ();           ! DECODE THE STATUS, IF ANY
: 4303 5  !
: 4304 5  !     RETRIES = TRUE;           ! SET RETRIES FLAG
: 4305 5  !     end
: 4306 4  ! else
: 4307 4  ! BLOCK1 :
: 4308 5  !     begin
: 4309 5  ! !
: 4310 5  !     BYTE_COUNT = ZERO;           ! BYTE COUNTS ZERO
: 4311 5  !     LBN_ST = .SWP_START*.SIZ_LBN;           ! STARTING LBN
: 4312 5  !     BUF_DESCRPT = ZERO;           ! CLEAR BUFFER DESCRIPTOR
: 4313 5  ! !
: 4314 5  ! ! FILL THE COMMUNICATION COMMAND RING SLOTS

```

```

: 4315 5 ! WITH READ COMMANDS
: 4316 5 OUT_BOUND = ZERO; ! INITIALIZE COMMAND COUNT
: 4317 5 IN_BOUND = ZERO; ! INIT RECEIVE COUNT
: 4318 5
: 4319 5 incru I from 0 to SND_ALLOCATE - 1 do ! FILL COMMAND BUFFER WITH
: 4320 6 begin ! SEEK COMMANDS (16 SLOTS
: 4321 6 CMD_REF = .CMD_SLOT; ! WILL BE FILLED TO GET
: 4322 6 ! A QUEUE LENGTH OF 15)
: 4323 6 OUT_BOUND = .OUT_BOUND + 1; ! UPDATE COMMAND COUNT
: 4324 6 READ_CMD (); ! ISSUE READ COMMAND
: 4325 6
: 4326 6 if GET_CMD_SLOT () then exitloop; ! GET NEXT COMMAND SLOT
: 4327 6
: 4328 5 end;
: 4329 5
: 4330 5 !
: 4331 5 ! INIT THE CLOCK AND START TIMING
: 4332 5 !
: 4333 5 CLOCK_INIT (); ! INIT CLOCK VARIABLES
: 4334 5 TEMP = .RC25_ADDR [RCIP, RC_ALL]; ! READ IP TO INITIATE
: 4335 5 ! CONTROLLER TO START POLLING
: 4336 5
: 4337 5 while .IN_BOUND lequ 100 do ! DO SEEK 100 TIMES
: 4338 6 begin
: 4339 6
: 4340 6 if REC_STATUS () ! POLL RECEIVE RING FOR HOST
: 4341 6 ! OWNERSHIP BIT.
: 4342 6 ! IF ERROR, REPORT ERROR
: 4343 7 then
: 4344 7 begin
: 4345 7 TEMP = .IN_BOUND;
: 4346 7 ERRDF (76, MSG_SEEK_ERR, 0); !
: 4347 7 PRINTB (FMT11, .TEMP, .LBN_ST); !
: 4348 7 DECODE (); ! DECODE END PACKET STATUS
: 4349 7 RETRIES = TRUE; ! AND ABORT TEST
: 4350 7 leave BLOCK1;
: 4351 7 end
: 4352 6 else
: 4353 7 begin
: 4354 7 while .OUT_BOUND lequ 100 do
: 4355 8 begin
: 4356 8 SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED; ! RETURN
: 4357 8 ! READ CMD BACK TO PORT
: 4358 8 OUT_BOUND = .OUT_BOUND + 1; ! INCREMENT COMMAND COUNT
: 4359 8
: 4360 8 if GET_CMD_SLOT () then exitloop;
: 4361 8
: 4362 8 if (.OUT_BOUND - .IN_BOUND) eglu 16 then exitloop; ! MAINTAIN A QUEUE LENGTH OF 15
: 4363 8
: 4364 7 end;
: 4365 7
: 4366 7 TEMP = .RC25_ADDR [RCIP, RC_ALL]; ! READ IP TO INITIATE
: 4367 7 ! CONTROLLER TO START POLLING
: 4368 6 end;
: 4369 6
: 4370 5 end;
: 4371 5

```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 B11ss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0308
Page 113
(22)

```

: 4372 5
: 4373 5
: 4374 5
: 4375 5
: 4376 5
: 4377 5
: 4378 5
: 4379 5
: 4380 5
: 4381 4
: 4382 4
: 4383 4
: 4384 4
: 4385 4
: 4386 4
: 4387 3
: 4388 3
: 4389 3
: 4390 1

! STOP THE CLOCK
!
! .CLK_CSR = ZERO;
! P6 = 100;
! P2 = (.MINUTES*60 + .SECONDS)*10;
! DATA4 = .P2 * (.TICKS*10/.CLK_HERTZ);
! DATA3 = .TICKS*1000 mod .CLK_HERTZ;
! PRINTB (MSG_PRO_TIME, .DATA4, .DATA3);
! end;
! STOP THE CLOCK
! NUMBER OF SEEKS
! SO MANY MSEC. PER SEEK
! TOTAL MSEC./SEEK
! 100TH OF MSEC. PER SEEK
! PRINT MESSAGE 'AVERAGE SEEK TIME'

if (.RETRIES) then DO_RETRIES ();
if (.NUM_RETRIES eqv ZERO) then exitloop;
end;
return;
ENDTST;

```

```

000000 010146          $T22: .SBTTL $T22 TEST SECTION
000002 024646          MOV R1,-(SP) ; 4256
000004 032767 000001 000000G CMP -(SP),-(SP) ;
000012 001407          BIT #1,SWP.TRACE ; 4287
000014 012746 000000G BEQ 1$
000020 012746 000001 MOV #DBM28,-(SP)
000024 010600          MOV #1,-(SP)
000026 104417          MOV SP,R0 ; SP,*
000030 022626          TRAP 17
000032 005067 000000G CMP (SP),,(SP).
000036 026767 000000G 000000G 1$: CLR NUM.RETRIES ; 4289
000044 101402          2$: CMP NUM.RETRIES,SWP.RETRIES ; 4291
000046 000167 000622 BLOS 3$
000052 012767 177777 000000G 3$: JMP 15$
000060 004767 000000G MOV #1,TIP ; 4293
000064 006000          JSR PC,AZTEC.READY ; 4297
000066 103017          ROR R0
000070 104455          BCC 5$
000072 000113          TRAP 55 ; 4300
000074 000000G .WORD 113
000076 000000 .WORD AZT.READY.ERR
000100 032767 000001 000000G BIT #1,RET.STATUS ; 4302
000106 001402          BEQ 4$
000110 004767 000000G JSR PC,DECODE
000114 012767 000001 000000G 4$: MOV #1,RETRIES ; 4304
000122 000167 000520          JMP 13$ ; 4297
000126 005067 000000G 5$: CLR BYTE.COUNT ; 4310
000132 016746 000000G MOV SWP.START,-(SP) ; 4311
000136 016746 000000G MOV SIZ.LBN,-(SP)
000142 004767 000000G JSR PC,BL#MUL
000146 010067 000000G MOV R0,LBN.ST
000152 005067 000000G CLR BUF.DESCRPTR ; 4312
000156 005067 000000G CLR OUT.BOUND ; 4316
000162 005067 000000G CLR IN.BOUND ; 4317
000166 005001          CLR R1 ; I 4319

```

ZRCFR3 V03.0	CZRCFC0 RC25 FR END TEST TEST SECTION		27-Mar-1985 15:27:28 27-Mar-1985 13:28:18	VAX-11 Bliss-16 V4.0-579 USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4		
000170	016767	000000G 000000G	6#:	MOV	CMD.SLOT,CMD.REF	4321
000176	005267	000000G		INC	OUT.BOUND	4323
000202	004767	000000G		JSR	PC,READ.CMD	4324
000206	004767	000000G		JSR	PC,GET.CMD.SLOT	4326
000212	006000			ROR	RO	
000214	103404			BLO	7#	
000216	005201			INC	R1	: I
000220	020127	000017		CMP	R1,#17	: I,*
000224	101761			BLOS	6#	
000226	004767	000000G	7#:	JSR	PC,CLOCK.INIT	4333
000232	017766	000000G 000004		MOV	@RC25.ADDR,4(SP)	: *,RC.REG
000240	016667	000004 000000G		MOV	4(SP),TEMP	: RC.REG,*
000246	026727	000000G 000144	8#:	CMP	IN.BOUND,#144	
000254	101100			BHI	11#	
000256	004767	000000G		JSR	PC,REC.STATUS	4340
000262	006000			ROR	RO	
000264	103033			BCC	9#	
000266	016767	000000G 000000G		MOV	IN.BOUND,TEMP	4344
000274	104455			TRAP	55	4345
000276	000114			.WORD	114	
000300	000000G			.WORD	MSG.SEEK.ERR	
000302	000000			.WORD	0	
000304	016716	000000G		MOV	LBN.ST,(SP)	4346
000310	016746	000000G		MOV	TEMP,-(SP)	
000314	012746	000000G		MOV	@FMT11,-(SP)	
000320	012746	000003		MOV	#3,-(SP)	
000324	010600			MOV	SP,RO	: SP,*
000326	104414			TRAP	14	
000330	062706	000006		ADD	#6,SP	
000334	004767	000000G		JSR	PC,DECODE	4347
000340	012767	000001 000000G		MOV	#1,RETRIES	4348
000346	162706	000022		SUB	#22,SP	4343
000352	000533			BR	12#	
000354	026727	000000G 000144	9#:	CMP	OUT.BOUND,#144	4354
000362	101026			BHI	10#	
000364	016700	000000G		MOV	CMD.SLOT,RO	4356
000370	006300			ASL	RO	
000372	006300			ASL	RO	
000374	066700	000000G		ADD	SEND.RING,RO	
000400	052760	100000 000002		BIS	#100000,2(RO)	
000406	005267	000000G		INC	OUT.BOUND	4358
000412	004767	000000G		JSR	PC,GET.CMD.SLOT	4360
000416	006000			ROR	RO	
000420	103407			BLO	10#	
000422	016700	000000G		MOV	IN.BOUND,RO	4362
000426	062700	000020		ADD	#20,RO	
000432	026700	000000G		CMP	OUT.BOUND,RO	
000436	001346			BNE	9#	
000440	017766	000000G 000006	10#:	MOV	@RC25.ADDR,6(SP)	: *,RC.REG
000446	016667	000006 000000G		MOV	6(SP),TEMP	: RC.REG,*
000454	000674			BR	8#	4337
000456	005077	000000G	11#:	CLR	@CLK.CSR	4375
000462	012767	000144 000000G		MOV	#144,P6	4376
000470	016716	000000G		MOV	MINUTES,(SP)	4377
000474	012746	000074		MOV	#74,-(SP)	
000500	004767	000000G		JSR	PC,BL\$MUL	
000504	066700	000000G		ADD	SECONDS,RO	


```

000510 010016          MOV    RO,(SP)
000512 012746 000012  MOV    #12,-(SP)
000516 004767 000000G  JSR    PC,BL#MUL
000522 010067 000000G  MOV    RO,P2
000526 016716 000000G  MOV    TICKS,(SP) ; 4378
000532 012746 000012  MOV    #12,-(SP)
000536 004767 000000G  JSR    PC,BL#MUL
000542 010016          MOV    RO,(SP)
000544 016746 000000G  MOV    CLK.HERTZ,-(SP)
000550 004767 000000G  JSR    PC,BL#DIV
000554 066700 000000G  ADD    P2,RO
000560 010067 000000G  MOV    RO,DATA4
000564 016716 000000G  MOV    TICKS,(SP) ; 4379
000570 012746 001750  MOV    #1750,-(SP)
000574 004767 000000G  JSR    PC,BL#MUL
000600 010016          MOV    RO,(SP)
000602 016746 000000G  MOV    CLK.HERTZ,-(SP)
000606 004767 000000G  JSR    PC,BL#MOD
000612 010067 000000G  MOV    RO,DATA3
000616 016716 000000G  MOV    DATA3,(SP) ; 4380
000622 016746 000000G  MOV    DATA4,-(SP)
000626 012746 000000G  MOV    #MSG.PRO.TIME,-(SP)
000632 012746 000003  MOV    #3,-(SP)
000636 010600          MOV    SP,RO ; SP,*
000640 104414          TRAP   14
000642 062706 000026 12#:  ADD    #26,SP ; 4297
000646 032767 000001 000000G 13#:  BIT    #1,RETRIES ; 4383
000654 001402          BEQ    14#
000656 004767 000000G  JSR    PC,DO.RETRIES
000662 005767 000000G 14#:  TST    NUM.RETRIES ; 4385
000666 001402          BEQ    15#
000670 000167 177142          JMP    2#
000674 022626 15#:  CMP    (SP),.(SP) ; 4256
000676 012601          MOV    (SP),.R1
000700 000207          RTS    PC
    
```

; Routine Size: 225 words, Routine Base: AC#CODE * 17074
 ; Maximum stack depth per invocation: 16 words

```

000000 004767 177072          .SBTTL T22 TEST SECTION
000000 122::          JSR    PC,#T22 ; 4389
000004 104466          TRAP   66
000006 006000          ROR    RO
000010 103773          BLO    1#
000012 000207          RTS    PC
    
```

; Routine Size: 6 words, Routine Base: AC#CODE * 17776
 ; Maximum stack depth per invocation: 2 words

```

; 4391 1 ;
; 4392 3 ! BGNTST;
; 4393 3 ! TEST #23 - ONE TRACK SEEK TIMING TEST
    
```

```

: 4394 3
: 4395 3
: 4396 3
: 4397 3
: 4398 3
: 4399 3
: 4400 3
: 4401 3
: 4402 3
: 4403 3
: 4404 3
: 4405 3
: 4406 3
: 4407 3
: 4408 3
: 4409 3
: 4410 3
: 4411 3
: 4412 3
: 4413 3
: 4414 3
: 4415 3
: 4416 3
: 4417 3
: 4418 3
: 4419 3
: 4420 3
: 4421 4
: 4422 4
: 4423 4
: 4424 4
: 4425 4
: 4426 4
: 4427 4
: 4428 5
: 4429 5
: 4430 5
: 4431 5
: 4432 5
: 4433 5
: 4434 5
: 4435 4
: 4436 4
: 4437 3
: 4438 3
: 4439 3
: 4440 3
: 4441 3
: 4442 3
: 4443 3
: 4444 3
: 4445 3
: 4446 3
: 4447 3
: 4448 3
: 4449 3
: 4450 3

```

DESCRIPTION:

THIS TEST BRINGS RC25 CONTROLLER AND SELECTED UNIT ONLINE.

ONE TRACK SEEK TIME IS THE AVERAGE OF ALL ONE TRACK SEEKS THAT DO NOT INCLUDE A HEAD SWITCH. ALL FORWARD ONE TRACK SEEKS WILL BE DONE AND TIMED AND THEN REVERSE ONE TRACK SEEKS WILL BE DONE AND TIMED. AVERAGE TIME WILL BE REPORTED.

THIS IS A SINGLE SURFACE TEST. TOP SURFACE WILL BE USED UNLESS THE OPERATOR CHOSE OTHERWISE. SEEKS WILL BE FROM START TO THE END OF TRACKS.

IF THERE WAS AN ERROR, ERROR WILL BE REPORTED AND THE TEST ABORTED UNLESS RETRIES ARE TURNED ON.

```

label
  BLOCK1;
if .SWP_TRACE then PRINTF (DBM29);           ! TEST 23
NUM_RETRIES = ZERO;
while (.NUM_RETRIES lequ .SWP_RETRIES) do
  begin
    TIP = ALL_ONES;                          ! THIS FLAG INHIBITS READ_CMD
                                              ! WAITING FOR END PACKET.
! GET AZTEC READY FOR OPERATION
    if AZTEC_READY ()                        ! IF FAILURE REPORT ERROR
    then
      begin
        ERRDF (77, AZT_READY_ERR, 0);        !
        if .RET_STATUS then DECODE ();       ! DECODE THE STATUS, IF ANY
        RETRIES = TRUE;                      ! SET RETRIES FLAG
      end
    else
      BLOCK1 :
      begin
        BYTE_COUNT = ZERO;                   ! BYTE COUNTS ZERO
        BUF_DESCRIPTOR = ZERO;               ! CLEAR BUFFER DESCRIPTOR
        LBN_SZ = .SIZ_LBN;                   ! LBN SIZE TO INCREMENT TRACK
        LBN_ST = (.OFFSET * 0) * .LBN_SZ;    ! STARTING LBN
        LBN_ED = (.OFFSET * 820) * .LBN_SZ; ! ENDING LBN
      end
! INIT AND START THE CLOCK
    CLOCK_INIT ();                            ! INIT CLOCK VARIABLES
    incu COUNT from 0 to 1 do                ! DO FORWARD AND REVERSE

```

```

: 4451 6          begin          ! ONE TRACK SEEKS
: 4452 6
: 4453 6          ! FORWARD DIRECTION SEEK FIRST AND THEN REVERSE SEEK
: 4454 6          ! SECOND TIME AROUND THIS LOOP.
: 4455 6
: 4456 6          OUT_BOUND = ZERO;          ! INITIALIZE COMMAND COUNT
: 4457 6          IN_BOUND = ZERO;          ! INIT RECEIVE COUNT
: 4458 6          ! FILL THE COMMUNICATION COMMAND RING SLOTS
: 4459 6          ! WITH READ COMMANDS
: 4460 6
: 4461 6          incru I from 0 to SND_ALLOCATE - 1 do          ! FILL COMMAND BUFFER WITH
: 4462 7          begin          ! SEEK COMMANDS (16 SLOTS
: 4463 7          CMD_REF = .LBN_ST;          ! WILL BE FILLED TO GET
: 4464 7          READ_CMD ();          ! A QUEUE LENGTH OF 15)
: 4465 7          OUT_BOUND = .OUT_BOUND + 1;
: 4466 7          LBN_ST = .LBN_ST + .LBN_SZ;          ! NEXT TRACK
: 4467 7
: 4468 7          if GET_CMD_SLOT () then exitloop;          ! GET NEXT COMMAND SLOT
: 4469 7
: 4470 6          end;
: 4471 6
: 4472 6          TEMP = .RC25_ADDR [RCIP, RC_ALL];          ! READ IP FOR CONTROLLER START
: 4473 6          ! POLLING
: 4474 6
: 4475 6          ! FORWARD DIRECTION SEEK
: 4476 6
: 4477 6
: 4478 6          while .IN_BOUND lequ 820 do          ! DO SEEK FROM STARTING TRACK
: 4479 7          begin
: 4480 7
: 4481 7          if REC_STATUS ()          ! POLL RECEIVE RING FOR HOST
: 4482 7          ! OWNERSHIP BIT.
: 4483 7          ! IF ERROR, REPORT ERROR
: 4484 8          then
: 4485 8          begin
: 4486 8          TEMP = .IN_BOUND;          ! SAVE RECEIVE COUNT
: 4487 8          ERRDF (78, MSG_SEEK_ERR, 0);          !
: 4488 8          PRINTB (FMT11, .TEMP, .LBN_ST);
: 4489 8          DECODE ();          ! DECODE END PACKET STATUS
: 4490 8          RETRIES = TRUE;
: 4491 8          leave BLOCK1;          ! AND ABORT TEST
: 4492 7          end
: 4493 7          else
: 4494 8          begin
: 4495 8          while .OUT_BOUND lequ 820 do
: 4496 9          begin
: 4497 9          READ_FILL_CMD ();          ! GIVE NEXT SEEK COMMAND
: 4498 9          ! AND MAINTAIN A QUEUE OF 15
: 4499 9          OUT_BOUND = .OUT_BOUND + 1;
: 4500 9          LBN_ST = .LBN_ST + .LBN_SZ;
: 4501 9
: 4502 9          if GET_CMD_SLOT () then exitloop;
: 4503 9
: 4504 9          if (.OUT_BOUND - .IN_BOUND) eq 16 then exitloop;
: 4505 9
: 4506 8          end;
: 4507 8

```

```

: 4508 8          TEMP = .RC25_ADDR [RCIP, RC_ALL]; ! READ IP & CONTROLLER START TO POLL
: 4509 7          end;
: 4510 7
: 4511 6          end;
: 4512 6
: 4513 6          ! REVERSE STARTING AND ENDING LBN NUMBERS AND REDO
: 4514 6          ! ONE TRACK SEEKS AS DONE BEFORE
: 4515 6          LBN_ST = .LBN_ED; ! START FROM HIGH TRACK NUMBER
: 4516 6          LBN_ED = (.OFFSET + 0)*.LBN_SZ; ! TO LOWEST ON THE SURFACE.
: 4517 6          LBN_SZ = (not .SIZ_LBN) * I; ! COMPLEMENT SIZE OF LBN
: 4518 5          end;
: 4519 5
: 4520 5          !
: 4521 5          ! STOP THE CLOCK
: 4522 5          !
: 4523 5          .CLK_CSR = ZERO; ! STOP THE CLOCK
: 4524 5          ! TOTAL SEKS = 1642
: 4525 5          DATA4 = (.MINUTES*60 + .SECONDS); ! TOTAL SEC.
: 4526 5          DATA3 = (.DATA4*3) mod 5; ! REMAINDER
: 4527 5          DATA4 = .DATA4*3/5; ! MSEC. PER SEEK
: 4528 5          DATA3 = .DATA3*100; ! 100TH OF MSEC.
: 4529 5          DATA3 = .DATA3 * (.TICKS*300/(5*.CLK_HERTZ)); ! 100TH OF MSEC./SEEK
: 4530 5          PRINTB (MSG_SK_TIME, .DATA4, .DATA3); ! PRINT MESSAGE 'AVERAGE SEEK TIME'
: 4531 4          end;
: 4532 4
: 4533 4          if (.RETRIES) then DO_RETRIES ();
: 4534 4
: 4535 4          if (.NUM_RETRIES eglu ZERO) then exitloop;
: 4536 4
: 4537 3          end;
: 4538 3
: 4539 3          return;
: 4540 1          ENDTST;

```

Address	Hex	Dec	Hex	Dec	Label	Hex	Dec
000000	004167	000000G	\$T23:	.SBTTL	\$T23 TEST SECTION		
000004	024646			JSR	R1,\$SAVE2		4390
000006	032767	000001 000000G		CMP	-(SP),-(SP)		
000014	001407			BIT	#1,SWP.TRACE		4416
000016	012746	000000G		BEQ	1\$		
000022	012746	000001		MOV	#DBM29,-(SP)		
000026	010600			MOV	#1,-(SP)		
000030	104417			MOV	SP,R0		; SP,*
000032	022626			TRAP	17		
000034	005067	000000G	1\$:	CMP	(SP)*,(SP)*		
000040	026767	000000G 000000G	2\$:	CLR	NUM.RETRIES		4418
000046	101402			CMP	NUM.RETRIES,SWP.RETRIES		4420
000050	000167	001002		BLOS	3\$		
000054	012767	177777 000000G	3\$:	JMP	16\$		
000062	004767	000000G		MOV	#-1,TIP		4422
000066	006000			JSR	PC,AZTEC.READY		4426
000070	103017			ROR	R0		
000072	104455			BCC	5\$		
000074	000115			TRAP	55		4429
000076	000000G			.WORD	115		
000100	000000			.WORD	AZT.READY.ERR		
				.WORD	0		

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0314
Page 119
(22)

000102	032767	000001	000000G		BIT	#1,RET.STATUS	:	4431
000110	001402				BEQ	4\$		
000112	004767	000000G			JSR	PC.DECODE		
000116	012767	000001	000000G	4\$:	MOV	#1,RETRIES	:	4433
000124	000167	000700			JMP	14\$:	4426
000130	005067	000000G		5\$:	CLR	BYTE.COUNT	:	4439
000134	005067	000000G			CLR	BUF.DESCRPTR	:	4440
000140	016767	000000G	000000G		MOV	SIZ.LBN,LBN.SZ	:	4442
000146	016746	000000G			MOV	OFFSET,-(SP)	:	4443
000152	016746	000000G			MOV	LBN.SZ,-(SP)		
000156	004767	000000G			JSR	PC,BL#MUL		
000162	010067	000000G			MOV	R0,LBN.ST		
000166	016716	000000G			MOV	OFFSET,(SP)	:	4444
000172	062716	001464			ADD	#1464,(SP)		
000176	016746	000000G			MOV	LBN.SZ,-(SP)		
000202	004767	000000G			JSR	PC,BL#MUL		
000206	010067	000000G			MOV	R0,LBN.ED		
000212	004767	000000G			JSR	PC,CLOCK.INIT	:	4448
000216	005002				CLR	R2	: COUNT	4450
000220	005067	000000G		6\$:	CLR	OUT.BOUND	:	4456
000224	005067	000000G			CLR	IN.BOUND	:	4457
000230	005001				CLR	R1	: I	4461
000232	016767	000000G	000000G	7\$:	MOV	LBN.ST,CMD.REF	:	4463
000240	004767	000000G			JSR	PC,READ.CMD	:	4464
000244	005267	000000G			INC	OUT.BOUND	:	4465
000250	066767	000000G	000000G		ADD	LBN.SZ,LBN.ST	:	4466
000256	004767	000000G			JSR	PC,GET.CMD.SLOT	:	4468
000262	006000				ROR	R0		
000264	103404				BLO	8\$		
000266	005201				INC	R1	: I	4461
000270	020127	000017			CMP	R1,#17	: I,*	
000274	101756				BLOS	7\$		
000276	017766	000000G	000006	8\$:	MOV	@RC25.ADDR,6(SP)	: *,RC.REG	4472
000304	016667	000006	000000G		MOV	6(SP),TEMP	: RC.REG,*	
000312	026727	000000G	001464	9\$:	CMP	IN.BOUND,#1464	:	4478
000320	101075				BHI	12\$		
000322	004767	000000G			JSR	PC,REC.STATUS	:	4481
000326	006000				ROR	R0		
000330	103034				BCC	10\$		
000332	016767	000000G	000000G		MOV	IN.BOUND,TEMP	:	4485
000340	104455				TRAP	55	:	4486
000342	000116				.WORD	116		
000344	000000G				.WORD	MSG.SEEK.ERR		
000346	000000				.WORD	0		
000350	016716	000000G			MOV	LBN.ST,(SP)	:	4487
000354	016746	000000G			MOV	TEMP,-(SP)		
000360	012746	000000G			MOV	#FMT11,-(SP)		
000364	012746	000003			MOV	#3,-(SP)		
000370	010600				MOV	SP,R0	: SP,*	
000372	104414				TRAP	14		
000374	062706	000006			ADD	#6,SP		
000400	004767	000000G			JSR	PC,DECODE	:	4488
000404	012767	000001	000000G		MOV	#1,RETRIES	:	4489
000412	162706	000026			SUB	#26,SP	:	4484
000416	000167	000402			JMP	13\$		
000422	026727	000000G	001464	10\$:	CMP	OUT.BOUND,#1464	:	4495
000430	101022				BHI	11\$		

E9

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0315
Page 120
(22)

000432	004767	000000G			JSR	PC,READ.FILL.CMD	:	4497
000436	005267	000000G			INC	OUT.BOUND	:	4499
000442	066767	000000G	000000G		ADD	LBN.SZ,LBN.ST	:	4500
000450	004767	000000G			JSR	PC,GET.CMD.SLOT	:	4502
000454	006000				ROR	R0	:	
000456	103407				BLO	11\$		
000460	016700	000000G			MOV	IN.BOUND,R0	:	4504
000464	062700	000020			ADD	#20,R0		
000470	026700	000000G			CMP	OUT.BOUND,R0		
000474	001352				BNE	10\$		
000476	017766	000000G	000010	11\$:	MOV	@RC25.ADDR,10(SP)	: *,RC.REG	4508
000504	016667	000010	000000G		MOV	10(SP),TEMP	: RC.REG,*	
000512	000677				BR	9\$		4478
000514	016767	000000G	000000G	12\$:	MOV	LBN.ED,LBN.ST	:	4515
000522	016716	000000G			MOV	OFFSET,(SP)	:	4516
000526	016746	000000G			MOV	LBN.SZ,-(SP)		
000532	004767	000000G			JSR	PC,BL\$MUL		
000536	010067	000000G			MOV	R0,LBN.ED		
000542	016700	000000G			MOV	SIZ.LBN,R0	:	4517
000546	005100				COM	R0		
000550	010001				MOV	R0,R1		
000552	005201				INC	R1		
000554	010167	000000G			MOV	R1,LBN.SZ		
000560	005726				TST	(SP)	:	4451
000562	005202				INC	R2	: COUNT	4450
000564	020227	000001			CMP	R2,#1	: COUNT,*	
000570	101613				BLOS	6\$		
000572	005077	000000G			CLR	@CLK.CSR	:	4523
000576	016716	000000G			MOV	MINUTES,(SP)	:	4525
000602	012746	000074			MOV	#74,-(SP)		
000606	004767	000000G			JSR	PC,BL\$MUL		
000612	066700	000000G			ADD	SECONDS,R0		
000616	010067	000000G			MOV	R0,DATA4		
000622	016716	000000G			MOV	DATA4,(SP)	:	4526
000626	012746	000003			MOV	#3,-(SP)		
000632	004767	000000G			JSR	PC,BL\$MUL		
000636	010016				MOV	R0,(SP)		
000640	012746	000005			MOV	#5,-(SP)		
000644	004767	000000G			JSR	PC,BL\$MOD		
000650	010067	000000G			MOV	R0,DATA3		
000654	016716	000000G			MOV	DATA4,(SP)	:	4527
000660	012746	000003			MOV	#3,-(SP)		
000664	004767	000000G			JSR	PC,BL\$MUL		
000670	010016				MOV	R0,(SP)		
000672	012746	000005			MOV	#5,-(SP)		
000676	004767	000000G			JSR	PC,BL\$DIV		
000702	010067	000000G			MOV	R0,DATA4		
000706	016716	000000G			MOV	DATA3,(SP)	:	4528
000712	012746	000144			MOV	#144,-(SP)		
000716	004767	000000G			JSR	PC,BL\$MUL		
000722	010067	000000G			MOV	R0,DATA3		
000726	016716	000000G			MOV	TICKS,(SP)	:	4529
000732	012746	000454			MOV	#454,-(SP)		
000736	004767	000000G			JSR	PC,BL\$MUL		
000742	010016				MOV	R0,(SP)		
000744	016746	000000G			MOV	CLK.HERTZ,-(SP)		
000750	012746	000005			MOV	#5,-(SP)		

```

000754 004767 000000G      JSR    PC,BL$MUL
000760 005726              TST    (SP).
000762 010016              MOV    RO,(SP)
000764 004767 000000G      JSR    PC,BL$DIV
000770 066700 000000G      ADD    DATA3,RO
000774 010067 000000G      MOV    RO,DATA3
001000 016716 000000G      MOV    DATA3,(SP)
001004 016746 000000G      MOV    DATA4,-(SP)
001010 012746 000000G      MOV    #MSG.SK.TIME,-(SP)
001014 012746 000003      MOV    #3,-(SP)
001020 010600              MOV    SP,RO
001022 104414              TRAP   14
001024 062706 000034      13$:  ADD    #34,SP
001030 032767 000001 000000G 14$:  BIT    #1,RETRIES
001036 001402              BEQ    15$
001040 004767 000000G      JSR    PC,DO.RETRIES
001044 005767 000000G      15$:  TST    NUM.RETRIES
001050 001402              BEQ    16$
001052 000167 176762      JMP    2$
001056 022626      16$:  CMP    (SP).,(SP).
001060 000207              RTS    PC
    
```

: Routine Size: 281 words, Routine Base: AC\$CODE + 20012
 : Maximum stack depth per invocation: 21 words

```

000000 004767 176712      T23:: JSR    PC,$T23
000000 1$:              TRAP   66
000004 104466              ROR    RO
000006 006000              BLO    1$
000010 103773              RTS    PC
000012 000207
    
```

: Routine Size: 6 words, Routine Base: AC\$CODE + 21074
 : Maximum stack depth per invocation: 2 words

: 4541 1 !<BLF/PAGE>

```

: 4542 1  !
: 4543 3  ! BGNTST;
: 4544 3  !
: 4545 3  !
: 4546 3  ! ** TEST #24 - AVERAGE SEEK TIMING TEST
: 4547 3  !
: 4548 3  ! DESCRIPTION:
: 4549 3  !
: 4550 3  !     THIS TEST BRINGS RC25 CONTROLLER AND SELECTED UNIT ONLINE.
: 4551 3  !
: 4552 3  !     TE AVERAGE SEEK TIME IS THE AVERAGE TIME IT TAKES TO DO
: 4553 3  !     A SEEK GIVEN THAT IT IS EQUALLY LIKELY TO START ON ANY TRACK
: 4554 3  !     AND ANY HEAD, AND EQUALLY LIKELY TO END ON ANY TRACK AND ANY
: 4555 3  !     HEAD.
: 4556 3  !
: 4557 3  !     ONE THOUSAND RANDOM SEEKS WILL BE DONE OVER THE RANGE
: 4558 3  !     OF LBN TRACK 0 THRU LBN TRACK 1641 TO COVER BOTH SURFACES OF THE
: 4559 3  !     SELECTED UNIT. FIRST TIME EXPRESS BIT IN COMMAND MODIFIER FIELD
: 4560 3  !     FOR READ CMD WILL BE SET SO THAT RANDOM SEEKS ARE TIMED AND
: 4561 3  !     IN THE SECOND TIME EXPRESS BIT WILL BE RESET SO THAT THE
: 4562 3  !     RANDOM LBN AVAILABLE TO THE CONTROLLER ARE ORDERED BY THE
: 4563 3  !     CONTROLLER FOR SEEKS. AVERAGE TIME FOR BOTH CASES WILL BE REPORTED.
: 4564 3  !
: 4565 3  !     AN ERROR REPORT FOR THIS TEST WILL REPORT THE NUMBER OF SEEKS AND
: 4566 3  !     DESIRED TRACK NUMBER. AFTER REPORTING A FAILURE, THE DAIGNOSTIC WILL
: 4567 3  !     PROCEED TO THE NEXT TEST UNLESS RETRIES IS TURNED ON.
: 4568 3  !
: 4569 3  ! --
: 4570 3  !
: 4571 3  ! label
: 4572 3  !     BLOCK1;
: 4573 3  !
: 4574 3  ! if .SWP_TRACE then PRINTF (DBM30);           ! TEST 24
: 4575 3  !
: 4576 3  ! NUM_RETRIES = ZERO;
: 4577 3  !
: 4578 3  ! while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 4579 4  !     begin
: 4580 4  !     ! GET AZTEC READY FOR OPERATION
: 4581 4  !
: 4582 4  !     if AZTEC_READY ( )                       ! IF FAILURE REPORT ERROR
: 4583 4  !     then
: 4584 5  !         begin
: 4585 5  !         ERRDF (79, AZT_READY_ERR, 0);       !
: 4586 5  !
: 4587 5  !         if .RET_STATUS then DECODE ( );    ! DECODE THE STATUS, IF ANY
: 4588 5  !
: 4589 5  !         RETRIES = TRUE;                    ! SET RETRIES FLAG
: 4590 5  !         end
: 4591 4  !     else
: 4592 4  ! BLOCK1 :
: 4593 5  !     begin
: 4594 5  !
: 4595 5  !     !
: 4596 5  !     ! BYTE_COUNT = ZERO;                    ! BYTE COUNTS ZERO
: 4597 5  !     ! BUF_DESCPTR = ZERO;                  ! CLEAR BUFFER DESCRIPTOR
: 4598 5  !
: 4598 5  !     incru COUNT from 0 to 1 do

```



```

: 4599 6          begin          ! FIRST FIND AVERAGE SEEK TIME
: 4600 6          !              ! NEXT TIME FIND ORDERED AVERAGE
: 4601 6          !              ! SEEK TIME.
: 4602 6          !
: 4603 6          ! FILL THE COMMUNICATION COMMAND RING SLOTS
: 4604 6          ! WITH READ COMMANDS
: 4605 6          OUT_BOUND = ZERO;      ! INIT COMMAND COUNT
: 4606 6          IN_BOUND = ZERO;      ! INIT RECEIVE COUNT
: 4607 6          TIP = ALL_ONES;      ! TELL READ_CMD NOT TO WAIT FOR
: 4608 6          !              ! END PACKETS.
: 4609 6          P2 = .TICKS;          !
: 4610 6          !
: 4611 6          incru I from 0 to SND_ALLOCATE - 1 do      ! FILL COMMAND BUFFER WITH
: 4612 7          begin          ! SEEK COMMANDS (16 SLOTS)
: 4613 7          RANDOM_NUM ();
: 4614 7          LBN_ST = .P3;          ! GET RANDOM LBN
: 4615 7          CMD_REF = .LBN_ST;    ! WILL BE FILLED TO GET
: 4616 7          !              ! A QUEUE LENGTH OF 15
: 4617 7          READ_CMD ();          ! ISSUE READ COMMAND
: 4618 7          OUT_BOUND = .OUT_BOUND + 1;
: 4619 7          !
: 4620 7          if GET_CMD_SLOT () then exitloop;          ! GET NEXT COMMAND SLOT
: 4621 7          !
: 4622 6          end;
: 4623 6          !
: 4624 6          !
: 4625 6          ! INIT AND START THE CLOCK
: 4626 6          !
: 4627 6          CLOCK_INIT ();          ! INIT CLOCK VARIABLES
: 4628 6          TEMP = .RC25_ADDR [RCIP, RC_ALL];          ! READ IP & CONTROLLER START TO POLL
: 4629 6          !
: 4630 6          while .IN_BOUND lequ 1000 do          ! DO SEEK FROM STARTING TRACK
: 4631 7          begin
: 4632 7          !
: 4633 7          if REC_STATUS ()          ! POLL RECEIVE RING FOR HOST
: 4634 7          !              ! OWNERSHIP BIT.
: 4635 7          !              ! IF ERROR, REPORT ERROR
: 4636 8          then
: 4637 8          begin
: 4638 8          TEMP = .IN_BOUND;          ! SAVE RECEIVE COUNT
: 4639 8          ERRDF (80, MSG_SEEK_ERR, 0);          !
: 4640 8          PRINTB (FMT11, .TEMP, .LBN_ST);
: 4641 8          DECODE ();          ! DECODE END PACKET STATUS
: 4642 8          RETRIES = TRUE;
: 4643 8          leave BLOCK1;          ! AND ABORT TEST
: 4644 7          end
: 4645 7          else
: 4646 8          begin
: 4647 8          !
: 4648 8          while .OUT_BOUND lequ 1000 do
: 4649 9          begin
: 4650 9          RANDOM_NUM ();          ! GET RANDOM LBN
: 4651 9          LBN_ST = .P3;          ! GIVE NEXT SEEK COMMAND
: 4652 9          READ_FILL_CMD ();          ! AND MAINTAIN A QUEUE OF 15
: 4653 9          !
: 4654 9          OUT_BOUND = .OUT_BOUND + 1;
: 4655 9          !
: 4655 9          if GET_CMD_SLOT () then exitloop;

```

```

: 4656 9
: 4657 9           if (.OUT_BOUND - .IN_BOUND) eqlu 16 then exitloop;
: 4658 9
: 4659 8           end;
: 4660 8
: 4661 8           TEMP = .RC25_ADDR [RCIP, RC_ALL];    ! READ IP & CONTROLLER START TO POLL
: 4662 7           end;
: 4663 7
: 4664 6           end;
: 4665 6
: 4666 6           !
: 4667 6           ! STOP THE CLOCK
: 4668 6           !
: 4669 6           .CLK_CSR = ZERO;                    ! STOP THE CLOCK
: 4670 6           ! TOTAL SEEKS = 1000
: 4671 6           DATA4 = (.MINUTES*60 + .SECONDS);  ! MSEC. PER SEEK
: 4672 6           DATA3 = .TICKS*100/.CLK_HERTZ;    ! 100TH OF M.SEC/SEEK
: 4673 6
: 4674 6           if .COUNT eqlu ZERO
: 4675 6           then
: 4676 7             begin
: 4677 7               DATA2 = .DATA4;                ! SAVE AVE. SEEK TIME
: 4678 7               DATA1 = .DATA3;
: 4679 7               CMOD = ZERO;                   ! RESET EXPRESS BIT FOR
: 4680 7               ! READ COMMAND TO DO ORDERED SEEKS
: 4681 6             end;
: 4682 6
: 4683 5           end;
: 4684 5
: 4685 5           PRINTB (MSG_AVE_TIME, .DATA2, .DATA1); ! PRINT MESSAGE 'AVERAGE SEEK TIME'
: 4686 5           PRINTB (MES_SKO_TIME, .DATA4, .DATA3); ! AVERAGE SEEK ORDERED TIME.
: 4687 4           end;
: 4688 4
: 4689 4           if (.RETRIES) then DO_RETRIES ();
: 4690 4
: 4691 4           if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 4692 4
: 4693 3           end;
: 4694 3
: 4695 3           return;
: 4696 1           ENDTST;

```

Address	Hex	Dec	Label	Op	Opnd	Comment	Address
000000	004167	000000G	\$T24:	.SBTTL	\$T24 TEST SECTION		
000004	024646			JSR	R1,\$SAVE2		4540
000006	032767	000001 000000G		CMP	-(SP),-(SP)		
000014	001407			BIT	#1,SWP.TRACE		4574
000016	012746	000000G		BEQ	1\$		
000022	012746	000001		MOV	#DBM30,-(SP)		
000026	010600			MOV	#1,-(SP)		
000030	104417			MOV	SP,R0	; SP,*	
000032	022626			TRAP	17		
000034	005067	000000G	1\$:	CMP	(SP)*,(SP)*		
000040	026767	000000G 000000G	2\$:	CLR	NUM.RETRIES		4576
000046	101402			CMP	NUM.RETRIES,SWP.RETRIES		4578
000050	000167	000626		BLOS	3\$		
				JMP	18\$		

ZRCFB3	CZRCFC0	RC25	FR	END TEST		27-Mar-1985 15:27:28	VAX-11 Bliss-16 V4.0-579	
V03.0	TEST SECTION					27-Mar-1985 13:28:18	USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4	
000054	004767	000000G			3#:	JSR	PC,AZTEC.READY	4582
000060	006000					ROR	RO	
000062	103017					BCC	5#	
000064	104455					TRAP	55	4585
000066	000117					.WORD	117	
000070	000000G					.WORD	AZT.READY.ERR	
000072	000000					.WORD	0	
000074	032767	000001	000000G			BIT	#1,RET.STATUS	4587
000102	001402					BEQ	4#	
000104	004767	000000G				JSR	PC,DECODE	
000110	012767	000001	000000G		4#:	MOV	#1,RETRIES	4589
000116	000167	000532				JMP	16#	4582
000122	005067	000000G			5#:	CLR	BYTE.COUNT	4595
000126	005067	000000G				CLR	BUF.DESCRPTR	4596
000132	005002					CLR	R2	4598
000134	005067	000000G			6#:	CLR	OUT.BOUND	4605
000140	005067	000000G				CLR	IN.BOUND	4606
000144	012767	177777	000000G			MOV	#-1,TIP	4607
000152	016767	000000G	000000G			MOV	TICKS,P2	4609
000160	005001					CLR	R1	4611
000162	004767	000000G			7#:	JSR	PC,RANDOM.NUM	4613
000166	016767	000000G	000000G			MOV	P3,LBN.ST	4614
000174	016767	000000G	000000G			MOV	LBN.ST,CMD.REF	4615
000202	004767	000000G				JSR	PC,READ.CMD	4617
000206	005267	000000G				INC	OUT.BOUND	4618
000212	004767	000000G				JSR	PC,GET.CMD.SLOT	4620
000216	006000					ROR	RO	
000220	103404					BLO	8#	
000222	005201					INC	R1	4611
000224	020127	000017				CMP	R1,#17	I,*
000230	101754					BLOS	7#	
000232	004767	000000G			8#:	JSR	PC,CLOCK.INIT	4627
000236	017716	000000G				MOV	@RC25.ADDR,(SP)	*,RC.REG
000242	011667	000000G				MOV	(SP),TEMP	RC.REG,*
000246	026727	000000G	001750		9#:	CMP	IN.BOUND,#1750	4630
000254	101076					BHI	12#	
000256	004767	000000G				JSR	PC,REC.STATUS	4633
000262	006000					ROR	RO	
000264	103033					BCC	10#	
000266	016767	000000G	000000G			MOV	IN.BOUND,TEMP	4637
000274	104455					TRAP	55	4638
000276	000120					.WORD	120	
000300	000000G					.WORD	MSG.SEEK.ERR	
000302	000000					.WORD	0	
000304	016746	000000G				MOV	LBN.ST,-(SP)	4639
000310	016746	000000G				MOV	TEMP,-(SP)	
000314	012746	000000G				MOV	#FMT11,-(SP)	
000320	012746	000003				MOV	#3,-(SP)	
000324	010600					MOV	SP,RO	SP,*
000326	104414					TRAP	14	
000330	062706	000010				ADD	#10,SP	
000334	004767	000000G				JSR	PC,DECODE	4640
000340	012767	000001	000000G			MOV	#1,RETRIES	4641
000346	162706	000016				SUB	#16,SP	4636
000352	000536					BR	15#	
000354	026727	000000G	001750		10#:	CMP	OUT.BOUND,#1750	4647
000362	101024					BHI	11#	

ZRCFR3	CZRCFC0	RC25	FR	END	TEST	27-Mar-1985	15:27:28	VAX-11	Bliss-16	V4.0-579	SEQ 0321
V03.0	TEST	SECTION				27-Mar-1985	13:28:18	USER#1:	[AZTEC.CZRCFC]	ZRCFC3.B16;4	Page 126
											(23)
000364	004767	000000G									4649
000370	016767	000000G	000000G								4650
000376	004767	000000G									4651
000402	005267	000000G									4653
000406	004767	000000G									4655
000412	006000										
000414	103407										
000416	016700	000000G									4657
000422	062700	000020									
000426	026700	000000G									
000432	001350										
000434	017766	000000G	000002	11\$:							4661
000442	016667	000002	000000G								
000450	000676										
000452	005077	000000G		12\$:							4630
000456	016746	000000G									4669
000462	012746	000074									4671
000466	004767	000000G									
000472	066700	000000G									
000476	010067	000000G									
000502	016716	000000G									4672
000506	012746	000144									
000512	004767	000000G									
000516	010016										
000520	016746	000000G									
000524	004767	000000G									
000530	010067	000000G									
000534	005702										
000536	001010										
000540	016767	000000G	000000G								4674
000546	016767	000000G	000000G								4677
000554	005067	000000G									4678
000560	062706	000010		13\$:							4679
000564	005202										4599
000566	020227	000001									4598
000572	101002										
000574	000167	177334									
000600	016746	000000G		14\$:							4685
000604	016746	000000G									
000610	012746	000000G									
000614	012746	000003									
000620	010600										
000622	104414										
000624	016716	000000G									4686
000630	016746	000000G									
000634	012746	000000G									
000640	012746	000003									
000644	010600										
000646	104414										
000650	062706	000016		15\$:							4582
000654	032767	000001	000000G	16\$:							4689
000662	001402										
000664	004767	000000G									
000670	005767	000000G		17\$:							4691
000674	001402										
000676	000167	177136									
000702	022626			18\$:							4540

ZRCFB3 CZRCFC0 RC25 FR END TEST
V03.0 TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

000704 000207 RTS PC

: Routine Size: 227 words, Routine Base: AC#CODE * 21110
: Maximum stack depth per invocation: 14 words

000000	004767	177066	T24::	.SBTTL	T24 TEST SECTION	
000000			1#:	JSR	PC,\$T24	4695
000004	104466			TRAP	66	
000006	006000			ROR	R0	
000010	103773			BLO	1#	
000012	000207			RTS	PC	

: Routine Size: 6 words, Routine Base: AC#CODE * 22016
: Maximum stack depth per invocation: 2 words

: 4697 1 !<BLF/PAGE>

ZRCFR3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4SEQ 0323
Page 128
(24)

```

: 4698 1
: 4699 1
: 4700 1
: 4701 1
: 4702 1
: 4703 1
: 4704 1
: 4705 1
: 4706 1
: 4707 1
: 4708 1
: 4709 1
: 4710 1
: 4711 1
: 4712 1
: 4713 1
: 4714 1
: 4715 1
: 4716 1
: 4717 1
: 4718 1
: 4719 1
: 4720 1
: 4721 1
: 4722 1
: 4723 1
: 4724 1
: 4725 1
: 4726 1
: 4727 1
: 4728 1
: 4729 1
: 4730 1
: 4731 1
: 4732 1
: 4733 1
: 4734 1
: 4735 1
: 4736 1
: 4737 1
: 4738 1
: 4739 1
: 4740 1
: 4741 1
: 4742 1
: 4743 1
: 4744 1
: 4745 1
: 4746 1
: 4747 1
: 4748 1
: 4749 1
: 4750 1
: 4751 1
: 4752 1
: 4753 1
: 4754 1

!
BGNTST;

!..
TEST #25 - FULL STROKE SEEK TIMING TEST

DESCRIPTION:

THIS TEST BRINGS RC25 CONTROLLER AND UNIT ONLINE.

THE FULL STROKE SEEK TIME IS THE AVERAGE TIME OF 1000 FULL STROKE
SEEKS THAT DO NOT INVOLVE HEAD SWITCHES.

THIS IS A SINGLE SURFACE TEST. TOP SURFACE WILL BE USED UNLESS THE
OPERATOR CHOSE OTHERWISE.

THE ERROR REPORT WILL INCLUDE NUMBER OF SEEKS AND DESIRED TRACK
NUMBER. AFTER FAILURE, THE TEST WILL BE ABORTED UNLESS RETRIES
ARE TURNED ON.

!..
label
BLOCK1,
BLOCK2;

if .SWP_TRACE then PRINTF (DBM31);          ! TEST 25

NUM_RETRIES = ZERO;

while (.NUM_RETRIES lequ .SWP_RETRIES) do
begin
TIP = ALL_ONES;                          ! TELL READ_CMD NOT TO WAIT FOR
! RECEIVE STATUS

! GET AZTEC READY FOR OPERATION

if AZTEC_READY ()                          ! IF FAILURE REPORT ERROR
then
begin
ERRDF (B1, AZT_READY_ERR, 0);             !
if .RET_STATUS then DECODE ();            ! DECODE THE STATUS, IF ANY
RETRIES = TRUE;                           ! SET RETRIES FLAG
end
else
BLOCK1 :
begin
! SEEK BETWEEN BEGINNING TRACK AND ENDING TRACK

BYTE_COUNT = ZERO;                        ! BYTE COUNTS ZERO
BUF_DESCPTR = ZERO;                       ! CLEAR BUFFER DESCRIPTOR
!
! FILL THE COMMUNICATION COMMAND RING SLOTS
! WITH READ COMMANDS
LBN_SZ = .SIZ_LBN;                        ! LBN INCREMENT SIZE
OUT_BOUND = ZERO;                         ! INIT COMMAND COUNT

```

```

: 4755 5      IN_BOUND = ZERO;                ! INIT RECEIVE COUNT
: 4756 5
: 4757 5      incru I from 0 to SND_ALLOCATE - 1 do ! FILL COMMAND BUFFER WITH
: 4758 6      begin                          ! SEEK COMMANDS (16 SLOTS)
: 4759 6
: 4760 6      if .OUT_BOUND then LBN_ED = 820 else LBN_ED = 0;
: 4761 6
: 4762 6      LBN_ST = (.OFFSET * .LBN_ED)*.LBN_SZ; ! GET STARTING LBN
: 4763 6      CMD_REF = .LBN_ST;                ! LBN_ST WILL BE USED CMD_LREF
: 4764 6      ! SO THAT FAILING LBN CAN BE
: 4765 6      ! FOUND IN RECEIVE ENVELOPE
: 4766 6      READ CMD ();                    ! ISSUE READ COMMAND
: 4767 6      OUT_BOUND = .OUT_BOUND + 1;
: 4768 6
: 4769 6      if GET_CMD_SLOT () then exitloop; ! GET NEXT COMMAND SLOT
: 4770 6
: 4771 5      end;
: 4772 5
: 4773 5      !
: 4774 5      ! INIT AND START THE CLOCK
: 4775 5      !
: 4776 5      CLOCK_INIT ();                  ! INIT CLOCK VARIABLES
: 4777 5      TEMP = .RC25_ADDR [RCIP, RC_ALL]; ! READ IP & CONTROLLER START TO POLL
: 4778 5      !
: 4779 5      !
: 4780 5      while .IN_BOUND lequ 1000 do    ! DO SEEK FROM STARTING TRACK
: 4781 6      begin
: 4782 6
: 4783 6      if REC_STATUS ()                ! POLL RECEIVE RING FOR HOST
: 4784 6      then                          ! OWNERSHIP BIT.
: 4785 6      then                          ! IF ERROR, REPORT ERROR
: 4786 7      begin
: 4787 7      TEMP = .IN_BOUND;                ! SAVE RECEIVE COUNT
: 4788 7      ERRDF (82, MSG_SEEK_ERR, 0);    !
: 4789 7      PRINTB (FMT11, .TEMP, .LBN_ST); !
: 4790 7      DECODE ();                      ! DECODE END PACKET STATUS
: 4791 7      RETRIES = TRUE;
: 4792 7      leave BLOCK1;                  ! AND ABORT TEST
: 4793 7      end
: 4794 6      else
: 4795 6      BLOCK2 :
: 4796 7      begin
: 4797 7
: 4798 7      while .OUT_BOUND lequ 1000 do
: 4799 8      begin
: 4800 8      LBN_ST = (.OFFSET * .LBN_ED)*.LBN_SZ; ! GET STARTING LBN
: 4801 8      READ_FILL_CMD ();                ! ISSUE READ COMMAND
: 4802 8      OUT_BOUND = .OUT_BOUND + 1;
: 4803 8
: 4804 8      if .OUT_BOUND then LBN_ED = 820 else LBN_ED = 0;
: 4805 8
: 4806 8      if GET_CMD_SLOT () then leave BLOCK2; ! GET NEXT COMMAND SLOT
: 4807 8
: 4808 8      if (.OUT_BOUND - .IN_BOUND) eglu 16 then leave BLOCK2;
: 4809 8
: 4810 8      ! MAINTAIN A QUEUE LENGTH OF 15
: 4811 7      end;

```

```

: 4812 7
: 4813 6
: 4814 6
: 4815 6
: 4816 6
: 4817 5
: 4818 5
: 4819 5
: 4820 5
: 4821 5
: 4822 5
: 4823 5
: 4824 5
: 4825 5
: 4826 5
: 4827 5
: 4828 4
: 4829 4
: 4830 4
: 4831 4
: 4832 4
: 4833 4
: 4834 3
: 4835 3
: 4836 3
: 4837 1

```

```

                end;
                TEMP = .RC25_ADDR [RCIP, RC_ALL];    ! READ IP AND CONTROLLER WILL
                                                    ! START TO POLL
                end;
!
! STOP THE CLOCK
!
                .CLK_CSR = ZERO;                    ! STOP THE CLOCK
                TOTAL SEEKS = 1000
                DATA4 = .MINUTES*60 * .SECONDS;    ! MSEC./SEEK
                DATA3 = .TICKS*100/.CLK_HERTZ;    ! 100TH OF MSEC./SEEK
                PRINTB (MG_SKF_TIME, .DATA4, .DATA3); ! PRINT MESSAGE 'AVERAGE SEEK TIME'
                end;
                if (.RETRIES) then DO_RETRIES ();
                if (.NUM_RETRIES eq 0) then exitloop;
                end;
return;
ENDTST;

```

```

000000 010146          $T25:  .SBTTL  $T25 TEST SECTION
000002 024646          MOV    R1, -(SP) ;
000004 032767 000001 000000G  CMP    -(SP), -(SP) ;
000012 001407          BIT    #1, SWP.TRACE ;
000014 012746 000000G  BEQ    1$ ;
000020 012746 000001  MOV    #DBM31, -(SP)
000024 010600          MOV    #1, -(SP)
000026 104417          MOV    SP, R0 ; SP,*
000030 022626          TRAP  17
000032 005067 000000G  1$:  CLR    NUM.RETRIES ;
000036 026767 000000G 000000G  2$:  CMP    NUM.RETRIES, SWP.RETRIES ;
000044 101402          BLOS  3$ ;
000046 000167 000652  JMP    22$ ;
000052 012767 177777 000000G  3$:  MOV    #-1, TIP ;
000060 004767 000000G  JSR    PC, AZTEC.READY ;
000064 006000          ROR    R0 ;
000066 103017          BCC  5$ ;
000070 104455          TRAP  55 ;
000072 000121          .WORD 121 ;
000074 000000G      .WORD AZT.READY.ERR ;
000076 000000          .WORD 0 ;
000100 032767 000001 000000G  BIT    #1, RET.STATUS ;
000106 001402          BEQ    4$ ;
000110 004767 000000G  JSR    PC, DECODE ;
000114 012767 000001 000000G  4$:  MOV    #1, RETRIES ;
000122 000167 000550  JMP    20$ ;
000126 005067 000000G  5$:  CLR    BYTE.COUNT ;
000132 005067 000000G  CLR    BUF.DESCPTR ;

```


C10

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0326
Page 131
(24)

000136	016767	000000G	000000G		MOV	SIZ.LBN,LBN.SZ	:		4753
000144	005067	000000G			CLR	OUT.BOUND	:		4754
000150	005067	000000G			CLR	IN.BOUND	:		4755
000154	005001				CLR	R1	:	I	4757
000156	032767	000001	000000G	6#:	BIT	#1,OUT.BOUND	:		4760
000164	001404				BEQ	7#	:		
000166	012767	001464	000000G		MOV	#1464,LBN.ED	:		
000174	000402				BR	8#	:		
000176	005067	000000G		7#:	CLR	LBN.ED	:		
000202	016746	000000G		8#:	MOV	OFFSET,-(SP)	:		4762
000206	066716	000000G			ADD	LBN.ED,(SP)	:		
000212	016746	000000G			MOV	LBN.SZ,-(SP)	:		
000216	004767	000000G			JSR	PC,BL#MUL	:		
000222	010067	000000G			MOV	RO,LBN.ST	:		
000226	016767	000000G	000000G		MOV	LBN.ST,CMD.REF	:		4763
000234	004767	000000G			JSR	PC,READ.CMD	:		4766
000240	005267	000000G			INC	OUT.BOUND	:		4767
000244	004767	000000G			JSR	PC,GET.CMD.SLOT	:		4769
000250	006000				ROR	RO	:		
000252	103002				BCC	9#	:		
000254	022626				CMP	(SP)*,(SP)*	:		
000256	000405				BR	10#	:		
000260	022626			9#:	CMP	(SP)*,(SP)*	:		4758
000262	005201				INC	R1	:	I	4757
000264	020127	000017			CMP	R1,#17	:	I,*	
000270	101732				BLOS	6#	:		
000272	004767	000000G		10#:	JSR	PC,CLOCK.INIT	:		4776
000276	017716	000000G			MOV	@RC25.ADDR,(SP)	:	*,RC.REG	4777
000302	011667	000000G			MOV	(SP),TEMP	:	RC.REG,*	
000306	026727	000000G	001750	11#:	CMP	IN.BOUND,#1750	:		4780
000314	101123				BHI	18#	:		
000316	004767	000000G			JSR	PC,REC.STATUS	:		4783
000322	006000				ROR	RO	:		
000324	103033				BCC	12#	:		
000326	016767	000000G	000000G		MOV	IN.BOUND,TEMP	:		4787
000334	104455				TRAP	55	:		4788
000336	000122				.WORD	122	:		
000340	000000G				.WORD	MSG.SEEK.ERR	:		
000342	000000				.WORD	0	:		
000344	016746	000000G			MOV	LBN.ST,-(SP)	:		4789
000350	016746	000000G			MOV	TEMP,-(SP)	:		
000354	012746	000000G			MOV	#FMT11,-(SP)	:		
000360	012746	000003			MOV	#3,-(SP)	:		
000364	010600				MOV	SP,RO	:	SP,*	
000366	104414				TRAP	14	:		
000370	062706	000010			ADD	#10,SP	:		
000374	004767	000000G			JSR	PC,DECODE	:		4790
000400	012767	000001	000000G		MOV	#1,RETRIES	:		4791
000406	162706	000016			SUB	#16,SP	:		4786
000412	000527				BR	19#	:		
000414	026727	000000G	001750	12#:	CMP	OUT.BOUND,#1750	:		4798
000422	101051				BHI	17#	:		
000424	016746	000000G			MOV	OFFSET,-(SP)	:		4800
000430	066716	000000G			ADD	LBN.ED,(SP)	:		
000434	016746	000000G			MOV	LBN.SZ,-(SP)	:		
000440	004767	000000G			JSR	PC,BL#MUL	:		
000444	010067	000000G			MOV	RO,LBN.ST	:		

D10

ZRCFR3	CZRCFC0	RC25	FR	END	TEST	27-Mar-1985	15:27:28	VAX-11	Bliss-16	V4.0-579	SEQ 0327
V03.0	TEST	SECTION				27-Mar-1985	13:28:18	USER#1:	[AZTEC.CZRCFC]	ZRCFC3.B16;4	Page 132
											(24)
000450	004767	000000G									4801
000454	005267	000000G									4802
000460	032767	000001 000000G									4804
000466	001404										
000470	012767	001464 000000G									
000476	000402										
000500	005067	000000G			13#:						
000504	004767	000000G			14#:						4806
000510	006000										
000512	103002										
000514	022626										
000516	000413										
000520	016700	000000G			15#:						4808
000524	062700	000020									
000530	026700	000000G									
000534	001002										
000536	022626										
000540	000402										
000542	022626				16#:						4799
000544	000723										4798
000546	017766	000000G 000002			17#:						4815
000554	016667	000002 000000G									
000562	000651										4780
000564	005077	000000G			18#:						4822
000570	016746	000000G									4824
000574	012746	000074									
000600	004767	000000G									
000604	066700	000000G									
000610	010067	000000G									
000614	016716	000000G									4825
000620	012746	000144									
000624	004767	000000G									
000630	010016										
000632	016746	000000G									
000636	004767	000000G									
000642	010067	000000G									
000646	016716	000000G									4826
000652	016746	000000G									
000656	012746	000000G									
000662	012746	000003									
000666	010600										
000670	104414										
000672	062706	000016			19#:						4733
000676	032767	000001 000000G			20#:						4830
000704	001402										
000706	004767	000000G									
000712	005767	000000G			21#:						4832
000716	001402										
000720	000167	177112									
000724	022626				22#:						4696
000726	012601										
000730	000207										

; Routine Size: 237 words, Routine Base: AC\$CODE + 22032
; Maximum stack depth per invocation: 12 words

```

000000 004767 177042          T25::      .SBTTL  T25 TEST SECTION
000000          1$:      JSR      PC,$T25
000004 104466          TRAP    66
000006 006000          ROR     RO
000010 103773          BLO    1$
000012 000207          RTS     PC

```

4836

```

; Routine Size: 6 words,      Routine Base: AC$CODE + 22764
; Maximum stack depth per invocation: 2 words

```

```

: 4838 1  !
: 4839 3  !BGNTST;
: 4840 3  ! TEST #26 - WRITE DATA TEST
: 4841 3  !
: 4842 3  ! DESCRIPTION:
: 4843 3  !
: 4844 3  ! THIS TEST BRINGS RC25 CONTROLLER AND SELECTED UNIT ONLINE.
: 4845 3  ! THEN LOADS DM CODE VECTOR ARRAY DM_26 TO THE CONTROLLERS
: 4846 3  ! MEMORY BY ISSUING EX_SUP_PROG COMMAND.
: 4847 3  !
: 4848 3  ! THE DMCODE GETS THE UNIT NUMBER FROM THE HOST AND ATTEMPTS
: 4849 3  ! TO FIND AT LEAST ONE GOOD DIAGNOSTIC BLOCK ON EACH SURFACE
: 4850 3  ! OF THE PLATTER SPECIFIED AND MAKE SURE THAT DMCODE CAN READ
: 4851 3  ! AND WRITE TO THE BLOCK IN ORDER TO VERIFY THAT THE HEADS ARE
: 4852 3  ! WORKING PROPERLY. FIRST TOP SURFACE WILL BE ATTEMPTED WITH
: 4853 3  ! ALL ONES DATA AND SECOND ALL ZERO DATA. THIS WILL BE
: 4854 3  ! REPEATED FOR BOTTOM SURFACE AS WELL. THE DATA WRITTEN WILL BE
: 4855 3  ! READ AND COMPARED.
: 4856 3  !
: 4857 3  ! THE ERROR REPORT ON THIS TEST WILL INCLUDE DATA WRITTEN, DATA READ
: 4858 3  ! PLUS THE TRACK, HEAD AND SECTOR NUMBER. ALSO ERROR STATUS
: 4859 3  ! FROM THE MICRO CODE IF ANY WILL BE REPORTED. ERROR STATUS OF
: 4860 3  ! ZERO WILL MEAN OTHER ERRORS TRAPPED IN DMCODE. AFTER REPORTING
: 4861 3  ! THE ERROR THE REST OF THE TEST WILL BE ABORTED UNLESS THE OPERATOR
: 4862 3  ! SELECTS RETRIES.
: 4863 3  !
: 4864 3  !--
: 4865 3  !
: 4866 3  label
: 4867 3  BLOCK1;
: 4868 3  !
: 4869 3  if .SWP_TRACE then PRINTF (DBM32);          ! TEST 26
: 4870 3  !
: 4871 3  NUM_RETRIES = ZERO;
: 4872 3  !
: 4873 3  while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 4874 4  begin
: 4875 4  TIP = 26;
: 4876 4  ! GET AZTEC READY FOR OPERATION
: 4877 4  !
: 4878 4  if AZTEC_READY ( )          ! IF FAILURE REPORT ERROR
: 4879 4  then
: 4880 5  begin

```

```

: 4881 5      ERRDF (83, AZT_READY_ERR, 0);      !
: 4882 5
: 4883 5      if .RET_STATUS then DECODE ();      ! DECODE THE STATUS, IF ANY
: 4884 5
: 4885 5      RETRIES = TRUE;                    ! SET RETRIES FLAG
: 4886 5      end
: 4887 4      else
: 4888 4      BLOCK1 :
: 4889 5      begin
: 4890 5
: 4891 5      ! ISSUE AN EX_SUP_PROG COMMAND WITH START ADDRESS OF
: 4892 5      ! DM_26 VECTOR ARRAY AND BYTE COUNT.
: 4893 5      CMD_REF = .CMD_SLOT;                ! COMMAND REFERENCE NUMBER
: 4894 5      BUF_DESCRPT = DM_26;                ! DM CODE STARTING ADDRESS
: 4895 5      BYTE_COUNT = 413*2;                ! BYTE COUNTS VER:C
: 4896 5
: 4897 5      if EX_SUP_PRG ()                    ! ISSUE AN EXECUTE SUPPLIED COMMAND
: 4898 5      then                                ! REPORT IF FAILED
: 4899 6      begin
: 4900 6      ERRDF (84, EXE_SUP_ERR, 0);
: 4901 6
: 4902 6      if .RET_STATUS then DECODE ();      ! DECODE STATUS
: 4903 6
: 4904 6      RETRIES = TRUE;
: 4905 6      leave BLOCK1;                       ! ABORT TEST
: 4906 5      end;
: 4907 5
: 4908 5      CMD_REF = .CMD_SLOT;                ! COMMAND REFERENCE
: 4909 5      BUF_DESCRPT = UNIT;                 ! DESCRIPTOR ADDRESS
: 4910 5      BYTE_COUNT = 02;                    ! TOTAL BYTES TO BE TRANSFERRED
: 4911 5
: 4912 5      if SEND_DATA ()                    ! ISSUE SEND DATA COMMAND
: 4913 5      then                                ! IF STATUS BIT INDICATES ERROR
: 4914 6      begin                                ! THEN REPORT ERROR
: 4915 6      ERRDF (85, SND_DATA_ERR, 0);        !
: 4916 6
: 4917 6      if .RET_STATUS then DECODE ();      ! DECODE RETURN STATUS
: 4918 6
: 4919 6      RETRIES = TRUE;
: 4920 6      leave BLOCK1;
: 4921 5      end;
: 4922 5
: 4923 5      ! ISSUE A REC_DATA COMMAND AND WAIT FOR END PACKET
: 4924 5      ! TO GET THE STATUS SENT BY DM CODE AFTER DOING
: 4925 5      ! HEAD SWITCH TEST.
: 4926 5      CMD_REF = .CMD_SLOT;                ! COMMAND REFERENCE #
: 4927 5      BUF_DESCRPT = RCV_DATA_BUF [0];     ! SET THE BUFFER AREA TO
: 4928 5      ! RECEIVE 6 WORDS FROM DM CODE
: 4929 5      BYTE_COUNT = 12;                    ! SET BYTE COUNTS = 12
: 4930 5
: 4931 5      if REC_DATA ()                      ! SEND A RECEIVE DATA COMMAND
: 4932 5      then                                ! IF FAILURE REPORT ERROR
: 4933 6      begin
: 4934 6      ERRDF (86, RE_DATA_ERR, 0);
: 4935 6
: 4936 6      if .RET_STATUS then DECODE ();      ! DECODE STATUS
: 4937 6

```

ZRCFB3
V03.0CZRCFC0 RC25 FR END TEST
TEST SECTION27-Mar-1985 15:27:28
27-Mar-1985 13:28:18VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC3.B16;4SEQ 0330
Page 135
(24)

```

: 4938 6      RETRIES = TRUE;
: 4939 6      leave BLOCK1;
: 4940 5      end;
: 4941 5
: 4942 5      ! CHECK DM CODE FLAG FOR SUCCESS. IF FAILURE REPORT ERROR
: 4943 5
: 4944 5      if .RCV_DATA_BUF [0] nequ #0'177777'      ! IF NOT SUCCESS, REPORT ERROR
: 4945 5      then
: 4946 6          begin
: 4947 6              ERRDF (87, MSG_WRITE_ERR, 0);          ! REPORT WRITE DATA FAILURE
: 4948 6              PRINTB (FMT16, .RCV_DATA_BUF [1], .RCV_DATA_BUF [2]);
: 4949 6              PRINTB (FMT17, .RCV_DATA_BUF [3], .RCV_DATA_BUF [4], .RCV_DATA_BUF [5]);      ! PRINT UNIT, HEAD AND
: 4950 6              PRINTB (FMT20, .RCV_DATA_BUF [0]);      ! TRACK NUMBER
: 4951 6              RETRIES = TRUE;
: 4952 6              end;
: 4953 5
: 4954 5      end;
: 4955 4
: 4956 4      if (.RETRIES) then DO_RETRIES ();
: 4957 4
: 4958 4      if (.NUM_RETRIES eqlu ZERO) then exitloop;
: 4959 4
: 4960 4      end;
: 4961 3      return;
: 4962 3      ENDTST;
: 4963 3
: 4964 1

```

000000	032767	000001	000000G	\$T26:	.SBTTL \$T26 TEST SECTION		
000006	001407				BIT #1,SWP.TRACE	:	4869
000010	012746	000000G			BEQ 1\$		
000014	012746	000001			MOV #DBM32,-(SP)		
000020	010600				MOV #1,-(SP)		
000022	104417				MOV SP,RO	: SP,*	
000024	022626				TRAP 17		
000026	005067	000000G		1\$:	CMP (SP)*,(SP)*		
000032	026767	000000G	000000G	2\$:	CLR NUM.RETRIES	:	4871
000040	101401				CMP NUM.RETRIES,SWP.RETRIES	:	4873
000042	000207				BLOS 3\$		
000044	012767	000032	000000G	3\$:	RTS PC		
000052	004767	000000G			MOV #32,TIP	:	4875
000056	006000				JSR PC,AZTEC.READY	:	4878
000060	103016				ROR RO		
000062	104455				BCC 5\$		
000064	000123				TRAP 55	:	4881
000066	000000G				.WORD 123		
000070	000000				.WORD AZT.READY.ERR		
000072	032767	000001	000000G		.WORD 0		
000100	001402				BIT #1,RET.STATUS	:	4883
000102	004767	000000G			BEQ 4\$		
000106	012767	000001	000000G	4\$:	JSR PC,DECODE		
000114	000574				MOV #1,RETRIES	:	4885
000116	016767	000000G	000000G	5\$:	BR 12\$:	4878
000124	012767	000000G	000000G		MOV CMD.SLOT,CMD.REF	:	4893
000132	012767	001472	000000G		MOV #DM.26,BUF.DESCRPTR	:	4894
					MOV #1472,BYTE.COUNT	:	4895

H10

ZRCFB3 V03.0	CZRCFC0 RC25 FR END TEST TEST SECTION	27-Mar-1985 15:27:28 27-Mar-1985 13:28:18	VAX-11 Bliss-16 V4.0-579 USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4	4897
000140	004767 000000G		JSR PC,EX.SUP.PRG ;	4897
000144	006000		ROR RO ;	
000146	103016		BCC 7\$;	
000150	104455		TRAP 55 ;	4900
000152	000124		.WORD 124 ;	
000154	000000G		.WORD EXE.SUP.ERR ;	
000156	000000		.WORD 0 ;	
000160	032767 000001 000000G		BIT #1,RET.STATUS ;	4902
000166	001402		BEQ 6\$;	
000170	004767 000000G		JSR PC,DECODE ;	
000174	012767 000001 000000G	6\$:	MOV #1,RETRIES ;	4904
000202	000541		BR 12\$;	4899
000204	016767 000000G 000000G	7\$:	MOV CMD.SLOT,CMD.REF ;	4908
000212	012767 000000G 000000G		MOV #UNIT,BUF.DESCRPTR ;	4909
000220	012767 000002 000000G		MOV #2,BYTE.COUNT ;	4910
000226	004767 000000G		JSR PC,SEND.DATA ;	4912
000232	006000		ROR RO ;	
000234	103016		BCC 9\$;	
000236	104455		TRAP 55 ;	4915
000240	000125		.WORD 125 ;	
000242	000000G		.WORD SND.DATA.ERR ;	
000244	000000		.WORD 0 ;	
000246	032767 000001 000000G		BIT #1,RET.STATUS ;	4917
000254	001402		BEQ 8\$;	
000256	004767 000000G		JSR PC,DECODE ;	
000262	012767 000001 000000G	8\$:	MOV #1,RETRIES ;	4919
000270	000506		BR 12\$;	4914
000272	016767 000000G 000000G	9\$:	MOV CMD.SLOT,CMD.REF ;	4926
000300	012767 000000G 000000G		MOV #RCV.DATA.BUF,BUF.DESCRPTR ;	4927
000306	012767 000014 000000G		MOV #14,BYTE.COUNT ;	4929
000314	004767 000000G		JSR PC,REC.DATA ;	4931
000320	006000		ROR RO ;	
000322	103016		BCC 11\$;	
000324	104455		TRAP 55 ;	4934
000326	000126		.WORD 126 ;	
000330	000000G		.WORD RE.DATA.ERR ;	
000332	000000		.WORD 0 ;	
000334	032767 000001 000000G		BIT #1,RET.STATUS ;	4936
000342	001402		BEQ 10\$;	
000344	004767 000000G		JSR PC,DECODE ;	
000350	012767 000001 000000G	10\$:	MOV #1,RETRIES ;	4938
000356	000453		BR 12\$;	4933
000360	026727 000000G 177777	11\$:	CMP RCV.DATA.BUF,#-1 ;	4944
000366	001447		BEQ 12\$;	
000370	104455		TRAP 55 ;	4947
000372	000127		.WORD 127 ;	
000374	000000G		.WORD MSG.WRITE.ERR ;	
000376	000000		.WORD 0 ;	
000400	016746 000004G		MOV RCV.DATA.BUF+4,-(SP) ;	4948
000404	016746 000002G		MOV RCV.DATA.BUF+2,-(SP) ;	
000410	012746 000000G		MOV #FMT16,-(SP) ;	
000414	012746 000003		MOV #3,-(SP) ;	
000420	010600		MOV SP,RO ; SP,*	
000422	104414		TRAP 14 ;	
000424	016716 000012G		MOV RCV.DATA.BUF+12,(SP) ;	4949
000430	016746 000010G		MOV RCV.DATA.BUF+10,-(SP) ;	
000434	016746 000006G		MOV RCV.DATA.BUF+6,-(SP) ;	

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER\$1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0332
Page 137
(24)

```

000440 012746 000000G      MOV      #FMT17,-(SP)
000444 012746 0000004      MOV      #4,-(SP)
000450 010600              MOV      SP,RO      ; SP,*
000452 104414              TRAP     14
000454 016716 000000G      MOV      RCV.DATA.BUF,(SP)      ;
000460 012746 000000G      MOV      #FMT20,-(SP)      ;
000464 012746 0000002      MOV      #2,-(SP)
000470 010600              MOV      SP,RO      ; SP,*
000472 104414              TRAP     14
000474 012767 0000001 000000G      MOV      #1,RETRIES      ;
000502 062706 0000024              ADD      #24,SP      ;
000506 032767 0000001 000000G      BIT      #1,RETRIES      ;
000514 001402              BEQ      13$
000516 004767 000000G      JSR      PC,DO.RETRIES
000522 005767 000000G      TST     NUM.RETRIES      ;
000526 001402              BEQ      14$
000530 000167 177276              JMP      2$
000534 000207              RTS      PC      ;

```

; Routine Size: 175 words, Routine Base: AC\$CODE + 23000
; Maximum stack depth per invocation: 12 words

```

000000 004767 177236      T26::   .SBTTL  T26 TEST SECTION
000000 1$:      JSR      PC,$T26      ;
000004 104466              TRAP     66
000006 006000              ROR      RO
000010 103773              BLO      1$
000012 000207              RTS      PC

```

; Routine Size: 6 words, Routine Base: AC\$CODE + 23536
; Maximum stack depth per invocation: 2 words

; 4965 1 !<BLF/PAGE>


```

: 5023 6      begin
: 5024 6      ERRDF (89, EXE_SUP_ERR, 0);
: 5025 6
: 5026 6      if .RET_STATUS then DECODE ();      ! DECODE STATUS
: 5027 6
: 5028 6      RETRIES = TRUE;
: 5029 6      leave BLOCK1;                          ! ABORT TEST
: 5030 5      end;
: 5031 5
: 5032 5      CMD_REF = .CMD_SLOT;                    ! COMMAND REFERENCE
: 5033 5      BUF_DESCRIPTOR = UNIT;                  ! DESCRIPTOR ADDRESS
: 5034 5      BYTE_COUNT = 02;                       ! TOTAL BYTES TO BE TRANSFERRED
: 5035 5
: 5036 5      if SEND_DATA ()                        ! ISSUE SEND DATA COMMAND
: 5037 5      then                                  ! IF STATUS BIT INDICATES ERROR
: 5038 6      begin                                  ! THEN REPORT ERROR
: 5039 6      ERRDF (90, SND_DATA_ERR, 0);          !
: 5040 6
: 5041 6      if .RET_STATUS then DECODE ();      ! DECODE RETURN STATUS
: 5042 6
: 5043 6      RETRIES = TRUE;
: 5044 6      leave BLOCK1;
: 5045 5      end;
: 5046 5
: 5047 5      ! ISSUE A REC_DATA COMMAND AND WAIT FOR END PACKET
: 5048 5      ! TO GET THE STATUS SENT BY DM CODE AFTER DOING
: 5049 5      ! HEAD SWITCH TEST.
: 5050 5      CMD_REF = .CMD_SLOT;                    ! COMMAND REFERENCE #
: 5051 5      BUF_DESCRIPTOR = RCV_DATA_BUF [0];    ! SET THE BUFFER AREA TO
: 5052 5      ! RECEIVE 2 WORDS FROM DM CODE
: 5053 5      BYTE_COUNT = 4;                        ! SET BYTE COUNTS = 4
: 5054 5
: 5055 5      if REC_DATA ()                        ! SEND A RECEIVE DATA COMMAND
: 5056 5      then                                  ! IF FAILURE REPORT ERROR
: 5057 6      begin
: 5058 6      ERRDF (91, RE_DATA_ERR, 0);
: 5059 6
: 5060 6      if .RET_STATUS then DECODE ();      ! DECODE STATUS
: 5061 6
: 5062 6      RETRIES = TRUE;
: 5063 6      leave BLOCK1;
: 5064 5      end;
: 5065 5
: 5066 5      ! CHECK DM CODE FLAG FOR SUCCESS. IF FAILURE REPORT ERROR
: 5067 5
: 5068 5      if .RCV_DATA_BUF [0] nequ #0'104'    ! IF NOT SUCCESS, REPORT ERROR
: 5069 5      then
: 5070 6      begin
: 5071 6      ERRDF (92, MSG_READ_ERR, 0);          ! REPORT READ ACCESS FAILURE
: 5072 6      RETRIES = TRUE;
: 5073 6      end
: 5074 5      else
: 5075 6      begin
: 5076 6      ! OFFSET RECEIVED MULTIPLIED BY 4/10 GIVES # OFFSET
: 5077 6      DATA4 = .RCV_DATA_BUF [1]*2/5;        ! OFFSET EXPRESSED AS #
: 5078 6      DATA3 = ((.RCV_DATA_BUF [1]*2) mod 5)*2;
: 5079 6      PRINTB (FMT12, .DATA4, .DATA3);        ! PRINT OFFSET VALUE

```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER\$1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0335
Page 140
(25)

```

: 5080 5      end;
: 5081 5
: 5082 4      end;
: 5083 4
: 5084 4      if (.RETRIES) then DO_RETRIES ();
: 5085 4
: 5086 4      if (.NUM_RETRIES eqv ZERO) then exitloop;
: 5087 4
: 5088 3      end;
: 5089 3
: 5090 3      return;
: 5091 1      ENDTST;
    
```

000000	010146			.SBTTL	\$T27 TEST SECTION		
000002	032767	000001	000000G	\$T27:	MOV R1,-(SP)	:	4964
000010	001407				BIT #1,SWP.TRACE	:	4993
000012	012746	000000G			BEQ 1\$		
000016	012746	000001			MOV #DBM36,-(SP)		
000022	010600				MOV #1,-(SP)		
000024	104417				MOV SP,R0	: SP,*	
000026	022626				TRAP 17		
000030	005067	000000G			CMP (SP), (SP)		
000034	026767	000000G	000000G	1\$:	CLR NUM.RETRIES	:	4995
000042	101402			2\$:	CMP NUM.RETRIES,SWP.RETRIES	:	4997
000044	000167	000472			BLOS 3\$		
000050	012767	000033	000000G	3\$:	JMP 15\$		
000056	004767	000000G			MOV #33,TIP	:	4999
000062	006000				JSR PC,AZTEC.READY	:	5002
000064	103016				ROR R0		
000066	104455				BCC 5\$		
000070	000130				TRAP 55	:	5005
000072	000000G				.WORD 130		
000074	000000				.WORD AZT.READY.ERR		
000076	032767	000001	000000G		.WORD 0		
000104	001402				BIT #1,RET.STATUS	:	5007
000106	004767	000000G			BEQ 4\$		
000112	012767	000001	0000J0G	4\$:	JSR PC,DECODE		
000120	000575				MOV #1,RETRIES	:	5009
000122	016767	000000G	000000G	5\$:	BR 13\$:	5002
000130	012767	000000G	000000G		MOV CMD.SLOT,CMD.REF	:	5017
000136	012767	001146	000000G		MOV #DM.27,BUF.DESCRPTR	:	5018
000144	004767	000000G			MOV #1146,BYTE.COUNT	:	5019
000150	006000				JSR PC,EX.SUP.PRG	:	5021
000152	103016				ROR R0		
000154	104455				BCC 7\$		
000156	000131				TRAP 55	:	5024
000160	000000G				.WORD 131		
000162	000000				.WORD EXE.SUP.ERR		
000164	032767	000001	000000G		.WORD 0		
000172	001402				BIT #1,RET.STATUS	:	5026
000174	004767	000000G			BEQ 6\$		
000200	012767	000001	000000G	6\$:	JSR PC,DECODE		
000206	000542				MOV #1,RETRIES	:	5028
000210	016767	000000G	000000G	7\$:	BR 13\$:	5023
000216	012767	000000G	000000G		MOV CMD.SLOT,CMD.REF	:	5032
					MOV #UNIT,BUF.DESCRPTR	:	5033

M10

ZRCFB3 V03.0	CZRCFC0 RC25 FR END TEST TEST SECTION	27-Mar-1985 15:27:28 27-Mar-1985 13:28:18	VAX-11 Bliss-16 V4.0-579 USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4	
000224	012767 000002 000000G			MOV #2,BYTE.COUNT ; 5034
000232	004767 000000G			JSR PC,SEND.DATA ; 5036
000236	006000			ROR R0 ;
000240	103016			BCC 9# ;
000242	104455			TRAP 55 ; 5039
000244	000132			.WORD 132 ;
000246	000000G			.WORD SND.DATA.ERR ;
000250	000000			.WORD 0 ;
000252	032767 000001 000000G			BIT #1,RET.STATUS ; 5041
000260	001402			BEQ 8# ;
000262	004767 000000G			JSR PC,DECODE ;
000266	012767 000001 000000G	8#:		MOV #1,RETRIES ; 5043
000274	000507			BR 13# ; 5038
000276	016767 000000G 000000G	9#:		MOV CMD.SLOT,CMD.REF ; 5050
000304	012767 000000G 000000G			MOV #RCV.DATA.BUF,BUF.DESCRPTR ; 5051
000312	012767 000004 000000G			MOV #4,BYTE.COUNT ; 5053
000320	004767 000000G			JSR PC,REC.DATA ; 5055
000324	006000			ROR R0 ;
000326	103016			BCC 11# ;
000330	104455			TRAP 55 ; 5058
000332	000133			.WORD 133 ;
000334	000000G			.WORD RE.DATA.ERR ;
000336	000000			.WORD 0 ;
000340	032767 000001 000000G			BIT #1,RET.STATUS ; 5060
000346	001402			BEQ 10# ;
000350	004767 000000G			JSR PC,DECODE ;
000354	012767 000001 000000G	10#:		MOV #1,RETRIES ; 5062
000362	000454			BR 13# ; 5057
000364	026727 000000G 000104	11#:		CMP RCV.DATA.BUF,#104 ; 5068
000372	001410			BEQ 12# ;
000374	104455			TRAP 55 ; 5071
000376	000134			.WORD 134 ;
000400	000000G			.WORD MSG.READ.ERR ;
000402	000000			.WORD 0 ;
000404	012767 000001 000000G			MOV #1,RETRIES ; 5072
000412	000440			BR 13# ; 5068
000414	016746 000002G	12#:		MOV RCV.DATA.BUF+2,-(SP) ; 5077
000420	006316			ASL (SP) ;
000422	012746 000005			MOV #5,-(SP) ;
000426	004767 000000G			JSR PC,BL\$DIV ;
000432	010067 000000G			MOV R0,DATA4 ;
000436	016716 000002G			MOV RCV.DATA.BUF+2,(SP) ; 5078
000442	006316			ASL (SP) ;
000444	012746 000005			MOV #5,-(SP) ;
000450	004767 000000G			JSR PC,BL\$MOD ;
000454	010001			MOV R0,R1 ;
000456	006301			ASL R1 ;
000460	010167 000000G			MOV R1,DATA3 ;
000464	016716 000000G			MOV DATA3,(SP) ; 5079
000470	016746 000000G			MOV DATA4,-(SP) ;
000474	012746 000000G			MOV #FMT12,-(SP) ;
000500	012746 000003			MOV #3,-(SP) ;
000504	010600			MOV SP,R0 ; SP,*
000506	104414			TRAP 14 ;
000510	062706 000014			ADD #14,SP ; 5075
000514	032767 000001 000000G	13#:		BIT #1,RETRIES ; 5084
000522	001402			BEQ 14# ;

N10

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

3337
Page 142
(25)

000524	004767	000000G		JSR	PC,DO.RETRIES		
000530	005767	000000G	144:	TST	NUM.RETRIES	:	5086
000534	001402			BEQ	154		
000536	000167	177272		JMP	24		
000542	012601		154:	MOV	(SP),R1	:	4964
000544	000207			RTS	PC		

: Routine Size: 179 words, Routine Base: AC\$CODE * 23552
: Maximum stack depth per invocation: 9 words

000000	004767	177226		.SBTTL	T27 TEST SECTION		
000000			T27::	JSR	PC,\$T27	:	5090
000004	104466		14:	TRAP	66		
000006	006000			ROR	RO		
000010	103773			BLO	14		
000012	000207			RTS	PC		

: Routine Size: 6 words, Routine Base: AC\$CODE * 24320
: Maximum stack depth per invocation: 2 words

: 5092 1 !<BLF/PAGE>

```

: 5093 1      !
: 5094 3      ! BGNTST;
: 5095 3      !
: 5096 3      !
: 5097 3      ! **
: 5098 3      ! TEST #28 - AVERAGE ROTATIONAL TIMING TEST
: 5099 3      !
: 5100 3      ! DESCRIPTION:
: 5101 3      !     THIS TEST WILL BRING RC25 CONTROLLER AND THE UNIT ONLINE.
: 5102 3      !
: 5103 3      !     THIS TEST WILL BE PERFORMED FROM THE HOST USING THE MSCP "READ"
: 5104 3      !     COMMAND. AN LBN WILL BE SELECTED RANDOMLY. ONE THOUSAND TWO
: 5105 3      !     BYTE COUNT READS OF THE SAME LBN WILL BE PERFORMED. THIS OPERATION
: 5106 3      !     WILL BE TIMED AND THE AVERAGE TIME WILL BE REPORTED.
: 5107 3      !
: 5108 3      !     IF THE OPERATOR HAS SELECTED RETRIES, THE TEST WILL BE REPEATED.
: 5109 3      ! --
: 5110 3      !
: 5111 3      ! label
: 5112 3      !     BLOCK1;
: 5113 3      !
: 5114 3      ! if .SWP_TRACE then PRINTF (DBM37);           ! TEST 28
: 5115 3      !
: 5116 3      ! NUM_RETRIES = ZERO;
: 5117 3      !
: 5118 3      ! while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 5119 4      !     begin
: 5120 4      !         TIP = ALL_ONES;           ! TELL READ_CMD NOT TO WAIT
: 5121 4      !                                     ! FOR REC_STATUS
: 5122 4      !     ! GET AZTEC READY FOR OPERATION
: 5123 4      !
: 5124 4      !         if AZTEC_READY ()           ! IF FAILURE REPORT ERROR
: 5125 4      !         then
: 5126 5      !             begin
: 5127 5      !                 ERRDF (93, AZT_READY_ERR, 0);           !
: 5128 5      !
: 5129 5      !                 if .RET_STATUS then DECODE ();           ! DECODE THE STATUS, IF ANY
: 5130 5      !
: 5131 5      !                 RETRIES = TRUE;           ! SET RETRIES FLAG
: 5132 5      !             end
: 5133 4      !         else
: 5134 4      !     BLOCK1 :
: 5135 5      !         begin
: 5136 5      !
: 5137 5      !         ! SEEK RANDOM SECTOR AND REPEAT SEEKING THE SAME SECTOR 1000 TIMES.
: 5138 5      !
: 5139 5      !             BYTE_COUNT = 2;           ! BYTE COUNTS
: 5140 5      !             BUF_DESCRPT = DATA1;           ! BUFFER DESCRIPTOR
: 5141 5      !
: 5142 5      !         ! FILL THE COMMUNICATION COMMAND RING SLOTS
: 5143 5      !         ! WITH READ COMMANDS
: 5144 5      !             P2 = .TICKS;           ! INIT P2 FOR RANDOM NUMBER
: 5145 5      !             RANDOM_NUM ();           ! GET RANDOM LBN
: 5146 5      !             LBN_ST = .P3;
: 5147 5      !             OUT_BOUND = ZERO;           ! CLEAR COMMAND COUNT
: 5148 5      !             IN_BOUND = ZERO;           ! CLEAR RECEIVE COUNT
: 5149 5

```


ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0340
Page 145
(26)

```

: 5207 5 ! TOTAL SEEKS = 1000
: 5208 5 DATA4 = .MINUTES*60 + .SECONDS; ! MSEC./SEEK
: 5209 5 DATA3 = .TICKS*100/.CLK HERTZ; ! 100TH OF MSEC./SEEK
: 5210 5 PRINTB (MSG_ROT_TIME, .DATA4, .DATA3); ! PRINT MESSAGE 'AVERAGE SEEK TIME'
: 5211 5 !
: 5212 4 end;
: 5213 4
: 5214 4 if (.RETRIES) then DO_RETRIES ();
: 5215 4
: 5216 4 if (.NUM_RETRIES eqv ZERO) then exitloop;
: 5217 4
: 5218 3 end;
: 5219 3
: 5220 3 return;
: 5221 1 ENDTST;

```

000000	010146			\$T28:	.SBTTL \$T28 TEST SECTION				
000002	024646				MOV R1, -(SP)				5091
000004	032767	000001	000000G		CMP -(SP), -(SP)				
000012	001407				BIT #1, SWP.TRACE				5114
000014	012746	000000G			BEQ 1\$				
000020	012746	000001			MOV #DBM37, -(SP)				
000024	010600				MOV #1, -(SP)				
000026	104417				MOV SP, R0				; SP, *
000030	022626				TRAP 17				
000032	005067	000000G		1\$:	CMP (SP), (SP)				
000036	026767	000000G	000000G	2\$:	CLR NUM.RETRIES				5116
000044	101402				CMP NUM.RETRIES, SWP.RETRIES				5118
000046	000167	000526			BLOS 3\$				
000052	012767	177777	000000G	3\$:	JMP 15\$				
000060	004767	000000G			MOV # -1, TIP				5120
000064	006000				JSR PC, AZTEC.READY				5124
000066	103017				ROR R0				
000070	104455				BCC 5\$				
000072	000135				TRAP 55				5127
000074	000000G				.WORD 135				
000076	000000				.WORD AZT.READY.ERR				
000100	032767	000001	000000G		.WORD 0				
000106	001402				BIT #1, RET.STATUS				5129
000110	004767	000000G			BEQ 4\$				
000114	012767	000001	000000G	4\$:	JSR PC, DECODE				
000122	000167	000424			MOV #1, RETRIES				5131
000126	012767	000002	000000G	5\$:	JMP 13\$				5124
000134	012767	000000G	000000G		MOV #2, BYTE.COUNT				5139
000142	016767	000000G	000000G		MOV #DATA1, BUF.DESCRPTR				5140
000150	004767	000000G			MOV TICKS, P2				5144
000154	016767	000000G	000000G		JSR PC, RANDOM.NUM				5145
000162	005067	000000G			MOV P3, LBN.ST				5146
000166	005067	000000G			CLR OUT.BOUND				5147
000172	005001				CLR IN.BOUND				5148
000174	016767	000000G	000000G	6\$:	CLR R1				5150
000202	004767	000000G			MOV LBN.ST, CMD.REF				5152
000206	005267	000000G			JSR PC, READ.CMD				5153
000212	004767	000000G			INC OUT.BOUND				5154
000216	006000				JSR PC, GET.CMD.SLOT				5156
					ROR R0				

E11

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC3.B16;4

SEQ 0341
Page 146
(26)

000220	103404			BLO	7\$			
000222	005201			INC	R1		; I	5150
000224	020127	000017		CMP	R1,#17		; I,*	
000230	101761			BLOS	6\$			
000232	004767	000000G	7\$:	JSR	PC,CLOCK.INIT			5163
000236	017716	000000G		MOV	@RC25.ADDR,(SP)		; *,RC.REG	5164
000242	011667	000000G		MOV	(SP),TEMP		; RC.REG,*	
000246	026727	000000G	001750	8\$:	CMP	IN.BOUND,#1750		5168
000254	101071			BHI	11\$			
000256	004767	000000G		JSR	PC,REC.STATUS			5171
000262	006000			ROR	RO			
000264	103033			BCC	9\$			
000266	016767	000000G	000000G	MOV	IN.BOUND,TEMP			5175
000274	104455			TRAP	55			5176
000276	000136			.WORD	136			
000300	000000G			.WORD	MSG.SEEK.ERR			
000302	000000			.WORD	0			
000304	016746	000000G		MOV	LBN.ST,-(SP)			5177
000310	016746	000000G		MOV	TEMP,-(SP)			
000314	012746	000000G		MOV	#FM11,-(SP)			
000320	012746	000003		MOV	#3,-(SP)			
000324	010600			MOV	SP,RO		; SP,*	
000326	104414			TRAP	14			
000330	062706	000010		ADD	#10,SP			
000334	004767	000000G		JSR	PC,DECODE			5178
000340	012767	000001	000000G	MOV	#1,RETRIES			5179
000346	162706	000016		SUB	#16,SP			5174
000352	000475			BR	12\$			
000354	026727	000000G	001750	9\$:	CMP	OUT.BOUND,#1750		5185
000362	101017			BHI	10\$			
000364	004767	000000G		JSR	PC,READ.FILL.CMD			5187
000370	005267	000000G		INC	OUT.BOUND			5189
000374	004767	000000G		JSR	PC,GET.CMD.SLOT			5191
000400	006000			ROR	RO			
000402	103407			BLO	10\$			
000404	016700	000000G		MOV	IN.BOUND,RO			5193
000410	062700	000020		ADD	#20,RO			
000414	026700	000000G		CMP	OUT.BOUND,RO			
000420	001355			BNE	9\$			
000422	017766	000000G	000002	10\$:	MOV	@RC25.ADDR,2(SP)		5197
000430	016667	000002	000000G	MOV	2(SP),TEMP		; RC.REG,*	
000436	000703			BR	8\$			5168
000440	005077	000000G		11\$:	CLR	@CLK.CSR		5206
000444	016746	000000G		MOV	MINUTES,-(SP)			5208
000450	012746	000074		MOV	#74,-(SP)			
000454	004767	000000G		JSR	PC,BL#MUL			
000460	066700	000000G		ADD	SECONDS,RO			
000464	010067	000000G		MOV	RO,DATA4			
000470	016716	000000G		MOV	TICKS,(SP)			5209
000474	012746	000144		MOV	#144,-(SP)			
000500	004767	000000G		JSR	PC,BL#MUL			
000504	010016			MOV	RO,(SP)			
000506	016746	000000G		MOV	CLK.HERTZ,-(SP)			
000512	004767	000000G		JSR	PC,BL#DIV			
000516	010067	000000G		MOV	RO,DATA3			
000522	016716	000000G		MOV	DATA3,(SP)			5210
000526	016746	000000G		MOV	DATA4,-(SP)			

F11

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0342
Page 147
(26)

000532	012746	000000G			MOV	#MSG.ROT.TIME,-(SP)		
000536	012746	000003			MOV	#3,-(SP)		
000542	010600				MOV	SP,R0		
000544	104414				TRAP	14		; SP,*
000546	062706	000016		12#:	ADD	#16,SP		
000552	032767	000001	000000G	13#:	BIT	#1,RETRIES		5124
000560	001402				BEQ	14#		5214
000562	004767	000000G			JSR	PC,DO.RETRIES		
000566	005767	000000G		14#:	TST	NUM.RETRIES		5216
000572	001402				BEQ	15#		
000574	000167	177236			JMP	2#		
000600	022626			15#:	CMP	(SP)*,(SP)*		5091
000602	012601				MOV	(SP)*,R1		
000604	000207				RTS	PC		

; Routine Size: 195 words, Routine Base: AC\$CODE * 24334
; Maximum stack depth per invocation: 12 words

000000	004767	177166			.SBTTL	T28 TEST SECTION		
000000				T28::	JSR	PC,\$T28		5220
000004	104466			1#:	TRAP	66		
000006	006000				ROR	R0		
000010	103773				BLO	1#		
000012	000207				RTS	PC		

; Routine Size: 6 words, Routine Base: AC\$CODE * 25142
; Maximum stack depth per invocation: 2 words

; 5222 1 !<BLF/PAGE>

```

: 5223 1 !
: 5224 3 ! BGNTST;
: 5225 3 !
: 5226 3 !
: 5227 3 ! **
: 5228 3 ! TEST #29 - WRITE PROTECT TEST
: 5229 3 !
: 5230 3 ! DESCRIPTION:
: 5231 3 !
: 5232 3 ! THIS TEST REQUIRES MANUAL INTERVENTION. IT WILL BE EXECUTED IF
: 5233 3 ! THE SOFTWARE PARAMETER QUESTIONS DO NOT CAUSE IT TO BE OMITTED.
: 5234 3 !
: 5235 3 ! THIS TEST BRINGS RC25 CONTROLLER AND THE UNIT ONLINE FIRST.
: 5236 3 ! THE TEST IS DONE FROM THE HOST USING THE MSCP COMMAND "GET UNIT
: 5237 3 ! STATUS" (GUS). THE TEST WILL ASK THE OPERATOR TO MAKE SURE THE
: 5238 3 ! WRITE PROTECT SWITCH FOR THE UNIT IS IN THE OFF POSITION. IT WILL
: 5239 3 ! DO THE GUS FOR THE UNIT TO VERIFY THAT THE CONTROLLER KNOWS IT IS
: 5240 3 ! NOT WRITE PROTECTED. THEN THE OPERATOR WILL BE ASKED TO PUT THE
: 5241 3 ! WRITE PROTECT SWITCH IN THE ON POSITION AND A GUS WILL BE DONE TO
: 5242 3 ! MAKE SURE THE CONTROLLER RECOGNIZES THAT THE UNIT IS WRITE PROTECTED.
: 5243 3 !
: 5244 3 ! THE ERROR REPORT FOR THIS TEST WILL CONTAIN THE UNIT NUMBER, EXPECTED
: 5245 3 ! AND ACTUAL POSITONS OF THE WRITE PROTECT SWITCH.
: 5246 3 !
: 5247 3 !
: 5248 3 !
: 5249 3 ! SKIP THIS TEST, IF MANUAL INTERVENTION SWITCH IS CLEARED.
: 5250 3 !
: 5251 3 if not .SWP_MANUAL
: 5252 3 then
: 5253 4 begin
: 5254 4 PRINTF (DBM39);
: 5255 4 return;
: 5256 3 end;
: 5257 3
: 5258 3 if .SWP_TRACE then PRINTF (DBM38); ! TEST 29
: 5259 3
: 5260 3 NUM_RETRIES = ZERO;
: 5261 3
: 5262 3 while (.NUM_RETRIES lequ .SWP_RETRIES) do
: 5263 4 begin
: 5264 4 TIP = 29;
: 5265 4 ! GET AZTEC READY FOR OPERATION
: 5266 4
: 5267 4 if AZTEC_READY () ! IF FAILURE REPORT ERROR
: 5268 4 then
: 5269 5 begin
: 5270 5 ERRDF (95, AZT_READY_ERR, 0); !
: 5271 5
: 5272 5 if .RET_STATUS then DECODE (); ! DECODE THE STATUS, IF ANY
: 5273 5
: 5274 5 RETRIES = TRUE; ! SET RETRIES FLAG
: 5275 5 end
: 5276 4 else
: 5277 5 begin
: 5278 5
: 5279 5 ! DISPLAY MESSAGE "TURN OFF WRITE PROTECT SWITCH" THEN ISSUE

```

```

: 5280 5 ! A GET UNIT STATUS COMMAND AND EXAMINE THE UNIT FLAG.
: 5281 5 !
: 5282 5     MANU_SW = ONE;           ! MANUAL SWITCH IS SET TO 'YES'
: 5283 5
: 5284 5     while TRUE do
: 5285 6         begin
: 5286 6             GMANIL (QST14, MANU_SW, 1, YES, 0); ! DISPLAY MESSAGE TURN OFF WRT
: 5287 6             ! PROTECT SWITCH
: 5288 6
: 5289 6             if (.MANU_SW) then exitloop;
: 5290 6
: 5291 5         end;
: 5292 5
: 5293 5 !
: 5294 5 ! PROGRAM WAITING FOR GO (CR) SIGNAL
: 5295 5 !
: 5296 5
: 5297 6     if (.MANU_SW eq1 YES)
: 5298 5     then           ! WAITING FOR CR SIGNAL
: 5299 6         begin
: 5300 6             CMD_REF = .CMD_SLOT;           ! COMMAND REFERENCE NUMBER
: 5301 6
: 5302 6             if GET_UNIT_STATUS ()
: 5303 6             then           ! ISSUE A GET UNIT STATUS COMMAND
: 5304 7                 begin           ! IF RESPONSE STATUS BIT ERROR, THEN
: 5305 7                     ERRDF (96, MSG_GUS_ERR, 0); ! GET UNIT STATUS ERROR
: 5306 7                     RETRIES = TRUE;
: 5307 7                 end
: 5308 6             else
: 5309 7                 begin
: 5310 7                     RET_UNIT_FLAG = .RET_UNIT_FLAG and %o'020000'; ! MASKED OUT OTHER BITS
: 5311 7
: 5312 7                     if .RET_UNIT_FLAG eq1 UF_WPH ! IF WRT PROT. FLAG SET
: 5313 7                     then           ! ERROR
: 5314 8                         begin
: 5315 8                             ERRDF (97, MSG_COM_WPT, 0); ! REPORT ERROR
: 5316 8                             PRINTB (FMT18, .UNIT); !
: 5317 8                             RETRIES = TRUE;
: 5318 7                         end;
: 5319 7
: 5320 6                     end;
: 5321 6
: 5322 5                 end;
: 5323 5
: 5324 5 !
: 5325 5 ! DISPLAY MESSAGE "TURN ON THE WRITE PROTECT SWITCH" THEN
: 5326 5 ! ISSUE A GET UNIT SATUS COMMAND AND EXAMINE THE RESPONSE
: 5327 5 ! UNIT FLAGS.
: 5328 5 !
: 5329 5     SWITCH2 = ONE;           ! SET MANUAL SWITCH
: 5330 5
: 5331 5     while TRUE do
: 5332 6         begin
: 5333 6             GMANIL (QST15, SWITCH2, 1, YES, 0); ! DISPLAY MESSAGE TURN OFF WRT
: 5334 6             ! PROTECT SWITCH
: 5335 6
: 5336 6             if (.SWITCH2) then exitloop;

```

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0345
Page 150
(27)

```

: 5337 6
: 5338 5
: 5339 5
: 5340 5
: 5341 5
: 5342 5
: 5343 5
: 5344 6
: 5345 5
: 5346 6
: 5347 6
: 5348 6
: 5349 6
: 5350 6
: 5351 7
: 5352 7
: 5353 7
: 5354 7
: 5355 6
: 5356 7
: 5357 7
: 5358 7
: 5359 7
: 5360 7
: 5361 8
: 5362 8
: 5363 8
: 5364 8
: 5365 7
: 5366 7
: 5367 6
: 5368 6
: 5369 5
: 5370 5
: 5371 4
: 5372 4
: 5373 4
: 5374 4
: 5375 4
: 5376 4
: 5377 3
: 5378 3
: 5379 3
: 5380 1

```

```

end;

PROGRAM WAITING FOR GO (CR) SIGNAL

if (.SWITCH2 eq1 YES)
then
begin
CMD_REF = .CMD_SLOT;
if GET_UNIT_STATUS ()
then
begin
ERRDF (98, MSG_GUS_ERR, 0);
RETRIES = TRUE;
end
else
begin
RET_UNIT_FLAG = .RET_UNIT_FLAG and %0'020000';
if .RET_UNIT_FLAG neq UF_WPH
then
begin
ERRDF (99, MSG_COM_WPT, 0);
PRINTB (FMT19, .UNIT);
RETRIES = TRUE;
end;
end;
end;

end;

if (.RETRIES) then DO_RETRIES ();
if (.NUM_RETRIES eq1 ZERO) then exitloop;

end;

return;
ENDTST;

```

000000	032767	000001	000000G	\$T29:	.SBTTL	\$T29 TEST SECTION		
000006	001010				BIT	#1,SWP.MANUAL	:	5251
000010	012746	000000G			BNE	1\$		
000014	012746	000001			MOV	#DBM39,-(SP)	:	5254
000020	010600				MOV	#1,-(SP)		
000022	104417				MOV	SP,RO	:	SP,*
000024	022626				TRAP	17		
000026	000207				CMP	(SP)*,(SP)*	:	5255
000030	032767	000001	000000G	1\$:	RTS	PC	:	5253
000036	001407				BIT	#1,SWP.TRACE	:	5258
					BEQ	2\$		

ZRCFB3
V03.0

CZRCFC0 RC25 FR END TEST
TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

SEQ 0346
Page 151
(27)

000040	012746	000000G			MOV	#DBM38,-(SP)		
000044	012746	000001			MOV	#1,-(SP)		
000050	010600				MOV	SP,R0	; SP,*	
000052	104417				TRAP	17		
000054	022626				CMP	(SP)*,(SP)*		
000056	005067	000000G		2\$:	CLR	NUM.RETRIES		5260
000062	026767	000000G	000000G	3\$:	CMP	NUM.RETRIES,SWP.RETRIES		5262
000070	101401				BLOS	4\$		
000072	000207				RTS	PC		
000074	012767	000035	000000G	4\$:	MOV	#35,TIP		5264
000102	004767	000000G			JSR	PC,AZTEC.READY		5267
000106	006000				ROR	R0		
000110	103016				BCC	6\$		
000112	104455				TRAP	55		5270
000114	000137				.WORD	137		
000116	000000G				.WORD	AZT.READY.ERR		
000120	000000				.WORD	0		
000122	032767	000001	000000G		BIT	#1,RET.STATUS		5272
000130	001402				BEQ	5\$		
000132	004767	000000G			JSR	PC,DECODE		
000136	012767	000001	000000G	5\$:	MOV	#1,RETRIES		5274
000144	000560				BR	12\$		5267
000146	012767	000001	000000G	6\$:	MOV	#1,MANU.SW		5282
000154	104443			7\$:	TRAP	43		5286
000156	000404				.WORD	404		
000160	000000G				.WORD	MANU.SW		
000162	000130				.WORD	130		
000164	000000G				.WORD	QST14		
000166	000001				.WORD	1		
000170	032767	000001	000000G		BIT	#1,MANU.SW		5289
000176	001766				BEQ	7\$		
000200	026727	000000G	000001		CMP	MANU.SW,#1		5297
000206	001047				BNE	9\$		
000210	016767	000000G	000000G		MOV	CMD.SLOT,CMD.REF		5300
000216	004767	000000G			JSR	PC,GET.UNIT.STATUS		5302
000222	006000				ROR	R0		
000224	103010				BCC	8\$		
000226	104455				TRAP	55		5305
000230	000140				.WORD	140		
000232	000000G				.WORD	MSG.GUS.ERR		
000234	000000				.WORD	0		
000236	012767	000001	000000G		MOV	#1,RETRIES		5306
000244	000430				BR	9\$		5302
000246	042767	157777	000000G	8\$:	BIC	#-20001,RET.UNIT.FLAG		5310
000254	026727	000000G	020000		CMP	RET.UNIT.FLAG,#20000		5312
000262	001021				BNE	9\$		
000264	104455				TRAP	55		5315
000266	000141				.WORD	141		
000270	000000G				.WORD	MSG.COM.WPT		
000272	000000				.WORD	0		
000274	016746	000000G			MOV	UNIT,-(SP)		5316
000300	012746	000000G			MOV	#FMT18,-(SP)		
000304	012746	000002			MOV	#2,-(SP)		
000310	010600				MOV	SP,R0	; SP,*	
000312	104414				TRAP	14		
000314	012767	000001	000000G		MOV	#1,RETRIES		5317
000322	062706	000006			ADD	#6,SP		5314

000326	012767	000001	000000G	9#:	MOV	#1, SWITCH2	:	5329
000334	104443			10#:	TRAP	43	:	5333
000336	000404				.WORD	404		
000340	000000G				.WORD	SWITCH2		
000342	000130				.WORD	130		
000344	000000G				.WORD	QST15		
000346	000001				.WORD	1		
000350	032767	000001	000000G		BIT	#1, SWITCH2	:	5336
000356	001766				BEQ	10#		
000360	026727	000000G	000001		CMP	SWITCH2, #1	:	5344
000366	001047				BNE	12#		
000370	016767	000000G	000000G		MOV	CMD.SLOT, CMD.REF	:	5347
000376	004767	000000G			JSR	PC, GET.UNIT.STATUS	:	5349
000402	006000				ROR	RO		
000404	103010				BCC	11#		
000406	104455				TRAP	55	:	5352
000410	000142				.WORD	142		
000412	000000G				.WORD	MSG.GUS.ERR		
000414	000000				.WORD	0		
000416	012767	000001	000000G		MOV	#1, RETRIES	:	5353
000424	000430				BR	12#	:	5349
000426	042767	157777	000000G	11#:	BIC	#-20001, RET.UNIT.FLAG	:	5357
000434	026727	000000G	020000		CMP	RET.UNIT.FLAG, #20000	:	5359
000442	001421				BEQ	12#		
000444	104455				TRAP	55	:	5362
000446	000143				.WORD	143		
000450	000000G				.WORD	MSG.COM.WPT		
000452	000000				.WORD	0		
000454	016746	000000G			MOV	UNIT, -(SP)	:	5363
000460	012746	000000G			MOV	#FMT19, -(SP)		
000464	012746	000002			MOV	#2, -(SP)		
000470	010600				MOV	SP, RO	: SP, *	
000472	104414				TRAP	14		
000474	012767	000001	000000G		MOV	#1, RETRIES	:	5364
000502	062706	000006			ADD	#6, SP	:	5361
000506	032767	000001	000000G	12#:	BIT	#1, RETRIES	:	5373
000514	001402				BEQ	13#		
000516	004767	000000G			JSR	PC, DO.RETRIES		
000522	005767	000000G		13#:	TST	NUM.RETRIES	:	5375
000526	001402				BEQ	14#		
000530	000167	177326			JMP	3#		
000534	000207			14#:	RTS	PC	:	5221

: Routine Size: 175 words, Routine Base: AC#CODE * 25156
 : Maximum stack depth per invocation: 5 words

000000	004767	177236		T29::	.SBTTL	T29 TEST SECTION		
000000				1#:	JSR	PC, #T29	:	5379
000004	104466				TRAP	66		
000006	006000				ROR	RO		
000010	103773				BLO	1#		
000012	000207				RTS	PC		

ZRCFB3 CZRCFC0 RC25 FR END TEST
V03.0 TEST SECTION

27-Mar-1985 15:27:28
27-Mar-1985 13:28:18

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC3.B16;4

: Routine Size: 6 words, Routine Base: AC\$CODE + 25714
: Maximum stack depth per invocation: 2 words

: 5381 1 end
: 5382 1
: 5383 0 eludom

: OTS external references

.GLOBL \$SAVE4, \$SAVE3, \$SAVE2, BL\$SHF
.GLOBL BL\$DIV, BL\$MOD, BL\$MUL

: PSECT SUMMARY

Psect Name	Words	Attributes
\$OWN\$	81	RW : D : LCL, REL, CON
AC\$CODE	5612	RO : I : LCL, REL, CON

: Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
USER#1:[AZTEC.CZRCFC]AZTECO.L16;2	485	238	49	24	00:00.2

: COMMAND QUALIFIERS

: BLISS/PDP11/LIST ZRCFC3.B16/EN:NOEIS

: Size: 5612 code + 81 data words
: Run Time: 05:18.2
: Elapsed Time: 05:31.4
: Lines/CPU Min: 1015
: Lexemes/CPU-Min: 8270
: Memory Used: 333 pages
: Compilation Complete

```

: 0001 0  MODULE ZRCFB4 (#TITLE 'CZRCFC0 RC25 FR END TEST'
: 0002 0  IDENT = 'V03.0',
: 0003 0  ADDRESSING_MODE (RELATIVE)
: 0004 0  ) =
: 0005 1  BEGIN
: 0006 1  !<BLF/LOWERCASE_KEY>
: 0007 1
: 0008 1  library 'AZTECO';           ! AZTEC LIBRARY
: 0009 1
: 0010 1  require 'BLSMAC.REQ';      ! DIAGNOSTIC SUPERVISR LIBRARY
: 1499 1
: 1500 1  !
: 1501 1  !
: 1502 1  #sbttl 'DM PROGRAM'
: 1503 1
: 1504 1  !..
: 1505 1  !
: 1506 1  !   THIS MODULE CONTAINS DM CODE FOR SOME OF THE TESTS
: 1507 1  !   AS GLOBAL DATA. THE HOST PROGRAM WILL DOWN LINE LOAD
: 1508 1  !   THESE TESTS IN AZTEC CONTROLLER'S MEMORY FOR EXECUTION.
: 1509 1  !   THE DM CODE WAS FIRST ASSEMBLED AND LINKED UNDER RT
: 1510 1  !   AND THEN MADE AS VECTOR ARRAYS BY USING DMCONV.EXE
: 1511 1  !   THIS MODULE IS A COLLECTION OF ARRAYS FOR SPECIFIC
: 1512 1  !   TESTS.
: 1513 1  !..
: 1514 1
: 1515 1  !<BLF/PAGE>

```


N11

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
DM PROGRAM

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0350
Page (2)

: 1516 1
: 1517 1
: 1518 1

psect
global = DM\$CODE(nowrite, noexecute, global, concatenate);

ZRCFB4
V03.0CZRCFC0 RC25 FR END TEST
DM CODE DOWN LINE LOAD TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 B1100-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3SEQ 0351
Page 3
(3)

```

: 1519 1 #sbttl 'DM CODE DOWN LINE LOAD TEST'
: 1520 1
: 1521 1
: 1522 1 global
: 1523 1 DM_09 : vector [93, word] preset (
: 1524 1 [0] = %0'000270' : ! THIS IS THE DM PROGRAM BYTE COUNT.
: 1525 1 [1] = %0'000000' :
: 1526 1 [2] = %0'000000' : ! THIS IS THE DM OVERLAY BYTE COUNT.
: 1527 1 [3] = %0'000000' :
: 1528 1 [4] = %0'042524' : ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 1529 1 [5] = %0'052123' : ! PROGRAM NAME IS 'TEST09'
: 1530 1 [6] = %0'034460' :
: 1531 1 [7] = %0'000000' : ! THIS IS THE PROGRAM VERSION
: 1532 1 [8] = %0'126411' : ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 1533 1 [9] = %0'000000' :
: 1534 1 [10] = %0'000000' :
: 1535 1 [11] = %0'000000' :
: 1536 1 [12] = %0'000000' :
: 1537 1 [13] = %0'000000' :
: 1538 1 [14] = %0'000000' :
: 1539 1 [15] = %0'000000' :
: 1540 1 [16] = %0'104206' : ! DM CODE STARTS HERE
: 1541 1 [17] = %0'003051' :
: 1542 1 [18] = %0'114000' :
: 1543 1 [19] = %0'003037' :
: 1544 1 [20] = %0'104207' :
: 1545 1 [21] = %0'003032' :
: 1546 1 [22] = %0'104201' :
: 1547 1 [23] = %0'000003' :
: 1548 1 [24] = %0'060023' :
: 1549 1 [25] = %0'103207' :
: 1550 1 [26] = %0'177740' :
: 1551 1 [27] = %0'115007' :
: 1552 1 [28] = %0'012756' :
: 1553 1 [29] = %0'003003' :
: 1554 1 [30] = %0'114000' :
: 1555 1 [31] = %0'003052' :
: 1556 1 [32] = %0'104307' :
: 1557 1 [33] = %0'003032' :
: 1558 1 [34] = %0'104301' :
: 1559 1 [35] = %0'003033' :
: 1560 1 [36] = %0'104302' :
: 1561 1 [37] = %0'003034' :
: 1562 1 [38] = %0'104203' :
: 1563 1 [39] = %0'003052' :
: 1564 1 [40] = %0'060020' :
: 1565 1 [41] = %0'103207' :
: 1566 1 [42] = %0'177740' :
: 1567 1 [43] = %0'115007' :
: 1568 1 [44] = %0'013007' :
: 1569 1 [45] = %0'115400' :
: 1570 1 [46] = %0'003037' :
: 1571 1 [47] = %0'106300' :
: 1572 1 [48] = %0'003035' :
: 1573 1 [49] = %0'003037' :
: 1574 1 [50] = %0'032756' :
: 1575 1 [51] = %0'104200' :
: 1575 1 [52] = %0'000106' :

```

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
DM CODE DOWN LINE LOAD TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

:	1576	1	[53]	=	%0'003040'
:	1577	1	[54]	=	%0'003024'
:	1578	1	[55]	=	%0'104207'
:	1579	1	[56]	=	%0'003052'
:	1580	1	[57]	=	%0'104201'
:	1581	1	[58]	=	%0'125252'
:	1582	1	[59]	=	%0'104302'
:	1583	1	[60]	=	%0'003034'
:	1584	1	[61]	=	%0'106271'
:	1585	1	[62]	=	%0'053003'
:	1586	1	[63]	=	%0'117402'
:	1587	1	[64]	=	%0'053015'
:	1588	1	[65]	=	%0'104200'
:	1589	1	[66]	=	%0'000104'
:	1590	1	[67]	=	%0'003040'
:	1591	1	[68]	=	%0'104207'
:	1592	1	[69]	=	%0'003040'
:	1593	1	[70]	=	%0'104201'
:	1594	1	[71]	=	%0'000001'
:	1595	1	[72]	=	%0'060022'
:	1596	1	[73]	=	%0'060010'
:	1597	1	[74]	=	%0'000000'
:	1598	1	[75]	=	%0'000000'
:	1599	1	[76]	=	%0'000000'
:	1600	1	[77]	=	%0'000012'
:	1601	1	[78]	=	%0'000000'
:	1602	1	[79]	=	%0'000000'
:	1603	1	[80]	=	%0'000000'
:	1604	1	[81]	=	%0'000000'
:	1605	1	[82]	=	%0'000000'
:	1606	1	[83]	=	%0'000000'
:	1607	1	[84]	=	%0'000000'
:	1608	1	[85]	=	%0'000000'
:	1609	1	[86]	=	%0'000000'
:	1610	1	[87]	=	%0'000000'
:	1611	1	[88]	=	%0'000000'
:	1612	1	[89]	=	%0'000000'
:	1613	1	[90]	=	%0'000000'
:	1614	1	[91]	=	%0'144423'
:	1615	1	[92]	=	%0'000000'):
:	1616	1			

ZRCFB4
V03.0CZRCFC0 RC25 FR END TEST
NONEXISTENT MEMORY TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0353
Page 5
(4)

```

: 1617 1 #sbttl 'NONEXISTENT MEMORY TEST'
: 1618 1
: 1619 1
: 1620 1 global
: 1621 1 DM_10 : vector [58, word] preset (
: 1622 1 [0] = %o'000162' : THIS IS THE DM PROGRAM BYTE COUNT.
: 1623 1 [1] = %o'000000' :
: 1624 1 [2] = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 1625 1 [3] = %o'000000' :
: 1626 1 [4] = %o'042524' : NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 1627 1 [5] = %o'052123' : PROGRAM NAME IS 'TEST10'
: 1628 1 [6] = %o'030061' :
: 1629 1 [7] = %o'000000' : THIS IS THE PROGRAM VERSION
: 1630 1 [8] = %o'126411' : UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 1631 1 [9] = %o'000000' :
: 1632 1 [10] = %o'000000' :
: 1633 1 [11] = %o'000000' :
: 1634 1 [12] = %o'000000' :
: 1635 1 [13] = %o'000000' :
: 1636 1 [14] = %o'000000' :
: 1637 1 [15] = %o'000000' :
: 1638 1 [16] = %o'104206' : DM CODE STARTS HERE
: 1639 1 [17] = %o'003007' :
: 1640 1 [18] = %o'104207' :
: 1641 1 [19] = %o'160000' :
: 1642 1 [20] = %o'104201' :
: 1643 1 [21] = %o'077777' !VER:C
: 1644 1 [22] = %o'104202' :
: 1645 1 [23] = %o'000001' :
: 1646 1 [24] = %o'104203' :
: 1647 1 [25] = %o'003500' :
: 1648 1 [26] = %o'060021' :
: 1649 1 [27] = %o'103207' :
: 1650 1 [28] = %o'177740' :
: 1651 1 [29] = %o'104070' :
: 1652 1 [30] = %o'002765' :
: 1653 1 [31] = %o'104207' :
: 1654 1 [32] = %o'002765' :
: 1655 1 [33] = %o'104201' :
: 1656 1 [34] = %o'000001' :
: 1657 1 [35] = %o'060022' :
: 1658 1 [36] = %o'060010' :
: 1659 1 [37] = %o'000000' :
: 1660 1 [38] = %o'000000' :
: 1661 1 [39] = %o'000000' :
: 1662 1 [40] = %o'000000' :
: 1663 1 [41] = %o'000000' :
: 1664 1 [42] = %o'000000' :
: 1665 1 [43] = %o'000000' :
: 1666 1 [44] = %o'000000' :
: 1667 1 [45] = %o'000000' :
: 1668 1 [46] = %o'000000' :
: 1669 1 [47] = %o'000000' :
: 1670 1 [48] = %o'000000' :
: 1671 1 [49] = %o'000000' :
: 1672 1 [50] = %o'000000' :
: 1673 1 [51] = %o'000000' :
: 1673 1 [52] = %o'000000' :

```

E12

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
NONEXISTENT MEMORY TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0354
Page 6
(4)

:	1674	1	[53]	=	%o'000000'	,
:	1675	1	[54]	=	%o'000000'	,
:	1676	1	[55]	=	%o'000000'	,
:	1677	1	[56]	=	%o'030037'	,
:	1678	1	[57]	=	%o'000000');
:	1679	1				

ZRCFR4
V03.0CZRCFC0 RC25 FR END TEST
BUS ADDRESSING/DATA TEST A27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0355
Page 7
(5)

```

: 1680 1 %sbttl 'BUS ADDRESSING/DATA TEST A'
: 1681 1
: 1682 1
: 1683 1 global
: 1684 1 DM_11 : vector [100, word] preset (
: 1685 1 [0] = %o'000306' : ! THIS IS THE DM PROGRAM BYTE COUNT.
: 1686 1 [1] = %o'000000' :
: 1687 1 [2] = %o'000000' : ! THIS IS THE DM OVERLAY BYTE COUNT.
: 1688 1 [3] = %o'000000' :
: 1689 1 [4] = %o'042524' : ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 1690 1 [5] = %o'052123' : ! PROGRAM NAME IS 'TEST11'
: 1691 1 [6] = %o'030461' :
: 1692 1 [7] = %o'000000' : ! THIS IS THE PROGRAM VERSION
: 1693 1 [8] = %o'126411' : ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 1694 1 [9] = %o'000000' :
: 1695 1 [10] = %o'000000' :
: 1696 1 [11] = %o'000000' :
: 1697 1 [12] = %o'000000' :
: 1698 1 [13] = %o'000000' :
: 1699 1 [14] = %o'000000' :
: 1700 1 [15] = %o'000000' :
: 1701 1 [16] = %o'104206' : ! DM CODE STARTS HERE
: 1702 1 [17] = %o'003061' :
: 1703 1 [18] = %o'104207' :
: 1704 1 [19] = %o'003040' :
: 1705 1 [20] = %o'104201' :
: 1706 1 [21] = %o'000003' :
: 1707 1 [22] = %o'060023' :
: 1708 1 [23] = %o'103207' :
: 1709 1 [24] = %o'177740' :
: 1710 1 [25] = %o'115007' :
: 1711 1 [26] = %o'012754' :
: 1712 1 [27] = %o'003023' :
: 1713 1 [28] = %o'104200' :
: 1714 1 [29] = %o'000001' :
: 1715 1 [30] = %o'003043' :
: 1716 1 [31] = %o'104300' :
: 1717 1 [32] = %o'003040' :
: 1718 1 [33] = %o'003044' :
: 1719 1 [34] = %o'104304' :
: 1720 1 [35] = %o'003042' :
: 1721 1 [36] = %o'114000' :
: 1722 1 [37] = %o'003046' :
: 1723 1 [38] = %o'104307' :
: 1724 1 [39] = %o'003040' :
: 1725 1 [40] = %o'104301' :
: 1726 1 [41] = %o'003041' :
: 1727 1 [42] = %o'104302' :
: 1728 1 [43] = %o'003043' :
: 1729 1 [44] = %o'104203' :
: 1730 1 [45] = %o'003044' :
: 1731 1 [46] = %o'060021' :
: 1732 1 [47] = %o'103207' :
: 1733 1 [48] = %o'177740' :
: 1734 1 [49] = %o'115007' :
: 1735 1 [50] = %o'013012' :
: 1736 1 [51] = %o'115400' :
: [52] = %o'003046' :

```

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
BUS ADDRESSING/DATA TEST A

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0356
Page 8
(5)

:	1737	1	[53]	=	'106200'	,
:	1738	1	[54]	=	'000012'	,
:	1739	1	[55]	=	'003046'	,
:	1740	1	[56]	=	'032766'	,
:	1741	1	[57]	=	'003023'	,
:	1742	1	[58]	=	'117404'	,
:	1743	1	[59]	=	'013027'	,
:	1744	1	[60]	=	'105200'	,
:	1745	1	[61]	=	'000002'	,
:	1746	1	[62]	=	'003040'	,
:	1747	1	[63]	=	'104300'	,
:	1748	1	[64]	=	'003040'	,
:	1749	1	[65]	=	'003044'	,
:	1750	1	[66]	=	'002764'	,
:	1751	1	[67]	=	'104200'	,
:	1752	1	[68]	=	'000106'	,
:	1753	1	[69]	=	'003045'	,
:	1754	1	[70]	=	'003032'	,
:	1755	1	[71]	=	'104200'	,
:	1756	1	[72]	=	'000104'	,
:	1757	1	[73]	=	'003045'	,
:	1758	1	[74]	=	'104207'	,
:	1759	1	[75]	=	'003045'	,
:	1760	1	[76]	=	'104201'	,
:	1761	1	[77]	=	'000001'	,
:	1762	1	[78]	=	'060022'	,
:	1763	1	[79]	=	'060010'	,
:	1764	1	[80]	=	'000000'	,
:	1765	1	[81]	=	'000000'	,
:	1766	1	[82]	=	'000000'	,
:	1767	1	[83]	=	'000000'	,
:	1768	1	[84]	=	'000000'	,
:	1769	1	[85]	=	'000000'	,
:	1770	1	[86]	=	'000000'	,
:	1771	1	[87]	=	'000000'	,
:	1772	1	[88]	=	'000000'	,
:	1773	1	[89]	=	'000000'	,
:	1774	1	[90]	=	'000000'	,
:	1775	1	[91]	=	'000000'	,
:	1776	1	[92]	=	'000000'	,
:	1777	1	[93]	=	'000000'	,
:	1778	1	[94]	=	'000000'	,
:	1779	1	[95]	=	'000000'	,
:	1780	1	[96]	=	'000000'	,
:	1781	1	[97]	=	'000000'	,
:	1782	1	[98]	=	'056247'	,
:	1783	1	[99]	=	'000000');
:	1784	1				

```

: 1785 1  %sbttl 'BUS ADDRESSING/DATA TEST B'
: 1786 1
: 1787 1
: 1788 1  global
: 1789 1  DM_12 : vector [202, word] preset (
: 1790 1  [0] = %o'000622' . ! THIS IS THE DM PROGRAM BYTE COUNT.
: 1791 1  [1] = %o'000000' .
: 1792 1  [2] = %o'000000' . ! THIS IS THE DM OVERLAY BYTE COUNT.
: 1793 1  [3] = %o'000000' .
: 1794 1  [4] = %o'042524' . ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 1795 1  [5] = %o'052123' . ! PROGRAM NAME IS 'TEST12'
: 1796 1  [6] = %o'031061' .
: 1797 1  [7] = %o'000000' . ! THIS IS THE PROGRAM VERSION
: 1798 1  [8] = %o'177411' . ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 1799 1  [9] = %o'000000' .
: 1800 1  [10] = %o'000000' .
: 1801 1  [11] = %o'000000' .
: 1802 1  [12] = %o'000000' .
: 1803 1  [13] = %o'000000' .
: 1804 1  [14] = %o'000000' .
: 1805 1  [15] = %o'000000' .
: 1806 1  [16] = %o'104206' . ! DM CODE STARTS HERE
: 1807 1  [17] = %o'002767' .
: 1808 1  [18] = %o'003004' .
: 1809 1  [19] = %o'000000' .
: 1810 1  [20] = %o'000000' .
: 1811 1  [21] = %o'000000' .
: 1812 1  [22] = %o'000000' .
: 1813 1  [23] = %o'000000' .
: 1814 1  [24] = %o'000000' .
: 1815 1  [25] = %o'000000' .
: 1816 1  [26] = %o'000000' .
: 1817 1  [27] = %o'000000' .
: 1818 1  [28] = %o'000000' .
: 1819 1  [29] = %o'000000' .
: 1820 1  [30] = %o'000000' .
: 1821 1  [31] = %o'000000' .
: 1822 1  [32] = %o'000000' .
: 1823 1  [33] = %o'000000' .
: 1824 1  [34] = %o'000000' .
: 1825 1  [35] = %o'000000' .
: 1826 1  [36] = %o'000000' .
: 1827 1  [37] = %o'000000' .
: 1828 1  [38] = %o'000000' .
: 1829 1  [39] = %o'000000' .
: 1830 1  [40] = %o'000000' .
: 1831 1  [41] = %o'000000' .
: 1832 1  [42] = %o'000000' .
: 1833 1  [43] = %o'000104' .
: 1834 1  [44] = %o'000106' .
: 1835 1  [45] = %o'000000' .
: 1836 1  [46] = %o'000000' .
: 1837 1  [47] = %o'000000' .
: 1838 1  [48] = %o'000000' .
: 1839 1  [49] = %o'000000' .
: 1840 1  [50] = %o'000000' .
: 1841 1  [51] = %o'000000' .
: 1841 1  [52] = %o'023016' .

```


ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
BUS ADDRESSING/DATA TEST B

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0358
Page 10
(6)

:	1842	1	[53]	=	'023031'
:	1843	1	[54]	=	'023210'
:	1844	1	[55]	=	'023120'
:	1845	1	[56]	=	'023126'
:	1846	1	[57]	=	'023210'
:	1847	1	[58]	=	'023120'
:	1848	1	[59]	=	'023155'
:	1849	1	[60]	=	'023210'
:	1850	1	[61]	=	'060010'
:	1851	1	[62]	=	'104207'
:	1852	1	[63]	=	'002770'
:	1853	1	[64]	=	'104201'
:	1854	1	[65]	=	'000003'
:	1855	1	[66]	=	'060023'
:	1856	1	[67]	=	'103207'
:	1857	1	[68]	=	'177740'
:	1858	1	[69]	=	'115007'
:	1859	1	[70]	=	'013030'
:	1860	1	[71]	=	'003203'
:	1861	1	[72]	=	'000000'
:	1862	1	[73]	=	'104300'
:	1863	1	[74]	=	'002770'
:	1864	1	[75]	=	'002777'
:	1865	1	[76]	=	'104300'
:	1866	1	[77]	=	'002771'
:	1867	1	[78]	=	'003000'
:	1868	1	[79]	=	'104301'
:	1869	1	[80]	=	'002772'
:	1870	1	[81]	=	'104207'
:	1871	1	[82]	=	'177777'
:	1872	1	[83]	=	'107307'
:	1873	1	[84]	=	'002777'
:	1874	1	[85]	=	'104070'
:	1875	1	[86]	=	'003002'
:	1876	1	[87]	=	'023063'
:	1877	1	[88]	=	'105200'
:	1878	1	[89]	=	'000002'
:	1879	1	[90]	=	'002777'
:	1880	1	[91]	=	'115000'
:	1881	1	[92]	=	'002777'
:	1882	1	[93]	=	'053060'
:	1883	1	[94]	=	'115400'
:	1884	1	[95]	=	'003000'
:	1885	1	[96]	=	'117401'
:	1886	1	[97]	=	'053041'
:	1887	1	[98]	=	'000000'
:	1888	1	[99]	=	'100467'
:	1889	1	[100]	=	'100461'
:	1890	1	[101]	=	'100462'
:	1891	1	[102]	=	'100463'
:	1892	1	[103]	=	'104307'
:	1893	1	[104]	=	'002777'
:	1894	1	[105]	=	'104301'
:	1895	1	[106]	=	'003000'
:	1896	1	[107]	=	'104202'
:	1897	1	[108]	=	'000001'
:	1898	1	[109]	=	'104203'

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
BUS ADDRESSING/DATA TEST B

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

:	1899	1	[110]	=	'003002'
:	1900	1	[111]	=	'060021'
:	1901	1	[112]	=	'103207'
:	1902	1	[113]	=	'177740'
:	1903	1	[114]	=	'115007'
:	1904	1	[115]	=	'013113'
:	1905	1	[116]	=	'115400'
:	1906	1	[117]	=	'003003'
:	1907	1	[118]	=	'106200'
:	1908	1	[119]	=	'000012'
:	1909	1	[120]	=	'003003'
:	1910	1	[121]	=	'033067'
:	1911	1	[122]	=	'003203'
:	1912	1	[123]	=	'104263'
:	1913	1	[124]	=	'104262'
:	1914	1	[125]	=	'104261'
:	1915	1	[126]	=	'104267'
:	1916	1	[127]	=	'000000'
:	1917	1	[128]	=	'104207'
:	1918	1	[129]	=	'002775'
:	1919	1	[130]	=	'104201'
:	1920	1	[131]	=	'000001'
:	1921	1	[132]	=	'060023'
:	1922	1	[133]	=	'000000'
:	1923	1	[134]	=	'104300'
:	1924	1	[135]	=	'002770'
:	1925	1	[136]	=	'002777'
:	1926	1	[137]	=	'104300'
:	1927	1	[138]	=	'002771'
:	1928	1	[139]	=	'003000'
:	1929	1	[140]	=	'104301'
:	1930	1	[141]	=	'002772'
:	1931	1	[142]	=	'104200'
:	1932	1	[143]	=	'177777'
:	1933	1	[144]	=	'003002'
:	1934	1	[145]	=	'023063'
:	1935	1	[146]	=	'105200'
:	1936	1	[147]	=	'000002'
:	1937	1	[148]	=	'002777'
:	1938	1	[149]	=	'115000'
:	1939	1	[150]	=	'002777'
:	1940	1	[151]	=	'053152'
:	1941	1	[152]	=	'115400'
:	1942	1	[153]	=	'003000'
:	1943	1	[154]	=	'117401'
:	1944	1	[155]	=	'053141'
:	1945	1	[156]	=	'000000'
:	1946	1	[157]	=	'104300'
:	1947	1	[158]	=	'002770'
:	1948	1	[159]	=	'002777'
:	1949	1	[160]	=	'104300'
:	1950	1	[161]	=	'002771'
:	1951	1	[162]	=	'003000'
:	1952	1	[163]	=	'104301'
:	1953	1	[164]	=	'002772'
:	1954	1	[165]	=	'114000'
:	1955	1	[166]	=	'003002'

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
BUS ADDRESSING/DATA TEST B

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0360
Page 12
(6)

```

: 1956 1 [167] = %0'023063'
: 1957 1 [168] = %0'105200'
: 1958 1 [169] = %0'000002'
: 1959 1 [170] = %0'002777'
: 1960 1 [171] = %0'115000'
: 1961 1 [172] = %0'002777'
: 1962 1 [173] = %0'053200'
: 1963 1 [174] = %0'115400'
: 1964 1 [175] = %0'003000'
: 1965 1 [176] = %0'117401'
: 1966 1 [177] = %0'053167'
: 1967 1 [178] = %0'000000'
: 1968 1 [179] = %0'104300'
: 1969 1 [180] = %0'002774'
: 1970 1 [181] = %0'002776'
: 1971 1 [182] = %0'023215'
: 1972 1 [183] = %0'060010'
: 1973 1 [184] = %0'104300'
: 1974 1 [185] = %0'002773'
: 1975 1 [186] = %0'002776'
: 1976 1 [187] = %0'023215'
: 1977 1 [188] = %0'000000'
: 1978 1 [189] = %0'104207'
: 1979 1 [190] = %0'002776'
: 1980 1 [191] = %0'104201'
: 1981 1 [192] = %0'000001'
: 1982 1 [193] = %0'060022'
: 1983 1 [194] = %0'103207'
: 1984 1 [195] = %0'177740'
: 1985 1 [196] = %0'115007'
: 1986 1 [197] = %0'013227'
: 1987 1 [198] = %0'060010'
: 1988 1 [199] = %0'000000'
: 1989 1 [200] = %0'165572'
: 1990 1 [201] = %0'000000' );
: 1991 1
: 1992 1
: 1993 1
: 1994 1
: 1995 1

```

```

:
: THE FOLLOWING 4 BLOCK VECTORS WITH DM_13,DM_19,DM_26,DM_27
: WERE APPENDED IN VERSION B0.
:

```

ZRCFR4
V03.0CZRCFC0 RC25 FR END TEST
BLOCK TRANSFER TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0361
Page 13
(7)

```

: 1996 1  %sbttl 'BLOCK TRANSFER TEST'
: 1997 1
: 1998 1
: 1999 1  global
: 2000 1  DM_13 : vector [105, word] preset (
: 2001 1  [0]   = %o'000320' : THIS IS THE DM PROGRAM BYTE COUNT.
: 2002 1  [1]   = %o'000000' :
: 2003 1  [2]   = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 2004 1  [3]   = %o'000000' :
: 2005 1  [4]   = %o'042524' : NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 2006 1  [5]   = %o'052123' : PROGRAM NAME IS 'TEST13'
: 2007 1  [6]   = %o'031461' :
: 2008 1  [7]   = %o'000000' : THIS IS THE PROGRAM VERSION
: 2009 1  [8]   = %o'126411' : UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 2010 1  [9]   = %o'000000' :
: 2011 1  [10]  = %o'000000' :
: 2012 1  [11]  = %o'000000' :
: 2013 1  [12]  = %o'000000' :
: 2014 1  [13]  = %o'000000' :
: 2015 1  [14]  = %o'000000' :
: 2016 1  [15]  = %o'000000' :
: 2017 1  [16]  = %o'104206' : DM CODE STARTS HERE
: 2018 1  [17]  = %o'003065' :
: 2019 1  [18]  = %o'002743' :
: 2020 1  [19]  = %o'104200' :
: 2021 1  [20]  = %o'000104' :
: 2022 1  [21]  = %o'003045' :
: 2023 1  [22]  = %o'104207' :
: 2024 1  [23]  = %o'003047' :
: 2025 1  [24]  = %o'104201' :
: 2026 1  [25]  = %o'000004' :
: 2027 1  [26]  = %o'060023' :
: 2028 1  [27]  = %o'103200' :
: 2029 1  [28]  = %o'000001' :
: 2030 1  [29]  = %o'003047' :
: 2031 1  [30]  = %o'103200' :
: 2032 1  [31]  = %o'000001' :
: 2033 1  [32]  = %o'003051' :
: 2034 1  [33]  = %o'114000' :
: 2035 1  [34]  = %o'003046' :
: 2036 1  [35]  = %o'104307' :
: 2037 1  [36]  = %o'003047' :
: 2038 1  [37]  = %o'104301' :
: 2039 1  [38]  = %o'003050' :
: 2040 1  [39]  = %o'104202' :
: 2041 1  [40]  = %o'000400' :
: 2042 1  [41]  = %o'104203' :
: 2043 1  [42]  = %o'003066' :
: 2044 1  [43]  = %o'060020' :
: 2045 1  [44]  = %o'115007' :
: 2046 1  [45]  = %o'013005' :
: 2047 1  [46]  = %o'115400' :
: 2048 1  [47]  = %o'003046' :
: 2049 1  [48]  = %o'106200' :
: 2050 1  [49]  = %o'000012' :
: 2051 1  [50]  = %o'003046' :
: 2052 1  [51]  = %o'032763' :
: 2052 1  [52]  = %o'003041' :

```

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
BLOCK TRANSFER TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 B16-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

:	2053	1	[53]	▪	No' 114000'
:	2054	1	54	▪	No' 003046'
:	2055	1	55	▪	No' 104307'
:	2056	1	56	▪	No' 003051'
:	2057	1	57	▪	No' 104301'
:	2058	1	58	▪	No' 003052'
:	2059	1	59	▪	No' 104202'
:	2060	1	60	▪	No' 000400'
:	2061	1	61	▪	No' 104203'
:	2062	1	62	▪	No' 003066'
:	2063	1	63	▪	No' 060021'
:	2064	1	64	▪	No' 103207'
:	2065	1	65	▪	No' 177740'
:	2066	1	66	▪	No' 115007'
:	2067	1	67	▪	No' 013033'
:	2068	1	68	▪	No' 115400'
:	2069	1	69	▪	No' 003046'
:	2070	1	70	▪	No' 106200'
:	2071	1	71	▪	No' 000012'
:	2072	1	72	▪	No' 003046'
:	2073	1	73	▪	No' 033007'
:	2074	1	74	▪	No' 003041'
:	2075	1	75	▪	No' 104207'
:	2076	1	76	▪	No' 003045'
:	2077	1	77	▪	No' 104201'
:	2078	1	78	▪	No' 000001'
:	2079	1	79	▪	No' 060022'
:	2080	1	80	▪	No' 002743'
:	2081	1	81	▪	No' 104200'
:	2082	1	82	▪	No' 000106'
:	2083	1	83	▪	No' 003045'
:	2084	1	84	▪	No' 003033'
:	2085	1	85	▪	No' 000104'
:	2086	1	86	▪	No' 000000'
:	2087	1	87	▪	No' 000000'
:	2088	1	88	▪	No' 000000'
:	2089	1	89	▪	No' 000000'
:	2090	1	90	▪	No' 000000'
:	2091	1	91	▪	No' 000000'
:	2092	1	92	▪	No' 000000'
:	2093	1	93	▪	No' 000000'
:	2094	1	94	▪	No' 000000'
:	2095	1	95	▪	No' 000000'
:	2096	1	96	▪	No' 000000'
:	2097	1	97	▪	No' 000000'
:	2098	1	98	▪	No' 000000'
:	2099	1	99	▪	No' 000000'
:	2100	1	100	▪	No' 000000'
:	2101	1	101	▪	No' 000000'
:	2102	1	102	▪	No' 000000'
:	2103	1	103	▪	No' 113606'
:	2104	1	104	▪	No' 000000'
:	2105	1			

```

: 2106 1  #sbttl 'HEAD SWITCH TEST'
: 2107 1
: 2108 1
: 2109 1  global
: 2110 1  DM_19 : vector [156, word] preset (
: 2111 1  [0]   = #o'000466' . ! THIS IS THE DM PROGRAM BYTE COUNT.
: 2112 1  [1]   = #o'000000' .
: 2113 1  [2]   = #o'000000' . ! THIS IS THE DM OVERLAY BYTE COUNT.
: 2114 1  [3]   = #o'000000' .
: 2115 1  [4]   = #o'042524' . ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 2116 1  [5]   = #o'052123' . ! PROGRAM NAME IS 'TEST19'
: 2117 1  [6]   = #o'034461' .
: 2118 1  [7]   = #o'000000' . ! THIS IS THE PROGRAM VERSION
: 2119 1  [8]   = #o'126411' . ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 2120 1  [9]   = #o'000000' .
: 2121 1  [10]  = #o'000000' .
: 2122 1  [11]  = #o'000000' .
: 2123 1  [12]  = #o'000000' .
: 2124 1  [13]  = #o'000000' .
: 2125 1  [14]  = #o'000000' .
: 2126 1  [15]  = #o'000000' .
: 2127 1  [16]  = #o'104206' . ! DM CODE STARTS HERE
: 2128 1  [17]  = #o'003151' .
: 2129 1  [18]  = #o'104207' .
: 2130 1  [19]  = #o'003127' .
: 2131 1  [20]  = #o'104201' .
: 2132 1  [21]  = #o'000001' .
: 2133 1  [22]  = #o'060023' .
: 2134 1  [23]  = #o'114000' .
: 2135 1  [24]  = #o'003134' .
: 2136 1  [25]  = #o'104200' .
: 2137 1  [26]  = #o'000040' .
: 2138 1  [27]  = #o'003135' .
: 2139 1  [28]  = #o'102200' .
: 2140 1  [29]  = #o'000001' .
: 2141 1  [30]  = #o'003127' .
: 2142 1  [31]  = #o'012766' .
: 2143 1  [32]  = #o'104200' .
: 2144 1  [33]  = #o'000100' .
: 2145 1  [34]  = #o'003134' .
: 2146 1  [35]  = #o'104200' .
: 2147 1  [36]  = #o'000140' .
: 2148 1  [37]  = #o'003135' .
: 2149 1  [38]  = #o'104300' .
: 2150 1  [39]  = #o'003134' .
: 2151 1  [40]  = #o'003130' .
: 2152 1  [41]  = #o'114000' .
: 2153 1  [42]  = #o'003131' .
: 2154 1  [43]  = #o'023006' .
: 2155 1  [44]  = #o'104300' .
: 2156 1  [45]  = #o'003135' .
: 2157 1  [46]  = #o'003130' .
: 2158 1  [47]  = #o'114000' .
: 2159 1  [48]  = #o'003131' .
: 2160 1  [49]  = #o'023006' .
: 2161 1  [50]  = #o'104200' .
: 2162 1  [51]  = #o'000104' .
: 2162 1  [52]  = #o'003126' .

```

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
HEAD SWITCH TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 B110-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0364
Page 16
(8)

:	2163	1	[53]	=	No' 003047'
:	2164	1	[54]	=	No' 104200'
:	2165	1	[55]	=	No' 000012'
:	2166	1	[56]	=	No' 003132'
:	2167	1	[57]	=	No' 104307'
:	2168	1	[58]	=	No' 003127'
:	2169	1	[59]	=	No' 104301'
:	2170	1	[60]	=	No' 003130'
:	2171	1	[61]	=	No' 104302'
:	2172	1	[62]	=	No' 003131'
:	2173	1	[63]	=	No' 060015'
:	2174	1	[64]	=	No' 103207'
:	2175	1	[65]	=	No' 177740'
:	2176	1	[66]	=	No' 115007'
:	2177	1	[67]	=	No' 053036'
:	2178	1	[68]	=	No' 103201'
:	2179	1	[69]	=	No' 177637'
:	2180	1	[70]	=	No' 106301'
:	2181	1	[71]	=	No' 003130'
:	2182	1	[72]	=	No' 053036'
:	2183	1	[73]	=	No' 106302'
:	2184	1	[74]	=	No' 003131'
:	2185	1	[75]	=	No' 053036'
:	2186	1	[76]	=	No' 023055'
:	2187	1	[77]	=	No' 000000'
:	2188	1	[78]	=	No' 117400'
:	2189	1	[79]	=	No' 003132'
:	2190	1	[80]	=	No' 115000'
:	2191	1	[81]	=	No' 003132'
:	2192	1	[82]	=	No' 073044'
:	2193	1	[83]	=	No' 003011'
:	2194	1	[84]	=	No' 104200'
:	2195	1	[85]	=	No' 000106'
:	2196	1	[86]	=	No' 003126'
:	2197	1	[87]	=	No' 104207'
:	2198	1	[88]	=	No' 003126'
:	2199	1	[89]	=	No' 104201'
:	2200	1	[90]	=	No' 000005'
:	2201	1	[91]	=	No' 060022'
:	2202	1	[92]	=	No' 060010'
:	2203	1	[93]	=	No' 114000'
:	2204	1	[94]	=	No' 003133'
:	2205	1	[95]	=	No' 104307'
:	2206	1	[96]	=	No' 003127'
:	2207	1	[97]	=	No' 060014'
:	2208	1	[98]	=	No' 103207'
:	2209	1	[99]	=	No' 177740'
:	2210	1	[100]	=	No' 115007'
:	2211	1	[101]	=	No' 053112'
:	2212	1	[102]	=	No' 104110'
:	2213	1	[103]	=	No' 003136'
:	2214	1	[104]	=	No' 102200'
:	2215	1	[105]	=	No' 000040'
:	2216	1	[106]	=	No' 003130'
:	2217	1	[107]	=	No' 053101'
:	2218	1	[108]	=	No' 106200'
:	2219	1	[109]	=	No' 000040'

```

: 2220 1 [110] = No'003136'
: 2221 1 [111] = No'033121'
: 2222 1 [112] = No'003112'
: 2223 1 [113] = No'106200'
: 2224 1 [114] = No'000437'
: 2225 1 [115] = No'003136'
: 2226 1 [116] = No'033121'
: 2227 1 [117] = No'003112'
: 2228 1 [118] = No'106200'
: 2229 1 [119] = No'000377'
: 2230 1 [120] = No'003136'
: 2231 1 [121] = No'073121'
: 2232 1 [122] = No'115400'
: 2233 1 [123] = No'003133'
: 2234 1 [124] = No'106200'
: 2235 1 [125] = No'000024'
: 2236 1 [126] = No'003133'
: 2237 1 [127] = No'033057'
: 2238 1 [128] = No'003122'
: 2239 1 [129] = No'000000'
: 2240 1 [130] = No'104200'
: 2241 1 [131] = No'000106'
: 2242 1 [132] = No'003126'
: 2243 1 [133] = No'003047'
: 2244 1 [134] = No'000000'
: 2245 1 [135] = No'000000'
: 2246 1 [136] = No'000000'
: 2247 1 [137] = No'000000'
: 2248 1 [138] = No'000000'
: 2249 1 [139] = No'000000'
: 2250 1 [140] = No'000000'
: 2251 1 [141] = No'000000'
: 2252 1 [142] = No'000000'
: 2253 1 [143] = No'000000'
: 2254 1 [144] = No'000000'
: 2255 1 [145] = No'000000'
: 2256 1 [146] = No'000000'
: 2257 1 [147] = No'000000'
: 2258 1 [148] = No'000000'
: 2259 1 [149] = No'000000'
: 2260 1 [150] = No'000000'
: 2261 1 [151] = No'000000'
: 2262 1 [152] = No'000000'
: 2263 1 [153] = No'000000'
: 2264 1 [154] = No'044310'
: 2265 1 [155] = No'000000'
: 2266 1

```


ZRCFR4
V03.0CZRCFC0 RC25 FR END TEST
BLOCK TRANSFER TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0366
Page 18
(9)

```

: 2267 1 #sbttl 'BLOCK TRANSFER TEST'
: 2268 1
: 2269 1
: 2270 1 global
: 2271 1 DM_21 : vector [213, word] preset (
: 2272 1 [0] = %o'000650' : THIS IS THE DM PROGRAM BYTE COUNT.
: 2273 1 [1] = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 2274 1 [2] = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 2275 1 [3] = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 2276 1 [4] = %o'042524' : NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 2277 1 [5] = %o'052123' : PROGRAM NAME IS 'TEST21'
: 2278 1 [6] = %o'030462' : PROGRAM NAME IS 'TEST21'
: 2279 1 [7] = %o'000000' : THIS IS THE PROGRAM VERSION
: 2280 1 [8] = %o'126411' : UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 2281 1 [9] = %o'000000'
: 2282 1 [10] = %o'000000'
: 2283 1 [11] = %o'000000'
: 2284 1 [12] = %o'000000'
: 2285 1 [13] = %o'000000'
: 2286 1 [14] = %o'000000'
: 2287 1 [15] = %o'000000'
: 2288 1 [16] = %o'104206' : DM CODE STARTS HERE
: 2289 1 [17] = %o'003223'
: 2290 1 [18] = %o'104207'
: 2291 1 [19] = %o'003176'
: 2292 1 [20] = %o'104201'
: 2293 1 [21] = %o'000002'
: 2294 1 [22] = %o'060023'
: 2295 1 [23] = %o'114000'
: 2296 1 [24] = %o'003206'
: 2297 1 [25] = %o'104200'
: 2298 1 [26] = %o'000040'
: 2299 1 [27] = %o'003207'
: 2300 1 [28] = %o'102200'
: 2301 1 [29] = %o'000001'
: 2302 1 [30] = %o'003176'
: 2303 1 [31] = %o'012771'
: 2304 1 [32] = %o'104200'
: 2305 1 [33] = %o'000100'
: 2306 1 [34] = %o'003206'
: 2307 1 [35] = %o'104200'
: 2308 1 [36] = %o'000140'
: 2309 1 [37] = %o'003207'
: 2310 1 [38] = %o'104200'
: 2311 1 [39] = %o'144000'
: 2312 1 [40] = %o'003202'
: 2313 1 [41] = %o'104300'
: 2314 1 [42] = %o'003177'
: 2315 1 [43] = %o'003205'
: 2316 1 [44] = %o'104300'
: 2317 1 [45] = %o'003206'
: 2318 1 [46] = %o'003177'
: 2319 1 [47] = %o'115000'
: 2320 1 [48] = %o'003205'
: 2321 1 [49] = %o'053005'
: 2322 1 [50] = %o'104300'
: 2323 1 [51] = %o'003207'
: 2323 1 [52] = %o'003177'

```

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
BLOCK TRANSFER TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0367
Page 19
(9)

:	2324	1	[53]	=	%o'114000'
:	2325	1	[54]	=	%o'003200'
:	2326	1	[55]	=	%o'023015'
:	2327	1	[56]	=	%o'023045'
:	2328	1	[57]	=	%o'104200'
:	2329	1	[58]	=	%o'000104'
:	2330	1	[59]	=	%o'003175'
:	2331	1	[60]	=	%o'003167'
:	2332	1	[61]	=	%o'114000'
:	2333	1	[62]	=	%o'003203'
:	2334	1	[63]	=	%o'104307'
:	2335	1	[64]	=	%o'003176'
:	2336	1	[65]	=	%o'104301'
:	2337	1	[66]	=	%o'003177'
:	2338	1	[67]	=	%o'104302'
:	2339	1	[68]	=	%o'003200'
:	2340	1	[69]	=	%o'060015'
:	2341	1	[70]	=	%o'103207'
:	2342	1	[71]	=	%o'177740'
:	2343	1	[72]	=	%o'106020'
:	2344	1	[73]	=	%o'003200'
:	2345	1	[74]	=	%o'053036'
:	2346	1	[75]	=	%o'115007'
:	2347	1	[76]	=	%o'053036'
:	2348	1	[77]	=	%o'000000'
:	2349	1	[78]	=	%o'115400'
:	2350	1	[79]	=	%o'003203'
:	2351	1	[80]	=	%o'106200'
:	2352	1	[81]	=	%o'000012'
:	2353	1	[82]	=	%o'003203'
:	2354	1	[83]	=	%o'033017'
:	2355	1	[84]	=	%o'003164'
:	2356	1	[85]	=	%o'104200'
:	2357	1	[86]	=	%o'000040'
:	2358	1	[87]	=	%o'003204'
:	2359	1	[88]	=	%o'023121'
:	2360	1	[89]	=	%o'114000'
:	2361	1	[90]	=	%o'003210'
:	2362	1	[91]	=	%o'104200'
:	2363	1	[92]	=	%o'100000'
:	2364	1	[93]	=	%o'003242'
:	2365	1	[94]	=	%o'104200'
:	2366	1	[95]	=	%o'003256'
:	2367	1	[96]	=	%o'003245'
:	2368	1	[97]	=	%o'104300'
:	2369	1	[98]	=	%o'003201'
:	2370	1	[99]	=	%o'003246'
:	2371	1	[100]	=	%o'104300'
:	2372	1	[101]	=	%o'003202'
:	2373	1	[102]	=	%o'003247'
:	2374	1	[103]	=	%o'104300'
:	2375	1	[104]	=	%o'003201'
:	2376	1	[105]	=	%o'003254'
:	2377	1	[106]	=	%o'104300'
:	2378	1	[107]	=	%o'003202'
:	2379	1	[108]	=	%o'003255'
:	2380	1	[109]	=	%o'104307'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
BLOCK TRANSFER TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0368
Page 20
(9)

:	2381	1	[110]	=	'003176'
:	2382	1	[111]	=	'104201'
:	2383	1	[112]	=	'003242'
:	2384	1	[113]	=	'060002'
:	2385	1	[114]	=	'103207'
:	2386	1	[115]	=	'177740'
:	2387	1	[116]	=	'115007'
:	2388	1	[117]	=	'013115'
:	2389	1	[118]	=	'115400'
:	2390	1	[119]	=	'003210'
:	2391	1	[120]	=	'106200'
:	2392	1	[121]	=	'000012'
:	2393	1	[122]	=	'003210'
:	2394	1	[123]	=	'033050'
:	2395	1	[124]	=	'003164'
:	2396	1	[125]	=	'117400'
:	2397	1	[126]	=	'003204'
:	2398	1	[127]	=	'053050'
:	2399	1	[128]	=	'000000'
:	2400	1	[129]	=	'104307'
:	2401	1	[130]	=	'003176'
:	2402	1	[131]	=	'060014'
:	2403	1	[132]	=	'103207'
:	2404	1	[133]	=	'177740'
:	2405	1	[134]	=	'115007'
:	2406	1	[135]	=	'053155'
:	2407	1	[136]	=	'104110'
:	2408	1	[137]	=	'003201'
:	2409	1	[138]	=	'106200'
:	2410	1	[139]	=	'000440'
:	2411	1	[140]	=	'003201'
:	2412	1	[141]	=	'033137'
:	2413	1	[142]	=	'003155'
:	2414	1	[143]	=	'115401'
:	2415	1	[144]	=	'104110'
:	2416	1	[145]	=	'003202'
:	2417	1	[146]	=	'104117'
:	2418	1	[147]	=	'103207'
:	2419	1	[148]	=	'144000'
:	2420	1	[149]	=	'053155'
:	2421	1	[150]	=	'104117'
:	2422	1	[151]	=	'103207'
:	2423	1	[152]	=	'104000'
:	2424	1	[153]	=	'106207'
:	2425	1	[154]	=	'040000'
:	2426	1	[155]	=	'053155'
:	2427	1	[156]	=	'000000'
:	2428	1	[157]	=	'115400'
:	2429	1	[158]	=	'003241'
:	2430	1	[159]	=	'106200'
:	2431	1	[160]	=	'000144'
:	2432	1	[161]	=	'003241'
:	2433	1	[162]	=	'033121'
:	2434	1	[163]	=	'003164'
:	2435	1	[164]	=	'104200'
:	2436	1	[165]	=	'000106'
:	2437	1	[166]	=	'003175'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
BLOCK TRANSFER TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

```

: 2438 1 [167] = %0'104207';
: 2439 1 [168] = %0'003175';
: 2440 1 [169] = %0'104201';
: 2441 1 [170] = %0'000005';
: 2442 1 [171] = %0'060022';
: 2443 1 [172] = %0'060010';
: 2444 1 [173] = %0'000000';
: 2445 1 [174] = %0'000000';
: 2446 1 [175] = %0'000000';
: 2447 1 [176] = %0'000000';
: 2448 1 [177] = %0'000000';
: 2449 1 [178] = %0'140000';
: 2450 1 [179] = %0'000000';
: 2451 1 [180] = %0'000000';
: 2452 1 [181] = %0'000000';
: 2453 1 [182] = %0'000000';
: 2454 1 [183] = %0'000000';
: 2455 1 [184] = %0'000000';
: 2456 1 [185] = %0'000000';
: 2457 1 [186] = %0'000000';
: 2458 1 [187] = %0'000000';
: 2459 1 [188] = %0'000000';
: 2460 1 [189] = %0'000000';
: 2461 1 [190] = %0'000000';
: 2462 1 [191] = %0'000000';
: 2463 1 [192] = %0'000000';
: 2464 1 [193] = %0'000000';
: 2465 1 [194] = %0'000000';
: 2466 1 [195] = %0'000000';
: 2467 1 [196] = %0'000000';
: 2468 1 [197] = %0'000000';
: 2469 1 [198] = %0'000000';
: 2470 1 [199] = %0'000000';
: 2471 1 [200] = %0'000000';
: 2472 1 [201] = %0'000000';
: 2473 1 [202] = %0'000000';
: 2474 1 [203] = %0'000000';
: 2475 1 [204] = %0'000000';
: 2476 1 [205] = %0'000000';
: 2477 1 [206] = %0'000000';
: 2478 1 [207] = %0'000000';
: 2479 1 [208] = %0'000000';
: 2480 1 [209] = %0'000000';
: 2481 1 [210] = %0'000000';
: 2482 1 [211] = %0'065461';
: 2483 1 [212] = %0'000000');
: 2484 1

```

```

: 2485 1 #sbttl 'WRITE DATA TEST'
: 2486 1 ! VER:C THIS TEST WAS REVISED
: 2487 1 global
: 2488 1
: 2489 1 DM_26:VECTOR[413,WORD]
: 2490 1 PRESET (
: 2491 1 [0] = %0'001470' ! THIS IS THE DM PROGRAM BYTE COUNT.
: 2492 1 [1] = %0'000000'
: 2493 1 [2] = %0'000000' ! THIS IS THE DM OVERLAY BYTE COUNT.
: 2494 1 [3] = %0'000000'
: 2495 1 [4] = %0'042524' ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 2496 1 [5] = %0'052123' ! PROGRAM NAME IS 'TEST26'
: 2497 1 [6] = %0'033062'
: 2498 1 [7] = %0'000000' ! THIS IS THE PROGRAM VERSION
: 2499 1 [8] = %0'126411' ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 2500 1 [9] = %0'000000'
: 2501 1 [10] = %0'000000'
: 2502 1 [11] = %0'000000'
: 2503 1 [12] = %0'000000'
: 2504 1 [13] = %0'000000'
: 2505 1 [14] = %0'000000'
: 2506 1 [15] = %0'000000'
: 2507 1 [16] = %0'104206' ! DM CODE STARTS HERE
: 2508 1 [17] = %0'003550'
: 2509 1 [18] = %0'022777'
: 2510 1 [19] = %0'023005'
: 2511 1 [20] = %0'104206'
: 2512 1 [21] = %0'003550'
: 2513 1 [22] = %0'106200'
: 2514 1 [23] = %0'177777'
: 2515 1 [24] = %0'003507'
: 2516 1 [25] = %0'012770'
: 2517 1 [26] = %0'023030'
: 2518 1 [27] = %0'023043'
: 2519 1 [28] = %0'104200'
: 2520 1 [29] = %0'177777'
: 2521 1 [30] = %0'003507'
: 2522 1 [31] = %0'114000'
: 2523 1 [32] = %0'003501'
: 2524 1 [33] = %0'114000'
: 2525 1 [34] = %0'003477'
: 2526 1 [35] = %0'114000'
: 2527 1 [36] = %0'003500'
: 2528 1 [37] = %0'104200'
: 2529 1 [38] = %0'000010'
: 2530 1 [39] = %0'003473'
: 2531 1 [40] = %0'023052'
: 2532 1 [41] = %0'023065'
: 2533 1 [42] = %0'104200'
: 2534 1 [43] = %0'177777'
: 2535 1 [44] = %0'003476'
: 2536 1 [45] = %0'023461'
: 2537 1 [46] = %0'060010'
: 2538 1 [47] = %0'104207'
: 2539 1 [48] = %0'003506'
: 2540 1 [49] = %0'104201'
: 2541 1 [50] = %0'000001'

```

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
WRITE DATA TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0371
Page 23
(10)

:	2542	1	[51]	=	'060023'
:	2543	1	[52]	=	'000000'
:	2544	1	[53]	=	'114000'
:	2545	1	[54]	=	'003504'
:	2546	1	[55]	=	'104200'
:	2547	1	[56]	=	'000040'
:	2548	1	[57]	=	'003505'
:	2549	1	[58]	=	'102200'
:	2550	1	[59]	=	'000001'
:	2551	1	[60]	=	'003506'
:	2552	1	[61]	=	'013027'
:	2553	1	[62]	=	'104200'
:	2554	1	[63]	=	'000100'
:	2555	1	[64]	=	'003504'
:	2556	1	[65]	=	'104200'
:	2557	1	[66]	=	'000140'
:	2558	1	[67]	=	'003505'
:	2559	1	[68]	=	'104200'
:	2560	1	[69]	=	'144000'
:	2561	1	[70]	=	'003472'
:	2562	1	[71]	=	'000000'
:	2563	1	[72]	=	'104300'
:	2564	1	[73]	=	'003504'
:	2565	1	[74]	=	'003503'
:	2566	1	[75]	=	'023236'
:	2567	1	[76]	=	'023177'
:	2568	1	[77]	=	'023116'
:	2569	1	[78]	=	'023362'
:	2570	1	[79]	=	'023137'
:	2571	1	[80]	=	'023177'
:	2572	1	[81]	=	'023160'
:	2573	1	[82]	=	'000000'
:	2574	1	[83]	=	'023177'
:	2575	1	[84]	=	'023137'
:	2576	1	[85]	=	'023362'
:	2577	1	[86]	=	'023116'
:	2578	1	[87]	=	'023177'
:	2579	1	[88]	=	'023160'
:	2580	1	[89]	=	'000000'
:	2581	1	[90]	=	'104300'
:	2582	1	[91]	=	'003505'
:	2583	1	[92]	=	'003503'
:	2584	1	[93]	=	'023236'
:	2585	1	[94]	=	'023177'
:	2586	1	[95]	=	'023116'
:	2587	1	[96]	=	'023362'
:	2588	1	[97]	=	'023137'
:	2589	1	[98]	=	'023177'
:	2590	1	[99]	=	'023160'
:	2591	1	[100]	=	'000000'
:	2592	1	[101]	=	'023177'
:	2593	1	[102]	=	'023137'
:	2594	1	[103]	=	'023362'
:	2595	1	[104]	=	'023116'
:	2596	1	[105]	=	'023177'
:	2597	1	[106]	=	'023160'
:	2598	1	[107]	=	'000000'

ZRCFB4
VO3.0

CZRCFC0 RC25 FR END TEST
WRITE DATA TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0372
Page 24
(10)

:	2599	1	[108]	=	#0'104307'
:	2600	1	[109]	=	#0'003506'
:	2601	1	[110]	=	#0'104301'
:	2602	1	[111]	=	#0'003503'
:	2603	1	[112]	=	#0'104302'
:	2604	1	[113]	=	#0'003501'
:	2605	1	[114]	=	#0'060015'
:	2606	1	[115]	=	#0'104070'
:	2607	1	[116]	=	#0'003476'
:	2608	1	[117]	=	#0'103207'
:	2609	1	[118]	=	#0'177740'
:	2610	1	[119]	=	#0'115007'
:	2611	1	[120]	=	#0'053115'
:	2612	1	[121]	=	#0'106020'
:	2613	1	[122]	=	#0'003501'
:	2614	1	[123]	=	#0'053115'
:	2615	1	[124]	=	#0'000000'
:	2616	1	[125]	=	#0'003424'
:	2617	1	[126]	=	#0'114007'
:	2618	1	[127]	=	#0'104301'
:	2619	1	[128]	=	#0'003475'
:	2620	1	[129]	=	#0'100671'
:	2621	1	[130]	=	#0'003566'
:	2622	1	[131]	=	#0'115407'
:	2623	1	[132]	=	#0'106207'
:	2624	1	[133]	=	#0'000377'
:	2625	1	[134]	=	#0'033121'
:	2626	1	[135]	=	#0'104207'
:	2627	1	[136]	=	#0'003566'
:	2628	1	[137]	=	#0'104201'
:	2629	1	[138]	=	#0'000400'
:	2630	1	[139]	=	#0'060004'
:	2631	1	[140]	=	#0'104010'
:	2632	1	[141]	=	#0'004166'
:	2633	1	[142]	=	#0'000000'
:	2634	1	[143]	=	#0'114007'
:	2635	1	[144]	=	#0'104301'
:	2636	1	[145]	=	#0'003474'
:	2637	1	[146]	=	#0'100671'
:	2638	1	[147]	=	#0'003566'
:	2639	1	[148]	=	#0'115407'
:	2640	1	[149]	=	#0'106207'
:	2641	1	[150]	=	#0'000377'
:	2642	1	[151]	=	#0'033142'
:	2643	1	[152]	=	#0'104207'
:	2644	1	[153]	=	#0'003566'
:	2645	1	[154]	=	#0'104201'
:	2646	1	[155]	=	#0'000400'
:	2647	1	[156]	=	#0'060004'
:	2648	1	[157]	=	#0'104010'
:	2649	1	[158]	=	#0'004166'
:	2650	1	[159]	=	#0'000000'
:	2651	1	[160]	=	#0'104301'
:	2652	1	[161]	=	#0'003477'
:	2653	1	[162]	=	#0'114007'
:	2654	1	[163]	=	#0'104672'
:	2655	1	[164]	=	#0'003566'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
WRITE DATA TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0373
Page 25
(10)

:	2656	1	[165]	=	%0'106012'
:	2657	1	[166]	=	%0'053174'
:	2658	1	[167]	=	%0'115407'
:	2659	1	[168]	=	%0'106207'
:	2660	1	[169]	=	%0'000377'
:	2661	1	[170]	=	%0'033163'
:	2662	1	[171]	=	%0'000000'
:	2663	1	[172]	=	%0'104020'
:	2664	1	[173]	=	%0'003500'
:	2665	1	[174]	=	%0'003424'
:	2666	1	[175]	=	%0'104200'
:	2667	1	[176]	=	%0'100000'
:	2668	1	[177]	=	%0'003552'
:	2669	1	[178]	=	%0'104200'
:	2670	1	[179]	=	%0'003566'
:	2671	1	[180]	=	%0'003555'
:	2672	1	[181]	=	%0'104300'
:	2673	1	[182]	=	%0'003467'
:	2674	1	[183]	=	%0'003556'
:	2675	1	[184]	=	%0'104300'
:	2676	1	[185]	=	%0'003470'
:	2677	1	[186]	=	%0'003557'
:	2678	1	[187]	=	%0'104300'
:	2679	1	[188]	=	%0'003467'
:	2680	1	[189]	=	%0'003564'
:	2681	1	[190]	=	%0'104300'
:	2682	1	[191]	=	%0'003470'
:	2683	1	[192]	=	%0'003565'
:	2684	1	[193]	=	%0'104307'
:	2685	1	[194]	=	%0'003506'
:	2686	1	[195]	=	%0'104201'
:	2687	1	[196]	=	%0'003552'
:	2688	1	[197]	=	%0'060002'
:	2689	1	[198]	=	%0'104070'
:	2690	1	[199]	=	%0'003476'
:	2691	1	[200]	=	%0'103207'
:	2692	1	[201]	=	%0'177740'
:	2693	1	[202]	=	%0'115007'
:	2694	1	[203]	=	%0'013235'
:	2695	1	[204]	=	%0'003424'
:	2696	1	[205]	=	%0'000000'
:	2697	1	[206]	=	%0'023074'
:	2698	1	[207]	=	%0'114000'
:	2699	1	[208]	=	%0'003502'
:	2700	1	[209]	=	%0'104307'
:	2701	1	[210]	=	%0'003506'
:	2702	1	[211]	=	%0'060014'
:	2703	1	[212]	=	%0'104070'
:	2704	1	[213]	=	%0'003476'
:	2705	1	[214]	=	%0'103207'
:	2706	1	[215]	=	%0'177740'
:	2707	1	[216]	=	%0'115007'
:	2708	1	[217]	=	%0'053361'
:	2709	1	[218]	=	%0'114002'
:	2710	1	[219]	=	%0'104113'
:	2711	1	[220]	=	%0'106613'
:	2712	1	[221]	=	%0'000002'

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
WRITE DATA TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0374
Page 26
(10)

:	2713	1	[222]	=	#0'053262'
:	2714	1	[223]	=	#0'106613'
:	2715	1	[224]	=	#0'000004'
:	2716	1	[225]	=	#0'013306'
:	2717	1	[226]	=	#0'106613'
:	2718	1	[227]	=	#0'000004'
:	2719	1	[228]	=	#0'053270'
:	2720	1	[229]	=	#0'106613'
:	2721	1	[230]	=	#0'000006'
:	2722	1	[231]	=	#0'013306'
:	2723	1	[232]	=	#0'106613'
:	2724	1	[233]	=	#0'000002'
:	2725	1	[234]	=	#0'053276'
:	2726	1	[235]	=	#0'106613'
:	2727	1	[236]	=	#0'000006'
:	2728	1	[237]	=	#0'013306'
:	2729	1	[238]	=	#0'104613'
:	2730	1	[239]	=	#0'000002'
:	2731	1	[240]	=	#0'106613'
:	2732	1	[241]	=	#0'000004'
:	2733	1	[242]	=	#0'053361'
:	2734	1	[243]	=	#0'106613'
:	2735	1	[244]	=	#0'000006'
:	2736	1	[245]	=	#0'053361'
:	2737	1	[246]	=	#0'115002'
:	2738	1	[247]	=	#0'053316'
:	2739	1	[248]	=	#0'104030'
:	2740	1	[249]	=	#0'003467'
:	2741	1	[250]	=	#0'115401'
:	2742	1	[251]	=	#0'104202'
:	2743	1	[252]	=	#0'177777'
:	2744	1	[253]	=	#0'003253'
:	2745	1	[254]	=	#0'104030'
:	2746	1	[255]	=	#0'003470'
:	2747	1	[256]	=	#0'104300'
:	2748	1	[257]	=	#0'003467'
:	2749	1	[258]	=	#0'003502'
:	2750	1	[259]	=	#0'103200'
:	2751	1	[260]	=	#0'177740'
:	2752	1	[261]	=	#0'003502'
:	2753	1	[262]	=	#0'102200'
:	2754	1	[263]	=	#0'000001'
:	2755	1	[264]	=	#0'003467'
:	2756	1	[265]	=	#0'053361'
:	2757	1	[266]	=	#0'102200'
:	2758	1	[267]	=	#0'010000'
:	2759	1	[268]	=	#0'003470'
:	2760	1	[269]	=	#0'053361'
:	2761	1	[270]	=	#0'106300'
:	2762	1	[271]	=	#0'003472'
:	2763	1	[272]	=	#0'003470'
:	2764	1	[273]	=	#0'053361'
:	2765	1	[274]	=	#0'104302'
:	2766	1	[275]	=	#0'003467'
:	2767	1	[276]	=	#0'105202'
:	2768	1	[277]	=	#0'000000'
:	2769	1	[278]	=	#0'103202'

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
WRITE DATA TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0375
Page 27
(10)

:	2770	1	[279]	▄	#0'000037'
:	2771	1	[280]	▄	#0'110602'
:	2772	1	[281]	▄	#0'110602'
:	2773	1	[282]	▄	#0'110602'
:	2774	1	[283]	▄	#0'110602'
:	2775	1	[284]	▄	#0'110602'
:	2776	1	[285]	▄	#0'106302'
:	2777	1	[286]	▄	#0'003473'
:	2778	1	[287]	▄	#0'053361'
:	2779	1	[288]	▄	#0'000000'
:	2780	1	[289]	▄	#0'003424'
:	2781	1	[290]	▄	#0'104300'
:	2782	1	[291]	▄	#0'003566'
:	2783	1	[292]	▄	#0'003477'
:	2784	1	[293]	▄	#0'104200'
:	2785	1	[294]	▄	#0'140000'
:	2786	1	[295]	▄	#0'003552'
:	2787	1	[296]	▄	#0'104200'
:	2788	1	[297]	▄	#0'003566'
:	2789	1	[298]	▄	#0'003555'
:	2790	1	[299]	▄	#0'104300'
:	2791	1	[300]	▄	#0'003467'
:	2792	1	[301]	▄	#0'003556'
:	2793	1	[302]	▄	#0'104300'
:	2794	1	[303]	▄	#0'003470'
:	2795	1	[304]	▄	#0'003557'
:	2796	1	[305]	▄	#0'104300'
:	2797	1	[306]	▄	#0'003467'
:	2798	1	[307]	▄	#0'003564'
:	2799	1	[308]	▄	#0'104300'
:	2800	1	[309]	▄	#0'003470'
:	2801	1	[310]	▄	#0'003565'
:	2802	1	[311]	▄	#0'104307'
:	2803	1	[312]	▄	#0'003506'
:	2804	1	[313]	▄	#0'104201'
:	2805	1	[314]	▄	#0'003552'
:	2806	1	[315]	▄	#0'060003'
:	2807	1	[316]	▄	#0'104070'
:	2808	1	[317]	▄	#0'003476'
:	2809	1	[318]	▄	#0'103207'
:	2810	1	[319]	▄	#0'177740'
:	2811	1	[320]	▄	#0'115007'
:	2812	1	[321]	▄	#0'013423'
:	2813	1	[322]	▄	#0'003424'
:	2814	1	[323]	▄	#0'000000'
:	2815	1	[324]	▄	#0'115400'
:	2816	1	[325]	▄	#0'003471'
:	2817	1	[326]	▄	#0'106200'
:	2818	1	[327]	▄	#0'000100'
:	2819	1	[328]	▄	#0'003471'
:	2820	1	[329]	▄	#0'013433'
:	2821	1	[330]	▄	#0'002744'
:	2822	1	[331]	▄	#0'115400'
:	2823	1	[332]	▄	#0'003501'
:	2824	1	[333]	▄	#0'115400'
:	2825	1	[334]	▄	#0'003473'
:	2826	1	[335]	▄	#0'114000'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
WRITE DATA TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0376
Page 28
(10)

:	2827	1	(336)	▪	#0'003471'
:	2828	1	(337)	▪	#0'106200'
:	2829	1	(338)	▪	#0'000004'
:	2830	1	(339)	▪	#0'003501'
:	2831	1	(340)	▪	#0'053450'
:	2832	1	(341)	▪	#0'104200'
:	2833	1	(342)	▪	#0'001473'
:	2834	1	(343)	▪	#0'003501'
:	2835	1	(344)	▪	#0'106200'
:	2836	1	(345)	▪	#0'001477'
:	2837	1	(346)	▪	#0'003501'
:	2838	1	(347)	▪	#0'013455'
:	2839	1	(348)	▪	#0'002744'
:	2840	1	(349)	▪	#0'117400'
:	2841	1	(350)	▪	#0'003501'
:	2842	1	(351)	▪	#0'023461'
:	2843	1	(352)	▪	#0'060010'
:	2844	1	(353)	▪	#0'104207'
:	2845	1	(354)	▪	#0'003476'
:	2846	1	(355)	▪	#0'104201'
:	2847	1	(356)	▪	#0'000006'
:	2848	1	(357)	▪	#0'060022'
:	2849	1	(358)	▪	#0'000000'
:	2850	1	(359)	▪	#0'000000'
:	2851	1	(360)	▪	#0'000000'
:	2852	1	(361)	▪	#0'000000'
:	2853	1	(362)	▪	#0'140000'
:	2854	1	(363)	▪	#0'000000'
:	2855	1	(364)	▪	#0'000000'
:	2856	1	(365)	▪	#0'177777'
:	2857	1	(366)	▪	#0'000000'
:	2858	1	(367)	▪	#0'000000'
:	2859	1	(368)	▪	#0'000000'
:	2860	1	(369)	▪	#0'000000'
:	2861	1	(370)	▪	#0'000000'
:	2862	1	(371)	▪	#0'000000'
:	2863	1	(372)	▪	#0'000000'
:	2864	1	(373)	▪	#0'000000'
:	2865	1	(374)	▪	#0'000000'
:	2866	1	(375)	▪	#0'000000'
:	2867	1	(376)	▪	#0'000000'
:	2868	1	(377)	▪	#0'000000'
:	2869	1	(378)	▪	#0'000000'
:	2870	1	(379)	▪	#0'000000'
:	2871	1	(380)	▪	#0'000000'
:	2872	1	(381)	▪	#0'000000'
:	2873	1	(382)	▪	#0'000000'
:	2874	1	(383)	▪	#0'000000'
:	2875	1	(384)	▪	#0'000000'
:	2876	1	(385)	▪	#0'000000'
:	2877	1	(386)	▪	#0'000000'
:	2878	1	(387)	▪	#0'000000'
:	2879	1	(388)	▪	#0'000000'
:	2880	1	(389)	▪	#0'000000'
:	2881	1	(390)	▪	#0'000000'
:	2882	1	(391)	▪	#0'000000'
:	2883	1	(392)	▪	#0'000000'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
WRITE DATA TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0377
Page 29
(10)

:	2884	1	[393]	=	%0'000000'	,
:	2885	1	[394]	=	%0'000000'	,
:	2886	1	[395]	=	%0'000000'	,
:	2887	1	[396]	=	%0'000000'	,
:	2888	1	[397]	=	%0'000000'	,
:	2889	1	[398]	=	%0'000000'	,
:	2890	1	[399]	=	%0'000000'	,
:	2891	1	[400]	=	%0'000000'	,
:	2892	1	[401]	=	%0'000000'	,
:	2893	1	[402]	=	%0'000000'	,
:	2894	1	[403]	=	%0'000000'	,
:	2895	1	[404]	=	%0'000000'	,
:	2896	1	[405]	=	%0'000000'	,
:	2897	1	[406]	=	%0'000000'	,
:	2898	1	[407]	=	%0'000000'	,
:	2899	1	[408]	=	%0'000000'	,
:	2900	1	[409]	=	%0'000000'	,
:	2901	1	[410]	=	%0'000000'	,
:	2902	1	[411]	=	%0'044552'	,
:	2903	1	[412]	=	%0'000000');
:	2904	1				

ZRCFR4
V03.0CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0378
Page 30
(11)

```

: 2905 1 #sbttl 'OFFSET TOLERANCE TEST'
: 2906 1
: 2907 1
: 2908 1 global
: 2909 1 DM_27 : vector [307, word] preset (
: 2910 1 [0] = %o'001144' : THIS IS THE DM PROGRAM BYTE COUNT.
: 2911 1 [1] = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 2912 1 [2] = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 2913 1 [3] = %o'000000' : THIS IS THE DM OVERLAY BYTE COUNT.
: 2914 1 [4] = %o'042524' : NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 2915 1 [5] = %o'052123' : PROGRAM NAME IS 'TEST27'
: 2916 1 [6] = %o'033462' : NEXT 3 WORDS = PROGRAM NAME (ASCII)
: 2917 1 [7] = %o'000000' : PROGRAM NAME IS 'TEST27'
: 2918 1 [8] = %o'126411' : THIS IS THE PROGRAM VERSION
: 2919 1 [9] = %o'000000' : UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
: 2920 1 [10] = %o'000000'
: 2921 1 [11] = %o'000000'
: 2922 1 [12] = %o'000000'
: 2923 1 [13] = %o'000000'
: 2924 1 [14] = %o'000000'
: 2925 1 [15] = %o'000000'
: 2926 1 [16] = %o'104206' : DM CODE STARTS HERE
: 2927 1 [17] = %o'003377'
: 2928 1 [18] = %o'022750'
: 2929 1 [19] = %o'022756'
: 2930 1 [20] = %o'022776'
: 2931 1 [21] = %o'023040'
: 2932 1 [22] = %o'023305'
: 2933 1 [23] = %o'060010'
: 2934 1 [24] = %o'104207'
: 2935 1 [25] = %o'003320'
: 2936 1 [26] = %o'104201'
: 2937 1 [27] = %o'000001'
: 2938 1 [28] = %o'060023'
: 2939 1 [29] = %o'000000'
: 2940 1 [30] = %o'114000'
: 2941 1 [31] = %o'003321'
: 2942 1 [32] = %o'104200'
: 2943 1 [33] = %o'000040'
: 2944 1 [34] = %o'003322'
: 2945 1 [35] = %o'102200'
: 2946 1 [36] = %o'000001'
: 2947 1 [37] = %o'003320'
: 2948 1 [38] = %o'012775'
: 2949 1 [39] = %o'104200'
: 2950 1 [40] = %o'000100'
: 2951 1 [41] = %o'003321'
: 2952 1 [42] = %o'104200'
: 2953 1 [43] = %o'000140'
: 2954 1 [44] = %o'003322'
: 2955 1 [45] = %o'000000'
: 2956 1 [46] = %o'114000'
: 2957 1 [47] = %o'003330'
: 2958 1 [48] = %o'114000'
: 2959 1 [49] = %o'003332'
: 2960 1 [50] = %o'104300'
: 2961 1 [51] = %o'003321'
: 2961 1 [52] = %o'003336'

```

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0379
Page 31
(11)

:	2962	1	[53]	=	No' 023170'
:	2963	1	[54]	=	No' 023125'
:	2964	1	[55]	=	No' 102200'
:	2965	1	[56]	=	No' 000001'
:	2966	1	[57]	=	No' 003313'
:	2967	1	[58]	=	No' 053021'
:	2968	1	[59]	=	No' 115400'
:	2969	1	[60]	=	No' 003330'
:	2970	1	[61]	=	No' 106200'
:	2971	1	[62]	=	No' 001000'
:	2972	1	[63]	=	No' 003330'
:	2973	1	[64]	=	No' 033006'
:	2974	1	[65]	=	No' 023225'
:	2975	1	[66]	=	No' 115000'
:	2976	1	[67]	=	No' 003316'
:	2977	1	[68]	=	No' 053030'
:	2978	1	[69]	=	No' 115000'
:	2979	1	[70]	=	No' 003317'
:	2980	1	[71]	=	No' 013037'
:	2981	1	[72]	=	No' 115400'
:	2982	1	[73]	=	No' 003332'
:	2983	1	[74]	=	No' 106200'
:	2984	1	[75]	=	No' 001000'
:	2985	1	[76]	=	No' 003332'
:	2986	1	[77]	=	No' 033002'
:	2987	1	[78]	=	No' 003300'
:	2988	1	[79]	=	No' 000000'
:	2989	1	[80]	=	No' 023170'
:	2990	1	[81]	=	No' 114000'
:	2991	1	[82]	=	No' 003324'
:	2992	1	[83]	=	No' 114000'
:	2993	1	[84]	=	No' 003325'
:	2994	1	[85]	=	No' 115400'
:	2995	1	[86]	=	No' 003324'
:	2996	1	[87]	=	No' 107200'
:	2997	1	[88]	=	No' 000001'
:	2998	1	[89]	=	No' 003325'
:	2999	1	[90]	=	No' 104300'
:	3000	1	[91]	=	No' 003324'
:	3001	1	[92]	=	No' 003326'
:	3002	1	[93]	=	No' 023100'
:	3003	1	[94]	=	No' 023225'
:	3004	1	[95]	=	No' 115000'
:	3005	1	[96]	=	No' 003316'
:	3006	1	[97]	=	No' 053077'
:	3007	1	[98]	=	No' 023170'
:	3008	1	[99]	=	No' 104300'
:	3009	1	[100]	=	No' 003325'
:	3010	1	[101]	=	No' 003326'
:	3011	1	[102]	=	No' 023100'
:	3012	1	[103]	=	No' 023225'
:	3013	1	[104]	=	No' 115000'
:	3014	1	[105]	=	No' 003316'
:	3015	1	[106]	=	No' 053077'
:	3016	1	[107]	=	No' 104300'
:	3017	1	[108]	=	No' 003324'
:	3018	1	[109]	=	No' 003334'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0380
Page 32
(11)

:	3019	1	[110]	=	no'003045'
:	3020	1	[111]	=	no'000000'
:	3021	1	[112]	=	no'114000'
:	3022	1	[113]	=	no'003327'
:	3023	1	[114]	=	no'104307'
:	3024	1	[115]	=	no'003320'
:	3025	1	[116]	=	no'104301'
:	3026	1	[117]	=	no'003326'
:	3027	1	[118]	=	no'104202'
:	3028	1	[119]	=	no'000174'
:	3029	1	[120]	=	no'060013'
:	3030	1	[121]	=	no'103207'
:	3031	1	[122]	=	no'177740'
:	3032	1	[123]	=	no'115007'
:	3033	1	[124]	=	no'013124'
:	3034	1	[125]	=	no'115400'
:	3035	1	[126]	=	no'003327'
:	3036	1	[127]	=	no'106200'
:	3037	1	[128]	=	no'000012'
:	3038	1	[129]	=	no'003327'
:	3039	1	[130]	=	no'033102'
:	3040	1	[131]	=	no'003300'
:	3041	1	[132]	=	no'000000'
:	3042	1	[133]	=	no'104307'
:	3043	1	[134]	=	no'003320'
:	3044	1	[135]	=	no'060014'
:	3045	1	[136]	=	no'103207'
:	3046	1	[137]	=	no'177740'
:	3047	1	[138]	=	no'115007'
:	3048	1	[139]	=	no'053161'
:	3049	1	[140]	=	no'104110'
:	3050	1	[141]	=	no'003313'
:	3051	1	[142]	=	no'106200'
:	3052	1	[143]	=	no'000440'
:	3053	1	[144]	=	no'003313'
:	3054	1	[145]	=	no'033143'
:	3055	1	[146]	=	no'003161'
:	3056	1	[147]	=	no'115401'
:	3057	1	[148]	=	no'104110'
:	3058	1	[149]	=	no'003314'
:	3059	1	[150]	=	no'104117'
:	3060	1	[151]	=	no'103207'
:	3061	1	[152]	=	no'144000'
:	3062	1	[153]	=	no'053161'
:	3063	1	[154]	=	no'104117'
:	3064	1	[155]	=	no'103207'
:	3065	1	[156]	=	no'104000'
:	3066	1	[157]	=	no'106207'
:	3067	1	[158]	=	no'040000'
:	3068	1	[159]	=	no'053161'
:	3069	1	[160]	=	no'000000'
:	3070	1	[161]	=	no'115400'
:	3071	1	[162]	=	no'003331'
:	3072	1	[163]	=	no'106200'
:	3073	1	[164]	=	no'000144'
:	3074	1	[165]	=	no'003331'
:	3075	1	[166]	=	no'033125'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0381
Page 33
(11)

:	3076	1	[167]	=	'003300'
:	3077	1	[168]	=	'100467'
:	3078	1	[169]	=	'114000'
:	3079	1	[170]	=	'003315'
:	3080	1	[171]	=	'104200'
:	3081	1	[172]	=	'001475'
:	3082	1	[173]	=	'003335'
:	3083	1	[174]	=	'104307'
:	3084	1	[175]	=	'003320'
:	3085	1	[176]	=	'104301'
:	3086	1	[177]	=	'003336'
:	3087	1	[178]	=	'104302'
:	3088	1	[179]	=	'003335'
:	3089	1	[180]	=	'060015'
:	3090	1	[181]	=	'103207'
:	3091	1	[182]	=	'177740'
:	3092	1	[183]	=	'106020'
:	3093	1	[184]	=	'003335'
:	3094	1	[185]	=	'053216'
:	3095	1	[186]	=	'115007'
:	3096	1	[187]	=	'053216'
:	3097	1	[188]	=	'104267'
:	3098	1	[189]	=	'000000'
:	3099	1	[190]	=	'115400'
:	3100	1	[191]	=	'003315'
:	3101	1	[192]	=	'106200'
:	3102	1	[193]	=	'000012'
:	3103	1	[194]	=	'003315'
:	3104	1	[195]	=	'033176'
:	3105	1	[196]	=	'003300'
:	3106	1	[197]	=	'100467'
:	3107	1	[198]	=	'114000'
:	3108	1	[199]	=	'003316'
:	3109	1	[200]	=	'114000'
:	3110	1	[201]	=	'003317'
:	3111	1	[202]	=	'104200'
:	3112	1	[203]	=	'100000'
:	3113	1	[204]	=	'003400'
:	3114	1	[205]	=	'104200'
:	3115	1	[206]	=	'003414'
:	3116	1	[207]	=	'003403'
:	3117	1	[208]	=	'104300'
:	3118	1	[209]	=	'003313'
:	3119	1	[210]	=	'003404'
:	3120	1	[211]	=	'104300'
:	3121	1	[212]	=	'003314'
:	3122	1	[213]	=	'003405'
:	3123	1	[214]	=	'104300'
:	3124	1	[215]	=	'003313'
:	3125	1	[216]	=	'003412'
:	3126	1	[217]	=	'104300'
:	3127	1	[218]	=	'003314'
:	3128	1	[219]	=	'003413'
:	3129	1	[220]	=	'104307'
:	3130	1	[221]	=	'003320'
:	3131	1	[222]	=	'104201'
:	3132	1	[223]	=	'003400'

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0382
Page 34
(11)

:	3133	1	[224]	=	%0'060002'
:	3134	1	[225]	=	%0'103207'
:	3135	1	[226]	=	%0'177740'
:	3136	1	[227]	=	%0'115007'
:	3137	1	[228]	=	%0'013267'
:	3138	1	[229]	=	%0'115400'
:	3139	1	[230]	=	%0'003316'
:	3140	1	[231]	=	%0'104307'
:	3141	1	[232]	=	%0'003400'
:	3142	1	[233]	=	%0'102207'
:	3143	1	[234]	=	%0'020000'
:	3144	1	[235]	=	%0'013276'
:	3145	1	[236]	=	%0'115400'
:	3146	1	[237]	=	%0'003317'
:	3147	1	[238]	=	%0'104267'
:	3148	1	[239]	=	%0'000000'
:	3149	1	[240]	=	%0'104200'
:	3150	1	[241]	=	%0'000106'
:	3151	1	[242]	=	%0'003333'
:	3152	1	[243]	=	%0'023305'
:	3153	1	[244]	=	%0'060010'
:	3154	1	[245]	=	%0'104207'
:	3155	1	[246]	=	%0'003333'
:	3156	1	[247]	=	%0'104201'
:	3157	1	[248]	=	%0'000002'
:	3158	1	[249]	=	%0'060022'
:	3159	1	[250]	=	%0'000000'
:	3160	1	[251]	=	%0'000000'
:	3161	1	[252]	=	%0'140000'
:	3162	1	[253]	=	%0'000000'
:	3163	1	[254]	=	%0'000000'
:	3164	1	[255]	=	%0'000000'
:	3165	1	[256]	=	%0'000000'
:	3166	1	[257]	=	%0'000000'
:	3167	1	[258]	=	%0'000000'
:	3168	1	[259]	=	%0'000000'
:	3169	1	[260]	=	%0'000000'
:	3170	1	[261]	=	%0'000000'
:	3171	1	[262]	=	%0'000000'
:	3172	1	[263]	=	%0'000000'
:	3173	1	[264]	=	%0'000000'
:	3174	1	[265]	=	%0'000000'
:	3175	1	[266]	=	%0'000000'
:	3176	1	[267]	=	%0'000104'
:	3177	1	[268]	=	%0'000000'
:	3178	1	[269]	=	%0'000000'
:	3179	1	[270]	=	%0'000000'
:	3180	1	[271]	=	%0'000000'
:	3181	1	[272]	=	%0'000000'
:	3182	1	[273]	=	%0'000000'
:	3183	1	[274]	=	%0'000000'
:	3184	1	[275]	=	%0'000000'
:	3185	1	[276]	=	%0'000000'
:	3186	1	[277]	=	%0'000000'
:	3187	1	[278]	=	%0'000000'
:	3188	1	[279]	=	%0'000000'
:	3189	1	[280]	=	%0'000000'

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

```

: 3190 1 [281] = %'000000';
: 3191 1 [282] = %'000000';
: 3192 1 [283] = %'000000';
: 3193 1 [284] = %'000000';
: 3194 1 [285] = %'000000';
: 3195 1 [286] = %'000000';
: 3196 1 [287] = %'000000';
: 3197 1 [288] = %'000000';
: 3198 1 [289] = %'000000';
: 3199 1 [290] = %'000000';
: 3200 1 [291] = %'000000';
: 3201 1 [292] = %'000000';
: 3202 1 [293] = %'000000';
: 3203 1 [294] = %'000000';
: 3204 1 [295] = %'000000';
: 3205 1 [296] = %'000000';
: 3206 1 [297] = %'000000';
: 3207 1 [298] = %'000000';
: 3208 1 [299] = %'000000';
: 3209 1 [300] = %'000000';
: 3210 1 [301] = %'000000';
: 3211 1 [302] = %'000000';
: 3212 1 [303] = %'000000';
: 3213 1 [304] = %'000000';
: 3214 1 [305] = %'120475';
: 3215 1 [306] = %'000000';
: 3216 1
: 3217 1 end
: 3218 1
: 3219 0 eludom

```

.TITLE ZRCFB4 CZRCFC0 RC25 FR END TEST
.IDENT /V03.0/

```

000000
000000 000270
000002 000000
000004 000000
000006 000000
000010 042524
000012 052123
000014 034460
000016 000000
000020 126411
000022 000000
000024 000000
000026 000000
000030 000000
000032 000000
000034 000000
000036 000000
000040 104206
000042 003051
000044 114000
000046 003037

DM.09: .PSECT DM$CODE, RO, D, GBL
        .WORD 270
        .WORD 0
        .WORD 0
        .WORD 0
        .WORD 42524
        .WORD 52123
        .WORD 34460
        .WORD 0
        .WORD -51367
        .WORD 0
        .WORD 0
        .WORD 0
        .WORD 0
        .WORD 0
        .WORD 0
        .WORD 0
        .WORD -73572
        .WORD 3051
        .WORD -64000
        .WORD 3037

```

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0384
Page 36
(11)

000050	104207	.WORD	-73571
000052	003032	.WORD	3032
000054	104201	.WORD	-73577
000056	000003	.WORD	3
000060	060023	.WORD	60023
000062	103207	.WORD	-74571
000064	177740	.WORD	-40
000066	115007	.WORD	-62771
000070	012756	.WORD	12756
000072	003003	.WORD	3003
000074	114000	.WORD	-64000
000076	003052	.WORD	3052
000100	104307	.WORD	-73471
000102	003032	.WORD	3032
000104	104301	.WORD	-73477
000106	003033	.WORD	3033
000110	104302	.WORD	-73476
000112	003034	.WORD	3034
000114	104203	.WORD	-73575
000116	003052	.WORD	3052
000120	060020	.WORD	60020
000122	103207	.WORD	-74571
000124	177740	.WORD	-40
000126	115007	.WORD	-62771
000130	013007	.WORD	13007
000132	115400	.WORD	-62400
000134	003037	.WORD	3037
000136	106300	.WORD	-71500
000140	003035	.WORD	3035
000142	003037	.WORD	3037
000144	032756	.WORD	32756
000146	104200	.WORD	-73600
000150	000106	.WORD	106
000152	003040	.WORD	3040
000154	003024	.WORD	3024
000156	104207	.WORD	-73571
000160	003052	.WORD	3052
000162	104201	.WORD	-73577
000164	125252	.WORD	-52526
000166	104302	.WORD	-73476
000170	003034	.WORD	3034
000172	106271	.WORD	-71507
000174	053003	.WORD	53003
000176	117402	.WORD	-60376
000200	053015	.WORD	53015
000202	104200	.WORD	-73600
000204	000104	.WORD	104
000206	003040	.WORD	3040
000210	104207	.WORD	-73571
000212	003040	.WORD	3040
000214	104201	.WORD	-73577
000216	000001	.WORD	1
000220	060022	.WORD	60022
000222	060010	.WORD	60010
000224	000000	.WORD	0
000226	000000	.WORD	0
000230	000000	.WORD	0

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0385
Page 37
(11)

000232	000012		.WORD	12
000234	000000		.WORD	0
000236	000000		.WORD	0
000240	000000		.WORD	0
000242	000000		.WORD	0
000244	000000		.WORD	0
000246	000000		.WORD	0
000250	000000		.WORD	0
000252	000000		.WORD	0
000254	000000		.WORD	0
000256	000000		.WORD	0
000260	000000		.WORD	0
000262	000000		.WORD	0
000264	000000		.WORD	0
000266	144423		.WORD	-33355
000270	000000		.WORD	0
000272	000162	DM.10::	.WORD	162
000274	000000		.WORD	0
000276	000000		.WORD	0
000300	000000		.WORD	0
000302	042524		.WORD	42524
000304	052123		.WORD	52123
000306	030061		.WORD	30061
000310	000000		.WORD	0
000312	126411		.WORD	-51367
000314	000000		.WORD	0
000316	000000		.WORD	0
000320	000000		.WORD	0
000322	000000		.WORD	0
000324	000000		.WORD	0
000326	000000		.WORD	0
000330	000000		.WORD	0
000332	104206		.WORD	-73572
000334	003007		.WORD	3007
000336	104207		.WORD	-73571
000340	160000		.WORD	-20000
000342	104201		.WORD	-73577
000344	077777		.WORD	77777
000346	104202		.WORD	-73576
000350	000001		.WORD	1
000352	104203		.WORD	-73575
000354	003500		.WORD	3500
000356	060021		.WORD	60021
000360	103207		.WORD	-74571
000362	177740		.WORD	-40
000364	104070		.WORD	-73710
000366	002765		.WORD	2765
000370	104207		.WORD	-73571
000372	002765		.WORD	2765
000374	104201		.WORD	-73577
000376	000001		.WORD	1
000400	060022		.WORD	60022
000402	060010		.WORD	60010
000404	000000		.WORD	0
000406	000000		.WORD	0
000410	000000		.WORD	0
000412	000000		.WORD	0

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0386
Page 38
(11)

000414	000000		.WORD	0
000416	000000		.WORD	0
000420	000000		.WORD	0
000422	000000		.WORD	0
000424	000000		.WORD	0
000426	000000		.WORD	0
000430	000000		.WORD	0
000432	000000		.WORD	0
000434	000000		.WORD	0
000436	000000		.WORD	0
000440	000000		.WORD	0
000442	000000		.WORD	0
000444	000000		.WORD	0
000446	000000		.WORD	0
000450	000000		.WORD	0
000452	030037		.WORD	30037
000454	000000		.WORD	0
000456	000306	DM.11::	.WORD	306
000460	000000		.WORD	0
000462	000000		.WORD	0
000464	000000		.WORD	0
000466	042524		.WORD	42524
000470	052123		.WORD	52123
000472	030461		.WORD	30461
000474	000000		.WORD	0
000476	126411		.WORD	-51367
000500	000000		.WORD	0
000502	000000		.WORD	0
000504	000000		.WORD	0
000506	000000		.WORD	0
000510	000000		.WORD	0
000512	000000		.WORD	0
000514	000000		.WORD	0
000516	104206		.WORD	-73572
000520	003061		.WORD	3061
000522	104207		.WORD	-73571
000524	003040		.WORD	3040
000526	104201		.WORD	-73577
000530	000003		.WORD	3
000532	060023		.WORD	60023
000534	103207		.WORD	-74571
000536	177740		.WORD	-40
000540	115007		.WORD	-62771
000542	012754		.WORD	12754
000544	003023		.WORD	3023
000546	104200		.WORD	-73600
000550	000001		.WORD	1
000552	003043		.WORD	3043
000554	104300		.WORD	-73500
000556	003040		.WORD	3040
000560	003044		.WORD	3044
000562	104304		.WORD	-73474
000564	003042		.WORD	3042
000566	114000		.WORD	-64000
000570	003046		.WORD	3046
000572	104307		.WORD	-73471
000574	003040		.WORD	3040

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0387
Page 39
(11)

000576	104301	.WORD	-73477
000600	003041	.WORD	3041
000602	104302	.WORD	-73476
000604	003043	.WORD	3043
000606	104203	.WORD	-73575
000610	003044	.WORD	3044
000612	060021	.WORD	60021
000614	103207	.WORD	-74571
000616	177740	.WORD	-40
000620	115007	.WORD	-62771
000622	013012	.WORD	13012
000624	115400	.WORD	-62400
000626	003046	.WORD	3046
000630	106200	.WORD	-71600
000632	000012	.WORD	12
000634	003046	.WORD	3046
000636	032766	.WORD	32766
000640	003023	.WORD	3023
000642	117404	.WORD	-60374
000644	013027	.WORD	13027
000646	105200	.WORD	-72600
000650	000002	.WORD	2
000652	003040	.WORD	3040
000654	104300	.WORD	-73500
000656	003040	.WORD	3040
000660	003044	.WORD	3044
000662	002764	.WORD	2764
000664	104200	.WORD	-73600
000666	000106	.WORD	106
000670	003045	.WORD	3045
000672	003032	.WORD	3032
000674	104200	.WORD	-73600
000676	000104	.WORD	104
000700	003045	.WORD	3045
000702	104207	.WORD	-73571
000704	003045	.WORD	3045
000706	104201	.WORD	-73577
000710	000001	.WORD	1
000712	060022	.WORD	60022
000714	060010	.WORD	60010
000716	000000	.WORD	0
000720	000000	.WORD	0
000722	000000	.WORD	0
000724	000000	.WORD	0
000726	000000	.WORD	0
000730	000000	.WORD	0
000732	000000	.WORD	0
000734	000000	.WORD	0
000736	000000	.WORD	0
000740	000000	.WORD	0
000742	000000	.WORD	0
000744	000000	.WORD	0
000746	000000	.WORD	0
000750	000000	.WORD	0
000752	000000	.WORD	0
000754	000000	.WORD	0
000756	000000	.WORD	0

M14

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0388
Page 40
(11)

000760	000000		.WORD	0
000762	056247		.WORD	56247
000764	000000		.WORD	0
000766	000622	DM.12::	.WORD	622
000770	000000		.WORD	0
000772	000000		.WORD	0
000774	000000		.WORD	0
000776	042524		.WORD	42524
001000	052123		.WORD	52123
001002	031061		.WORD	31061
001004	000000		.WORD	0
001006	177411		.WORD	-367
001010	000000		.WORD	0
001012	000000		.WORD	0
001014	000000		.WORD	0
001016	000000		.WORD	0
001020	000000		.WORD	0
001022	000000		.WORD	0
001024	000000		.WORD	0
001026	104206		.WORD	-73572
001030	002767		.WORD	2767
001032	003004		.WORD	3004
001034	000000		.WORD	0
001036	000000		.WORD	0
001040	000000		.WORD	0
001042	000000		.WORD	0
001044	000000		.WORD	0
001046	000000		.WORD	0
001050	000000		.WORD	0
001052	000000		.WORD	0
001054	000000		.WORD	0
001056	000000		.WORD	0
001060	000000		.WORD	0
001062	000000		.WORD	0
001064	000000		.WORD	0
001066	000000		.WORD	0
001070	000000		.WORD	0
001072	000000		.WORD	0
001074	000000		.WORD	0
001076	000000		.WORD	0
001100	000000		.WORD	0
001102	000000		.WORD	0
001104	000000		.WORD	0
001106	000000		.WORD	0
001110	000000		.WORD	0
001112	000000		.WORD	0
001114	000104		.WORD	104
001116	000106		.WORD	106
001120	000000		.WORD	0
001122	000000		.WORD	0
001124	000000		.WORD	0
001126	000000		.WORD	0
001130	000000		.WORD	0
001132	000000		.WORD	0
001134	000000		.WORD	0
001136	023016		.WORD	23016
001140	023031		.WORD	23031

N14

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0389
Page 41
(11)

001142	023210	.WORD	23210
001144	023120	.WORD	23120
001146	023126	.WORD	23126
001150	023210	.WORD	23210
001152	023120	.WORD	23120
001154	023155	.WORD	23155
001156	023210	.WORD	23210
001160	060010	.WORD	60010
001162	104207	.WORD	-73571
001164	002770	.WORD	2770
001166	104201	.WORD	-73577
001170	000003	.WORD	3
001172	060023	.WORD	60023
001174	103207	.WORD	-74571
001176	177740	.WORD	-40
001200	115007	.WORD	-62771
001202	013030	.WORD	13030
001204	003203	.WORD	3203
001206	000000	.WORD	0
001210	104300	.WORD	-73500
001212	002770	.WORD	2770
001214	002777	.WORD	2777
001216	104300	.WORD	-73500
001220	002771	.WORD	2771
001222	003000	.WORD	3000
001224	104301	.WORD	-73477
001226	002772	.WORD	2772
001230	104207	.WORD	-73571
001232	177777	.WORD	-1
001234	107307	.WORD	-70471
001236	002777	.WORD	2777
001240	104070	.WORD	-73710
001242	003002	.WORD	3002
001244	023063	.WORD	23063
001246	105200	.WORD	-72600
001250	000002	.WORD	2
001252	002777	.WORD	2777
001254	115000	.WORD	-63000
001256	002777	.WORD	2777
001260	053060	.WORD	53060
001262	115400	.WORD	-62400
001264	003000	.WORD	3000
001266	117401	.WORD	-60377
001270	053041	.WORD	53041
001272	000000	.WORD	0
001274	100467	.WORD	-77311
001276	100461	.WORD	-77317
001300	100462	.WORD	-77316
001302	100463	.WORD	-77315
001304	104307	.WORD	-73471
001306	002777	.WORD	2777
001310	104301	.WORD	-73477
001312	003000	.WORD	3000
001314	104202	.WORD	-73576
001316	000001	.WORD	1
001320	104203	.WORD	-73575
001322	003002	.WORD	3002

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0390
Page 42
(11)

001324	060021	.WORD	60021
001326	103207	.WORD	-74571
001330	177740	.WORD	-40
001332	115007	.WORD	-62771
001334	013113	.WORD	13113
001336	115400	.WORD	-62400
001340	003003	.WORD	3003
001342	106200	.WORD	-71600
001344	000012	.WORD	12
001346	003003	.WORD	3003
001350	033067	.WORD	33067
001352	003203	.WORD	3203
001354	104263	.WORD	-73515
001356	104262	.WORD	-73516
001360	104261	.WORD	-73517
001362	104267	.WORD	-73511
001364	000000	.WORD	0
001366	104207	.WORD	-73571
001370	002775	.WORD	2775
001372	104201	.WORD	-73577
001374	000001	.WORD	1
001376	060023	.WORD	60023
001400	000000	.WORD	0
001402	104300	.WORD	-73500
001404	002770	.WORD	2770
001406	002777	.WORD	2777
001410	104300	.WORD	-73500
001412	002771	.WORD	2771
001414	003000	.WORD	3000
001416	104301	.WORD	-73477
001420	002772	.WORD	2772
001422	104200	.WORD	-73600
001424	177777	.WORD	-1
001426	003002	.WORD	3002
001430	023063	.WORD	23063
001432	105200	.WORD	-72600
001434	000002	.WORD	2
001436	002777	.WORD	2777
001440	115000	.WORD	-63000
001442	002777	.WORD	2777
001444	053152	.WORD	53152
001446	115400	.WORD	-62400
001450	003000	.WORD	3000
001452	117401	.WORD	-60377
001454	053141	.WORD	53141
001456	000000	.WORD	0
001460	104300	.WORD	-73500
001462	002770	.WORD	2770
001464	002777	.WORD	2777
001466	104300	.WORD	-73500
001470	002771	.WORD	2771
001472	003000	.WORD	3000
001474	104301	.WORD	-73477
001476	002772	.WORD	2772
001500	114000	.WORD	-64000
001502	003002	.WORD	3002
001504	023063	.WORD	23063

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

001506	105200	.WORD	-72600
001510	000002	.WORD	2
001512	002777	.WORD	2777
001514	115000	.WORD	-63000
001516	002777	.WORD	2777
001520	053200	.WORD	53200
001522	115400	.WORD	-62400
001524	003000	.WORD	3000
001526	117401	.WORD	-60377
001530	053167	.WORD	53167
001532	000000	.WORD	0
001534	104300	.WORD	-73500
001536	002774	.WORD	2774
001540	002776	.WORD	2776
001542	023215	.WORD	23215
001544	060010	.WORD	60010
001546	104300	.WORD	-73500
001550	002773	.WORD	2773
001552	002776	.WORD	2776
001554	023215	.WORD	23215
001556	000000	.WORD	0
001560	104207	.WORD	-73571
001562	002776	.WORD	2776
001564	104201	.WORD	-73577
001566	000001	.WORD	1
001570	060022	.WORD	60022
001572	103207	.WORD	-74571
001574	177740	.WORD	-40
001576	115007	.WORD	-62771
001600	013227	.WORD	13227
001602	060010	.WORD	60010
001604	000000	.WORD	0
001606	165572	.WORD	-12206
001610	000000	.WORD	0
001612	000320	DM.13:: .WORD	320
001614	000000	.WORD	0
001616	000000	.WORD	0
001620	000000	.WORD	0
001622	042524	.WORD	42524
001624	052123	.WORD	52123
001626	031461	.WORD	31461
001630	000000	.WORD	0
001632	126411	.WORD	-51367
001634	000000	.WORD	0
001636	000000	.WORD	0
001640	000000	.WORD	0
001642	000000	.WORD	0
001644	000000	.WORD	0
001646	000000	.WORD	0
001650	000000	.WORD	0
001652	104206	.WORD	-73572
001654	003065	.WORD	3065
001656	002743	.WORD	2743
001660	104200	.WORD	-73600
001662	000104	.WORD	104
001664	003045	.WORD	3045
001666	104207	.WORD	-73571

ZRCFR4
V03.0CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0392
Page 44
(11)

001670	003047	.WORD	3047
001672	104201	.WORD	-73577
001674	000004	.WORD	4
001676	060023	.WORD	60023
001700	103200	.WORD	-74600
001702	000001	.WORD	1
001704	003047	.WORD	3047
001706	103200	.WORD	-74600
001710	000001	.WORD	1
001712	003051	.WORD	3051
001714	114000	.WORD	-64000
001716	003046	.WORD	3046
001720	104307	.WORD	-73471
001722	003047	.WORD	3047
001724	104301	.WORD	-73477
001726	003050	.WORD	3050
001730	104202	.WORD	-73576
001732	000400	.WORD	400
001734	104203	.WORD	-73575
001736	003066	.WORD	3066
001740	060020	.WORD	60020
001742	115007	.WORD	-62771
001744	013005	.WORD	13005
001746	115400	.WORD	-62400
001750	003046	.WORD	3046
001752	106200	.WORD	-71600
001754	000012	.WORD	12
001756	003046	.WORD	3046
001760	032763	.WORD	32763
001762	003041	.WORD	3041
001764	114000	.WORD	-64000
001766	003046	.WORD	3046
001770	104307	.WORD	-73471
001772	003051	.WORD	3051
001774	104301	.WORD	-73477
001776	003052	.WORD	3052
002000	104202	.WORD	-73576
002002	000400	.WORD	400
002004	104203	.WORD	-73575
002006	003066	.WORD	3066
002010	060021	.WORD	60021
002012	103207	.WORD	-74571
002014	177740	.WORD	-40
002016	115007	.WORD	-62771
002020	013033	.WORD	13033
002022	115400	.WORD	-62400
002024	003046	.WORD	3046
002026	106200	.WORD	-71600
002030	000012	.WORD	12
002032	003046	.WORD	3046
002034	033007	.WORD	33007
002036	003041	.WORD	3041
002040	104207	.WORD	-73571
002042	003045	.WORD	3045
002044	104201	.WORD	-73577
002046	000001	.WORD	1
002050	060022	.WORD	60022

E15

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0393
Page 45
(11)

002052	002743	.WORD	2743
002054	104200	.WORD	-73600
002056	000106	.WORD	106
002060	003045	.WORD	3045
002062	003033	.WORD	3033
002064	000104	.WORD	104
002066	000000	.WORD	0
002070	000000	.WORD	0
002072	000000	.WORD	0
002074	000000	.WORD	0
002076	000000	.WORD	0
002100	000000	.WORD	0
002102	000000	.WORD	0
002104	000000	.WORD	0
002106	000000	.WORD	0
002110	000000	.WORD	0
002112	000000	.WORD	0
002114	000000	.WORD	0
002116	000000	.WORD	0
002120	000000	.WORD	0
002122	000000	.WORD	0
002124	000000	.WORD	0
002126	000000	.WORD	0
002130	113606	.WORD	-64172
002132	000000	.WORD	0
002134	000466	DM.19:: .WORD	466
002136	000000	.WORD	0
002140	000000	.WORD	0
002142	000000	.WORD	0
002144	042524	.WORD	42524
002146	052123	.WORD	52123
002150	034461	.WORD	34461
002152	000000	.WORD	0
002154	126411	.WORD	-51367
002156	000000	.WORD	0
002160	000000	.WORD	0
002162	000000	.WORD	0
002164	000000	.WORD	0
002166	000000	.WORD	0
002170	000000	.WORD	0
002172	000000	.WORD	0
002174	104206	.WORD	-73572
002176	003151	.WORD	3151
002200	104207	.WORD	-73571
002202	003127	.WORD	3127
002204	104201	.WORD	-73577
002206	000001	.WORD	1
002210	060023	.WORD	60023
002212	114000	.WORD	-64000
002214	003134	.WORD	3134
002216	104200	.WORD	-73600
002220	000040	.WORD	40
002222	003135	.WORD	3135
002224	102200	.WORD	-75600
002226	000001	.WORD	1
002230	003127	.WORD	3127
002232	012766	.WORD	12766

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0394
Page 46
(11)

002234	104200	.WORD	-73600
002236	000100	.WORD	100
002240	003134	.WORD	3134
002242	104200	.WORD	-73600
002244	000140	.WORD	140
002246	003135	.WORD	3135
002250	104300	.WORD	-73500
002252	003134	.WORD	3134
002254	003130	.WORD	3130
002256	114000	.WORD	-64000
002260	003131	.WORD	3131
002262	023006	.WORD	23006
002264	104300	.WORD	-73500
002266	003135	.WORD	3135
002270	003130	.WORD	3130
002272	114000	.WORD	-64000
002274	003131	.WORD	3131
002276	023006	.WORD	23006
002300	104200	.WORD	-73600
002302	000104	.WORD	104
002304	003126	.WORD	3126
002306	003047	.WORD	3047
002310	104200	.WORD	-73600
002312	000012	.WORD	12
002314	003132	.WORD	3132
002316	104307	.WORD	-73471
002320	003127	.WORD	3127
002322	104301	.WORD	-73477
002324	003130	.WORD	3130
002326	104302	.WORD	-73476
002330	003131	.WORD	3131
002332	060015	.WORD	60015
002334	103207	.WORD	-74571
002336	177740	.WORD	-40
002340	115007	.WORD	-62771
002342	053036	.WORD	53036
002344	103201	.WORD	-74577
002346	177637	.WORD	-141
002350	106301	.WORD	-71477
002352	003130	.WORD	3130
002354	053036	.WORD	53036
002356	106302	.WORD	-71476
002360	003131	.WORD	3131
002362	053036	.WORD	53036
002364	023055	.WORD	23055
002366	000000	.WORD	0
002370	117400	.WORD	-60400
002372	003132	.WORD	3132
002374	115000	.WORD	-63000
002376	003132	.WORD	3132
002400	073044	.WORD	73044
002402	003011	.WORD	3011
002404	104200	.WORD	-73600
002406	000106	.WORD	106
002410	003126	.WORD	3126
002412	104207	.WORD	-73571
002414	003126	.WORD	3126

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0395
Page 47
(11)

002416	104201	.WORD	-73577
002420	000005	.WORD	5
002422	060022	.WORD	60022
002424	060010	.WORD	60010
002426	114000	.WORD	-64000
002430	003133	.WORD	3133
002432	104307	.WORD	-73471
002434	003127	.WORD	3127
002436	060014	.WORD	60014
002440	103207	.WORD	-74571
002442	177740	.WORD	-40
002444	115007	.WORD	-62771
002446	053112	.WORD	53112
002450	104110	.WORD	-73670
002452	003136	.WORD	3136
002454	102200	.WORD	-75600
002456	000040	.WORD	40
002460	003130	.WORD	3130
002462	053101	.WORD	53101
002464	106200	.WORD	-71600
002466	000040	.WORD	40
002470	003136	.WORD	3136
002472	033121	.WORD	33121
002474	003112	.WORD	3112
002476	106200	.WORD	-71600
002500	000437	.WORD	437
002502	003136	.WORD	3136
002504	033121	.WORD	33121
002506	003112	.WORD	3112
002510	106200	.WORD	-71600
002512	000377	.WORD	377
002514	003136	.WORD	3136
002516	073121	.WORD	73121
002520	115400	.WORD	-62400
002522	003133	.WORD	3133
002524	106200	.WORD	-71600
002526	000024	.WORD	24
002530	003133	.WORD	3133
002532	033057	.WORD	33057
002534	003122	.WORD	3122
002536	000000	.WORD	0
002540	104200	.WORD	-73600
002542	000106	.WORD	106
002544	003126	.WORD	3126
002546	003047	.WORD	3047
002550	000000	.WORD	0
002552	000000	.WORD	0
002554	000000	.WORD	0
002556	000000	.WORD	0
002560	000000	.WORD	0
002562	000000	.WORD	0
002564	000000	.WORD	0
002566	000000	.WORD	0
002570	000000	.WORD	0
002572	000000	.WORD	0
002574	000000	.WORD	0
002576	000000	.WORD	0

H15

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 B11es-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0396
Page 48
(11)

002600	000000		.WORD	0
002602	000000		.WORD	0
002604	000000		.WORD	0
002606	000000		.WORD	0
002610	000000		.WORD	0
002612	000000		.WORD	0
002614	000000		.WORD	0
002616	000000		.WORD	0
002620	044310		.WORD	44310
002622	000000		.WORD	0
002624	000650	DM.21::	.WORD	650
002626	000000		.WORD	0
002630	000000		.WORD	0
002632	000000		.WORD	0
002634	042524		.WORD	42524
002636	052123		.WORD	52123
002640	030462		.WORD	30462
002642	000000		.WORD	0
002644	126411		.WORD	-51367
002646	000000		.WORD	0
002650	000000		.WORD	0
002652	000000		.WORD	0
002654	000000		.WORD	0
002656	000000		.WORD	0
002660	000000		.WORD	0
002662	000000		.WORD	0
002664	104206		.WORD	-73572
002666	003223		.WORD	3223
002670	104207		.WORD	-73571
002672	003176		.WORD	3176
002674	104201		.WORD	-73577
002676	000002		.WORD	2
002700	060023		.WORD	60023
002702	114000		.WORD	-64000
002704	003206		.WORD	3206
002706	104200		.WORD	-73600
002710	000040		.WORD	40
002712	003207		.WORD	3207
002714	102200		.WORD	-75600
002716	000001		.WORD	1
002720	003176		.WORD	3176
002722	012771		.WORD	12771
002724	104200		.WORD	-73600
002726	000100		.WORD	100
002730	003206		.WORD	3206
002732	104200		.WORD	-73600
002734	000140		.WORD	140
002736	003207		.WORD	3207
002740	104200		.WORD	-73600
002742	144000		.WORD	-34000
002744	003202		.WORD	3202
002746	104300		.WORD	-73500
002750	003177		.WORD	3177
002752	003205		.WORD	3205
002754	104300		.WORD	-73500
002756	003206		.WORD	3206
002760	003177		.WORD	3177

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0397
Page 49
(11)

002762	115000	.WORD	-63000
002764	003205	.WORD	3205
002766	053005	.WORD	53005
002770	104300	.WORD	-73500
002772	003207	.WORD	3207
002774	003177	.WORD	3177
002776	114000	.WORD	-64000
003000	003200	.WORD	3200
003002	023015	.WORD	23015
003004	023045	.WORD	23045
003006	104200	.WORD	-73600
003010	000104	.WORD	104
003012	003175	.WORD	3175
003014	003167	.WORD	3167
003016	114000	.WORD	-64000
003020	003203	.WORD	3203
003022	104307	.WORD	-73471
003024	003176	.WORD	3176
003026	104301	.WORD	-73477
003030	003177	.WORD	3177
003032	104302	.WORD	-73476
003034	003200	.WORD	3200
003036	060015	.WORD	60015
003040	103207	.WORD	-74571
003042	177740	.WORD	-40
003044	106020	.WORD	-71760
003046	003200	.WORD	3200
003050	053036	.WORD	53036
003052	115007	.WORD	-62771
003054	053036	.WORD	53036
003056	000000	.WORD	0
003060	115400	.WORD	-62400
003062	003203	.WORD	3203
003064	106200	.WORD	-71600
003066	000012	.WORD	12
003070	003203	.WORD	3203
003072	033017	.WORD	33017
003074	003164	.WORD	3164
003076	104200	.WORD	-73600
003100	000040	.WORD	40
003102	003204	.WORD	3204
003104	023121	.WORD	23121
003106	114000	.WORD	-64000
003110	003210	.WORD	3210
003112	104200	.WORD	-73600
003114	100000	.WORD	-100000
003116	003242	.WORD	3242
003120	104200	.WORD	-73600
003122	003256	.WORD	3256
003124	003245	.WORD	3245
003126	104300	.WORD	-73500
003130	003201	.WORD	3201
003132	003246	.WORD	3246
003134	104300	.WORD	-73500
003136	003202	.WORD	3202
003140	003247	.WORD	3247
003142	104300	.WORD	-73500

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0398
Page 50
(11)

003144	003201	.WORD	3201
003146	003254	.WORD	3254
003150	104300	.WORD	-73500
003152	003202	.WORD	3202
003154	003255	.WORD	3255
003156	104307	.WORD	-73471
003160	003176	.WORD	3176
003162	104201	.WORD	-73577
003164	003242	.WORD	3242
003166	060002	.WORD	60002
003170	103207	.WORD	-74571
003172	177740	.WORD	-40
003174	115007	.WORD	-62771
003176	013115	.WORD	13115
003200	115400	.WORD	-62400
003202	003210	.WORD	3210
003204	106200	.WORD	-71600
003206	000012	.WORD	12
003210	003210	.WORD	3210
003212	033050	.WORD	33050
003214	003164	.WORD	3164
003216	117400	.WORD	-60400
003220	003204	.WORD	3204
003222	053050	.WORD	53050
003224	000000	.WORD	0
003226	104307	.WORD	-73471
003230	003176	.WORD	3176
003232	060014	.WORD	60014
003234	103207	.WORD	-74571
003236	177740	.WORD	-40
003240	115007	.WORD	-62771
003242	053155	.WORD	53155
003244	104110	.WORD	-73670
003246	003201	.WORD	3201
003250	106200	.WORD	-71600
003252	000440	.WORD	440
003254	003201	.WORD	3201
003256	033137	.WORD	33137
003260	003155	.WORD	3155
003262	115401	.WORD	-62377
003264	104110	.WORD	-73670
003266	003202	.WORD	3202
003270	104117	.WORD	-73661
003272	103207	.WORD	-74571
003274	144000	.WORD	-34000
003276	053155	.WORD	53155
003300	104117	.WORD	-73661
003302	103207	.WORD	-74571
003304	104000	.WORD	-74000
003306	106207	.WORD	-71571
003310	040000	.WORD	40000
003312	053155	.WORD	53155
003314	000000	.WORD	0
003316	115400	.WORD	-62400
003320	003241	.WORD	3241
003322	106200	.WORD	-71600
003324	000144	.WORD	144

K15

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0399
Page 51
(11)

003326	003241	.WORD	3241
003330	033121	.WORD	33121
003332	003164	.WORD	3164
003334	104200	.WORD	-73600
003336	000106	.WORD	106
003340	003175	.WORD	3175
003342	104207	.WORD	-73571
003344	003175	.WORD	3175
003346	104201	.WORD	-73577
003350	000005	.WORD	5
003352	060022	.WORD	60022
003354	060010	.WORD	60010
003356	000000	.WORD	0
003360	000000	.WORD	0
003362	000000	.WORD	0
003364	000000	.WORD	0
003366	000000	.WORD	0
003370	140000	.WORD	-40000
003372	000000	.WORD	0
003374	000000	.WORD	0
003376	000000	.WORD	0
003400	000000	.WORD	0
003402	000000	.WORD	0
003404	000000	.WORD	0
003406	000000	.WORD	0
003410	000000	.WORD	0
003412	000000	.WORD	0
003414	000000	.WORD	0
003416	000000	.WORD	0
003420	000000	.WORD	0
003422	000000	.WORD	0
003424	000000	.WORD	0
003426	000000	.WORD	0
003430	000000	.WORD	0
003432	000000	.WORD	0
003434	000000	.WORD	0
003436	000000	.WORD	0
003440	000000	.WORD	0
003442	000000	.WORD	0
003444	000000	.WORD	0
003446	000000	.WORD	0
003450	000000	.WORD	0
003452	000000	.WORD	0
003454	000000	.WORD	0
003456	000000	.WORD	0
003460	000000	.WORD	0
003462	000000	.WORD	0
003464	000000	.WORD	0
003466	000000	.WORD	0
003470	000000	.WORD	0
003472	065461	.WORD	65461
003474	000000	.WORD	0
003476	001470	DM.26:: .WORD	1470
003500	000000	.WORD	0
003502	000000	.WORD	0
003504	000000	.WORD	0
003506	042524	.WORD	42524

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0400
Page 52
(11)

003510	052123	.WORD	52123
003512	033062	.WORD	33062
003514	000000	.WORD	0
003516	126411	.WORD	-51367
003520	000000	.WORD	0
003522	000000	.WORD	0
003524	000000	.WORD	0
003526	000000	.WORD	0
003530	000000	.WORD	0
003532	000000	.WORD	0
003534	000000	.WORD	0
003536	104206	.WORD	-73572
003540	003550	.WORD	3550
003542	022777	.WORD	22777
003544	023005	.WORD	23005
003546	104206	.WORD	-73572
003550	003550	.WORD	3550
003552	106200	.WORD	-71600
003554	177777	.WORD	-1
003556	003507	.WORD	3507
003560	012770	.WORD	12770
003562	023030	.WORD	23030
003564	023043	.WORD	23043
003566	104200	.WORD	-73600
003570	177777	.WORD	-1
003572	003507	.WORD	3507
003574	114000	.WORD	-64000
003576	003501	.WORD	3501
003600	114000	.WORD	-64000
003602	003477	.WORD	3477
003604	114000	.WORD	-64000
003606	003500	.WORD	3500
003610	104200	.WORD	-73600
003612	000010	.WORD	10
003614	003473	.WORD	3473
003616	023052	.WORD	23052
003620	023065	.WORD	23065
003622	104200	.WORD	-73600
003624	177777	.WORD	-1
003626	003476	.WORD	3476
003630	023461	.WORD	23461
003632	060010	.WORD	60010
003634	104207	.WORD	-73571
003636	003506	.WORD	3506
003640	104201	.WORD	-73577
003642	000001	.WORD	1
003644	060023	.WORD	60023
003646	000000	.WORD	0
003650	114000	.WORD	-64000
003652	003504	.WORD	3504
003654	104200	.WORD	-73600
003656	000040	.WORD	40
003660	003505	.WORD	3505
003662	102200	.WORD	-75600
003664	000001	.WORD	1
003666	003506	.WORD	3506
003670	013027	.WORD	13027

M15

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0401
Page 53
(11)

003672	104200	.WORD	-73600
003674	000100	.WORD	100
003676	003504	.WORD	3504
003700	104200	.WORD	-73600
003702	000140	.WORD	140
003704	003505	.WORD	3505
003706	104200	.WORD	-73600
003710	144000	.WORD	-34000
003712	003472	.WORD	3472
003714	000000	.WORD	0
003716	104300	.WORD	-73500
003720	003504	.WORD	3504
003722	003503	.WORD	3503
003724	023236	.WORD	23236
003726	023177	.WORD	23177
003730	023116	.WORD	23116
003732	023362	.WORD	23362
003734	023137	.WORD	23137
003736	023177	.WORD	23177
003740	023160	.WORD	23160
003742	000000	.WORD	0
003744	023177	.WORD	23177
003746	023137	.WORD	23137
003750	023362	.WORD	23362
003752	023116	.WORD	23116
003754	023177	.WORD	23177
003756	023160	.WORD	23160
003760	000000	.WORD	0
003762	104300	.WORD	-73500
003764	003505	.WORD	3505
003766	003503	.WORD	3503
003770	023236	.WORD	23236
003772	023177	.WORD	23177
003774	023116	.WORD	23116
003776	023362	.WORD	23362
004000	023137	.WORD	23137
004002	023177	.WORD	23177
004004	023160	.WORD	23160
004006	000000	.WORD	0
004010	023177	.WORD	23177
004012	023137	.WORD	23137
004014	023362	.WORD	23362
004016	023116	.WORD	23116
004020	023177	.WORD	23177
004022	023160	.WORD	23160
004024	000000	.WORD	0
004026	104307	.WORD	-73471
004030	003506	.WORD	3506
004032	104301	.WORD	-73477
004034	003503	.WORD	3503
004036	104302	.WORD	-73476
004040	003501	.WORD	3501
004042	060015	.WORD	60015
004044	104070	.WORD	-73710
004046	003476	.WORD	3476
004050	103207	.WORD	-74571
004052	177740	.WORD	-40

N15

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0402
Page 54
(11)

004054	115007	.WORD	-62771
004056	053115	.WORD	53115
004060	106020	.WORD	-71760
004062	003501	.WORD	3501
004064	053115	.WORD	53115
004066	000000	.WORD	0
004070	003424	.WORD	3424
004072	114007	.WORD	-63771
004074	104301	.WORD	-73477
004076	003475	.WORD	3475
004100	100671	.WORD	-77107
004102	003566	.WORD	3566
004104	115407	.WORD	-62371
004106	106207	.WORD	-71571
004110	000377	.WORD	377
004112	033121	.WORD	33121
004114	104207	.WORD	-73571
004116	003566	.WORD	3566
004120	104201	.WORD	-73577
004122	000400	.WORD	400
004124	060004	.WORD	60004
004126	104010	.WORD	-73770
004130	004166	.WORD	4166
004132	000000	.WORD	0
004134	114007	.WORD	-63771
004136	104301	.WORD	-73477
004140	003474	.WORD	3474
004142	100671	.WORD	-77107
004144	003566	.WORD	3566
004146	115407	.WORD	-62371
004150	106207	.WORD	-71571
004152	000377	.WORD	377
004154	033142	.WORD	33142
004156	104207	.WORD	-73571
004160	003566	.WORD	3566
004162	104201	.WORD	-73577
004164	000400	.WORD	400
004166	060004	.WORD	60004
004170	104010	.WORD	-73770
004172	004166	.WORD	4166
004174	000000	.WORD	0
004176	104301	.WORD	-73477
004200	003477	.WORD	3477
004202	114007	.WORD	-63771
004204	104672	.WORD	-73106
004206	003566	.WORD	3566
004210	106012	.WORD	-71766
004212	053174	.WORD	53174
004214	115407	.WORD	-62371
004216	106207	.WORD	-71571
004220	000377	.WORD	377
004222	033163	.WORD	33163
004224	000000	.WORD	0
004226	104020	.WORD	-73760
004230	003500	.WORD	3500
004232	003424	.WORD	3424
004234	104200	.WORD	-73600

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC4.B16;3

SEQ 0403
Page 55
(11)

004236	100000	.WORD	-100000
004240	003552	.WORD	3552
004242	104200	.WORD	-73600
004244	003566	.WORD	3566
004246	003555	.WORD	3555
004250	104300	.WORD	-73500
004252	003467	.WORD	3467
004254	003556	.WORD	3556
004256	104300	.WORD	-73500
004260	003470	.WORD	3470
004262	003557	.WORD	3557
004264	104300	.WORD	-73500
004266	003467	.WORD	3467
004270	003564	.WORD	3564
004272	104300	.WORD	-73500
004274	003470	.WORD	3470
004276	003565	.WORD	3565
004300	104307	.WORD	-73471
004302	003506	.WORD	3506
004304	104201	.WORD	-73577
004306	003552	.WORD	3552
004310	060002	.WORD	60002
004312	104070	.WORD	-73710
004314	003476	.WORD	3476
004316	103207	.WORD	-74571
004320	177740	.WORD	-40
004322	115007	.WORD	-62771
004324	013235	.WORD	13235
004326	003424	.WORD	3424
004330	000000	.WORD	0
004332	023074	.WORD	23074
004334	114000	.WORD	-64000
004336	003502	.WORD	3502
004340	104307	.WORD	-73471
004342	003506	.WORD	3506
004344	060014	.WORD	60014
004346	104070	.WORD	-73710
004350	003476	.WORD	3476
004352	103207	.WORD	-74571
004354	177740	.WORD	-40
004356	115007	.WORD	-62771
004360	053361	.WORD	53361
004362	114002	.WORD	-63776
004364	104113	.WORD	-73665
004366	106613	.WORD	-71165
004370	000002	.WORD	2
004372	053262	.WORD	53262
004374	106613	.WORD	-71165
004376	000004	.WORD	4
004400	013306	.WORD	13306
004402	106613	.WORD	-71165
004404	000004	.WORD	4
004406	053270	.WORD	53270
004410	106613	.WORD	-71165
004412	000006	.WORD	6
004414	013306	.WORD	13306
004416	106613	.WORD	-71165

C16

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0404
Page 56
(11)

004420	000002	.WORD	2
004422	053276	.WORD	53276
004424	106613	.WORD	-71165
004426	000006	.WORD	6
004430	013306	.WORD	13306
004432	104613	.WORD	-73165
004434	000002	.WORD	2
004436	106613	.WORD	-71165
004440	000004	.WORD	4
004442	053361	.WORD	53361
004444	106613	.WORD	-71165
004446	000006	.WORD	6
004450	053361	.WORD	53361
004452	115002	.WORD	-62776
004454	053316	.WORD	53316
004456	104030	.WORD	-73750
004460	003467	.WORD	3467
004462	115401	.WORD	-62377
004464	104202	.WORD	-73576
004466	177777	.WORD	-1
004470	003253	.WORD	3253
004472	104030	.WORD	-73750
004474	003470	.WORD	3470
004476	104300	.WORD	-73500
004500	003467	.WORD	3467
004502	003502	.WORD	3502
004504	103200	.WORD	-74600
004506	177740	.WORD	-40
004510	003502	.WORD	3502
004512	102200	.WORD	-75600
004514	000001	.WORD	1
004516	003467	.WORD	3467
004520	053361	.WORD	53361
004522	102200	.WORD	-75600
004524	010000	.WORD	10000
004526	003470	.WORD	3470
004530	053361	.WORD	53361
004532	106300	.WORD	-71500
004534	003472	.WORD	3472
004536	003470	.WORD	3470
004540	053361	.WORD	53361
004542	104302	.WORD	-73476
004544	003467	.WORD	3467
004546	105202	.WORD	-72576
004550	000000	.WORD	0
004552	103202	.WORD	-74576
004554	000037	.WORD	37
004556	110602	.WORD	-67176
004560	110602	.WORD	-67176
004562	110602	.WORD	-67176
004564	110602	.WORD	-67176
004566	110602	.WORD	-67176
004570	106302	.WORD	-71476
004572	003473	.WORD	3473
004574	053361	.WORD	53361
004576	000000	.WORD	0
004600	003424	.WORD	3424

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0405
Page 57
(11)

004602	104300	.WORD	-73500
004604	003566	.WORD	3566
004606	003477	.WORD	3477
004610	104200	.WORD	-73600
004612	140000	.WORD	-40000
004614	003552	.WORD	3552
004616	104200	.WORD	-73600
004620	003566	.WORD	3566
004622	003555	.WORD	3555
004624	104300	.WORD	-73500
004626	003467	.WORD	3467
004630	003556	.WORD	3556
004632	104300	.WORD	-73500
004634	003470	.WORD	3470
004636	003557	.WORD	3557
004640	104300	.WORD	-73500
004642	003467	.WORD	3467
004644	003564	.WORD	3564
004646	104300	.WORD	-73500
004650	003470	.WORD	3470
004652	003565	.WORD	3565
004654	104307	.WORD	-73471
004656	003506	.WORD	3506
004660	104201	.WORD	-73577
004662	003552	.WORD	3552
004664	060003	.WORD	60003
004666	104070	.WORD	-73710
004670	003476	.WORD	3476
004672	103207	.WORD	-74571
004674	177740	.WORD	-40
004676	115007	.WORD	-62771
004700	013423	.WORD	13423
004702	003424	.WORD	3424
004704	000000	.WORD	0
004706	115400	.WORD	-62400
004710	003471	.WORD	3471
004712	106200	.WORD	-71600
004714	000100	.WORD	100
004716	003471	.WORD	3471
004720	013433	.WORD	13433
004722	002744	.WORD	2744
004724	115400	.WORD	-62400
004726	003501	.WORD	3501
004730	115400	.WORD	-62400
004732	003473	.WORD	3473
004734	114000	.WORD	-64000
004736	003471	.WORD	3471
004740	106200	.WORD	-71600
004742	000004	.WORD	4
004744	003501	.WORD	3501
004746	053450	.WORD	53450
004750	104200	.WORD	-73600
004752	001473	.WORD	1473
004754	003501	.WORD	3501
004756	106200	.WORD	-71600
004760	001477	.WORD	1477
004762	003501	.WORD	3501

E16

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0406
Page 58
(11)

004764	013455	.WORD	13455
004766	002744	.WORD	2744
004770	117400	.WORD	-60400
004772	003501	.WORD	3501
004774	023461	.WORD	23461
004776	060010	.WORD	60010
005000	104207	.WORD	-73571
005002	003476	.WORD	3476
005004	104201	.WORD	-73577
005006	000006	.WORD	6
005010	060022	.WORD	60022
005012	000000	.WORD	0
005014	000000	.WORD	0
005016	000000	.WORD	0
005020	000000	.WORD	0
005022	140000	.WORD	-40000
005024	000000	.WORD	0
005026	000000	.WORD	0
005030	177777	.WORD	-1
005032	000000	.WORD	0
005034	000000	.WORD	0
005036	000000	.WORD	0
005040	000000	.WORD	0
005042	000000	.WORD	0
005044	000000	.WORD	0
005046	000000	.WORD	0
005050	000000	.WORD	0
005052	000000	.WORD	0
005054	000000	.WORD	0
005056	000000	.WORD	0
005060	000000	.WORD	0
005062	000000	.WORD	0
005064	000000	.WORD	0
005066	000000	.WORD	0
005070	000000	.WORD	0
005072	000000	.WORD	0
005074	000000	.WORD	0
005076	000000	.WORD	0
005100	000000	.WORD	0
005102	000000	.WORD	0
005104	000000	.WORD	0
005106	000000	.WORD	0
005110	000000	.WORD	0
005112	000000	.WORD	0
005114	000000	.WORD	0
005116	000000	.WORD	0
005120	000000	.WORD	0
005122	000000	.WORD	0
005124	000000	.WORD	0
005126	000000	.WORD	0
005130	000000	.WORD	0
005132	000000	.WORD	0
005134	000000	.WORD	0
005136	000000	.WORD	0
005140	000000	.WORD	0
005142	000000	.WORD	0
005144	000000	.WORD	0

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0407
Page 59
(11)

005146	000000		.WORD	0
005150	000000		.WORD	0
005152	000000		.WORD	0
005154	000000		.WORD	0
005156	000000		.WORD	0
005160	000000		.WORD	0
005162	000000		.WORD	0
005164	044552		.WORD	44552
005166	000000		.WORD	0
005170	001144	DM.27::	.WORD	1144
005172	000000		.WORD	0
005174	000000		.WORD	0
005176	000000		.WORD	0
005200	042524		.WORD	42524
005202	052123		.WORD	52123
005204	033462		.WORD	33462
005206	000000		.WORD	0
005210	126411		.WORD	-51367
005212	000000		.WORD	0
005214	000000		.WORD	0
005216	000000		.WORD	0
005220	000000		.WORD	0
005222	000000		.WORD	0
005224	000000		.WORD	0
005226	000000		.WORD	0
005230	104206		.WORD	-73572
005232	003377		.WORD	3377
005234	022750		.WORD	22750
005236	022756		.WORD	22756
005240	022776		.WORD	22776
005242	023040		.WORD	23040
005244	023305		.WORD	23305
005246	060010		.WORD	60010
005250	104207		.WORD	-73571
005252	003320		.WORD	3320
005254	104201		.WORD	-73577
005256	000001		.WORD	1
005260	060023		.WORD	60023
005262	000000		.WORD	0
005264	114000		.WORD	-64000
005266	003321		.WORD	3321
005270	104200		.WORD	-73600
005272	000040		.WORD	40
005274	003322		.WORD	3322
005276	102200		.WORD	-75600
005300	000001		.WORD	1
005302	003320		.WORD	3320
005304	012775		.WORD	12775
005306	104200		.WORD	-73600
005310	000100		.WORD	100
005312	003321		.WORD	3321
005314	104200		.WORD	-73600
005316	000140		.WORD	140
005320	003322		.WORD	3322
005322	000000		.WORD	0
005324	114000		.WORD	-64000
005326	003330		.WORD	3330

ZRCFB4
V03.0CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0408
Page 60
(11)

005330	114000	.WORD	-64000
005332	003332	.WORD	3332
005334	104300	.WORD	-73500
005336	003321	.WORD	3321
005340	003336	.WORD	3336
005342	023170	.WORD	23170
005344	023125	.WORD	23125
005346	102200	.WORD	-75600
005350	000001	.WORD	1
005352	003313	.WORD	3313
005354	053021	.WORD	53021
005356	115400	.WORD	-62400
005360	003330	.WORD	3330
005362	106200	.WORD	-71600
005364	001000	.WORD	1000
005366	003330	.WORD	3330
005370	033006	.WORD	33006
005372	023225	.WORD	23225
005374	115000	.WORD	-63000
005376	003316	.WORD	3316
005400	053030	.WORD	53030
005402	115000	.WORD	-63000
005404	003317	.WORD	3317
005406	013037	.WORD	13037
005410	115400	.WORD	-62400
005412	003332	.WORD	3332
005414	106200	.WORD	-71600
005416	001000	.WORD	1000
005420	003332	.WORD	3332
005422	033002	.WORD	33002
005424	003300	.WORD	3300
005426	000000	.WORD	0
005430	023170	.WORD	23170
005432	114000	.WORD	-64000
005434	003324	.WORD	3324
005436	114000	.WORD	-64000
005440	003325	.WORD	3325
005442	115400	.WORD	-62400
005444	003324	.WORD	3324
005446	107200	.WORD	-70600
005450	000001	.WORD	1
005452	003325	.WORD	3325
005454	104300	.WORD	-73500
005456	003324	.WORD	3324
005460	003326	.WORD	3326
005462	023100	.WORD	23100
005464	023225	.WORD	23225
005466	115000	.WORD	-63000
005470	003316	.WORD	3316
005472	053077	.WORD	53077
005474	023170	.WORD	23170
005476	104300	.WORD	-73500
005500	003325	.WORD	3325
005502	003326	.WORD	3326
005504	023100	.WORD	23100
005506	023225	.WORD	23225
005510	115000	.WORD	-63000

H16

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0409
Page 61
(11)

005512	003316	.WORD	3316
005514	053077	.WORD	53077
005516	104300	.WORD	-73500
005520	003324	.WORD	3324
005522	003334	.WORD	3334
005524	003045	.WORD	3045
005526	000000	.WORD	0
005530	114000	.WORD	-64000
005532	003327	.WORD	3327
005534	104307	.WORD	-73471
005536	003320	.WORD	3320
005540	104301	.WORD	-73477
005542	003326	.WORD	3326
005544	104202	.WORD	-73576
005546	000174	.WORD	174
005550	060013	.WORD	60013
005552	103207	.WORD	-74571
005554	177740	.WORD	-40
005556	115007	.WORD	-62771
005560	013124	.WORD	13124
005562	115400	.WORD	-62400
005564	003327	.WORD	3327
005566	106200	.WORD	-71600
005570	000012	.WORD	12
005572	003327	.WORD	3327
005574	033102	.WORD	33102
005576	003300	.WORD	3300
005600	000000	.WORD	0
005602	104307	.WORD	-73471
005604	003320	.WORD	3320
005606	060014	.WORD	60014
005610	103207	.WORD	-74571
005612	177740	.WORD	-40
005614	115007	.WORD	-62771
005616	053161	.WORD	53161
005620	104110	.WORD	-73670
005622	003313	.WORD	3313
005624	106200	.WORD	-71600
005626	000440	.WORD	440
005630	003313	.WORD	3313
005632	033143	.WORD	33143
005634	003161	.WORD	3161
005636	115401	.WORD	-62377
005640	104110	.WORD	-73670
005642	003314	.WORD	3314
005644	104117	.WORD	-73661
005646	103207	.WORD	-74571
005650	144000	.WORD	-34000
005652	053161	.WORD	53161
005654	104117	.WORD	-73661
005656	103207	.WORD	-74571
005660	104000	.WORD	-74000
005662	106207	.WORD	-71571
005664	040000	.WORD	40000
005666	053161	.WORD	53161
005670	000000	.WORD	0
005672	115400	.WORD	-62400

ZRCFB4
V03.0CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST27-Mar-1985 15:33:05
11-Jan-1985 08:19:20VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3SEQ 0410
Page 62
(11)

005674	003331	.WORD	3331
005676	106200	.WORD	-71600
005700	000144	.WORD	144
005702	003331	.WORD	3331
005704	033125	.WORD	33125
005706	003300	.WORD	3300
005710	100467	.WORD	-77311
005712	114000	.WORD	-64000
005714	003315	.WORD	3315
005716	104200	.WORD	-73600
005720	001475	.WORD	1475
005722	003335	.WORD	3335
005724	104307	.WORD	-73471
005726	003320	.WORD	3320
005730	104301	.WORD	-73477
005732	003336	.WORD	3336
005734	104302	.WORD	-73476
005736	003335	.WORD	3335
005740	060015	.WORD	60015
005742	103207	.WORD	-74571
005744	177740	.WORD	-40
005746	106020	.WORD	-71760
005750	003335	.WORD	3335
005752	053216	.WORD	53216
005754	115007	.WORD	-62771
005756	053216	.WORD	53216
005760	104267	.WORD	-73511
005762	000000	.WORD	0
005764	115400	.WORD	-62400
005766	003315	.WORD	3315
005770	106200	.WORD	-71600
005772	000012	.WORD	12
005774	003315	.WORD	3315
005776	033176	.WORD	33176
006000	003300	.WORD	3300
006002	100467	.WORD	-77311
006004	114000	.WORD	-64000
006006	003316	.WORD	3316
006010	114000	.WORD	-64000
006012	003317	.WORD	3317
006014	104200	.WORD	-73600
006016	100000	.WORD	-100000
006020	003400	.WORD	3400
006022	104200	.WORD	-73600
006024	003414	.WORD	3414
006026	003403	.WORD	3403
006030	104300	.WORD	-73500
006032	003313	.WORD	3313
006034	003404	.WORD	3404
006036	104300	.WORD	-73500
006040	003314	.WORD	3314
006042	003405	.WORD	3405
006044	104300	.WORD	-73500
006046	003313	.WORD	3313
006050	003412	.WORD	3412
006052	104300	.WORD	-73500
006054	003314	.WORD	3314

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0411
Page 63
(11)

006056	003413	.WORD	3413
006060	104307	.WORD	-73471
006062	003320	.WORD	3320
006064	104201	.WORD	-73577
006066	003400	.WORD	3400
006070	060002	.WORD	60002
006072	103207	.WORD	-74571
006074	177740	.WORD	-40
006076	115007	.WORD	-62771
006100	013267	.WORD	13267
006102	115400	.WORD	-62400
006104	003316	.WORD	3316
006106	104307	.WORD	-73471
006110	003400	.WORD	3400
006112	102207	.WORD	-75571
006114	020000	.WORD	20000
006116	013276	.WORD	13276
006120	115400	.WORD	-62400
006122	003317	.WORD	3317
006124	104267	.WORD	-73511
006126	000000	.WORD	0
006130	104200	.WORD	-73600
006132	000106	.WORD	106
006134	003333	.WORD	3333
006136	023305	.WORD	23305
006140	060010	.WORD	60010
006142	104207	.WORD	-73571
006144	003333	.WORD	3333
006146	104201	.WORD	-73577
006150	000002	.WORD	2
006152	060022	.WORD	60022
006154	000000	.WORD	0
006156	000000	.WORD	0
006160	140000	.WORD	-40000
006162	000000	.WORD	0
006164	000000	.WORD	0
006166	000000	.WORD	0
006170	000000	.WORD	0
006172	000000	.WORD	0
006174	000000	.WORD	0
006176	000000	.WORD	0
006200	000000	.WORD	0
006202	000000	.WORD	0
006204	000000	.WORD	0
006206	000000	.WORD	0
006210	000000	.WORD	0
006212	000000	.WORD	0
006214	000000	.WORD	0
006216	000104	.WORD	104
006220	000000	.WORD	0
006222	000000	.WORD	0
006224	000000	.WORD	0
006226	000000	.WORD	0
006230	000000	.WORD	0
006232	000000	.WORD	0
006234	000000	.WORD	0
006236	000000	.WORD	0

ZRCFR4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0412
Page 64
(11)

006240	000000	.WORD	0
006242	000000	.WORD	0
006244	000000	.WORD	0
006246	000000	.WORD	0
006250	000000	.WORD	0
006252	000000	.WORD	0
006254	000000	.WORD	0
006256	000000	.WORD	0
006260	000000	.WORD	0
006262	000000	.WORD	0
006264	000000	.WORD	0
006266	000000	.WORD	0
006270	000000	.WORD	0
006272	000000	.WORD	0
006274	000000	.WORD	0
006276	000000	.WORD	0
006300	000000	.WORD	0
006302	000000	.WORD	0
006304	000000	.WORD	0
006306	000000	.WORD	0
006310	000000	.WORD	0
006312	000000	.WORD	0
006314	000000	.WORD	0
006316	000000	.WORD	0
006320	000000	.WORD	0
006322	000000	.WORD	0
006324	000000	.WORD	0
006326	000000	.WORD	0
006330	000000	.WORD	0
006332	120475	.WORD	-57303
006334	000000	.WORD	0

PSECT SUMMARY

```

:
:
: Psect Name      Words  Attributes
: DM$CODE        1647   RO , D , GBL, REL, CON
:

```

Library Statistics

```

:
:
: File           Total  Symbols  Percent  Pages  Processing
:                Total  Loaded   Percent  Mapped  Time
:
: USER#1:[AZTEC.CZRCFC]AZTECO.L16;2  485    4        0       24    00:00.2
:

```

COMMAND QUALIFIERS

L16

ZRCFB4
V03.0

CZRCFC0 RC25 FR END TEST
OFFSET TOLERANCE TEST

27-Mar-1985 15:33:05
11-Jan-1985 08:19:20

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC4.B16;3

SEQ 0413
Page 65
(11)

: BLISS/PDP11/LIST ZRCFC4.B16/EN:NOEIS

: Size: 0 code + 1647 data words
: Run Time: 03:08.2
: Elapsed Time: 03:20.4
: Lines/CPU Min: 1026
: Lexemes/CPU-Min: 20841
: Memory Used: 442 pages
: Compilation Complete

M16

ZRCF85

27-Mar-1985 15:36:31
11-Jan-1985 08:19:21

VAX-11 Bliss-16 V4.0-579
USER#1:(AZTEC.CZRCFC)ZRCFC5.B16;1

SEQ 0414
Page 1
(1)

: 0001 0 MODULE ZRCF85 =
: 0002 1 BEGIN

ZRCF85

LASTAD AND SETUP

27-Mar-1985 15:36:31
11-Jan-1985 08:19:21

VAX-11 B100-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC5.B16;1

```

: 0003 1 *TITLE 'LASTAD AND SETUP'
: 0004 1 !
: 0005 1 !
: 0006 1 !
: 0007 1 REQUIRE 'BLSMAC.REQ';
: 1496 1
: 1497 1 LIBRARY 'AZTECO';
: 1498 1
: 1499 1 *SBTTL 'LAST ADDRESS AND SETUP SECTION'
: 1500 2 LASTAD;
: 1501 2 BGNSETUP (0);
: 1502 1 ENDSETUP;

```

.TITLE ZRCF85 LASTAD AND SETUP

```

000000 .PSECT $XYZ$, RO
000000 000004' BL$LAS::.WORD T$FREE
000002 000000C .WORD <<T$FREE-<BL$LAS*4>>/2>
000004 000000 T$FREE::.WORD 0

```

```

000004' L$LAST== BL$LAS*4
000000 T$PTHV== 0

```

```

000000 000207 .SBTTL $END.LINK LAST ADDRESS AND SETUP SECTION
$END.LINK::
RTS PC ;

```

1497

```

: Routine Size: 1 word, Routine Base: $XYZ$ * 0006
: Maximum stack depth per invocation: 0 words

```

```

: 1503 1 END
: 1504 1
: 1505 0 ELUDOM

```

PSECT SUMMARY

```

: Psect Name Words Attributes
: $XYZ$ 4 RO , I , LCL, REL, CON

```

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
USER#1:[AZTEC.CZRCFC]AZTECO.L16;2	485	0	0	24	00:00.3

C1

ZRCF85

LASTAD AND SETUP
LAST ADDRESS AND SETUP SECTION

27-Mar-1985 15:36:31
11-Jan-1985 08:19:21

VAX-11 Bliss-16 V4.0-579
USER#1:[AZTEC.CZRCFC]ZRCFC5.B16;1

SEQ 0416
Page 3
(2)

COMMAND QUALIFIERS

:
: BLISS/PDP11/LIST ZRCFC5.B16/EN:NOEIS

: Size: 1 code * 3 data words
: Run time: 00:19.0
: Elapsed Time: 00:20.3
: Lines/CPU Min: 4752
: Lexemes/CPU-Min: 24031
: Memory Used: 95 pages
: Compilation Complete