RC25

RC25 FR END TEST
CZRCFA0

AH-T271A-MC
FICHE 1 OF 2

d03t03

RC25

RC25 FR END TEST
CZRCFA0

AH-T271A-MC
FICHE 2 OF 2

```
MODULE AZTECO ( %TITLE'CZRCFA0 RC25 FR END TEST'
                IDENT = 'V01.0',
                ADDRESSING_MODE (RELATIVE))=
BEGIN
LIBRARY 'library';
REQUIRE 'BLSMAC.REQ';
%SBTTL  'USER DOCUMENTATION'
%(
```

IDENTIFICATION

PRODUCT CODE: AC-T270A-MC

PRODUCT NAME: CZRCFA0 RC25 FR END TEST

PRODUCT DATE: JULY 13, 1983

MAINTAINER: DISK ENGINEERING

AUTHOR: SING LAKSHMANAN

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

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 - 12 **base level**

module 4 - tests 9 _ 12 (dm code)

module 5 - last address and setup section

AZTECO.R16 is a file containing literals and field delarations 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 HIERARCY 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 rurtime 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 "DDDDD".

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

Example of switch usage:

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

The effect of this command will be:

1. Tests 1 through 5 will be executed.
2. All units will tested 1000 times.
3. The end of pass messages will be printed after each 100 passes only.

A Switch can be reccgnized 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 wiith 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 ?

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 followng 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.

Do you wish to limit the area tested in tests #13 through #15 (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 #13 through #15. The following two questions will be asked only if this one is answered yes.

Starting track (D) 0 ?

Answer with the beginning track number of the area you wish to select for testing. This applies to tests #13 through #15 only.

Ending track (d) 799 ?

Answer with the last track number in the area you wish to select for testing. This applies to tests #13 through #15 only.

Do you want 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 need 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 (O)  ?  160000<CR>
SUB-DEVICE # (O) ?  0<CR>
Q-FACTOR (O) 0   ?  1<CR>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.7 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
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

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 WAITRV 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
```

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 INITALZATION 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 (O)

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 (O)

xx 0 means UNIBUS/QBUS storage systems port.

3) At step 4 read the following will be given:
MICRO CODE: MODEL = xx (O) VERSION = yy (O)

xx = 0 UDA50
1 RC25 Integrated Controller
5 TU81 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, therby 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 communciations area packet and buffer space (a 5 in message length and a 5 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 transfered data.

If the transfered 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.

7.0 MAINTENANCE HISTORY

Modified By: Date: Version:

****
This is a base level release with tests 1 thru 12. A complete diagnostic with all 29 tests will be released in the following release cycle.
****

)%
ELUDOM

```
MODULE ZRCFA1 (%TITLE 'CZRCFA0 RC25 FR END TEST'
                IDENT = 'V01.0',
                ADDRESSING_MODE (RELATIVE)
                ) =
BEGIN
!
!<BLF/LOWERCASE_KEY>
!

library 'AZTEC0';                              ! AZTEC LIBRARY

require 'BLSMAC.REQ';                          ! DIAGNOSTIC SUPERVISOR LIBRARY

%sbttl 'PROGRAM HEADER AND TABLES'
!
! DEFINE THE NUMBER OF TESTS IN THIS DIAGNOSTIC
!
psect
    code = AA$CODE;

literal
    DS$NBR_OF_TESTS = 12;

POINTER (ALL);

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

HEADER (%ascii'CZRCF ', %ascii'A', %ascii'0', 120, 0, PRI00);
!;      ARGUMENTS ARE: NAME,REV,PATCH,LONGEST TEST TIME,TYPE
!;      WHERE "TYPE" = 0 FOR SEQUENTIAL DIAGNOSTIC AND =1
!;      FOR EXERCISER.  THERE IS ALSO AN OPTIONAL SIXTH ARGUMENT
!;      WHICH SPECIFIES THE PROCESSOR PRIORITY TO BE SET WHEN
!;      STARTING THE DIAGNOSTIC (DEFAULT IS 0).
```

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

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

```
;       1552  %sbttl 'SOFTWARE P-TABLE'
;       1553
;       1554  !++
;       1555  ! THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
;       1556  ! PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
;       1557  ! SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
;       1558  ! AT RUN TIME.
;       1559  !--
;       1560
;       1561  BGNSW (SFPTBL);
;       1562
;       1563  global
;       1564      SWP_TOP : word initial (YES),          !USE TOP SURFACE FOR SINGLE SURFACE TESTS
;       1565      SWP_LIMIT : word initial (NO),         !LIMIT AREA TESTED
;       1566      SWP_START : word initial (1),          !STARTING TRACK
;       1567      SWP_END : word initial (796),          !ENDING TRACK
;       1568      SWP_RETRIES : word initial (0),        !NUMBER OF RETRIES BEFORE DROPPING UNIT
;       1569      SWP_CONTINUE : word initial (NO),      !DO YOU NEED TO CONTINUE TESTING?
;       1570      SWP_MANUAL : word initial (NO),        !DO MANUAL INTERVENTION TEST
;       1571      SWP_TRACE : word initial (YES);        !DO YOU NEED TRACE MODE?
;       1572
;       1573  ENDSW;
```

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

```
%sbttl 'GLOBAL DATA SECTION'

!++
! THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
! IN MORE THAN ONE TEST.
!--

psect
    plit = $plit$( global),
    global = $GLOB$(nowrite, noexecute, global, concatenate),
    own = $own$;

structure                                   ! DEFINE ACCESS ALGORITHM
    RC25 [O, P, S, E] =                     ! TO ALLOW FIELD REFERANCES
        begin                               ! TO THE AZTEC

        local
            RC_REG;

        RC_REG = .(RC25 + %upval*O)<0, %bpval, 0>;
        RC_REG
        end
        <P, S, E>;

global
    RT : vector [WORD1_IN_RT_TAB, word],    !RUNTIME TABLE STORAGE
    RT_TABLE : ref block [WORD1_IN_RT_TAB, word] field (RT_FIELDS),     !RUNTIME TABLE POINTER
    HWP_TABLE : ref block [WORD2_IN_HWP_TAB, word] field (HWP_FIELDS),
    XMT_DATA_BUF : vector [256, word],      ! TRANSMITTING  DATA BUFFER 1
    RCV_DATA_BUF : vector [256, word],      ! RECEIVING DATA BUFFER 2
    CLK_ADR : word,                         ! LOC. TO RETURN CLOCK ADDR.
    CLK_TYPE : word,                        ! TYPE OF CLOCK ON SYSTEM
    CLK_CSR : word,                         ! STORE CSR ADDRESS FOR CLOCK HERE
    CLK_HERTZ : word,                       ! STORE CLOCK HERTZ RATE
    CLK_START : word,                       ! STORE CLOCK START VALUE
    UNIT : word,                            ! UNIT UNDER TEST THIS PASS
    LOG_UNIT : word,
    VEC_AD : byte volatile,                 ! VECTOR ADDRESS OF AZTEC
    RC25_ADDR : ref RC25 field (RC_REG),    ! DEFINE REFERANCE TO AZTEC FIELDS
    RC25_DATA : block [2, word] field (RC_REG),
    COM_AREA : blockvector [REC_ALLOCATE + SND_ALLOCATE + HDR_SIZ, 2, word],
    HEAD_AREA : ref block [4, word] field (HDR_FIELD),
    RECEIVE_RING : ref blockvector [REC_ALLOCATE, 2, word] field (DSC_FIELD),
    SEND_RING : ref blockvector [SND_ALLOCATE, 2, word] field (DSC_FIELD),
    REC_ENVELOPE : blockvector [REC_ALLOCATE, RB_SIZE + 2, word] field (ENV_FIELD),
    SND_ENVELOPE : blockvector [SND_ALLOCATE, SB_SIZE + 2, word] field (ENV_FIELD),
    BUF_DESCRPTR : word volatile,           ! BUFFER DESCRIPTOR AREA
    CMD_REF : word volatile,                ! COMMAND REFERENCE BUFFER
    BYTE_COUNT : word volatile,             ! BYTE COUNT BUFFER
    TICKS : word initial (1) volatile,      ! SOTRE THE NUMBERS OF CLOCK INTERRUPTED
    SECONDS : word initial (0) volatile,    ! STORE SECONDS
    MINUTES : word initial (0) volatile,    ! STORE MINUTES
    TIP : word,                             !STORAGE FOR NUMBER OF TEST IN PROGRESS
    DATA1 : word volatile,                  ! AZTEC STEP 1 WRITE DATA
    DATA2 : word volatile,                  ! AZTEC STEP 2 WRITE DATA
    DATA3 : word volatile,                  ! AZTEC STEP 3 WRITE DATA
    DATA4 : word volatile,                  ! AZTEC STEP 4 WRITE DATA
```

```
;       1643      I_AM_NEX : word initial (0) volatile,       ! INTERUPT FLAG
;       1644      MSGADR : word volatile,
;       1645      END_LBN : word initial (1593) volatile,     ! ENDING LBN TRACK
;       1646      P_MASK : byte volatile,
;       1647      B_MASK : byte volatile,
;       1648      MANU_SW : word volatile,
;       1649      SWITCH2 : word volatile,
;       1650      RET_UNIT_FLAG : word volatile,
;       1651      P1 : word volatile,
;       1652      P2 : word volatile,
;       1653      P3 : word volatile,
;       1654      P4 : word volatile,
;       1655      P5 : word volatile,
;       1656      P6 : word volatile,
;       1657      RET_STATUS : word volatile,                 ! SAVES VARIOUS RETURN STATUS
;       1658      CANCEL_TIMER : word volatile,               ! INIT SEQUENCE INTERRUPT
;       1659      CMD_SLOT : word volatile,                   ! COMMAND DESCRIPTOR SLOT
;       1660      RES_SLOT : word volatile,                   ! RECEIVE DESCRIPTOR SLOT
;       1661      LBN : word volatile,
;       1662      LBN_ST : word volatile,                     ! STARTING LOGICAL BLOCK #
;       1663      LBN_ED : word volatile,                     ! ENDING LOGICAL BLOCK #
;       1664      LBN_SZ : word volatile,                     ! INCREMENTING LBN SIZE
;       1665      FREE_MEM_ADDR,                              ! STARTING FREE MEMORY ADDR.
;       1666      MEM_SIZE : word volatile,                   ! FREE MEMORY SIZE
;       1667      H_SADD : word volatile,                     ! LOW-BYTE FREE MEMORY ADDR.
;       1668      H_EADD : word volatile,                     ! HIGH-BYTE FREE MEMORY ADDR.
;       1669      BUF_LENGTH : word volatile,                 ! BUFFER LENGTH
;       1670      NUM_RETRIES : word volatile,
;       1671      RETRIES : word initial (FALSE),
;       1672      FAL_CODE : word initial (1),                ! FAIL STATUS
;       1673      DMC_TEST : word,
;       1674      BYT_CNT : word,
;       1675      DM_REC : word,
;       1676      DM_XMT : word,
;       1677      TEMP : word volatile;
;       1678
```

```
;       1679  %sbttl 'GLOBAL TEXT SECTION'
;       1680
;       1681  !++
;       1682  ! THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
;       1683  ! MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
;       1684  ! MORE THAN ONE TEST.
;       1685  !--
;       1686
;       1687  global bind
;       1688      RINGBASE = COM_AREA [REC_BASE],
;       1689      TIME = plit (P4, P5),
;       1690      !
;       1691      ! FAILING FRU'S
;       1692      !
;       1693      FRU = uplit (%asciz'%AFAILING FRU = %T%D3%N'),
;       1694      ADAPTO = uplit (%asciz'ADAPTOR BOARD FOR UNIT #:'),
;       1695      CONTRO = uplit (%asciz'CONTROLER BOARD FOR UNIT #:'),
;       1696      DRIVE_ = uplit (%asciz'DRIVE BOARD FOR UNIT #:'),
;       1697      MECHAN = uplit (%asciz'MECHANIC SET FOR UNIT #:'),
;       1698      !
;       1699      ! HARDWARE AND SOFTWARE QUESTIONS
;       1700      !
;       1701      QST1 = uplit (%asciz'IP ADDRESS'),
;       1702      QST2 = uplit (%asciz'VECTOR'),
;       1703      QST3 = uplit (%asciz'BR LEVEL'),
;       1704      QST4 = uplit (%asciz'PLATTER ADDRESS(ES)'),
;       1705      QST6 = uplit (%asciz'USE TOP SURFACE FOR SINGLE SURFACE TESTS'),
;       1706      QST7 = uplit (%asciz'DO YOU WISH TO LIMIT AREA TESTED IN TESTS #13 THRU #15'),
;       1707      QST8 = uplit (%asciz'STARTING TRACK'),
;       1708      QST9 = uplit (%asciz'ENDING TRACK'),
;       1709      QST10 = uplit (%asciz'DO YOU WANT TO DO THE MANUAL INTERVENTION TEST?'),
;       1710      QS10_1 = uplit (%asciz'DO YOU NEED TRACE MODE?'),
;       1711      QS10_2 = uplit (%asciz'DO YOU WISH TO CONTINUE TESTING AFTER RETRIES?'),
;       1712      QST11 = uplit (%asciz'NUMBER OF RETRIES FOR TEST IF ERROR OCCURED'),
;       1713      QST12 = uplit (%asciz'UNIT STARTING TRACK #'),
;       1714      QST13 = uplit (%asciz'UNIT ENDING TRACK #'),
;       1715      QST14 = uplit (%asciz'TURN OFF WRITE PROTECT SWITCH AND DO <CR>'),
;       1716      QST15 = uplit (%asciz'TURN ON WRITE PROTECT SWITCH AND DO <CR>'),
;       1717
;       1718  !++
;       1719  !  THE FOLLOWING MESSAGES INCLUDE THE NAMES OF EACH ROUTINE, PLUS
;       1720  !  FORMAT STATEMENTS FOR PRINTING OUT OTHER INFORMATION.
;       1721  !--
;       1722
;       1723      DBM1 = uplit (%ASCIZ'%N%N%N%ATESTING UNIT#:%D3%A  IP_REGISTER:%O6%A  PLATTER#:%D3%N'),
;       1724      DBM2 = uplit (%asciz'%N%AREPORT'),
;       1725      DBM3 = uplit (%asciz'%N%AAUTO'),
;       1726      DBM4 = uplit (%asciz'%N%ACLEANUP'),
;       1727      DBM5 = uplit (%asciz'%N%ADROPPED'),
;       1728      DBM6 = uplit (%asciz'%N%AADDED'),
;       1729      DBM7 = uplit (%asciz'%N%ATEST  1 REGISTER EXISTENCE TEST'),
;       1730      DBM8 = uplit (%asciz'%N%ATEST  2 STEP 1 READ/WRITE POWERUP DIAGNOSTICS'),
;       1731      DBM9 = uplit (%asciz'%N%ATEST  5 STEP 1 THROUGH STEP 3 READ/WRITE TEST'),
;       1732      DBM10 = uplit (%asciz'%N%ATEST  3 DIAGNOSTIC WRAP TEST'),
;       1733      DBM11 = uplit (%asciz'%N%ATEST  4 VECTOR AND BR LEVEL TEST'),
;       1734      DBM12 = uplit (%asciz'%N%ATEST  6 PURGE AND POLL TEST'),
;       1735      DBM13 = uplit (%asciz'%N%ATEST  7 SMALL RING TEST'),
```

```
;         1736      DBM14 = uplit (%asciz'%N%ATEST  8 LARGE RING TEST'),
;         1737      DBM15 = uplit (%asciz'%N%ATEST  9 DM CODE OVERLAY TEST'),
;         1738      DBM16 = uplit (%asciz'%N%ATEST 10 NONEXISTENT MEMORY TEST'),
;         1739      DBM17 = uplit (%asciz'%N%ATEST 11 BUS ADDRESSING/DATA TEST A'),
;         1740      DBM18 = uplit (%asciz'%N%ATEST 12 BUS ADDRESSING/DATA TEST B'),
;         1741      DBM19 = uplit (%asciz'%N%ATEST 13 BLOCK TRANSFER TEST'),
;         1742      DBM20 = uplit (%asciz'%N%ATEST 14 SPIN UP HEAD LOAD SEQUENCE'),
;         1743      DBM21 = uplit (%asciz'%N%ATEST 15 SEQUENTIAL SEEK AND VERIFY'),
;         1744      DBM22 = uplit (%asciz'%N%ATEST 16 SAWTOOTH SEEK AND VERIFY'),
;         1745      DBM23 = uplit (%asciz'%N%ATEST 17 CONVERGING/DIVERGING SEEK AND VERIFY'),
;         1746      DBM24 = uplit (%asciz'%N%ATEST 18 TOGGLE SEEK AND VERIFY'),
;         1747      DBM25 = uplit (%asciz'%N%ATEST 19 HEAD SWITCH TEST'),
;         1748      DBM26 = uplit (%asciz'%N%ATEST 20 RANDOM SEEK AND VERIFY'),
;         1749      DBM27 = uplit (%asciz'%N%ATEST 21 SECTOR ACCESS TEST'),
;         1750      DBM28 = uplit (%asciz'%N%ATEST 22 CONTROLLER PROCESSING TIME'),
;         1751      DBM29 = uplit (%asciz'%N%ATEST 23 ONE TRACK SEEK TIME'),
;         1752      DBM30 = uplit (%asciz'%N%ATEST 24 AVERAGE SEEK TIME'),
;         1753      DBM31 = uplit (%asciz'%N%ATEST 25 FULL STROKE SEEK TIME'),
;         1754      DBM32 = uplit (%asciz'%N%ATEST 26 WRITE DATA TEST'),
;         1755      DBM33 = uplit (%asciz'%N%AEVENT START'),
;         1756      DBM34 = uplit (%asciz'%N%AEVENT RESTART'),
;         1757      DBM35 = uplit (%asciz'%N%AEVENT CONTINUE'),
;         1758      DBM36 = uplit (%asciz'%N%ATEST 27 OFFSET TOLERANCE TEST'),
;         1759      DBM37 = uplit (%asciz'%N%ATEST 28 AVERAGE ROTATIONAL TIME'),
;         1760      DBM38 = uplit (%asciz'%N%ATEST 29 WRITE PROTECT TEST'),
;         1761      DBM39 = uplit (%asciz'%N%A          MANUAL INTERVENTION TEST NOT PERFORMED'),
;         1762      !
;         1763      ! SYSTEM ERROR MESSAGES
;         1764      !
;         1765      MSG_01 = uplit (%asciz'%N%APOWER DELAY - WAITING'),
;         1766      ERR_01 = uplit (%asciz'%N%ATOO MANY UNITS'),
;         1767      ERR_02 = uplit (%asciz'%N%ANO CLOCK WAS FOUND ON THE SYSTEM'),
;         1768      !
;         1769      ! FORMATTED ASCIC STRINGS
;         1770      !
;         1771      FMT$C = uplit (%asciz'%N%N'),
;         1772      FMT1 = uplit (%asciz'%N%A    REGISTER FAILED TO RESPOND AT ADDRESS:  %O6%N'),
;         1773      FMT2 = uplit (%asciz'%N%AADDRESS: %O6%A     EXPECTED: %O6%A READ: %O6%N'),
;         1774      FMT3 = uplit (%asciz'%N%ASTEP MASK = %O2%A  FAILING REGISTER = %O6%A DATA = %O6%N'),
;         1775      FMT4 = uplit (%asciz'%N%A    PORT TYPE NUMBER = %O2'),
;         1776      FMT5 = uplit (%asciz'%N%A    PORT SPECIFIC INFO:/NV/QB/DI/OD/MP/ = %O2'),
;         1777      FMT6 = uplit (%asciz'%N%A    MICRO CODE: MODEL = %O2%A   VERSION = %O2'),
;         1778      FMT$A = uplit (%asciz'%N%A   NUMBER OF RETRIES =%D4'),
;         1779      !
;         1780      ! INIT ERROR MESSAGES
;         1781      !
;         1782      MSG_PWR = uplit (%asciz'    WAIT - POWER FAIL RECOVERY'),
;         1783      MSG_1 = uplit (%asciz'RCSA FAILED TO RESPOND'),
;         1784      MSG_2 = uplit (%asciz'RCIP FAILED TO RESPOND'),
;         1785      MSG_7 = uplit (%asciz'TEST PATTERN ECHOED IN RCSA IS INCORRECT'),
;         1786      MSG_8 = uplit (%asciz'VECTOR AND BR LEVEL TEST FAILURE'),
;         1787      MSG_9 = uplit (%asciz'HOST DETECTED TIME OUT ERROR'),
;         1788      MSG_10 = uplit (%asciz'RING BUFFERS NOT CLEARED BY THE PORT'),
;         1789      MSG_11 = uplit (%asciz'STEP READ DATA DOES NOT MATCH'),
;         1790      MSG_13 = uplit (%asciz'PORT FATAL ERROR'),
;         1791      MSG_14 = uplit (%asciz'INIT STEP READ ERROR'),
;         1792      BUFF_ERR = uplit (%asciz'MEMORY BUFFER DOES NOT CONTAIN EXPECTED DATA'),
```

```
;       1793      DMC_ERR = uplit (%asciz'DM CODE RETURNED FAILURE CODE'),
;       1794      INI_MSG = uplit (%asciz'%N%A          INTERRUPT AT VEC= %O3%A BR LEVEL= %O1'),
;       1795      END_MSG = uplit (%asciz'%N%A          NO INTERRUPT FROM PORT / CONTROLLER'),
;       1796      BRERR = UPLIT (%ASCIZ'%N%A            BR LEVEL RECEIVED/TYPED IS INCORRECT !'),
;       1797      MSG_17 = uplit (%asciz'PURGE AND POLE TEST SET ERROR BIT 15'),
;       1798      MSG_18 = uplit (%asciz'PURGE AND POLE TEST DID NOT SET STEP 4 BIT 14'),
;       1799      MSG_19 = uplit (%asciz'INIT DID NO CLEAR RING BUFFER'),
;       1800      MSG_20 = uplit (%asciz'FAILED POLLING ERROR IN RESPONCE RING'),
;       1801      MSG_21 = uplit (%asciz'AVAILABLE COMMAND SPIN-DOWN FAILURE'),
;       1802      MSG_28 = uplit (%asciz'SPIN UP TEST FAILURE'),
;       1803      MSG_29 = uplit (%asciz'SEQUENTIAL FORWARD SEEK FAILURE'),
;       1804      MSG_30 = uplit (%asciz'SEQUENTIAL REVERSE SEEK FAILURE'),
;       1805      CTO_ERR = uplit (%asciz'%N%ATIME EXPIRED'),
;       1806      PFE_ERR = uplit (%asciz'%N%AFATAL ERROR'),
;       1807      AHEAD_MSG = uplit (%asciz'AHEAD A OFFSET VALUE = %O3'),
;       1808      BHEAD_MSG = uplit (%asciz'AHEAD B OFFSET VALUE = %O3'),
;       1809      CHEAD_MSG = uplit (%asciz'AHEAD C OFFSET VALUE = %O3'),
;       1810      DHEAD_MSG = uplit (%asciz'AHEAD D OFFSET VALUE = %O3'),
;       1811      MSG_TK_DSP = uplit (%asciz'N%ACURRENT TRACK = %O4%A NUMBER OF SEEKS = %O5'),
;       1812      MSG_LBN_DSP = uplit (%asciz'%N%ASTARTING TRACK = %O4%A CURRENT TRACK = %O4%A ENDING TRACK = %O4'),
;       1813      MSG_STATUS_ERR = uplit (%asciz'%N%AEND PACKET STATUS ERROR = %O6%A REF # = %O2'),
;       1814      MSG_BUSA_ERR = uplit (%asciz'%N%ABUS ADDRESSING DATA TEST ERROR'),
;       1815      MSG_ADDR_ERR = uplit (%asciz'%N%AFAILING ADDR = %O6%A DATA = %O6%N'),
;       1816      MSG_DATA_ERR = uplit (%asciz'%N%ABLOCK DATA TRANSFER FAILED'),
;       1817      MSG_SEEK_ERR = uplit (%asciz'RC25 SEEK FAILURE'),
;       1818      MSG_ERR_CONT = uplit (%asciz'%N%ABLOCK LENGTH = %O6%N'),
;       1819      MSG_HSWICH_ERR = uplit (%asciz'%N%AHEAD SWITCH FAILED'),
;       1820      MSG_SURFACE_ERR = uplit (%asciz'FAILING SURFACE = %O3%A TRACK # = %O6%N'),
;       1821      MSG_READ_ERR = uplit (%asciz'READ SECTOR FAILED'),
;       1822      MSG_SAC_ERR = uplit (%asciz'AFAILING  TRACK # = %O6%A SECTOR # = %O6%N'),
;       1823      MSG_COM_WPT = uplit (%asciz'WRITE PROTECT TEST FAILED'),
;       1824      MSG_PT_ERR1 = uplit (%asciz'EXPECTED SW = OFF  ACTUAL SW = ON  UNIT # = %D3%N'),
;       1825      MSG_WRP_ERR2 = uplit (%asciz'AEXPECTED SW = ON  ACTUAL SW = OFF  UNIT # = %D3%N'),
;       1826      MSG_AVE_TIME = uplit (%asciz'%N%AAVERAGE SEEK TIME (ms) = %O2%A.%O2'),
;       1827      AZT_READY_ERR = uplit (%asciz'RC25 UNIT DOES NOT COME ONLINE'),
;       1828      EXE_SUP_ERR = uplit (%asciz'EX_SUP_PROG DUP COMMAND FAILURE'),
;       1829      SND_DATA_ERR = uplit (%asciz'SEND_DATA DUP COMMAND FAILURE'),
;       1830      RE_DATA_ERR = uplit (%asciz'REC_DATA DUP COMMAND FAILURE'),
;       1831  !<BLF/PAGE>
```

```
;          1832
;          1833      !
;          1834      ! Self-detected fatal port/controller errors
;          1835      !
;          1836  PFE_STRUCT = uplit (
;          1837                       uplit (%asciz'%N%A$FTLERR- UNRECOGNIZABLE ERROR CODE'),
;          1838                       uplit (%asciz'%N%A$FTLERR- ENVELOPE/PACKET READ (PARITY OR TIMEOUT)'),
;          1839                       uplit (%asciz'%N%A$FTLERR- ENVELOPE/PACKET WRITE (PARITY OR TIMEOUT)'),
;          1840                       uplit (%asciz'%N%A$FTLERR- CONTROLLER ROM AND RAM PARITY'),
;          1841                       uplit (%asciz'%N%A$FTLERR- CONTROLLER RAM PARITY'),
;          1842                       uplit (%asciz'%N%A$FTLERR- CONTROLLER ROM PARITY'),
;          1843                       uplit (%asciz'%N%A$FTLERR- RING READ (PARITY OR TIMEOUT)'),
;          1844                       uplit (%asciz'%N%A$FTLERR- RING WRITE (PARITY OR TIMEOUT)'),
;          1845                       uplit (%asciz'%N%A$FTLERR- INTERRUPT MASTER'),
;          1846                       uplit (%asciz'%N%A$FTLERR- HOST ACCESS TIMEOUT'),
;          1847                       uplit (%asciz'%N%A$FTLERR- CREDIT LIMIT EXCEEDED'),
;          1848                       uplit (%asciz'%N%A$FTLERR- BUS MASTER ERROR'),
;          1849                       uplit (%asciz'%N%A$FTLERR- DIAGNOSTIC CONTROLLER FATAL ERROR'),
;          1850                       uplit (%asciz'%N%A$FTLERR- INSTRUCTION LOOP TIMEOUT'),
;          1851                       uplit (%asciz'%N%A$FTLERR- INVALID CONNECTION IDENTIFIER'),
;          1852                       uplit (%asciz'%N%A$FTLERR- INTERRUPT WRITE'),
;          1853                       uplit (%asciz'%N%A$FTLERR- MAINTENANCE READ/WRITE INVALID REGION IDENTIFIER'),
;          1854                       uplit (%asciz'%N%A$FTLERR- MAINTENANCE WRITE LOAD TO NON-LOADABLE CONTROLLER'),
;          1855                       uplit (%asciz'%N%A$FTLERR- CONTROLLER RAM ERROR (NON-PARITY)'),
;          1856                       uplit (%asciz'%N%A$FTLERR- INIT SEQUENCE ERROR'),
;          1857                       uplit (%asciz'%N%A$FTLERR- HIGH LEVEL PROTOCOL INCOMPATIBILITY ERROR'),
;          1858                       uplit (%asciz'%N%A$FTLERR- PURGE/POLL HARDWARE FAILURE '),
;          1859                       uplit (%asciz'%N%A$FTLERR- MAPPING REGISTER READ ERROR (PARITY OR TIMEOUT)')
;          1860                                                                                ) : vector [23],
;          1861  !<BLF/PAGE>
```

```
;          1862      !
;          1863      ! Error message structure
;          1864      !
;          1865      EMSG_STRUCT = uplit (
;          1866  uplit (%asciz'%N%A$FTLERR- RESPONSE STATUS ERROR:%S'),
;          1867  uplit (%asciz'%N%A$FTLERR- SUPERVISOR SERVICE CALL FAILED'),
;          1868  uplit (%asciz'%N%A$FTLERR- PORT/CONTROLLER TIMEOUT ERROR'),
;          1869  uplit (%asciz'%N%A$FTLERR- UNKNOWN RETURN STATUS CODE')) : vector [4],
;          1870  !<blf/page>
```

```
;       1871      !
;       1872      ! Self-detected fatal port/controller errors
;       1873      !
;       1874      RC_STRUCTURE = uplit (
;       1875  uplit (%asciz'%N%A$FTLERR- VAX READ/WRITE ERROR ON INTERRUPT'),
;       1876  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.BFIL'),
;       1877  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.BMTY'),
;       1878  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.ALOC'),
;       1879  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT SERVO ENTRY (PIP SET)'),
;       1880  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT SERVO ENTRY (ERR SET)'),
;       1881  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.SEND'),
;       1882  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.RECV'),
;       1883  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.ATTN'),
;       1884  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.ONLN'),
;       1885  uplit (%asciz'%N%A$FTLERR- ILLEGAL D REQUEST (U.QDRQ)'),
;       1886  uplit (%asciz'%N%A$FTLERR- FENCE-POST ERROR AT PROTAB'),
;       1887  uplit (%asciz'%N%A$FTLERR- BAD PACKET DEQUEUED AT U.DONE'),
;       1888  uplit (%asciz'%N%A$FTLERR- UNEXPLAINED D-PROC SUSPENSION (U..TDS)'),
;       1889  uplit (%asciz'%N%A$FTLERR- DUP PACKET D-Q FAILED (XFC 34/35)'),
;       1890  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.HTST'),
;       1891  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.SEKO'),
;       1892  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT U.CKSV'),
;       1893  uplit (%asciz'%N%A$FTLERR- D.OPCD FOUND ILLEGAL OPCODE'),
;       1894  uplit (%asciz'%N%A$FTLERR- D.CSF FOUND ILLEGAL OPCODE'),
;       1895  uplit (%asciz'%N%A$FTLERR- UNKNOWN BAD DRIVE STATUS AT D.DSTS'),
;       1896  uplit (%asciz'%N%A$FTLERR- ILLEGAL XFC EXECUTED BY DM'),
;       1897  uplit (%asciz'%N%A$FTLERR- D PICKED UP A ZERO SCB.DB'),
;       1898  uplit (%asciz'%N%A$FTLERR- INCONSISTENCY AT D IDLE LOOP'),
;       1899  uplit (%asciz'%N%A$FTLERR- DM WORD COUNT ERROR ON HOST DMA/SEND/RECV'),
;       1900  uplit (%asciz'%N%A$FTLERR- UNKNOWN DISPLAY FAULT CODE AT D.DFLT'),
;       1901  uplit (%asciz'%N%A$FTLERR- DRIVE NOT FAULTING IN P.OFLN STATE'),
;       1902  uplit (%asciz'%N%A$FTLERR- U POWER UP DIAGNOSTICS FAILED'),
;       1903  uplit (%asciz'%N%A$FTLERR- D POWER UP DIAGNOSTICS FAILED'),
;       1904  uplit (%asciz'%N%A$FTLERR- ADAPTER CARD FAILURE'),
;       1905  uplit (%asciz'%N%A$FTLERR- EC.TMR TIMED OUT'),
;       1906  uplit (%asciz'%N%A$FTLERR- U.SEND/U.RECV RING READ INCONSISTENCY'),
;       1907  uplit (%asciz'%N%A$FTLERR- UNKNOWN WAITRV REASON AT D.RVCT'),
;       1908  uplit (%asciz'%N%A$FTLERR- D.ARCS DID NOT FIND CLOSEST UNDONE ZONE'),
;       1909  uplit (%asciz'%N%A$FTLERR- U.SEEK FOUND SEEK TO ILLEGAL TRACK'),
;       1910  uplit (%asciz'%N%A$FTLERR- U.HTST INIT DIAG DMA WRITE FAILED'),
;       1911  uplit (%asciz'%N%A$FTLERR- U.HTST INIT DIAG DMA COMPARE FAILED'),
;       1912  uplit (%asciz'%N%A$FTLERR- U.SYDR FOUND SS.DER SET AND SS.SPN NOT SET'),
;       1913  uplit (%asciz'%N%A$FTLERR- MASTER DRIVES ACLO ASSERTED')
;       1914                          ) : vector [39],
;       1915  !<blf/page>
```

```
;            !
;            ! Dup return status codes
;            !
;            SDUP_STRUCT = uplit (
;     uplit (%asciz'%A SUCCESSFUL%N'),
;     uplit (%asciz'%AINVALID COMMAND%N'),
;     uplit (%asciz'%ANO REGION AVAILABLE%N'),
;     uplit (%asciz'%ANO REGION SUITABLE%N'),
;     uplit (%asciz'%APROGRAM NOT KNOWN%N'),
;     uplit (%asciz'%ALOAD FAILURE%N'),
;     uplit (%asciz'%ASTANDALONE%N')
;                ) : vector [7],
;     !<blf/page>
```

```
        !
        ! MSCP return status codes
        !
        SMSCP_STRUCT = uplit (
uplit (%asciz'%ASUCCESS%N'),
uplit (%asciz'%AINVALID COMMAND%N'),
uplit (%asciz'%ACOMMAND ABORTED%N'),
uplit (%asciz'%AUNIT-OFFLINE%N'),
uplit (%asciz'%AUNIT-AVAILABLE%N'),
uplit (%asciz'%AMEDIA FORMAT ERROR%N'),
uplit (%asciz'%AWRITE PROTECTED%N'),
uplit (%asciz'%ACOMPARE ERROR%N'),
uplit (%asciz'%ADATA ERROR%N'),
uplit (%asciz'%AHOST BUFFER ACCESS ERROR%N'),
uplit (%asciz'%ACONTROLLER ERROR%N'),
uplit (%asciz'%ADRIVE ERROR%N'),
uplit (%asciz'%AMESSAGE FROM AN INTERNAL DIAGNOSTIC%N')
        ) : vector [13];

end

eludom


                                          .TITLE  ZRCFA1 CZRCFA0 RC25 FR END TEST
                                          .IDENT  /V01.0/


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$REV::
000010     101                          .ASCII  /A/
000011     060                          .ASCII  /0/
000012  000000G                  L$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  000166'                  L$HPTP::.WORD   L$HW
000024  000202'                  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
```

```
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  000154'                         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  000224'                         L$PRT:: .WORD   L$PROT
000114  000000                          L$TEST::.WORD   0
000116  000000                          L$DLY:: .WORD   0
000120  000000                          L$HIME::.WORD   0
000122  000014                          D$PCNT::.WORD   14
000124  000000G                         L$DISPATCH::
                                                .WORD   T1
000126  000000G                                 .WORD   T2
000130  000000G                                 .WORD   T3
000132  000000G                                 .WORD   T4
000134  000000G                                 .WORD   T5
000136  000000G                                 .WORD   T6
000140  000000G                                 .WORD   T7
000142  000000G                                 .WORD   T8
000144  000000G                                 .WORD   T9
000146  000000G                                 .WORD   T10
000150  000000G                                 .WORD   T11
000152  000000G                                 .WORD   T12
000154                                  ERRTYP::.BLKW   1
000156                                  ERRNBR::.BLKW   1
000160                                  ERRMSG::.BLKW   1
000162                                  ERRBLK::.BLKW   1
000164  000000C                         L$HWLEN::
                                                .WORD   <<L$NDHW-L$HWLEN>/2>
000166  172150                          P.IP.ADDRESS::
                                                .WORD   -5630
000170  000154                          P.VECTOR::
                                                .WORD   154
000172  000005                          P.BR.LEVEL::
                                                .WORD   5
000174  000000                          P.UNIT.NUMBER::
                                                .WORD   0
000176                                  L$NDHW::.BLKW   1
000200  000000C                         L$SWLEN::
                                                .WORD   <<L$NDSW-L$SWLEN>/2>
000202  000001                          SWP.TOP::
                                                .WORD   1
000204  000000                          SWP.LIMIT::
                                                .WORD   0
000206  000001                          SWP.START::
                                                .WORD   1
000210  001434                          SWP.END::
                                                .WORD   1434
000212  000000                          SWP.RETRIES::
```

```
                                                        .WORD   0
000214  000000                                SWP.CONTINUE::
                                                        .WORD   0
000216  000000                                SWP.MANUAL::
                                                        .WORD   0
000220  000001                                SWP.TRACE::
                                                        .WORD   1
000222                                          L$NDSW::.BLKW   1
000224  177777                                L$PROT::.WORD   -1
000226  177777                                        .WORD   -1
000230  177777                                        .WORD   -1


000000                                                  .PSECT  $PLIT$,  RO , D  , GBL
000000  000002                                        .WORD   2                       ; Plit count word
000002  005634'                                 P.AAA:  .WORD   P4
000004  005636'                                        .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  /PTO/
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/
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  050  105  123                .ASCII  /(ES/
000266  051  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/
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  043  061  063                .ASCII  /#13/
000417  040  124  110                .ASCII  / TH/
000422  122  125  040                .ASCII  /RU /
000425  043  061  065                .ASCII  /#15/
000430  000  000                     .ASCII  <00><00>
000432  123  124  101       P.AAM:   .ASCII  /STA/
```

```
000435  122  124  111                    .ASCII  /RTI/
000440  116  107  040                    .ASCII  /NG /
000443  124  122  101                    .ASCII  /TRA/
000446  103  113  000                    .ASCII  /CK/<00>
000451  000                             .ASCII  <00>
000452  105  116  104           P.AAN:   .ASCII  /END/
000455  111  116  107                    .ASCII  /ING/
000460  040  124  122                    .ASCII  / TR/
000463  101  103  113                    .ASCII  /ACK/
000466  000  000                        .ASCII  <00><00>
000470  104  117  040           P.AAO:   .ASCII  /DO /
000473  131  117  125                    .ASCII  /YOU/
000476  040  127  101                    .ASCII  / WA/
000501  116  124  040                    .ASCII  /NT /
000504  124  117  040                    .ASCII  /TO /
000507  104  117  040                    .ASCII  /DO /
000512  124  110  105                    .ASCII  /THE/
000515  040  115  101                    .ASCII  / MA/
000520  116  125  101                    .ASCII  /NUA/
000523  114  040  111                    .ASCII  /L I/
000526  116  124  105                    .ASCII  /NTE/
000531  122  126  105                    .ASCII  /RVE/
000534  116  124  111                    .ASCII  /NTI/
000537  117  116  040                    .ASCII  /ON /
000542  124  105  123                    .ASCII  /TES/
000545  124  077  000                    .ASCII  /T?/<00>
000550  104  117  040           P.AAP:   .ASCII  /DO /
000553  131  117  125                    .ASCII  /YOU/
000556  040  116  105                    .ASCII  / NE/
000561  105  104  040                    .ASCII  /ED /
000564  124  122  101                    .ASCII  /TRA/
000567  103  105  040                    .ASCII  /CE /
000572  115  117  104                    .ASCII  /MOD/
000575  105  077  000                    .ASCII  /E?/<00>
000600  104  117  040           P.AAQ:   .ASCII  /DO /
000603  131  117  125                    .ASCII  /YOU/
000606  040  127  111                    .ASCII  / WI/
000611  123  110  040                    .ASCII  /SH /
000614  124  117  040                    .ASCII  /TO /
000617  103  117  116                    .ASCII  /CON/
000622  124  111  116                    .ASCII  /TIN/
000625  125  105  040                    .ASCII  /UE /
000630  124  105  123                    .ASCII  /TES/
000633  124  111  116                    .ASCII  /TIN/
000636  107  040  101                    .ASCII  /G A/
000641  106  124  105                    .ASCII  /FTE/
000644  122  040  122                    .ASCII  /R R/
000647  105  124  122                    .ASCII  /ETR/
000652  111  105  123                    .ASCII  /IES/
000655  077  000  000                    .ASCII  /?/<00><00>
000660  116  125  115           P.AAR:   .ASCII  /NUM/
000663  102  105  122                    .ASCII  /BER/
000666  040  117  106                    .ASCII  / OF/
000671  040  122  105                    .ASCII  / RE/
000674  124  122  111                    .ASCII  /TRI/
000677  105  123  040                    .ASCII  /ES /
000702  106  117  122                    .ASCII  /FOR/
```

```
000705  040  124  105                    .ASCII  / TE/
000710  123  124  040                    .ASCII  /ST /
000713  111  106  040                    .ASCII  /IF /
000716  105  122  122                    .ASCII  /ERR/
000721  117  122  040                    .ASCII  /OR /
000724  117  103  103                    .ASCII  /OCC/
000727  125  122  105                    .ASCII  /URE/
000732  104  000                        .ASCII  /D/<00>
000734  125  116  111          P.AAS:    .ASCII  /UNI/
000737  124  040  123                    .ASCII  /T S/
000742  124  101  122                    .ASCII  /TAR/
000745  124  111  116                    .ASCII  /TIN/
000750  107  040  124                    .ASCII  /G T/
000753  122  101  103                    .ASCII  /RAC/
000756  113  040  043                    .ASCII  /K #/
000761  000                            .ASCII  <00>
000762  125  116  111          P.AAT:    .ASCII  /UNI/
000765  124  040  105                    .ASCII  /T E/
000770  116  104  111                    .ASCII  /NDI/
000773  116  107  040                    .ASCII  /NG /
000776  124  122  101                    .ASCII  /TRA/
001001  103  113  040                    .ASCII  /CK /
001004  043  000                        .ASCII  /#/<00>
001006  124  125  122          P.AAU:    .ASCII  /TUR/
001011  116  040  117                    .ASCII  /N O/
001014  106  106  040                    .ASCII  /FF /
001017  127  122  111                    .ASCII  /WRI/
001022  124  105  040                    .ASCII  /TE /
001025  120  122  117                    .ASCII  /PRO/
001030  124  105  103                    .ASCII  /TEC/
001033  124  040  123                    .ASCII  /T S/
001036  127  111  124                    .ASCII  /WIT/
001041  103  110  040                    .ASCII  /CH /
001044  101  116  104                    .ASCII  /AND/
001047  040  104  117                    .ASCII  / DO/
001052  040  074  103                    .ASCII  / <C/
001055  122  076  000                    .ASCII  /R>/<00>
001060  124  125  122          P.AAV:    .ASCII  /TUR/
001063  116  040  117                    .ASCII  /N O/
001066  116  040  127                    .ASCII  /N W/
001071  122  111  124                    .ASCII  /RIT/
001074  105  040  120                    .ASCII  /E P/
001077  122  117  124                    .ASCII  /ROT/
001102  105  103  124                    .ASCII  /ECT/
001105  040  123  127                    .ASCII  / SW/
001110  111  124  103                    .ASCII  /ITC/
001113  110  040  101                    .ASCII  /H A/
001116  116  104  040                    .ASCII  /ND /
001121  104  117  040                    .ASCII  /DO /
001124  074  103  122                    .ASCII  /<CR/
001127  076  000  000                    .ASCII  />/<00><00>
001132  045  116  045          P.AAW:    .ASCII  /%N%/
001135  116  045  116                    .ASCII  /N%N/
001140  045  101  124                    .ASCII  /%AT/
001143  105  123  124                    .ASCII  /EST/
001146  111  116  107                    .ASCII  /ING/
001151  040  125  116                    .ASCII  / UN/
```

```
001154  111  124  043                    .ASCII  /IT#/
001157  072  045  104                    .ASCII  /:%D/
001162  063  045  101                    .ASCII  /3%A/
001165  040  040  111                    .ASCII  /  I/
001170  120  137  122                    .ASCII  /P_R/
001173  105  107  111                    .ASCII  /EGI/
001176  123  124  105                    .ASCII  /STE/
001201  122  072  045                    .ASCII  /R:%/
001204  117  066  045                    .ASCII  /06%/
001207  101  040  040                    .ASCII  /A  /
001212  120  114  101                    .ASCII  /PLA/
001215  124  124  105                    .ASCII  /TTE/
001220  122  043  072                    .ASCII  /R#:/
001223  045  104  063                    .ASCII  /%D3/
001226  045  116  000                    .ASCII  /%N/<00>
001231  000                            .ASCII  <00>
001232  045  116  045            P.AAX:  .ASCII  /%N%/
001235  101  122  105                    .ASCII  /ARE/
001240  120  117  122                    .ASCII  /POR/
001243  124  000  000                    .ASCII  /T/<00><00>
001246  045  116  045            P.AAY:  .ASCII  /%N%/
001251  101  101  125                    .ASCII  /AAU/
001254  124  117  000                    .ASCII  /TO/<00>
001257  000                            .ASCII  <00>
001260  045  116  045            P.AAZ:  .ASCII  /%N%/
001263  101  103  114                    .ASCII  /ACL/
001266  105  101  116                    .ASCII  /EAN/
001271  125  120  000                    .ASCII  /UP/<00>
001274  045  116  045            P.ABA:  .ASCII  /%N%/
001277  101  104  122                    .ASCII  /ADR/
001302  117  120  120                    .ASCII  /OPP/
001305  105  104  000                    .ASCII  /ED/<00>
001310  045  116  045            P.ABB:  .ASCII  /%N%/
001313  101  101  104                    .ASCII  /AAD/
001316  104  105  104                    .ASCII  /DED/
001321  000                            .ASCII  <00>
001322  045  116  045            P.ABC:  .ASCII  /%N%/
001325  101  124  105                    .ASCII  /ATE/
001330  123  124  040                    .ASCII  /ST /
001333  040  061  040                    .ASCII  / 1 /
001336  122  105  107                    .ASCII  /REG/
001341  111  123  124                    .ASCII  /IST/
001344  105  122  040                    .ASCII  /ER /
001347  105  130  111                    .ASCII  /EXI/
001352  123  124  105                    .ASCII  /STE/
001355  116  103  105                    .ASCII  /NCE/
001360  040  124  105                    .ASCII  / TE/
001363  123  124  000                    .ASCII  /ST/<00>
001366  045  116  045            P.ABD:  .ASCII  /%N%/
001371  101  124  105                    .ASCII  /ATE/
001374  123  124  040                    .ASCII  /ST /
001377  040  062  040                    .ASCII  / 2 /
001402  123  124  105                    .ASCII  /STE/
001405  120  040  061                    .ASCII  /P 1/
001410  040  122  105                    .ASCII  / RE/
001413  101  104  057                    .ASCII  /AD/<57>
001416  127  122  111                    .ASCII  /WRI/
```

```
001421  124  105  040                    .ASCII  /TE /
001424  120  117  127                    .ASCII  /POW/
001427  105  122  125                    .ASCII  /ERU/
001432  120  040  104                    .ASCII  /P D/
001435  111  101  107                    .ASCII  /IAG/
001440  116  117  123                    .ASCII  /NOS/
001443  124  111  103                    .ASCII  /TIC/
001446  123  000                        .ASCII  /S/<00>
001450  045  116  045          P.ABE:    .ASCII  /%N%/
001453  101  124  105                    .ASCII  /ATE/
001456  123  124  040                    .ASCII  /ST /
001461  040  065  040                    .ASCII  / 5 /
001464  123  124  105                    .ASCII  /STE/
001467  120  040  061                    .ASCII  /P 1/
001472  040  124  110                    .ASCII  / TH/
001475  122  117  125                    .ASCII  /ROU/
001500  107  110  040                    .ASCII  /GH /
001503  123  124  105                    .ASCII  /STE/
001506  120  040  063                    .ASCII  /P 3/
001511  040  122  105                    .ASCII  / RE/
001514  101  104  057                    .ASCII  /AD/<57>
001517  127  122  111                    .ASCII  /WRI/
001522  124  105  040                    .ASCII  /TE /
001525  124  105  123                    .ASCII  /TES/
001530  124  000                        .ASCII  /T/<00>
001532  045  116  045          P.ABF:    .ASCII  /%N%/
001535  101  124  105                    .ASCII  /ATE/
001540  123  124  040                    .ASCII  /ST /
001543  040  063  040                    .ASCII  / 3 /
001546  104  111  101                    .ASCII  /DIA/
001551  107  116  117                    .ASCII  /GNO/
001554  123  124  111                    .ASCII  /STI/
001557  103  040  127                    .ASCII  /C W/
001562  122  101  120                    .ASCII  /RAP/
001565  040  124  105                    .ASCII  / TE/
001570  123  124  000                    .ASCII  /ST/<00>
001573  000                            .ASCII  <00>
001574  045  116  045          P.ABG:    .ASCII  /%N%/
001577  101  124  105                    .ASCII  /ATE/
001602  123  124  040                    .ASCII  /ST /
001605  040  064  040                    .ASCII  / 4 /
001610  126  105  103                    .ASCII  /VEC/
001613  124  117  122                    .ASCII  /TOR/
001616  040  101  116                    .ASCII  / AN/
001621  104  040  102                    .ASCII  /D B/
001624  122  040  114                    .ASCII  /R L/
001627  105  126  105                    .ASCII  /EVE/
001632  114  040  124                    .ASCII  /L T/
001635  105  123  124                    .ASCII  /EST/
001640  000  000                        .ASCII  <00><00>
001642  045  116  045          P.ABH:    .ASCII  /%N%/
001645  101  124  105                    .ASCII  /ATE/
001650  123  124  040                    .ASCII  /ST /
001653  040  066  040                    .ASCII  / 6 /
001656  120  125  122                    .ASCII  /PUR/
001661  107  105  040                    .ASCII  /GE /
001664  101  116  104                    .ASCII  /AND/
```

```
001667  040  120  117                    .ASCII  / PO/
001672  114  114  040                    .ASCII  /LL /
001675  124  105  123                    .ASCII  /TES/
001700  124  000                         .ASCII  /T/<00>
001702  045  116  045          P.ABI:    .ASCII  /%N%/
001705  101  124  105                    .ASCII  /ATE/
001710  123  124  040                    .ASCII  /ST /
001713  040  067  040                    .ASCII  / 7 /
001716  123  115  101                    .ASCII  /SMA/
001721  114  114  040                    .ASCII  /LL /
001724  122  111  116                    .ASCII  /RIN/
001727  107  040  124                    .ASCII  /G T/
001732  105  123  124                    .ASCII  /EST/
001735  000                            .ASCII  <00>
001736  045  116  045          P.ABJ:    .ASCII  /%N%/
001741  101  124  105                    .ASCII  /ATE/
001744  123  124  040                    .ASCII  /ST /
001747  040  070  040                    .ASCII  / 8 /
001752  114  101  122                    .ASCII  /LAR/
001755  107  105  040                    .ASCII  /GE /
001760  122  111  116                    .ASCII  /RIN/
001763  107  040  124                    .ASCII  /G T/
001766  105  123  124                    .ASCII  /EST/
001771  000                            .ASCII  <00>
001772  045  116  045          P.ABK:    .ASCII  /%N%/
001775  101  124  105                    .ASCII  /ATE/
002000  123  124  040                    .ASCII  /ST /
002003  040  071  040                    .ASCII  / 9 /
002006  104  115  040                    .ASCII  /DM /
002011  103  117  104                    .ASCII  /COD/
002014  105  040  117                    .ASCII  /E O/
002017  126  105  122                    .ASCII  /VER/
002022  114  101  131                    .ASCII  /LAY/
002025  040  124  105                    .ASCII  / TE/
002030  123  124  000                    .ASCII  /ST/<00>
002033  000                            .ASCII  <00>
002034  045  116  045          P.ABL:    .ASCII  /%N%/
002037  101  124  105                    .ASCII  /ATE/
002042  123  124  040                    .ASCII  /ST /
002045  061  060  040                    .ASCII  /10 /
002050  116  117  116                    .ASCII  /NON/
002053  105  130  111                    .ASCII  /EXI/
002056  123  124  105                    .ASCII  /STE/
002061  116  124  040                    .ASCII  /NT /
002064  115  105  115                    .ASCII  /MEM/
002067  117  122  131                    .ASCII  /ORY/
002072  040  124  105                    .ASCII  / TE/
002075  123  124  000                    .ASCII  /ST/<00>
002100  045  116  045          P.ABM:    .ASCII  /%N%/
002103  101  124  105                    .ASCII  /ATE/
002106  123  124  040                    .ASCII  /ST /
002111  061  061  040                    .ASCII  /11 /
002114  102  125  123                    .ASCII  /BUS/
002117  040  101  104                    .ASCII  / AD/
002122  104  122  105                    .ASCII  /DRE/
002125  123  123  111                    .ASCII  /SSI/
002130  116  107  057                    .ASCII  /NG/<57>
```

```
002133  104  101  124                    .ASCII  /DAT/
002136  101  040  124                    .ASCII  /A T/
002141  105  123  124                    .ASCII  /EST/
002144  040  101  000                    .ASCII  / A/<00>
002147  000                             .ASCII  <00>
002150  045  116  045          P.ABN:    .ASCII  /%N%/
002153  101  124  105                    .ASCII  /ATE/
002156  123  124  040                    .ASCII  /ST /
002161  061  062  040                    .ASCII  /12 /
002164  102  125  123                    .ASCII  /BUS/
002167  040  101  104                    .ASCII  / AD/
002172  104  122  105                    .ASCII  /DRE/
002175  123  123  111                    .ASCII  /SSI/
002200  116  107  057                    .ASCII  /NG/<57>
002203  104  101  124                    .ASCII  /DAT/
002206  101  040  124                    .ASCII  /A T/
002211  105  123  124                    .ASCII  /EST/
002214  040  102  000                    .ASCII  / B/<00>
002217  000                             .ASCII  <00>
002220  045  116  045          P.ABO:    .ASCII  /%N%/
002223  101  124  105                    .ASCII  /ATE/
002226  123  124  040                    .ASCII  /ST /
002231  061  063  040                    .ASCII  /13 /
002234  102  114  117                    .ASCII  /BLO/
002237  103  113  040                    .ASCII  /CK /
002242  124  122  101                    .ASCII  /TRA/
002245  116  123  106                    .ASCII  /NSF/
002250  105  122  040                    .ASCII  /ER /
002253  124  105  123                    .ASCII  /TES/
002256  124  000                        .ASCII  /T/<00>
002260  045  116  045          P.ABP:    .ASCII  /%N%/
002263  101  124  105                    .ASCII  /ATE/
002266  123  124  040                    .ASCII  /ST /
002271  061  064  040                    .ASCII  /14 /
002274  123  120  111                    .ASCII  /SPI/
002277  116  040  125                    .ASCII  /N U/
002302  120  040  110                    .ASCII  /P H/
002305  105  101  104                    .ASCII  /EAD/
002310  040  114  117                    .ASCII  / LO/
002313  101  104  040                    .ASCII  /AD /
002316  123  105  121                    .ASCII  /SEQ/
002321  125  105  116                    .ASCII  /UEN/
002324  103  105  000                    .ASCII  /CE/<00>
002327  000                             .ASCII  <00>
002330  045  116  045          P.ABQ:    .ASCII  /%N%/
002333  101  124  105                    .ASCII  /ATE/
002336  123  124  040                    .ASCII  /ST /
002341  061  065  040                    .ASCII  /15 /
002344  123  105  121                    .ASCII  /SEQ/
002347  125  105  116                    .ASCII  /UEN/
002352  124  111  101                    .ASCII  /TIA/
002355  114  040  123                    .ASCII  /L S/
002360  105  105  113                    .ASCII  /EEK/
002363  040  101  116                    .ASCII  / AN/
002366  104  040  126                    .ASCII  /D V/
002371  105  122  111                    .ASCII  /ERI/
002374  106  131  000                    .ASCII  /FY/<00>
```

```
002377  000                             .ASCII  <00>
002400  045  116  045          P.ABR:    .ASCII  /%N%/
002403  101  124  105                    .ASCII  /ATE/
002406  123  124  040                    .ASCII  /ST /
002411  061  066  040                    .ASCII  /16 /
002414  123  101  127                    .ASCII  /SAW/
002417  124  117  117                    .ASCII  /TOO/
002422  124  110  040                    .ASCII  /TH /
002425  123  105  105                    .ASCII  /SEE/
002430  113  040  101                    .ASCII  /K A/
002433  116  104  040                    .ASCII  /ND /
002436  126  105  122                    .ASCII  /VER/
002441  111  106  131                    .ASCII  /IFY/
002444  000  000                        .ASCII  <00><00>
002446  045  116  045          P.ABS:    .ASCII  /%N%/
002451  101  124  105                    .ASCII  /ATE/
002454  123  124  040                    .ASCII  /ST /
002457  061  067  040                    .ASCII  /17 /
002462  103  117  116                    .ASCII  /CON/
002465  126  105  122                    .ASCII  /VER/
002470  107  111  116                    .ASCII  /GIN/
002473  107  057  104                    .ASCII  /G/<57>/D/
002476  111  126  105                    .ASCII  /IVE/
002501  122  107  111                    .ASCII  /RGI/
002504  116  107  040                    .ASCII  /NG /
002507  123  105  105                    .ASCII  /SEE/
002512  113  040  101                    .ASCII  /K A/
002515  116  104  040                    .ASCII  /ND /
002520  126  105  122                    .ASCII  /VER/
002523  111  106  131                    .ASCII  /IFY/
002526  000  000                        .ASCII  <00><00>
002530  045  116  045          P.ABT:    .ASCII  /%N%/
002533  101  124  105                    .ASCII  /ATE/
002536  123  124  040                    .ASCII  /ST /
002541  061  070  040                    .ASCII  /18 /
002544  124  117  107                    .ASCII  /TOG/
002547  107  114  105                    .ASCII  /GLE/
002552  040  123  105                    .ASCII  / SE/
002555  105  113  040                    .ASCII  /EK /
002560  101  116  104                    .ASCII  /AND/
002563  040  126  105                    .ASCII  / VE/
002566  122  111  106                    .ASCII  /RIF/
002571  131  000  000                    .ASCII  /Y/<00><00>
002574  045  116  045          P.ABU:    .ASCII  /%N%/
002577  101  124  105                    .ASCII  /ATE/
002602  123  124  040                    .ASCII  /ST /
002605  061  071  040                    .ASCII  /19 /
002610  110  105  101                    .ASCII  /HEA/
002613  104  040  123                    .ASCII  /D S/
002616  127  111  124                    .ASCII  /WIT/
002621  103  110  040                    .ASCII  /CH /
002624  124  105  123                    .ASCII  /TES/
002627  124  000  000                    .ASCII  /T/<00><00>
002632  045  116  045          P.ABV:    .ASCII  /%N%/
002635  101  124  105                    .ASCII  /ATE/
002640  123  124  040                    .ASCII  /ST /
002643  062  060  040                    .ASCII  /20 /
```

```
002646  122  101  116                    .ASCII  /RAN/
002651  104  117  115                    .ASCII  /DOM/
002654  040  123  105                    .ASCII  / SE/
002657  105  113  040                    .ASCII  /EK /
002662  101  116  104                    .ASCII  /AND/
002665  040  126  105                    .ASCII  / VE/
002670  122  111  106                    .ASCII  /RIF/
002673  131  000  000                    .ASCII  /Y/<00><00>
002676  045  116  045          P.ABW:    .ASCII  /%N%/
002701  101  124  105                    .ASCII  /ATE/
002704  123  124  040                    .ASCII  /ST /
002707  062  061  040                    .ASCII  /21 /
002712  123  105  103                    .ASCII  /SEC/
002715  124  117  122                    .ASCII  /TOR/
002720  040  101  103                    .ASCII  / AC/
002723  103  105  123                    .ASCII  /CES/
002726  123  040  124                    .ASCII  /S T/
002731  105  123  124                    .ASCII  /EST/
002734  000  000                        .ASCII  <00><00>
002736  045  116  045          P.ABX:    .ASCII  /%N%/
002741  101  124  105                    .ASCII  /ATE/
002744  123  124  040                    .ASCII  /ST /
002747  062  062  040                    .ASCII  /22 /
002752  103  117  116                    .ASCII  /CON/
002755  124  122  117                    .ASCII  /TRO/
002760  114  114  105                    .ASCII  /LLE/
002763  122  040  120                    .ASCII  /R P/
002766  122  117  103                    .ASCII  /ROC/
002771  105  123  123                    .ASCII  /ESS/
002774  111  116  107                    .ASCII  /ING/
002777  040  124  111                    .ASCII  / TI/
003002  115  105  000                    .ASCII  /ME/<00>
003005  000                            .ASCII  <00>
003006  045  116  045          P.ABY:    .ASCII  /%N%/
003011  101  124  105                    .ASCII  /ATE/
003014  123  124  040                    .ASCII  /ST /
003017  062  063  040                    .ASCII  /23 /
003022  117  116  105                    .ASCII  /ONE/
003025  040  124  122                    .ASCII  / TR/
003030  101  103  113                    .ASCII  /ACK/
003033  040  123  105                    .ASCII  / SE/
003036  105  113  040                    .ASCII  /EK /
003041  124  111  115                    .ASCII  /TIM/
003044  105  000                        .ASCII  /E/<00>
003046  045  116  045          P.ABZ:    .ASCII  /%N%/
003051  101  124  105                    .ASCII  /ATE/
003054  123  124  040                    .ASCII  /ST /
003057  062  064  040                    .ASCII  /24 /
003062  101  126  105                    .ASCII  /AVE/
003065  122  101  107                    .ASCII  /RAG/
003070  105  040  123                    .ASCII  /E S/
003073  105  105  113                    .ASCII  /EEK/
003076  040  124  111                    .ASCII  / TI/
003101  115  105  000                    .ASCII  /ME/<00>
003104  045  116  045          P.ACA:    .ASCII  /%N%/
003107  101  124  105                    .ASCII  /ATE/
003112  123  124  040                    .ASCII  /ST /
```

```
003115  062  065  040                    .ASCII  /25 /
003120  106  125  114                    .ASCII  /FUL/
003123  114  040  123                    .ASCII  /L S/
003126  124  122  117                    .ASCII  /TRO/
003131  113  105  040                    .ASCII  /KE /
003134  123  105  105                    .ASCII  /SEE/
003137  113  040  124                    .ASCII  /K T/
003142  111  115  105                    .ASCII  /IME/
003145  000                             .ASCII  <00>
003146  045  116  045          P.ACB:    .ASCII  /%N%/
003151  101  124  105                    .ASCII  /ATE/
003154  123  124  040                    .ASCII  /ST /
003157  062  066  040                    .ASCII  /26 /
003162  127  122  111                    .ASCII  /WRI/
003165  124  105  040                    .ASCII  /TE /
003170  104  101  124                    .ASCII  /DAT/
003173  101  040  124                    .ASCII  /A T/
003176  105  123  124                    .ASCII  /EST/
003201  000                             .ASCII  <00>
003202  045  116  045          P.ACC:    .ASCII  /%N%/
003205  101  105  126                    .ASCII  /AEV/
003210  105  116  124                    .ASCII  /ENT/
003213  040  123  124                    .ASCII  / ST/
003216  101  122  124                    .ASCII  /ART/
003221  000                             .ASCII  <00>
003222  045  116  045          P.ACD:    .ASCII  /%N%/
003225  101  105  126                    .ASCII  /AEV/
003230  105  116  124                    .ASCII  /ENT/
003233  040  122  105                    .ASCII  / RE/
003236  123  124  101                    .ASCII  /STA/
003241  122  124  000                    .ASCII  /RT/<00>
003244  045  116  045          P.ACE:    .ASCII  /%N%/
003247  101  105  126                    .ASCII  /AEV/
003252  105  116  124                    .ASCII  /ENT/
003255  040  103  117                    .ASCII  / CO/
003260  116  124  111                    .ASCII  /NTI/
003263  116  125  105                    .ASCII  /NUE/
003266  000  000                        .ASCII  <00><00>
003270  045  116  045          P.ACF:    .ASCII  /%N%/
003273  101  124  105                    .ASCII  /ATE/
003276  123  124  040                    .ASCII  /ST /
003301  062  067  040                    .ASCII  /27 /
003304  117  106  106                    .ASCII  /OFF/
003307  123  105  124                    .ASCII  /SET/
003312  040  124  117                    .ASCII  / TO/
003315  114  105  122                    .ASCII  /LER/
003320  101  116  103                    .ASCII  /ANC/
003323  105  040  124                    .ASCII  /E T/
003326  105  123  124                    .ASCII  /EST/
003331  000                             .ASCII  <00>
003332  045  116  045          P.ACG:    .ASCII  /%N%/
003335  101  124  105                    .ASCII  /ATE/
003340  123  124  040                    .ASCII  /ST /
003343  062  070  040                    .ASCII  /28 /
003346  101  126  105                    .ASCII  /AVE/
003351  122  101  107                    .ASCII  /RAG/
003354  105  040  122                    .ASCII  /E R/
```

```
003357  117  124  101                    .ASCII  /OTA/
003362  124  111  117                    .ASCII  /TIO/
003365  116  101  114                    .ASCII  /NAL/
003370  040  124  111                    .ASCII  / TI/
003373  115  105  000                    .ASCII  /ME/<00>
003376  045  116  045          P.ACH:    .ASCII  /%N%/
003401  101  124  105                    .ASCII  /ATE/
003404  123  124  040                    .ASCII  /ST /
003407  062  071  040                    .ASCII  /29 /
003412  127  122  111                    .ASCII  /WRI/
003415  124  105  040                    .ASCII  /TE /
003420  120  122  117                    .ASCII  /PRO/
003423  124  105  103                    .ASCII  /TEC/
003426  124  040  124                    .ASCII  /T T/
003431  105  123  124                    .ASCII  /EST/
003434  000  000                        .ASCII  <00><00>
003436  045  116  045          P.ACI:    .ASCII  /%N%/
003441  101  011  011                    .ASCII  /A/<11><11>
003444  115  101  116                    .ASCII  /MAN/
003447  125  101  114                    .ASCII  /UAL/
003452  040  111  116                    .ASCII  / IN/
003455  124  105  122                    .ASCII  /TER/
003460  126  105  116                    .ASCII  /VEN/
003463  124  111  117                    .ASCII  /TIO/
003466  116  040  124                    .ASCII  /N T/
003471  105  123  124                    .ASCII  /EST/
003474  040  116  117                    .ASCII  / NO/
003477  124  040  120                    .ASCII  /T P/
003502  105  122  106                    .ASCII  /ERF/
003505  117  122  115                    .ASCII  /ORM/
003510  105  104  000                    .ASCII  /ED/<00>
003513  000                            .ASCII  <00>
003514  045  116  045          P.ACJ:    .ASCII  /%N%/
003517  101  120  117                    .ASCII  /APO/
003522  127  105  122                    .ASCII  /WER/
003525  040  104  105                    .ASCII  / DE/
003530  114  101  131                    .ASCII  /LAY/
003533  040  055  040                    .ASCII  / - /
003536  127  101  111                    .ASCII  /WAI/
003541  124  111  116                    .ASCII  /TIN/
003544  107  000                        .ASCII  /G/<00>
003546  045  116  045          P.ACK:    .ASCII  /%N%/
003551  101  124  117                    .ASCII  /ATO/
003554  117  040  115                    .ASCII  /O M/
003557  101  116  131                    .ASCII  /ANY/
003562  040  125  116                    .ASCII  / UN/
003565  111  124  123                    .ASCII  /ITS/
003570  000  000                        .ASCII  <00><00>
003572  045  116  045          P.ACL:    .ASCII  /%N%/
003575  101  116  117                    .ASCII  /ANO/
003600  040  103  114                    .ASCII  / CL/
003603  117  103  113                    .ASCII  /OCK/
003606  040  127  101                    .ASCII  / WA/
003611  123  040  106                    .ASCII  /S F/
003614  117  125  116                    .ASCII  /OUN/
003617  104  040  117                    .ASCII  /D O/
003622  116  040  124                    .ASCII  /N T/
```

```
003625  110  105  040                    .ASCII  /HE /
003630  123  131  123                    .ASCII  /SYS/
003633  124  105  115                    .ASCII  /TEM/
003636  000  000                         .ASCII  <00><00>
003640  045  116  045          P.ACM:    .ASCII  /%N%/
003643  116  000  000                    .ASCII  /N/<00><00>
003646  045  116  045          P.ACN:    .ASCII  /%N%/
003651  101  011  040                    .ASCII  /A/<11>/ /
003654  122  105  107                    .ASCII  /REG/
003657  111  123  124                    .ASCII  /IST/
003662  105  122  040                    .ASCII  /ER /
003665  106  101  111                    .ASCII  /FAI/
003670  114  105  104                    .ASCII  /LED/
003673  040  124  117                    .ASCII  / TO/
003676  040  122  105                    .ASCII  / RE/
003701  123  120  117                    .ASCII  /SPO/
003704  116  104  040                    .ASCII  /ND /
003707  101  124  040                    .ASCII  /AT /
003712  101  104  104                    .ASCII  /ADD/
003715  122  105  123                    .ASCII  /RES/
003720  123  072  040                    .ASCII  /S: /
003723  040  045  117                    .ASCII  / %O/
003726  066  045  116                    .ASCII  /6%N/
003731  000                             .ASCII  <00>
003732  045  116  045          P.ACO:    .ASCII  /%N%/
003735  101  101  104                    .ASCII  /AAD/
003740  104  122  105                    .ASCII  /DRE/
003743  123  123  072                    .ASCII  /SS:/
003746  040  045  117                    .ASCII  / %O/
003751  066  045  101                    .ASCII  /6%A/
003754  011  105  130                    .ASCII  <11>/EX/
003757  120  105  103                    .ASCII  /PEC/
003762  124  105  104                    .ASCII  /TED/
003765  072  040  045                    .ASCII  /: %/
003770  117  066  045                    .ASCII  /O6%/
003773  101  011  122                    .ASCII  /A/<11>/R/
003776  105  101  104                    .ASCII  /EAD/
004001  072  040  045                    .ASCII  /: %/
004004  117  066  045                    .ASCII  /O6%/
004007  116  000  000                    .ASCII  /N/<00><00>
004012  045  116  045          P.ACP:    .ASCII  /%N%/
004015  101  123  124                    .ASCII  /AST/
004020  105  120  040                    .ASCII  /EP /
004023  115  101  123                    .ASCII  /MAS/
004026  113  040  075                    .ASCII  /K =/
004031  040  045  117                    .ASCII  / %O/
004034  062  045  101                    .ASCII  /2%A/
004037  011  106  101                    .ASCII  <11>/FA/
004042  111  114  111                    .ASCII  /ILI/
004045  116  107  040                    .ASCII  /NG /
004050  122  105  107                    .ASCII  /REG/
004053  111  123  124                    .ASCII  /IST/
004056  105  122  040                    .ASCII  /ER /
004061  075  040  045                    .ASCII  /= %/
004064  117  066  045                    .ASCII  /O6%/
004067  101  040  104                    .ASCII  /A D/
004072  101  124  101                    .ASCII  /ATA/
```

```
004075  040  075  040                    .ASCII  / = /
004100  045  117  066                    .ASCII  /%O6/
004103  045  116  000                    .ASCII  /%N/<00>
004106  045  116  045          P.ACQ:    .ASCII  /%N%/
004111  101  011  040                    .ASCII  /A/<11>/ /
004114  120  117  122                    .ASCII  /POR/
004117  124  040  124                    .ASCII  /T T/
004122  131  120  105                    .ASCII  /YPE/
004125  040  116  125                    .ASCII  / NU/
004130  115  102  105                    .ASCII  /MBE/
004133  122  040  075                    .ASCII  /R =/
004136  040  045  117                    .ASCII  / %O/
004141  062  000  000                    .ASCII  /2/<00><00>
004144  045  116  045          P.ACR:    .ASCII  /%N%/
004147  101  011  040                    .ASCII  /A/<11>/ /
004152  120  117  122                    .ASCII  /POR/
004155  124  040  123                    .ASCII  /T S/
004160  120  105  103                    .ASCII  /PEC/
004163  111  106  111                    .ASCII  /IFI/
004166  103  040  111                    .ASCII  /C I/
004171  116  106  117                    .ASCII  /NFO/
004174  072  057  116                    .ASCII  /:/<57>/N/
004177  126  057  121                    .ASCII  /V/<57>/Q/
004202  102  057  104                    .ASCII  /B/<57>/D/
004205  111  057  117                    .ASCII  /I/<57>/O/
004210  104  057  115                    .ASCII  /D/<57>/M/
004213  120  057  040                    .ASCII  /P/<57>/ /
004216  075  040  045                    .ASCII  /= %/
004221  117  062  000                    .ASCII  /O2/<00>
004224  045  116  045          P.ACS:    .ASCII  /%N%/
004227  101  011  040                    .ASCII  /A/<11>/ /
004232  115  111  103                    .ASCII  /MIC/
004235  122  117  040                    .ASCII  /RO /
004240  103  117  104                    .ASCII  /COD/
004243  105  072  040                    .ASCII  /E: /
004246  115  117  104                    .ASCII  /MOD/
004251  105  114  040                    .ASCII  /EL /
004254  075  040  045                    .ASCII  /= %/
004257  117  062  045                    .ASCII  /O2%/
004262  101  040  040                    .ASCII  /A  /
004265  040  126  105                    .ASCII  / VE/
004270  122  123  111                    .ASCII  /RSI/
004273  117  116  040                    .ASCII  /ON /
004276  075  040  045                    .ASCII  /= %/
004301  117  062  000                    .ASCII  /O2/<00>
004304  045  116  045          P.ACT:    .ASCII  /%N%/
004307  101  011  040                    .ASCII  /A/<11>/ /
004312  116  125  115                    .ASCII  /NUM/
004315  102  105  122                    .ASCII  /BER/
004320  040  117  106                    .ASCII  / OF/
004323  040  122  105                    .ASCII  / RE/
004326  124  122  111                    .ASCII  /TRI/
004331  105  123  040                    .ASCII  /ES /
004334  075  045  104                    .ASCII  /=%D/
004337  064  000  000                    .ASCII  /4/<00><00>
004342  011  127  101          P.ACU:    .ASCII  <11>/WA/
004345  111  124  040                    .ASCII  /IT /
```

```
004350  137  040  120                    .ASCII  / P/
004353  117  127  105                    .ASCII  /OWE/
004356  122  040  106                    .ASCII  /R F/
004361  101  111  114                    .ASCII  /AIL/
004364  040  122  105                    .ASCII  / RE/
004367  103  117  126                    .ASCII  /COV/
004372  105  122  131                    .ASCII  /ERY/
004375  000                             .ASCII  <00>
004376  122  103  123          P.ACV:    .ASCII  /RCS/
004401  101  040  106                    .ASCII  /A F/
004404  101  111  114                    .ASCII  /AIL/
004407  105  104  040                    .ASCII  /ED /
004412  124  117  040                    .ASCII  /TO /
004415  122  105  123                    .ASCII  /RES/
004420  120  117  116                    .ASCII  /PON/
004423  104  000  000                    .ASCII  /D/<00><00>
004426  122  103  111          P.ACW:    .ASCII  /RCI/
004431  120  040  106                    .ASCII  /P F/
004434  101  111  114                    .ASCII  /AIL/
004437  105  104  040                    .ASCII  /ED /
004442  124  117  040                    .ASCII  /TO /
004445  122  105  123                    .ASCII  /RES/
004450  120  117  116                    .ASCII  /PON/
004453  104  000  000                    .ASCII  /D/<00><00>
004456  124  105  123          P.ACX:    .ASCII  /TES/
004461  124  040  120                    .ASCII  /T P/
004464  101  124  124                    .ASCII  /ATT/
004467  105  122  116                    .ASCII  /ERN/
004472  040  105  103                    .ASCII  / EC/
004475  110  117  105                    .ASCII  /HOE/
004500  104  040  111                    .ASCII  /D I/
004503  116  040  122                    .ASCII  /N R/
004506  103  123  101                    .ASCII  /CSA/
004511  040  111  123                    .ASCII  / IS/
004514  040  111  116                    .ASCII  / IN/
004517  103  117  122                    .ASCII  /COR/
004522  122  105  103                    .ASCII  /REC/
004525  124  000  000                    .ASCII  /T/<00><00>
004530  126  105  103          P.ACY:    .ASCII  /VEC/
004533  124  117  122                    .ASCII  /TOR/
004536  040  101  116                    .ASCII  / AN/
004541  104  040  102                    .ASCII  /D B/
004544  122  040  114                    .ASCII  /R L/
004547  105  126  105                    .ASCII  /EVE/
004552  114  040  124                    .ASCII  /L T/
004555  105  123  124                    .ASCII  /EST/
004560  040  106  101                    .ASCII  / FA/
004563  111  114  125                    .ASCII  /ILU/
004566  122  105  000                    .ASCII  /RE/<00>
004571  000                             .ASCII  <00>
004572  110  117  123          P.ACZ:    .ASCII  /HOS/
004575  124  040  104                    .ASCII  /T D/
004600  105  124  105                    .ASCII  /ETE/
004603  103  124  105                    .ASCII  /CTE/
004606  104  040  124                    .ASCII  /D T/
004611  111  115  105                    .ASCII  /IME/
004614  040  117  125                    .ASCII  / OU/
```

```
004617  124  040  105                .ASCII  /T E/
004622  122  122  117                .ASCII  /RRO/
004625  122  000  000                .ASCII  /R/<00><00>
004630  122  111  116        P.ADA:  .ASCII  /RIN/
004633  107  040  102                .ASCII  /G B/
004636  125  106  106                .ASCII  /UFF/
004641  105  122  123                .ASCII  /ERS/
004644  040  116  117                .ASCII  / NO/
004647  124  040  103                .ASCII  /T C/
004652  114  105  101                .ASCII  /LEA/
004655  122  105  104                .ASCII  /RED/
004660  040  102  131                .ASCII  / BY/
004663  040  124  110                .ASCII  / TH/
004666  105  040  120                .ASCII  /E P/
004671  117  122  124                .ASCII  /ORT/
004674  000  000                     .ASCII  <00><00>
004676  123  124  105        P.ADB:  .ASCII  /STE/
004701  120  040  122                .ASCII  /P R/
004704  105  101  104                .ASCII  /EAD/
004707  040  104  101                .ASCII  / DA/
004712  124  101  040                .ASCII  /TA /
004715  104  117  105                .ASCII  /DOE/
004720  123  040  116                .ASCII  /S N/
004723  117  124  040                .ASCII  /OT /
004726  115  101  124                .ASCII  /MAT/
004731  103  110  000                .ASCII  /CH/<00>
004734  120  117  122        P.ADC:  .ASCII  /POR/
004737  124  040  106                .ASCII  /T F/
004742  101  124  101                .ASCII  /ATA/
004745  114  040  105                .ASCII  /L E/
004750  122  122  117                .ASCII  /RRO/
004753  122  000  000                .ASCII  /R/<00><00>
004756  111  116  111        P.ADD:  .ASCII  /INI/
004761  124  040  123                .ASCII  /T S/
004764  124  105  120                .ASCII  /TEP/
004767  040  122  105                .ASCII  / RE/
004772  101  104  040                .ASCII  /AD /
004775  105  122  122                .ASCII  /ERR/
005000  117  122  000                .ASCII  /OR/<00>
005003  000                         .ASCII  <00>
005004  115  105  115        P.ADE:  .ASCII  /MEM/
005007  117  122  131                .ASCII  /ORY/
005012  040  102  125                .ASCII  / BU/
005015  106  106  105                .ASCII  /FFE/
005020  122  040  104                .ASCII  /R D/
005023  117  105  123                .ASCII  /OES/
005026  040  116  117                .ASCII  / NO/
005031  124  040  103                .ASCII  /T C/
005034  117  116  124                .ASCII  /ONT/
005037  101  111  116                .ASCII  /AIN/
005042  040  105  130                .ASCII  / EX/
005045  120  105  103                .ASCII  /PEC/
005050  124  105  104                .ASCII  /TED/
005053  040  104  101                .ASCII  / DA/
005056  124  101  000                .ASCII  /TA/<00>
005061  000                         .ASCII  <00>
005062  104  115  040        P.ADF:  .ASCII  /DM /
```

```
005065  103  117  104                .ASCII  /COD/
005070  105  040  122                .ASCII  /E R/
005073  105  124  125                .ASCII  /ETU/
005076  122  116  105                .ASCII  /RNE/
005101  104  040  106                .ASCII  /D F/
005104  101  111  114                .ASCII  /AIL/
005107  125  122  105                .ASCII  /URE/
005112  040  103  117                .ASCII  / CO/
005115  104  105  000                .ASCII  /DE/<00>
005120  045  116  045      P.ADG:    .ASCII  /%N%/
005123  101  040  040                .ASCII  /A  /
005126  040  040  040                .ASCII  /   /
005131  040  040  040                .ASCII  /   /
005134  111  116  124                .ASCII  /INT/
005137  105  122  122                .ASCII  /ERR/
005142  125  120  124                .ASCII  /UPT/
005145  040  101  124                .ASCII  / AT/
005150  040  126  105                .ASCII  / VE/
005153  103  075  040                .ASCII  /C= /
005156  045  117  063                .ASCII  /%03/
005161  045  101  040                .ASCII  /%A /
005164  102  122  040                .ASCII  /BR /
005167  114  105  126                .ASCII  /LEV/
005172  105  114  075                .ASCII  /EL=/
005175  040  045  117                .ASCII  / %O/
005200  061  000                     .ASCII  /1/<00>
005202  045  116  045      P.ADH:    .ASCII  /%N%/
005205  101  011  116                .ASCII  /A/<11>/N/
005210  117  040  111                .ASCII  /O I/
005213  116  124  105                .ASCII  /NTE/
005216  122  122  125                .ASCII  /RRU/
005221  120  124  040                .ASCII  /PT /
005224  106  122  117                .ASCII  /FRO/
005227  115  040  120                .ASCII  /M P/
005232  117  122  124                .ASCII  /ORT/
005235  040  057  040                .ASCII  / /<57>/ /
005240  103  117  116                .ASCII  /CON/
005243  124  122  117                .ASCII  /TRO/
005246  114  114  105                .ASCII  /LLE/
005251  122  000  000                .ASCII  /R/<00><00>
005254  045  116  045      P.ADI:    .ASCII  /%N%/
005257  101  011  011                .ASCII  /A/<11><11>
005262  102  122  040                .ASCII  /BR /
005265  114  105  126                .ASCII  /LEV/
005270  105  114  040                .ASCII  /EL /
005273  122  105  103                .ASCII  /REC/
005276  105  111  126                .ASCII  /EIV/
005301  105  104  057                .ASCII  /ED/<57>
005304  124  131  120                .ASCII  /TYP/
005307  105  104  040                .ASCII  /ED /
005312  111  123  040                .ASCII  /IS /
005315  111  116  103                .ASCII  /INC/
005320  117  122  122                .ASCII  /ORR/
005323  105  103  124                .ASCII  /ECT/
005326  040  041  000                .ASCII  / !/<00>
005331  000                         .ASCII  <00>
005332  120  125  122      P.ADJ:    .ASCII  /PUR/
```

```
005335  107  105  040                    .ASCII  /GE /
005340  101  116  104                    .ASCII  /AND/
005343  040  120  117                    .ASCII  / PO/
005346  114  105  040                    .ASCII  /LE /
005351  124  105  123                    .ASCII  /TES/
005354  124  040  123                    .ASCII  /T S/
005357  105  124  040                    .ASCII  /ET /
005362  105  122  122                    .ASCII  /ERR/
005365  117  122  040                    .ASCII  /OR /
005370  102  111  124                    .ASCII  /BIT/
005373  040  061  065                    .ASCII  / 15/
005376  000  000                         .ASCII  <00><00>
005400  120  125  122          P.ADK:    .ASCII  /PUR/
005403  107  105  040                    .ASCII  /GE /
005406  101  116  104                    .ASCII  /AND/
005411  040  120  117                    .ASCII  / PO/
005414  114  105  040                    .ASCII  /LE /
005417  124  105  123                    .ASCII  /TES/
005422  124  040  104                    .ASCII  /T D/
005425  111  104  040                    .ASCII  /ID /
005430  116  117  124                    .ASCII  /NOT/
005433  040  123  105                    .ASCII  / SE/
005436  124  040  123                    .ASCII  /T S/
005441  124  105  120                    .ASCII  /TEP/
005444  040  064  040                    .ASCII  / 4 /
005447  102  111  124                    .ASCII  /BIT/
005452  040  061  064                    .ASCII  / 14/
005455  000                             .ASCII  <00>
005456  111  116  111          P.ADL:    .ASCII  /INI/
005461  124  040  104                    .ASCII  /T D/
005464  111  104  040                    .ASCII  /ID /
005467  116  117  040                    .ASCII  /NO /
005472  103  114  105                    .ASCII  /CLE/
005475  101  122  040                    .ASCII  /AR /
005500  122  111  116                    .ASCII  /RIN/
005503  107  040  102                    .ASCII  /G B/
005506  125  106  106                    .ASCII  /UFF/
005511  105  122  000                    .ASCII  /ER/<00>
005514  106  101  111          P.ADM:    .ASCII  /FAI/
005517  114  105  104                    .ASCII  /LED/
005522  040  120  117                    .ASCII  / PO/
005525  114  114  111                    .ASCII  /LLI/
005530  116  107  040                    .ASCII  /NG /
005533  105  122  122                    .ASCII  /ERR/
005536  117  122  040                    .ASCII  /OR /
005541  111  116  040                    .ASCII  /IN /
005544  122  105  123                    .ASCII  /RES/
005547  120  117  116                    .ASCII  /PON/
005552  103  105  040                    .ASCII  /CE /
005555  122  111  116                    .ASCII  /RIN/
005560  107  000                         .ASCII  /G/<00>
005562  101  126  101          P.ADN:    .ASCII  /AVA/
005565  111  114  101                    .ASCII  /ILA/
005570  102  114  105                    .ASCII  /BLE/
005573  040  103  117                    .ASCII  / CO/
005576  115  115  101                    .ASCII  /MMA/
005601  116  104  040                    .ASCII  /ND /
```

```
005604  123  120  111            .ASCII  /SPI/
005607  116  055  104            .ASCII  /N-D/
005612  117  127  116            .ASCII  /OWN/
005615  040  106  101            .ASCII  / FA/
005620  111  114  125            .ASCII  /ILU/
005623  122  105  000            .ASCII  /RE/<00>
005626  123  120  111  P.ADO:     .ASCII  /SPI/
005631  116  040  125            .ASCII  /N U/
005634  120  040  124            .ASCII  /P T/
005637  105  123  124            .ASCII  /EST/
005642  040  106  101            .ASCII  / FA/
005645  111  114  125            .ASCII  /ILU/
005650  122  105  000            .ASCII  /RE/<00>
005653  000                      .ASCII  <00>
005654  123  105  121  P.ADP:     .ASCII  /SEQ/
005657  125  105  116            .ASCII  /UEN/
005662  124  111  101            .ASCII  /TIA/
005665  114  040  106            .ASCII  /L F/
005670  117  122  127            .ASCII  /ORW/
005673  101  122  104            .ASCII  /ARD/
005676  040  123  105            .ASCII  / SE/
005701  105  113  040            .ASCII  /EK /
005704  106  101  111            .ASCII  /FAI/
005707  114  125  122            .ASCII  /LUR/
005712  105  000                 .ASCII  /E/<00>
005714  123  105  121  P.ADQ:     .ASCII  /SEQ/
005717  125  105  116            .ASCII  /UEN/
005722  124  111  101            .ASCII  /TIA/
005725  114  040  122            .ASCII  /L R/
005730  105  126  105            .ASCII  /EVE/
005733  122  123  105            .ASCII  /RSE/
005736  040  123  105            .ASCII  / SE/
005741  105  113  040            .ASCII  /EK /
005744  106  101  111            .ASCII  /FAI/
005747  114  125  122            .ASCII  /LUR/
005752  105  000                 .ASCII  /E/<00>
005754  045  116  045  P.ADR:     .ASCII  /%N%/
005757  101  124  111            .ASCII  /ATI/
005762  115  105  040            .ASCII  /ME /
005765  105  130  120            .ASCII  /EXP/
005770  111  122  105            .ASCII  /IRE/
005773  104  000  000            .ASCII  /D/<00><00>
005776  045  116  045  P.ADS:     .ASCII  /%N%/
006001  101  106  101            .ASCII  /AFA/
006004  124  101  114            .ASCII  /TAL/
006007  040  105  122            .ASCII  / ER/
006012  122  117  122            .ASCII  /ROR/
006015  000                      .ASCII  <00>
006016  101  110  105  P.ADT:     .ASCII  /AHE/
006021  101  104  040            .ASCII  /AD /
006024  101  040  117            .ASCII  /A O/
006027  106  106  123            .ASCII  /FFS/
006032  105  124  040            .ASCII  /ET /
006035  126  101  114            .ASCII  /VAL/
006040  125  105  040            .ASCII  /UE /
006043  075  040  045            .ASCII  /= %/
006046  117  063  000            .ASCII  /03/<00>
```

```
006051  000                             .ASCII  <00>
006052  101     110     105     P.ADU:  .ASCII  /AHE/
006055  101     104     040             .ASCII  /AD /
006060  102     040     117             .ASCII  /B O/
006063  106     106     123             .ASCII  /FFS/
006066  105     124     040             .ASCII  /ET /
006071  126     101     114             .ASCII  /VAL/
006074  125     105     040             .ASCII  /UE /
006077  075     040     045             .ASCII  /= %/
006102  117     063     000             .ASCII  /03/<00>
006105  000                             .ASCII  <00>
006106  101     110     105     P.ADV:  .ASCII  /AHE/
006111  101     104     040             .ASCII  /AD /
006114  103     040     117             .ASCII  /C O/
006117  106     106     123             .ASCII  /FFS/
006122  105     124     040             .ASCII  /ET /
006125  126     101     114             .ASCII  /VAL/
006130  125     105     040             .ASCII  /UE /
006133  075     040     045             .ASCII  /= %/
006136  117     063     000             .ASCII  /03/<00>
006141  000                             .ASCII  <00>
006142  101     110     105     P.ADW:  .ASCII  /AHE/
006145  101     104     040             .ASCII  /AD /
006150  104     040     117             .ASCII  /D O/
006153  106     106     123             .ASCII  /FFS/
006156  105     124     040             .ASCII  /ET /
006161  126     101     114             .ASCII  /VAL/
006164  125     105     040             .ASCII  /UE /
006167  075     040     045             .ASCII  /= %/
006172  117     063     000             .ASCII  /03/<00>
006175  000                             .ASCII  <00>
006176  116     045     101     P.ADX:  .ASCII  /N%A/
006201  103     125     122             .ASCII  /CUR/
006204  122     105     116             .ASCII  /REN/
006207  124     040     124             .ASCII  /T T/
006212  122     101     103             .ASCII  /RAC/
006215  113     040     075             .ASCII  /K =/
006220  040     045     117             .ASCII  / %O/
006223  064     045     101             .ASCII  /4%A/
006226  040     116     125             .ASCII  / NU/
006231  115     102     105             .ASCII  /MBE/
006234  122     040     117             .ASCII  /R O/
006237  106     040     123             .ASCII  /F S/
006242  105     105     113             .ASCII  /EEK/
006245  123     040     075             .ASCII  /S =/
006250  040     045     117             .ASCII  / %O/
006253  065     000     000             .ASCII  /5/<00><00>
006256  045     116     045     P.ADY:  .ASCII  /%N%/
006261  101     123     124             .ASCII  /AST/
006264  101     122     124             .ASCII  /ART/
006267  111     116     107             .ASCII  /ING/
006272  040     124     122             .ASCII  / TR/
006275  101     103     113             .ASCII  /ACK/
006300  040     075     040             .ASCII  / = /
006303  045     117     064             .ASCII  /%04/
006306  045     101     040             .ASCII  /%A /
006311  103     125     122             .ASCII  /CUR/
```

```
006314  122  105  116                    .ASCII  /REN/
006317  124  040  124                    .ASCII  /T T/
006322  122  101  103                    .ASCII  /RAC/
006325  113  040  075                    .ASCII  /K =/
006330  040  045  117                    .ASCII  / %O/
006333  064  045  101                    .ASCII  /4%A/
006336  040  105  116                    .ASCII  / EN/
006341  104  111  116                    .ASCII  /DIN/
006344  107  040  124                    .ASCII  /G T/
006347  122  101  103                    .ASCII  /RAC/
006352  113  040  075                    .ASCII  /K =/
006355  040  045  117                    .ASCII  / %O/
006360  064  000                        .ASCII  /4/<00>
006362  045  116  045          P.ADZ:    .ASCII  /%N%/
006365  101  105  116                    .ASCII  /AEN/
006370  104  040  120                    .ASCII  /D P/
006373  101  103  113                    .ASCII  /ACK/
006376  105  124  040                    .ASCII  /ET /
006401  123  124  101                    .ASCII  /STA/
006404  124  125  123                    .ASCII  /TUS/
006407  040  105  122                    .ASCII  / ER/
006412  122  117  122                    .ASCII  /ROR/
006415  040  075  040                    .ASCII  / = /
006420  045  117  066                    .ASCII  /%O6/
006423  045  101  040                    .ASCII  /%A /
006426  122  105  106                    .ASCII  /REF/
006431  040  043  040                    .ASCII  / # /
006434  075  040  045                    .ASCII  /= %/
006437  117  062  000                    .ASCII  /O2/<00>
006442  045  116  045          P.AEA:    .ASCII  /%N%/
006445  101  102  125                    .ASCII  /ABU/
006450  123  040  101                    .ASCII  /S A/
006453  104  104  122                    .ASCII  /DDR/
006456  105  123  123                    .ASCII  /ESS/
006461  111  116  107                    .ASCII  /ING/
006464  040  104  101                    .ASCII  / DA/
006467  124  101  040                    .ASCII  /TA /
006472  124  105  123                    .ASCII  /TES/
006475  124  040  105                    .ASCII  /T E/
006500  122  122  117                    .ASCII  /RRO/
006503  122  000  000                    .ASCII  /R/<00><00>
006506  045  116  045          P.AEB:    .ASCII  /%N%/
006511  101  106  101                    .ASCII  /AFA/
006514  111  114  111                    .ASCII  /ILI/
006517  116  107  040                    .ASCII  /NG /
006522  101  104  104                    .ASCII  /ADD/
006525  122  040  075                    .ASCII  /R =/
006530  040  045  117                    .ASCII  / %O/
006533  066  045  101                    .ASCII  /6%A/
006536  040  104  101                    .ASCII  / DA/
006541  124  101  040                    .ASCII  /TA /
006544  075  040  045                    .ASCII  /= %/
006547  117  066  045                    .ASCII  /O6%/
006552  116  000                        .ASCII  /N/<00>
006554  045  116  045          P.AEC:    .ASCII  /%N%/
006557  101  102  114                    .ASCII  /ABL/
006562  117  103  113                    .ASCII  /OCK/
```

```
006565  040  104  101                    .ASCII  / DA/
006570  124  101  040                    .ASCII  /TA /
006573  124  122  101                    .ASCII  /TRA/
006576  116  123  106                    .ASCII  /NSF/
006601  105  122  040                    .ASCII  /ER /
006604  106  101  111                    .ASCII  /FAI/
006607  114  105  104                    .ASCII  /LED/
006612  000  000                        .ASCII  <00><00>
006614  122  103  062          P.AED:    .ASCII  /RC2/
006617  065  040  123                    .ASCII  /5 S/
006622  105  105  113                    .ASCII  /EEK/
006625  040  106  101                    .ASCII  / FA/
006630  111  114  125                    .ASCII  /ILU/
006633  122  105  000                    .ASCII  /RE/<00>
006636  045  116  045          P.AEE:    .ASCII  /%N%/
006641  101  102  114                    .ASCII  /ABL/
006644  117  103  113                    .ASCII  /OCK/
006647  040  114  105                    .ASCII  / LE/
006652  116  107  124                    .ASCII  /NGT/
006655  110  040  075                    .ASCII  /H =/
006660  040  045  117                    .ASCII  / %O/
006663  066  045  116                    .ASCII  /6%N/
006666  000  000                        .ASCII  <00><00>
006670  045  116  045          P.AEF:    .ASCII  /%N%/
006673  101  110  105                    .ASCII  /AHE/
006676  101  104  040                    .ASCII  /AD /
006701  123  127  111                    .ASCII  /SWI/
006704  124  103  110                    .ASCII  /TCH/
006707  040  106  101                    .ASCII  / FA/
006712  111  114  105                    .ASCII  /ILE/
006715  104  000  000                    .ASCII  /D/<00><00>
006720  106  101  111          P.AEG:    .ASCII  /FAI/
006723  114  111  116                    .ASCII  /LIN/
006726  107  040  123                    .ASCII  /G S/
006731  125  122  106                    .ASCII  /URF/
006734  101  103  105                    .ASCII  /ACE/
006737  040  075  040                    .ASCII  / = /
006742  045  117  063                    .ASCII  /%O3/
006745  045  101  040                    .ASCII  /%A /
006750  124  122  101                    .ASCII  /TRA/
006753  103  113  040                    .ASCII  /CK /
006756  043  040  075                    .ASCII  /# =/
006761  040  045  117                    .ASCII  / %O/
006764  066  045  116                    .ASCII  /6%N/
006767  000                            .ASCII  <00>
006770  122  105  101          P.AEH:    .ASCII  /REA/
006773  104  040  123                    .ASCII  /D S/
006776  105  103  124                    .ASCII  /ECT/
007001  117  122  040                    .ASCII  /OR /
007004  106  101  111                    .ASCII  /FAI/
007007  114  105  104                    .ASCII  /LED/
007012  000  000                        .ASCII  <00><00>
007014  101  106  101          P.AEI:    .ASCII  /AFA/
007017  111  114  111                    .ASCII  /ILI/
007022  116  107  040                    .ASCII  /NG /
007025  040  124  122                    .ASCII  / TR/
007030  101  103  113                    .ASCII  /ACK/
```

```
007033  040  043  040                    .ASCII  / # /
007036  075  040  045                    .ASCII  /= %/
007041  117  066  045                    .ASCII  /O6%/
007044  101  040  123                    .ASCII  /A S/
007047  105  103  124                    .ASCII  /ECT/
007052  117  122  040                    .ASCII  /OR /
007055  043  040  075                    .ASCII  /# =/
007060  040  045  117                    .ASCII  / %O/
007063  066  045  116                    .ASCII  /6%N/
007066  000  000                        .ASCII  <00><00>
007070  127  122  111          P.AEJ:    .ASCII  /WRI/
007073  124  105  040                    .ASCII  /TE /
007076  120  122  117                    .ASCII  /PRO/
007101  124  105  103                    .ASCII  /TEC/
007104  124  040  124                    .ASCII  /T T/
007107  105  123  124                    .ASCII  /EST/
007112  040  106  101                    .ASCII  / FA/
007115  111  114  105                    .ASCII  /ILE/
007120  104  000                        .ASCII  /D/<00>
007122  105  130  120          P.AEK:    .ASCII  /EXP/
007125  105  103  124                    .ASCII  /ECT/
007130  105  104  040                    .ASCII  /ED /
007133  123  127  040                    .ASCII  /SW /
007136  075  040  117                    .ASCII  /= O/
007141  106  106  040                    .ASCII  /FF /
007144  040  101  103                    .ASCII  / AC/
007147  124  125  101                    .ASCII  /TUA/
007152  114  040  123                    .ASCII  /L S/
007155  127  040  075                    .ASCII  /W =/
007160  040  117  116                    .ASCII  / ON/
007163  040  040  125                    .ASCII  /  U/
007166  116  111  124                    .ASCII  /NIT/
007171  040  043  040                    .ASCII  / # /
007174  075  040  045                    .ASCII  /= %/
007177  104  063  045                    .ASCII  /D3%/
007202  116  000                        .ASCII  /N/<00>
007204  101  105  130          P.AEL:    .ASCII  /AEX/
007207  120  105  103                    .ASCII  /PEC/
007212  124  105  104                    .ASCII  /TED/
007215  040  123  127                    .ASCII  / SW/
007220  040  075  040                    .ASCII  / = /
007223  117  116  040                    .ASCII  /ON /
007226  040  101  103                    .ASCII  / AC/
007231  124  125  101                    .ASCII  /TUA/
007234  114  040  123                    .ASCII  /L S/
007237  127  040  075                    .ASCII  /W =/
007242  040  117  106                    .ASCII  / OF/
007245  106  040  040                    .ASCII  /F  /
007250  125  116  111                    .ASCII  /UNI/
007253  124  040  043                    .ASCII  /T #/
007256  040  075  040                    .ASCII  / = /
007261  045  104  063                    .ASCII  /%D3/
007264  045  116  000                    .ASCII  /%N/<00>
007267  000                            .ASCII  <00>
007270  045  116  045          P.AEM:    .ASCII  /%N%/
007273  101  101  126                    .ASCII  /AAV/
007276  105  122  101                    .ASCII  /ERA/
```

```
007301  107  105  040                .ASCII  /GE /
007304  123  105  105                .ASCII  /SEE/
007307  113  040  124                .ASCII  /K T/
007312  111  115  105                .ASCII  /IME/
007315  040  050  155                .ASCII  / (m/
007320  163  051  040                .ASCII  /s) /
007323  075  040  045                .ASCII  /= %/
007326  117  062  045                .ASCII  /O2%/
007331  101  056  045                .ASCII  /A.%/
007334  117  062  000                .ASCII  /O2/<00>
007337  000                        .ASCII  <00>
007340  122  103  062        P.AEN:  .ASCII  /RC2/
007343  065  040  125                .ASCII  /5 U/
007346  116  111  124                .ASCII  /NIT/
007351  040  104  117                .ASCII  / DO/
007354  105  123  040                .ASCII  /ES /
007357  116  117  124                .ASCII  /NOT/
007362  040  103  117                .ASCII  / CO/
007365  115  105  040                .ASCII  /ME /
007370  117  116  114                .ASCII  /ONL/
007373  111  116  105                .ASCII  /INE/
007376  000  000                    .ASCII  <00><00>
007400  105  130  137        P.AEO:  .ASCII  /EX_/
007403  123  125  120                .ASCII  /SUP/
007406  137  120  122                .ASCII  /_PR/
007411  117  107  040                .ASCII  /OG /
007414  104  125  120                .ASCII  /DUP/
007417  040  103  117                .ASCII  / CO/
007422  115  115  101                .ASCII  /MMA/
007425  116  104  040                .ASCII  /ND /
007430  106  101  111                .ASCII  /FAI/
007433  114  125  122                .ASCII  /LUR/
007436  105  000                    .ASCII  /E/<00>
007440  123  105  116        P.AEP:  .ASCII  /SEN/
007443  104  137  104                .ASCII  /D_D/
007446  101  124  101                .ASCII  /ATA/
007451  040  104  125                .ASCII  / DU/
007454  120  040  103                .ASCII  /P C/
007457  117  115  115                .ASCII  /OMM/
007462  101  116  104                .ASCII  /AND/
007465  040  106  101                .ASCII  / FA/
007470  111  114  125                .ASCII  /ILU/
007473  122  105  000                .ASCII  /RE/<00>
007476  122  105  103        P.AEQ:  .ASCII  /REC/
007501  137  104  101                .ASCII  /_DA/
007504  124  101  040                .ASCII  /TA /
007507  104  125  120                .ASCII  /DUP/
007512  040  103  117                .ASCII  / CO/
007515  115  115  101                .ASCII  /MMA/
007520  116  104  040                .ASCII  /ND /
007523  106  101  111                .ASCII  /FAI/
007526  114  125  122                .ASCII  /LUR/
007531  105  000  000                .ASCII  /E/<00><00>
007534  045  116  045        P.AES:  .ASCII  /%N%/
007537  101  044  106                .ASCII  /A$F/
007542  124  114  105                .ASCII  /TLE/
007545  122  122  055                .ASCII  /RR-/
```

```
007550  040     125     116                     .ASCII  / UN/
007553  122     105     103                     .ASCII  /REC/
007556  117     107     116                     .ASCII  /OGN/
007561  111     132     101                     .ASCII  /IZA/
007564  102     114     105                     .ASCII  /BLE/
007567  040     105     122                     .ASCII  / ER/
007572  122     117     122                     .ASCII  /ROR/
007575  040     103     117                     .ASCII  / CO/
007600  104     105     000                     .ASCII  /DE/<00>
007603  000                                     .ASCII  <00>
007604  045     116     045             P.AET:  .ASCII  /%N%/
007607  101     044     106                     .ASCII  /A$F/
007612  124     114     105                     .ASCII  /TLE/
007615  122     122     055                     .ASCII  /RR-/
007620  040     105     116                     .ASCII  / EN/
007623  126     105     114                     .ASCII  /VEL/
007626  117     120     105                     .ASCII  /OPE/
007631  057     120     101                     .ASCII  <57>/PA/
007634  103     113     105                     .ASCII  /CKE/
007637  124     040     122                     .ASCII  /T R/
007642  105     101     104                     .ASCII  /EAD/
007645  040     050     120                     .ASCII  / (P/
007650  101     122     111                     .ASCII  /ARI/
007653  124     131     040                     .ASCII  /TY /
007656  117     122     040                     .ASCII  /OR /
007661  124     111     115                     .ASCII  /TIM/
007664  105     117     125                     .ASCII  /EOU/
007667  124     051     000                     .ASCII  /T)/<00>
007672  045     116     045             P.AEU:  .ASCII  /%N%/
007675  101     044     106                     .ASCII  /A$F/
007700  124     114     105                     .ASCII  /TLE/
007703  122     122     055                     .ASCII  /RR-/
007706  040     105     116                     .ASCII  / EN/
007711  126     105     114                     .ASCII  /VEL/
007714  117     120     105                     .ASCII  /OPE/
007717  057     120     101                     .ASCII  <57>/PA/
007722  103     113     105                     .ASCII  /CKE/
007725  124     040     127                     .ASCII  /T W/
007730  122     111     124                     .ASCII  /RIT/
007733  105     040     050                     .ASCII  /E (/
007736  120     101     122                     .ASCII  /PAR/
007741  111     124     131                     .ASCII  /ITY/
007744  040     117     122                     .ASCII  / OR/
007747  040     124     111                     .ASCII  / TI/
007752  115     105     117                     .ASCII  /MEO/
007755  125     124     051                     .ASCII  /UT)/
007760  000     000                             .ASCII  <00><00>
007762  045     116     045             P.AEV:  .ASCII  /%N%/
007765  101     044     106                     .ASCII  /A$F/
007770  124     114     105                     .ASCII  /TLE/
007773  122     122     055                     .ASCII  /RR-/
007776  040     103     117                     .ASCII  / CO/
010001  116     124     122                     .ASCII  /NTR/
010004  117     114     114                     .ASCII  /OLL/
010007  105     122     040                     .ASCII  /ER /
010012  122     117     115                     .ASCII  /ROM/
010015  040     101     116                     .ASCII  / AN/
```

```
010020  104  040  122                    .ASCII  /D R/
010023  101  115  040                    .ASCII  /AM /
010026  120  101  122                    .ASCII  /PAR/
010031  111  124  131                    .ASCII  /ITY/
010034  000  000                         .ASCII  <00><00>
010036  045  116  045          P.AEW:    .ASCII  /%N%/
010041  101  044  106                    .ASCII  /A$F/
010044  124  114  105                    .ASCII  /TLE/
010047  122  122  055                    .ASCII  /RR-/
010052  040  103  117                    .ASCII  / CO/
010055  116  124  122                    .ASCII  /NTR/
010060  117  114  114                    .ASCII  /OLL/
010063  105  122  040                    .ASCII  /ER /
010066  122  101  115                    .ASCII  /RAM/
010071  040  120  101                    .ASCII  / PA/
010074  122  111  124                    .ASCII  /RIT/
010077  131  000  000                    .ASCII  /Y/<00><00>
010102  045  116  045          P.AEX:    .ASCII  /%N%/
010105  101  044  106                    .ASCII  /A$F/
010110  124  114  105                    .ASCII  /TLE/
010113  122  122  055                    .ASCII  /RR-/
010116  040  103  117                    .ASCII  / CO/
010121  116  124  122                    .ASCII  /NTR/
010124  117  114  114                    .ASCII  /OLL/
010127  105  122  040                    .ASCII  /ER /
010132  122  117  115                    .ASCII  /ROM/
010135  040  120  101                    .ASCII  / PA/
010140  122  111  124                    .ASCII  /RIT/
010143  131  000  000                    .ASCII  /Y/<00><00>
010146  045  116  045          P.AEY:    .ASCII  /%N%/
010151  101  044  106                    .ASCII  /A$F/
010154  124  114  105                    .ASCII  /TLE/
010157  122  122  055                    .ASCII  /RR-/
010162  040  122  111                    .ASCII  / RI/
010165  116  107  040                    .ASCII  /NG /
010170  122  105  101                    .ASCII  /REA/
010173  104  040  050                    .ASCII  /D (/
010176  120  101  122                    .ASCII  /PAR/
010201  111  124  131                    .ASCII  /ITY/
010204  040  117  122                    .ASCII  / OR/
010207  040  124  111                    .ASCII  / TI/
010212  115  105  117                    .ASCII  /MEO/
010215  125  124  051                    .ASCII  /UT)/
010220  000  000                         .ASCII  <00><00>
010222  045  116  045          P.AEZ:    .ASCII  /%N%/
010225  101  044  106                    .ASCII  /A$F/
010230  124  114  105                    .ASCII  /TLE/
010233  122  122  055                    .ASCII  /RR-/
010236  040  122  111                    .ASCII  / RI/
010241  116  107  040                    .ASCII  /NG /
010244  127  122  111                    .ASCII  /WRI/
010247  124  105  040                    .ASCII  /TE /
010252  050  120  101                    .ASCII  /(PA/
010255  122  111  124                    .ASCII  /RIT/
010260  131  040  117                    .ASCII  /Y O/
010263  122  040  124                    .ASCII  /R T/
010266  111  115  105                    .ASCII  /IME/
```

```
010271  117  125  124                    .ASCII  /OUT/
010274  051  000                         .ASCII  /)/<00>
010276  045  116  045          P.AFA:    .ASCII  /%N%/
010301  101  044  106                    .ASCII  /A$F/
010304  124  114  105                    .ASCII  /TLE/
010307  122  122  055                    .ASCII  /RR-/
010312  040  111  116                    .ASCII  / IN/
010315  124  105  122                    .ASCII  /TER/
010320  122  125  120                    .ASCII  /RUP/
010323  124  040  115                    .ASCII  /T M/
010326  101  123  124                    .ASCII  /AST/
010331  105  122  000                    .ASCII  /ER/<00>
010334  045  116  045          P.AFB:    .ASCII  /%N%/
010337  101  044  106                    .ASCII  /A$F/
010342  124  114  105                    .ASCII  /TLE/
010345  122  122  055                    .ASCII  /RR-/
010350  040  110  117                    .ASCII  / HO/
010353  123  124  040                    .ASCII  /ST /
010356  101  103  103                    .ASCII  /ACC/
010361  105  123  123                    .ASCII  /ESS/
010364  040  124  111                    .ASCII  / TI/
010367  115  105  117                    .ASCII  /MEO/
010372  125  124  000                    .ASCII  /UT/<00>
010375  000                            .ASCII  <00>
010376  045  116  045          P.AFC:    .ASCII  /%N%/
010401  101  044  106                    .ASCII  /A$F/
010404  124  114  105                    .ASCII  /TLE/
010407  122  122  055                    .ASCII  /RR-/
010412  040  103  122                    .ASCII  / CR/
010415  105  104  111                    .ASCII  /EDI/
010420  124  040  114                    .ASCII  /T L/
010423  111  115  111                    .ASCII  /IMI/
010426  124  040  105                    .ASCII  /T E/
010431  130  103  105                    .ASCII  /XCE/
010434  105  104  105                    .ASCII  /EDE/
010437  104  000  000                    .ASCII  /D/<00><00>
010442  045  116  045          P.AFD:    .ASCII  /%N%/
010445  101  044  106                    .ASCII  /A$F/
010450  124  114  105                    .ASCII  /TLE/
010453  122  122  055                    .ASCII  /RR-/
010456  040  102  125                    .ASCII  / BU/
010461  123  040  115                    .ASCII  /S M/
010464  101  123  124                    .ASCII  /AST/
010467  105  122  040                    .ASCII  /ER /
010472  105  122  122                    .ASCII  /ERR/
010475  117  122  000                    .ASCII  /OR/<00>
010500  045  116  045          P.AFE:    .ASCII  /%N%/
010503  101  044  106                    .ASCII  /A$F/
010506  124  114  105                    .ASCII  /TLE/
010511  122  122  055                    .ASCII  /RR-/
010514  040  104  111                    .ASCII  / DI/
010517  101  107  116                    .ASCII  /AGN/
010522  117  123  124                    .ASCII  /OST/
010525  111  103  040                    .ASCII  /IC /
010530  103  117  116                    .ASCII  /CON/
010533  124  122  117                    .ASCII  /TRO/
010536  114  114  105                    .ASCII  /LLE/
```

```
010541  122  040  106                .ASCII  /R F/
010544  101  124  101                .ASCII  /ATA/
010547  114  040  105                .ASCII  /L E/
010552  122  122  117                .ASCII  /RRO/
010555  122  000  000                .ASCII  /R/<00><00>
010560  045  116  045      P.AFF:    .ASCII  /%N%/
010563  101  044  106                .ASCII  /A$F/
010566  124  114  105                .ASCII  /TLE/
010571  122  122  055                .ASCII  /RR-/
010574  040  111  116                .ASCII  / IN/
010577  123  124  122                .ASCII  /STR/
010602  125  103  124                .ASCII  /UCT/
010605  111  117  116                .ASCII  /ION/
010610  040  114  117                .ASCII  / LO/
010613  117  120  040                .ASCII  /OP /
010616  124  111  115                .ASCII  /TIM/
010621  105  117  125                .ASCII  /EOU/
010624  124  000                     .ASCII  /T/<00>
010626  045  116  045      P.AFG:    .ASCII  /%N%/
010631  101  044  106                .ASCII  /A$F/
010634  124  114  105                .ASCII  /TLE/
010637  122  122  055                .ASCII  /RR-/
010642  040  111  116                .ASCII  / IN/
010645  126  101  114                .ASCII  /VAL/
010650  111  104  040                .ASCII  /ID /
010653  103  117  116                .ASCII  /CON/
010656  116  105  103                .ASCII  /NEC/
010661  124  111  117                .ASCII  /TIO/
010664  116  040  111                .ASCII  /N I/
010667  104  105  116                .ASCII  /DEN/
010672  124  111  106                .ASCII  /TIF/
010675  111  105  122                .ASCII  /IER/
010700  000  000                     .ASCII  <00><00>
010702  045  116  045      P.AFH:    .ASCII  /%N%/
010705  101  044  106                .ASCII  /A$F/
010710  124  114  105                .ASCII  /TLE/
010713  122  122  055                .ASCII  /RR-/
010716  040  111  116                .ASCII  / IN/
010721  124  105  122                .ASCII  /TER/
010724  122  125  120                .ASCII  /RUP/
010727  124  040  127                .ASCII  /T W/
010732  122  111  124                .ASCII  /RIT/
010735  105  000  000                .ASCII  /E/<00><00>
010740  045  116  045      P.AFI:    .ASCII  /%N%/
010743  101  044  106                .ASCII  /A$F/
010746  124  114  105                .ASCII  /TLE/
010751  122  122  055                .ASCII  /RR-/
010754  040  115  101                .ASCII  / MA/
010757  111  116  124                .ASCII  /INT/
010762  105  116  101                .ASCII  /ENA/
010765  116  103  105                .ASCII  /NCE/
010770  040  122  105                .ASCII  / RE/
010773  101  104  057                .ASCII  /AD/<57>
010776  127  122  111                .ASCII  /WRI/
011001  124  105  040                .ASCII  /TE /
011004  111  116  126                .ASCII  /INV/
011007  101  114  111                .ASCII  /ALI/
```

```
011012  104  040  122                 .ASCII  /D R/
011015  105  107  111                 .ASCII  /EGI/
011020  117  116  040                 .ASCII  /ON /
011023  111  104  105                 .ASCII  /IDE/
011026  116  124  111                 .ASCII  /NTI/
011031  106  111  105                 .ASCII  /FIE/
011034  122  000                      .ASCII  /R/<00>
011036  045  116  045       P.AFJ:    .ASCII  /%N%/
011041  101  044  106                 .ASCII  /A$F/
011044  124  114  105                 .ASCII  /TLE/
011047  122  122  055                 .ASCII  /RR-/
011052  040  115  101                 .ASCII  / MA/
011055  111  116  124                 .ASCII  /INT/
011060  105  116  101                 .ASCII  /ENA/
011063  116  103  105                 .ASCII  /NCE/
011066  040  127  122                 .ASCII  / WR/
011071  111  124  105                 .ASCII  /ITE/
011074  040  114  117                 .ASCII  / LO/
011077  101  104  040                 .ASCII  /AD /
011102  124  117  040                 .ASCII  /TO /
011105  116  117  116                 .ASCII  /NON/
011110  055  114  117                 .ASCII  /-LO/
011113  101  104  101                 .ASCII  /ADA/
011116  102  114  105                 .ASCII  /BLE/
011121  040  103  117                 .ASCII  / CO/
011124  116  124  122                 .ASCII  /NTR/
011127  117  114  114                 .ASCII  /OLL/
011132  105  122  000                 .ASCII  /ER/<00>
011135  000                           .ASCII  <00>
011136  045  116  045       P.AFK:    .ASCII  /%N%/
011141  101  044  106                 .ASCII  /A$F/
011144  124  114  105                 .ASCII  /TLE/
011147  122  122  055                 .ASCII  /RR-/
011152  040  103  117                 .ASCII  / CO/
011155  116  124  122                 .ASCII  /NTR/
011160  117  114  114                 .ASCII  /OLL/
011163  105  122  040                 .ASCII  /ER /
011166  122  101  115                 .ASCII  /RAM/
011171  040  105  122                 .ASCII  / ER/
011174  122  117  122                 .ASCII  /ROR/
011177  040  050  116                 .ASCII  / (N/
011202  117  116  055                 .ASCII  /ON-/
011205  120  101  122                 .ASCII  /PAR/
011210  111  124  131                 .ASCII  /ITY/
011213  051  000  000                 .ASCII  /)/<00><00>
011216  045  116  045       P.AFL:    .ASCII  /%N%/
011221  101  044  106                 .ASCII  /A$F/
011224  124  114  105                 .ASCII  /TLE/
011227  122  122  055                 .ASCII  /RR-/
011232  040  111  116                 .ASCII  / IN/
011235  111  124  040                 .ASCII  /IT /
011240  123  105  121                 .ASCII  /SEQ/
011243  125  105  116                 .ASCII  /UEN/
011246  103  105  040                 .ASCII  /CE /
011251  105  122  122                 .ASCII  /ERR/
011254  117  122  000                 .ASCII  /OR/<00>
011257  000                           .ASCII  <00>
```

```
011260  045  116  045        P.AFM:  .ASCII  /%N%/
011263  101  044  106                .ASCII  /A$F/
011266  124  114  105                .ASCII  /TLE/
011271  122  122  055                .ASCII  /RR-/
011274  040  110  111                .ASCII  / HI/
011277  107  110  040                .ASCII  /GH /
011302  114  105  126                .ASCII  /LEV/
011305  105  114  040                .ASCII  /EL /
011310  120  122  117                .ASCII  /PRO/
011313  124  117  103                .ASCII  /TOC/
011316  117  114  040                .ASCII  /OL /
011321  111  116  103                .ASCII  /INC/
011324  117  115  120                .ASCII  /OMP/
011327  101  124  111                .ASCII  /ATI/
011332  102  111  114                .ASCII  /BIL/
011335  111  124  131                .ASCII  /ITY/
011340  040  105  122                .ASCII  / ER/
011343  122  117  122                .ASCII  /ROR/
011346  000  000                    .ASCII  <00><00>
011350  045  116  045        P.AFN:  .ASCII  /%N%/
011353  101  044  106                .ASCII  /A$F/
011356  124  114  105                .ASCII  /TLE/
011361  122  122  055                .ASCII  /RR-/
011364  040  120  125                .ASCII  / PU/
011367  122  107  105                .ASCII  /RGE/
011372  057  120  117                .ASCII  <57>/PO/
011375  114  114  040                .ASCII  /LL /
011400  110  101  122                .ASCII  /HAR/
011403  104  127  101                .ASCII  /DWA/
011406  122  105  040                .ASCII  /RE /
011411  106  101  111                .ASCII  /FAI/
011414  114  125  122                .ASCII  /LUR/
011417  105  040  000                .ASCII  /E /<00>
011422  045  116  045        P.AFO:  .ASCII  /%N%/
011425  101  044  106                .ASCII  /A$F/
011430  124  114  105                .ASCII  /TLE/
011433  122  122  055                .ASCII  /RR-/
011436  040  115  101                .ASCII  / MA/
011441  120  120  111                .ASCII  /PPI/
011444  116  107  040                .ASCII  /NG /
011447  122  105  107                .ASCII  /REG/
011452  111  123  124                .ASCII  /IST/
011455  105  122  040                .ASCII  /ER /
011460  122  105  101                .ASCII  /REA/
011463  104  040  105                .ASCII  /D E/
011466  122  122  117                .ASCII  /RRO/
011471  122  040  050                .ASCII  /R (/
011474  120  101  122                .ASCII  /PAR/
011477  111  124  131                .ASCII  /ITY/
011502  040  117  122                .ASCII  / OR/
011505  040  124  111                .ASCII  / TI/
011510  115  105  117                .ASCII  /MEO/
011513  125  124  051                .ASCII  /UT)/
011516  000  000                    .ASCII  <00><00>
011520  007534'            P.AER:  .WORD   P.AES
011522  007604'                    .WORD   P.AET
011524  007672'                    .WORD   P.AEU
```

```
011526  007762'                                 .WORD   P.AEV
011530  010036'                                 .WORD   P.AEW
011532  010102'                                 .WORD   P.AEX
011534  010146'                                 .WORD   P.AEY
011536  010222'                                 .WORD   P.AEZ
011540  010276'                                 .WORD   P.AFA
011542  010334'                                 .WORD   P.AFB
011544  010376'                                 .WORD   P.AFC
011546  010442'                                 .WORD   P.AFD
011550  010500'                                 .WORD   P.AFE
011552  010560'                                 .WORD   P.AFF
011554  010626'                                 .WORD   P.AFG
011556  010702'                                 .WORD   P.AFH
011560  010740'                                 .WORD   P.AFI
011562  011036'                                 .WORD   P.AFJ
011564  011136'                                 .WORD   P.AFK
011566  011216'                                 .WORD   P.AFL
011570  011260'                                 .WORD   P.AFM
011572  011350'                                 .WORD   P.AFN
011574  011422'                                 .WORD   P.AFO
011576      045     116     045         P.AFQ:  .ASCII  /%N%/
011601      101     044     106                 .ASCII  /A$F/
011604      124     114     105                 .ASCII  /TLE/
011607      122     122     055                 .ASCII  /RR-/
011612      040     122     105                 .ASCII  / RE/
011615      123     120     117                 .ASCII  /SPO/
011620      116     123     105                 .ASCII  /NSE/
011623      040     123     124                 .ASCII  / ST/
011626      101     124     125                 .ASCII  /ATU/
011631      123     040     105                 .ASCII  /S E/
011634      122     122     117                 .ASCII  /RRO/
011637      122     072     045                 .ASCII  /R:%/
011642      123     000                         .ASCII  /S/<00>
011644      045     116     045         P.AFR:  .ASCII  /%N%/
011647      101     044     106                 .ASCII  /A$F/
011652      124     114     105                 .ASCII  /TLE/
011655      122     122     055                 .ASCII  /RR-/
011660      040     123     125                 .ASCII  / SU/
011663      120     105     122                 .ASCII  /PER/
011666      126     111     123                 .ASCII  /VIS/
011671      117     122     040                 .ASCII  /OR /
011674      123     105     122                 .ASCII  /SER/
011677      126     111     103                 .ASCII  /VIC/
011702      105     040     103                 .ASCII  /E C/
011705      101     114     114                 .ASCII  /ALL/
011710      040     106     101                 .ASCII  / FA/
011713      111     114     105                 .ASCII  /ILE/
011716      104     000                         .ASCII  /D/<00>
011720      045     116     045         P.AFS:  .ASCII  /%N%/
011723      101     044     106                 .ASCII  /A$F/
011726      124     114     105                 .ASCII  /TLE/
011731      122     122     055                 .ASCII  /RR-/
011734      040     120     117                 .ASCII  / PO/
011737      122     124     057                 .ASCII  /RT/<57>
011742      103     117     116                 .ASCII  /CON/
011745      124     122     117                 .ASCII  /TRO/
011750      114     114     105                 .ASCII  /LLE/
```

```
011753      122     040     124                .ASCII  /R T/
011756      111     115     105                .ASCII  /IME/
011761      117     125     124                .ASCII  /OUT/
011764      040     105     122                .ASCII  / ER/
011767      122     117     122                .ASCII  /ROR/
011772      000     000                        .ASCII  <00><00>
011774      045     116     045        P.AFT:  .ASCII  /%N%/
011777      101     044     106                .ASCII  /A$F/
012002      124     114     105                .ASCII  /TLE/
012005      122     122     055                .ASCII  /RR-/
012010      040     125     116                .ASCII  / UN/
012013      113     116     117                .ASCII  /KNO/
012016      127     116     040                .ASCII  /WN /
012021      122     105     124                .ASCII  /RET/
012024      125     122     116                .ASCII  /URN/
012027      040     123     124                .ASCII  / ST/
012032      101     124     125                .ASCII  /ATU/
012035      123     040     103                .ASCII  /S C/
012040      117     104     105                .ASCII  /ODE/
012043      000                                .ASCII  <00>
012044  011576'                        P.AFP:  .WORD   P.AFQ
012046  011644'                                .WORD   P.AFR
012050  011720'                                .WORD   P.AFS
012052  011774'                                .WORD   P.AFT
012054      045     116     045        P.AFV:  .ASCII  /%N%/
012057      101     044     106                .ASCII  /A$F/
012062      124     114     105                .ASCII  /TLE/
012065      122     122     055                .ASCII  /RR-/
012070      040     126     101                .ASCII  / VA/
012073      130     040     122                .ASCII  /X R/
012076      105     101     104                .ASCII  /EAD/
012101      057     127     122                .ASCII  <57>/WR/
012104      111     124     105                .ASCII  /ITE/
012107      040     105     122                .ASCII  / ER/
012112      122     117     122                .ASCII  /ROR/
012115      040     117     116                .ASCII  / ON/
012120      040     111     116                .ASCII  / IN/
012123      124     105     122                .ASCII  /TER/
012126      122     125     120                .ASCII  /RUP/
012131      124     000     000                .ASCII  /T/<00><00>
012134      045     116     045        P.AFW:  .ASCII  /%N%/
012137      101     044     106                .ASCII  /A$F/
012142      124     114     105                .ASCII  /TLE/
012145      122     122     055                .ASCII  /RR-/
012150      040     111     116                .ASCII  / IN/
012153      103     117     116                .ASCII  /CON/
012156      123     111     123                .ASCII  /SIS/
012161      124     105     116                .ASCII  /TEN/
012164      103     131     040                .ASCII  /CY /
012167      101     124     040                .ASCII  /AT /
012172      125     056     102                .ASCII  /U.B/
012175      106     111     114                .ASCII  /FIL/
012200      000     000                        .ASCII  <00><00>
012202      045     116     045        P.AFX:  .ASCII  /%N%/
012205      101     044     106                .ASCII  /A$F/
012210      124     114     105                .ASCII  /TLE/
012213      122     122     055                .ASCII  /RR-/
```

```
012216  040  111  116                .ASCII  / IN/
012221  103  117  116                .ASCII  /CON/
012224  123  111  123                .ASCII  /SIS/
012227  124  105  116                .ASCII  /TEN/
012232  103  131  040                .ASCII  /CY /
012235  101  124  040                .ASCII  /AT /
012240  125  056  102                .ASCII  /U.B/
012243  115  124  131                .ASCII  /MTY/
012246  000  000                     .ASCII  <00><00>
012250  045  116  045          P.AFY: .ASCII  /%N%/
012253  101  044  106                .ASCII  /A$F/
012256  124  114  105                .ASCII  /TLE/
012261  122  122  055                .ASCII  /RR-/
012264  040  111  116                .ASCII  / IN/
012267  103  117  116                .ASCII  /CON/
012272  123  111  123                .ASCII  /SIS/
012275  124  105  116                .ASCII  /TEN/
012300  103  131  040                .ASCII  /CY /
012303  101  124  040                .ASCII  /AT /
012306  125  056  101                .ASCII  /U.A/
012311  114  117  103                .ASCII  /LOC/
012314  000  000                     .ASCII  <00><00>
012316  045  116  045          P.AFZ: .ASCII  /%N%/
012321  101  044  106                .ASCII  /A$F/
012324  124  114  105                .ASCII  /TLE/
012327  122  122  055                .ASCII  /RR-/
012332  040  111  116                .ASCII  / IN/
012335  103  117  116                .ASCII  /CON/
012340  123  111  123                .ASCII  /SIS/
012343  124  105  116                .ASCII  /TEN/
012346  103  131  040                .ASCII  /CY /
012351  101  124  040                .ASCII  /AT /
012354  123  105  122                .ASCII  /SER/
012357  126  117  040                .ASCII  /VO /
012362  105  116  124                .ASCII  /ENT/
012365  122  131  040                .ASCII  /RY /
012370  050  120  111                .ASCII  /(PI/
012373  120  040  123                .ASCII  /P S/
012376  105  124  051                .ASCII  /ET)/
012401  000                         .ASCII  <00>
012402  045  116  045          P.AGA: .ASCII  /%N%/
012405  101  044  106                .ASCII  /A$F/
012410  124  114  105                .ASCII  /TLE/
012413  122  122  055                .ASCII  /RR-/
012416  040  111  116                .ASCII  / IN/
012421  103  117  116                .ASCII  /CON/
012424  123  111  123                .ASCII  /SIS/
012427  124  105  116                .ASCII  /TEN/
012432  103  131  040                .ASCII  /CY /
012435  101  124  040                .ASCII  /AT /
012440  123  105  122                .ASCII  /SER/
012443  126  117  040                .ASCII  /VO /
012446  105  116  124                .ASCII  /ENT/
012451  122  131  040                .ASCII  /RY /
012454  050  105  122                .ASCII  /(ER/
012457  122  040  123                .ASCII  /R S/
012462  105  124  051                .ASCII  /ET)/
```

```
012465  000                             .ASCII  <00>
012466  045  116  045          P.AGB:    .ASCII  /%N%/
012471  101  044  106                    .ASCII  /A$F/
012474  124  114  105                    .ASCII  /TLE/
012477  122  122  055                    .ASCII  /RR-/
012502  040  111  116                    .ASCII  / IN/
012505  103  117  116                    .ASCII  /CON/
012510  123  111  123                    .ASCII  /SIS/
012513  124  105  116                    .ASCII  /TEN/
012516  103  131  040                    .ASCII  /CY /
012521  101  124  040                    .ASCII  /AT /
012524  125  056  123                    .ASCII  /U.S/
012527  105  116  104                    .ASCII  /END/
012532  000  000                        .ASCII  <00><00>
012534  045  116  045          P.AGC:    .ASCII  /%N%/
012537  101  044  106                    .ASCII  /A$F/
012542  124  114  105                    .ASCII  /TLE/
012545  122  122  055                    .ASCII  /RR-/
012550  040  111  116                    .ASCII  / IN/
012553  103  117  116                    .ASCII  /CON/
012556  123  111  123                    .ASCII  /SIS/
012561  124  105  116                    .ASCII  /TEN/
012564  103  131  040                    .ASCII  /CY /
012567  101  124  040                    .ASCII  /AT /
012572  125  056  122                    .ASCII  /U.R/
012575  105  103  126                    .ASCII  /ECV/
012600  000  000                        .ASCII  <00><00>
012602  045  116  045          P.AGD:    .ASCII  /%N%/
012605  101  044  106                    .ASCII  /A$F/
012610  124  114  105                    .ASCII  /TLE/
012613  122  122  055                    .ASCII  /RR-/
012616  040  111  116                    .ASCII  / IN/
012621  103  117  116                    .ASCII  /CON/
012624  123  111  123                    .ASCII  /SIS/
012627  124  105  116                    .ASCII  /TEN/
012632  103  131  040                    .ASCII  /CY /
012635  101  124  040                    .ASCII  /AT /
012640  125  056  101                    .ASCII  /U.A/
012643  124  124  116                    .ASCII  /TTN/
012646  000  000                        .ASCII  <00><00>
012650  045  116  045          P.AGE:    .ASCII  /%N%/
012653  101  044  106                    .ASCII  /A$F/
012656  124  114  105                    .ASCII  /TLE/
012661  122  122  055                    .ASCII  /RR-/
012664  040  111  116                    .ASCII  / IN/
012667  103  117  116                    .ASCII  /CON/
012672  123  111  123                    .ASCII  /SIS/
012675  124  105  116                    .ASCII  /TEN/
012700  103  131  040                    .ASCII  /CY /
012703  101  124  040                    .ASCII  /AT /
012706  125  056  117                    .ASCII  /U.O/
012711  116  114  116                    .ASCII  /NLN/
012714  000  000                        .ASCII  <00><00>
012716  045  116  045          P.AGF:    .ASCII  /%N%/
012721  101  044  106                    .ASCII  /A$F/
012724  124  114  105                    .ASCII  /TLE/
012727  122  122  055                    .ASCII  /RR-/
```

```
012732  040  111  114                .ASCII  / IL/
012735  114  105  107                .ASCII  /LEG/
012740  101  114  040                .ASCII  /AL /
012743  104  040  122                .ASCII  /D R/
012746  105  121  125                .ASCII  /EQU/
012751  105  123  124                .ASCII  /EST/
012754  040  050  125                .ASCII  / (U/
012757  056  121  104                .ASCII  /.QD/
012762  122  121  051                .ASCII  /RQ)/
012765  000                        .ASCII  <00>
012766  045  116  045     P.AGG:     .ASCII  /%N%/
012771  101  044  106                .ASCII  /A$F/
012774  124  114  105                .ASCII  /TLE/
012777  122  122  055                .ASCII  /RR-/
013002  040  106  105                .ASCII  / FE/
013005  116  103  105                .ASCII  /NCE/
013010  055  120  117                .ASCII  /-PO/
013013  123  124  040                .ASCII  /ST /
013016  105  122  122                .ASCII  /ERR/
013021  117  122  040                .ASCII  /OR /
013024  101  124  040                .ASCII  /AT /
013027  120  122  117                .ASCII  /PRO/
013032  124  101  102                .ASCII  /TAB/
013035  000                        .ASCII  <00>
013036  045  116  045     P.AGH:     .ASCII  /%N%/
013041  101  044  106                .ASCII  /A$F/
013044  124  114  105                .ASCII  /TLE/
013047  122  122  055                .ASCII  /RR-/
013052  040  102  101                .ASCII  / BA/
013055  104  040  120                .ASCII  /D P/
013060  101  103  113                .ASCII  /ACK/
013063  105  124  040                .ASCII  /ET /
013066  104  105  121                .ASCII  /DEQ/
013071  125  105  125                .ASCII  /UEU/
013074  105  104  040                .ASCII  /ED /
013077  101  124  040                .ASCII  /AT /
013102  125  056  104                .ASCII  /U.D/
013105  117  116  105                .ASCII  /ONE/
013110  000  000                    .ASCII  <00><00>
013112  045  116  045     P.AGI:     .ASCII  /%N%/
013115  101  044  106                .ASCII  /A$F/
013120  124  114  105                .ASCII  /TLE/
013123  122  122  055                .ASCII  /RR-/
013126  040  125  116                .ASCII  / UN/
013131  105  130  120                .ASCII  /EXP/
013134  114  101  111                .ASCII  /LAI/
013137  116  105  104                .ASCII  /NED/
013142  040  104  055                .ASCII  / D-/
013145  120  122  117                .ASCII  /PRO/
013150  103  040  123                .ASCII  /C S/
013153  125  123  120                .ASCII  /USP/
013156  105  116  123                .ASCII  /ENS/
013161  111  117  116                .ASCII  /ION/
013164  040  050  125                .ASCII  / (U/
013167  056  056  124                .ASCII  /..T/
013172  104  123  051                .ASCII  /DS)/
013175  000                        .ASCII  <00>
```

```
013176  045  116  045        P.AGJ:  .ASCII  /%N%/
013201  101  044  106                .ASCII  /A$F/
013204  124  114  105                .ASCII  /TLE/
013207  122  122  055                .ASCII  /RR-/
013212  040  104  125                .ASCII  / DU/
013215  120  040  120                .ASCII  /P P/
013220  101  103  113                .ASCII  /ACK/
013223  105  124  040                .ASCII  /ET /
013226  104  055  121                .ASCII  /D-Q/
013231  040  106  101                .ASCII  / FA/
013234  111  114  105                .ASCII  /ILE/
013237  104  040  050                .ASCII  /D (/
013242  130  106  103                .ASCII  /XFC/
013245  040  063  064                .ASCII  / 34/
013250  057  063  065                .ASCII  <57>/35/
013253  051  000  000                .ASCII  )/<00><00>
013256  045  116  045        P.AGK:  .ASCII  /%N%/
013261  101  044  106                .ASCII  /A$F/
013264  124  114  105                .ASCII  /TLE/
013267  122  122  055                .ASCII  /RR-/
013272  040  111  116                .ASCII  / IN/
013275  103  117  116                .ASCII  /CON/
013300  123  111  123                .ASCII  /SIS/
013303  124  105  116                .ASCII  /TEN/
013306  103  131  040                .ASCII  /CY /
013311  101  124  040                .ASCII  /AT /
013314  125  056  110                .ASCII  /U.H/
013317  124  123  124                .ASCII  /TST/
013322  000  000                    .ASCII  <00><00>
013324  045  116  045        P.AGL:  .ASCII  /%N%/
013327  101  044  106                .ASCII  /A$F/
013332  124  114  105                .ASCII  /TLE/
013335  122  122  055                .ASCII  /RR-/
013340  040  111  116                .ASCII  / IN/
013343  103  117  116                .ASCII  /CON/
013346  123  111  123                .ASCII  /SIS/
013351  124  105  116                .ASCII  /TEN/
013354  103  131  040                .ASCII  /CY /
013357  101  124  040                .ASCII  /AT /
013362  125  056  123                .ASCII  /U.S/
013365  105  113  117                .ASCII  /EKO/
013370  000  000                    .ASCII  <00><00>
013372  045  116  045        P.AGM:  .ASCII  /%N%/
013375  101  044  106                .ASCII  /A$F/
013400  124  114  105                .ASCII  /TLE/
013403  122  122  055                .ASCII  /RR-/
013406  040  111  116                .ASCII  / IN/
013411  103  117  116                .ASCII  /CON/
013414  123  111  123                .ASCII  /SIS/
013417  124  105  116                .ASCII  /TEN/
013422  103  131  040                .ASCII  /CY /
013425  101  124  040                .ASCII  /AT /
013430  125  056  103                .ASCII  /U.C/
013433  113  123  126                .ASCII  /KSV/
013436  000  000                    .ASCII  <00><00>
013440  045  116  045        P.AGN:  .ASCII  /%N%/
013443  101  044  106                .ASCII  /A$F/
```

```
013446  124  114  105                .ASCII  /TLE/
013451  122  122  055                .ASCII  /RR-/
013454  040  104  056                .ASCII  / D./
013457  117  120  103                .ASCII  /OPC/
013462  104  040  106                .ASCII  /D F/
013465  117  125  116                .ASCII  /OUN/
013470  104  040  111                .ASCII  /D I/
013473  114  114  105                .ASCII  /LLE/
013476  107  101  114                .ASCII  /GAL/
013501  040  117  120                .ASCII  / OP/
013504  103  117  104                .ASCII  /COD/
013507  105  000  000                .ASCII  /E/<00><00>
013512  045  116  045          P.AGO:.ASCII  /%N%/
013515  101  044  106                .ASCII  /A$F/
013520  124  114  105                .ASCII  /TLE/
013523  122  122  055                .ASCII  /RR-/
013526  040  104  056                .ASCII  / D./
013531  103  123  106                .ASCII  /CSF/
013534  040  106  117                .ASCII  / FO/
013537  125  116  104                .ASCII  /UND/
013542  040  111  114                .ASCII  / IL/
013545  114  105  107                .ASCII  /LEG/
013550  101  114  040                .ASCII  /AL /
013553  117  120  103                .ASCII  /OPC/
013556  117  104  105                .ASCII  /ODE/
013561  000                         .ASCII  <00>
013562  045  116  045          P.AGP:.ASCII  /%N%/
013565  101  044  106                .ASCII  /A$F/
013570  124  114  105                .ASCII  /TLE/
013573  122  122  055                .ASCII  /RR-/
013576  040  125  116                .ASCII  / UN/
013601  113  116  117                .ASCII  /KNO/
013604  127  116  040                .ASCII  /WN /
013607  102  101  104                .ASCII  /BAD/
013612  040  104  122                .ASCII  / DR/
013615  111  126  105                .ASCII  /IVE/
013620  040  123  124                .ASCII  / ST/
013623  101  124  125                .ASCII  /ATU/
013626  123  040  101                .ASCII  /S A/
013631  124  040  104                .ASCII  /T D/
013634  056  104  123                .ASCII  /.DS/
013637  124  123  000                .ASCII  /TS/<00>
013642  045  116  045          P.AGQ:.ASCII  /%N%/
013645  101  044  106                .ASCII  /A$F/
013650  124  114  105                .ASCII  /TLE/
013653  122  122  055                .ASCII  /RR-/
013656  040  111  114                .ASCII  / IL/
013661  114  105  107                .ASCII  /LEG/
013664  101  114  040                .ASCII  /AL /
013667  130  106  103                .ASCII  /XFC/
013672  040  105  130                .ASCII  / EX/
013675  105  103  125                .ASCII  /ECU/
013700  124  105  104                .ASCII  /TED/
013703  040  102  131                .ASCII  / BY/
013706  040  104  115                .ASCII  / DM/
013711  000                         .ASCII  <00>
013712  045  116  045          P.AGR:.ASCII  /%N%/
```

```
013715  101  044  106                    .ASCII  /A$F/
013720  124  114  105                    .ASCII  /TLE/
013723  122  122  055                    .ASCII  /RR-/
013726  040  104  040                    .ASCII  / D /
013731  120  111  103                    .ASCII  /PIC/
013734  113  105  104                    .ASCII  /KED/
013737  040  125  120                    .ASCII  / UP/
013742  040  101  040                    .ASCII  / A /
013745  132  105  122                    .ASCII  /ZER/
013750  117  040  123                    .ASCII  /O S/
013753  103  102  056                    .ASCII  /CB./
013756  104  102  000                    .ASCII  /DB/<00>
013761  000                             .ASCII  <00>
013762  045  116  045          P.AGS:    .ASCII  /%N%/
013765  101  044  106                    .ASCII  /A$F/
013770  124  114  105                    .ASCII  /TLE/
013773  122  122  055                    .ASCII  /RR-/
013776  040  111  116                    .ASCII  / IN/
014001  103  117  116                    .ASCII  /CON/
014004  123  111  123                    .ASCII  /SIS/
014007  124  105  116                    .ASCII  /TEN/
014012  103  131  040                    .ASCII  /CY /
014015  101  124  040                    .ASCII  /AT /
014020  104  040  111                    .ASCII  /D I/
014023  104  114  105                    .ASCII  /DLE/
014026  040  114  117                    .ASCII  / LO/
014031  117  120  000                    .ASCII  /OP/<00>
014034  045  116  045          P.AGT:    .ASCII  /%N%/
014037  101  044  106                    .ASCII  /A$F/
014042  124  114  105                    .ASCII  /TLE/
014045  122  122  055                    .ASCII  /RR-/
014050  040  104  115                    .ASCII  / DM/
014053  040  127  117                    .ASCII  / WO/
014056  122  104  040                    .ASCII  /RD /
014061  103  117  125                    .ASCII  /COU/
014064  116  124  040                    .ASCII  /NT /
014067  105  122  122                    .ASCII  /ERR/
014072  117  122  040                    .ASCII  /OR /
014075  117  116  040                    .ASCII  /ON /
014100  110  117  123                    .ASCII  /HOS/
014103  124  040  104                    .ASCII  /T D/
014106  115  101  057                    .ASCII  /MA/<57>
014111  123  105  116                    .ASCII  /SEN/
014114  104  057  122                    .ASCII  /D/<57>/R/
014117  105  103  126                    .ASCII  /ECV/
014122  000  000                        .ASCII  <00><00>
014124  045  116  045          P.AGU:    .ASCII  /%N%/
014127  101  044  106                    .ASCII  /A$F/
014132  124  114  105                    .ASCII  /TLE/
014135  122  122  055                    .ASCII  /RR-/
014140  040  125  116                    .ASCII  / UN/
014143  113  116  117                    .ASCII  /KNO/
014146  127  116  040                    .ASCII  /WN /
014151  104  111  123                    .ASCII  /DIS/
014154  120  114  101                    .ASCII  /PLA/
014157  131  040  106                    .ASCII  /Y F/
014162  101  125  114                    .ASCII  /AUL/
```

```
014165  124  040  103                    .ASCII  /T C/
014170  117  104  105                    .ASCII  /ODE/
014173  040  101  124                    .ASCII  / AT/
014176  040  104  056                    .ASCII  / D./
014201  104  106  114                    .ASCII  /DFL/
014204  124  000                        .ASCII  /T/<00>
014206  045  116  045          P.AGV:    .ASCII  /%N%/
014211  101  044  106                    .ASCII  /A$F/
014214  124  114  105                    .ASCII  /TLE/
014217  122  122  055                    .ASCII  /RR-/
014222  040  104  122                    .ASCII  / DR/
014225  111  126  105                    .ASCII  /IVE/
014230  040  116  117                    .ASCII  / NO/
014233  124  040  106                    .ASCII  /T F/
014236  101  125  114                    .ASCII  /AUL/
014241  124  111  116                    .ASCII  /TIN/
014244  107  040  111                    .ASCII  /G I/
014247  116  040  120                    .ASCII  /N P/
014252  056  117  106                    .ASCII  /.OF/
014255  114  116  040                    .ASCII  /LN /
014260  123  124  101                    .ASCII  /STA/
014263  124  105  000                    .ASCII  /TE/<00>
014266  045  116  045          P.AGW:    .ASCII  /%N%/
014271  101  044  106                    .ASCII  /A$F/
014274  124  114  105                    .ASCII  /TLE/
014277  122  122  055                    .ASCII  /RR-/
014302  040  125  040                    .ASCII  / U /
014305  120  117  127                    .ASCII  /POW/
014310  105  122  040                    .ASCII  /ER /
014313  125  120  040                    .ASCII  /UP /
014316  104  111  101                    .ASCII  /DIA/
014321  107  116  117                    .ASCII  /GNO/
014324  123  124  111                    .ASCII  /STI/
014327  103  123  040                    .ASCII  /CS /
014332  106  101  111                    .ASCII  /FAI/
014335  114  105  104                    .ASCII  /LED/
014340  000  000                        .ASCII  <00><00>
014342  045  116  045          P.AGX:    .ASCII  /%N%/
014345  101  044  106                    .ASCII  /A$F/
014350  124  114  105                    .ASCII  /TLE/
014353  122  122  055                    .ASCII  /RR-/
014356  040  104  040                    .ASCII  / D /
014361  120  117  127                    .ASCII  /POW/
014364  105  122  040                    .ASCII  /ER /
014367  125  120  040                    .ASCII  /UP /
014372  104  111  101                    .ASCII  /DIA/
014375  107  116  117                    .ASCII  /GNO/
014400  123  124  111                    .ASCII  /STI/
014403  103  123  040                    .ASCII  /CS /
014406  106  101  111                    .ASCII  /FAI/
014411  114  105  104                    .ASCII  /LED/
014414  000  000                        .ASCII  <00><00>
014416  045  116  045          P.AGY:    .ASCII  /%N%/
014421  101  044  106                    .ASCII  /A$F/
014424  124  114  105                    .ASCII  /TLE/
014427  122  122  055                    .ASCII  /RR-/
014432  040  101  104                    .ASCII  / AD/
```

```
014435  101  120  124                .ASCII  /APT/
014440  105  122  040                .ASCII  /ER /
014443  103  101  122                .ASCII  /CAR/
014446  104  040  106                .ASCII  /D F/
014451  101  111  114                .ASCII  /AIL/
014454  125  122  105                .ASCII  /URE/
014457  000                        .ASCII  <00>
014460  045  116  045        P.AGZ:  .ASCII  /%N%/
014463  101  044  106                .ASCII  /A$F/
014466  124  114  105                .ASCII  /TLE/
014471  122  122  055                .ASCII  /RR-/
014474  040  105  103                .ASCII  / EC/
014477  056  124  115                .ASCII  /.TM/
014502  122  040  124                .ASCII  /R T/
014505  111  115  105                .ASCII  /IME/
014510  104  040  117                .ASCII  /D O/
014513  125  124  000                .ASCII  /UT/<00>
014516  045  116  045        P.AHA:  .ASCII  /%N%/
014521  101  044  106                .ASCII  /A$F/
014524  124  114  105                .ASCII  /TLE/
014527  122  122  055                .ASCII  /RR-/
014532  040  125  056                .ASCII  / U./
014535  123  105  116                .ASCII  /SEN/
014540  104  057  125                .ASCII  /D/<57>/U/
014543  056  122  105                .ASCII  /.RE/
014546  103  126  040                .ASCII  /CV /
014551  122  111  116                .ASCII  /RIN/
014554  107  040  122                .ASCII  /G R/
014557  105  101  104                .ASCII  /EAD/
014562  040  111  116                .ASCII  / IN/
014565  103  117  116                .ASCII  /CON/
014570  123  111  123                .ASCII  /SIS/
014573  124  105  116                .ASCII  /TEN/
014576  103  131  000                .ASCII  /CY/<00>
014601  000                        .ASCII  <00>
014602  045  116  045        P.AHB:  .ASCII  /%N%/
014605  101  044  106                .ASCII  /A$F/
014610  124  114  105                .ASCII  /TLE/
014613  122  122  055                .ASCII  /RR-/
014616  040  125  116                .ASCII  / UN/
014621  113  116  117                .ASCII  /KNO/
014624  127  116  040                .ASCII  /WN /
014627  127  101  111                .ASCII  /WAI/
014632  124  122  126                .ASCII  /TRV/
014635  040  122  105                .ASCII  / RE/
014640  101  123  117                .ASCII  /ASO/
014643  116  040  101                .ASCII  /N A/
014646  124  040  104                .ASCII  /T D/
014651  056  122  126                .ASCII  /.RV/
014654  103  124  000                .ASCII  /CT/<00>
014657  000                        .ASCII  <00>
014660  045  116  045        P.AHC:  .ASCII  /%N%/
014663  101  044  106                .ASCII  /A$F/
014666  124  114  105                .ASCII  /TLE/
014671  122  122  055                .ASCII  /RR-/
014674  040  104  056                .ASCII  / D./
014677  101  122  103                .ASCII  /ARC/
```

```
014702  123  040  104                    .ASCII  /S D/
014705  111  104  040                    .ASCII  /ID /
014710  116  117  124                    .ASCII  /NOT/
014713  040  106  111                    .ASCII  / FI/
014716  116  104  040                    .ASCII  /ND /
014721  103  114  117                    .ASCII  /CLO/
014724  123  105  123                    .ASCII  /SES/
014727  124  040  125                    .ASCII  /T U/
014732  116  104  117                    .ASCII  /NDO/
014735  116  105  040                    .ASCII  /NE /
014740  132  117  116                    .ASCII  /ZON/
014743  105  000  000                    .ASCII  /E/<00><00>
014746  045  116  045          P.AHD:    .ASCII  /%N%/
014751  101  044  106                    .ASCII  /A$F/
014754  124  114  105                    .ASCII  /TLE/
014757  122  122  055                    .ASCII  /RR-/
014762  040  125  056                    .ASCII  / U./
014765  123  105  105                    .ASCII  /SEE/
014770  113  040  106                    .ASCII  /K F/
014773  117  125  116                    .ASCII  /OUN/
014776  104  040  123                    .ASCII  /D S/
015001  105  105  113                    .ASCII  /EEK/
015004  040  124  117                    .ASCII  / TO/
015007  040  111  114                    .ASCII  / IL/
015012  114  105  107                    .ASCII  /LEG/
015015  101  114  040                    .ASCII  /AL /
015020  124  122  101                    .ASCII  /TRA/
015023  103  113  000                    .ASCII  /CK/<00>
015026  045  116  045          P.AHE:    .ASCII  /%N%/
015031  101  044  106                    .ASCII  /A$F/
015034  124  114  105                    .ASCII  /TLE/
015037  122  122  055                    .ASCII  /RR-/
015042  040  125  056                    .ASCII  / U./
015045  110  124  123                    .ASCII  /HTS/
015050  124  040  111                    .ASCII  /T I/
015053  116  111  124                    .ASCII  /NIT/
015056  040  104  111                    .ASCII  / DI/
015061  101  107  040                    .ASCII  /AG /
015064  104  115  101                    .ASCII  /DMA/
015067  040  127  122                    .ASCII  / WR/
015072  111  124  105                    .ASCII  /ITE/
015075  040  106  101                    .ASCII  / FA/
015100  111  114  105                    .ASCII  /ILE/
015103  104  000  000                    .ASCII  /D/<00><00>
015106  045  116  045          P.AHF:    .ASCII  /%N%/
015111  101  044  106                    .ASCII  /A$F/
015114  124  114  105                    .ASCII  /TLE/
015117  122  122  055                    .ASCII  /RR-/
015122  040  125  056                    .ASCII  / U./
015125  110  124  123                    .ASCII  /HTS/
015130  124  040  111                    .ASCII  /T I/
015133  116  111  124                    .ASCII  /NIT/
015136  040  104  111                    .ASCII  / DI/
015141  101  107  040                    .ASCII  /AG /
015144  104  115  101                    .ASCII  /DMA/
015147  040  103  117                    .ASCII  / CO/
015152  115  120  101                    .ASCII  /MPA/
```

```
015155  122  105  040                  .ASCII  /RE /
015160  106  101  111                  .ASCII  /FAI/
015163  114  105  104                  .ASCII  /LED/
015166  000  000                       .ASCII  <00><00>
015170  045  116  045        P.AHG:    .ASCII  /%N%/
015173  101  044  106                  .ASCII  /A$F/
015176  124  114  105                  .ASCII  /TLE/
015201  122  122  055                  .ASCII  /RR-/
015204  040  125  056                  .ASCII  / U./
015207  123  131  104                  .ASCII  /SYD/
015212  122  040  106                  .ASCII  /R F/
015215  117  125  116                  .ASCII  /OUN/
015220  104  040  123                  .ASCII  /D S/
015223  123  056  104                  .ASCII  /S.D/
015226  105  122  040                  .ASCII  /ER /
015231  123  105  124                  .ASCII  /SET/
015234  040  101  116                  .ASCII  / AN/
015237  104  040  123                  .ASCII  /D S/
015242  123  056  123                  .ASCII  /S.S/
015245  120  116  040                  .ASCII  /PN /
015250  116  117  124                  .ASCII  /NOT/
015253  040  123  105                  .ASCII  / SE/
015256  124  000                       .ASCII  /T/<00>
015260  045  116  045        P.AHH:    .ASCII  /%N%/
015263  101  044  106                  .ASCII  /A$F/
015266  124  114  105                  .ASCII  /TLE/
015271  122  122  055                  .ASCII  /RR-/
015274  040  115  101                  .ASCII  / MA/
015277  123  124  105                  .ASCII  /STE/
015302  122  040  104                  .ASCII  /R D/
015305  122  111  126                  .ASCII  /RIV/
015310  105  123  040                  .ASCII  /ES /
015313  101  103  114                  .ASCII  /ACL/
015316  117  040  101                  .ASCII  /O A/
015321  123  123  105                  .ASCII  /SSE/
015324  122  124  105                  .ASCII  /RTE/
015327  104  000  000                  .ASCII  /D/<00><00>
015332  012054'             P.AFU:    .WORD   P.AFV
015334  012134'                       .WORD   P.AFW
015336  012202'                       .WORD   P.AFX
015340  012250'                       .WORD   P.AFY
015342  012316'                       .WORD   P.AFZ
015344  012402'                       .WORD   P.AGA
015346  012466'                       .WORD   P.AGB
015350  012534'                       .WORD   P.AGC
015352  012602'                       .WORD   P.AGD
015354  012650'                       .WORD   P.AGE
015356  012716'                       .WORD   P.AGF
015360  012766'                       .WORD   P.AGG
015362  013036'                       .WORD   P.AGH
015364  013112'                       .WORD   P.AGI
015366  013176'                       .WORD   P.AGJ
015370  013256'                       .WORD   P.AGK
015372  013324'                       .WORD   P.AGL
015374  013372'                       .WORD   P.AGM
015376  013440'                       .WORD   P.AGN
015400  013512'                       .WORD   P.AGO
```

```
015402  013562'                                 .WORD   P.AGP
015404  013642'                                 .WORD   P.AGQ
015406  013712'                                 .WORD   P.AGR
015410  013762'                                 .WORD   P.AGS
015412  014034'                                 .WORD   P.AGT
015414  014124'                                 .WORD   P.AGU
015416  014206'                                 .WORD   P.AGV
015420  014266'                                 .WORD   P.AGW
015422  014342'                                 .WORD   P.AGX
015424  014416'                                 .WORD   P.AGY
015426  014460'                                 .WORD   P.AGZ
015430  014516'                                 .WORD   P.AHA
015432  014602'                                 .WORD   P.AHB
015434  014660'                                 .WORD   P.AHC
015436  014746'                                 .WORD   P.AHD
015440  015026'                                 .WORD   P.AHE
015442  015106'                                 .WORD   P.AHF
015444  015170'                                 .WORD   P.AHG
015446  015260'                                 .WORD   P.AHH
015450      045     101     040         P.AHJ:  .ASCII  /%A /
015453      123     125     103                 .ASCII  /SUC/
015456      103     105     123                 .ASCII  /CES/
015461      123     106     125                 .ASCII  /SFU/
015464      114     045     116                 .ASCII  /L%N/
015467      000                                 .ASCII  <00>
015470      045     101     111         P.AHK:  .ASCII  /%AI/
015473      116     126     101                 .ASCII  /NVA/
015476      114     111     104                 .ASCII  /LID/
015501      040     103     117                 .ASCII  / CO/
015504      115     115     101                 .ASCII  /MMA/
015507      116     104     045                 .ASCII  /ND%/
015512      116     000                         .ASCII  /N/<00>
015514      045     101     116         P.AHL:  .ASCII  /%AN/
015517      117     040     122                 .ASCII  /O R/
015522      105     107     111                 .ASCII  /EGI/
015525      117     116     040                 .ASCII  /ON /
015530      101     126     101                 .ASCII  /AVA/
015533      111     114     101                 .ASCII  /ILA/
015536      102     114     105                 .ASCII  /BLE/
015541      045     116     000                 .ASCII  /%N/<00>
015544      045     101     116         P.AHM:  .ASCII  /%AN/
015547      117     040     122                 .ASCII  /O R/
015552      105     107     111                 .ASCII  /EGI/
015555      117     116     040                 .ASCII  /ON /
015560      123     125     111                 .ASCII  /SUI/
015563      124     101     102                 .ASCII  /TAB/
015566      114     105     045                 .ASCII  /LE%/
015571      116     000     000                 .ASCII  /N/<00><00>
015574      045     101     120         P.AHN:  .ASCII  /%AP/
015577      122     117     107                 .ASCII  /ROG/
015602      122     101     115                 .ASCII  /RAM/
015605      040     116     117                 .ASCII  / NO/
015610      124     040     113                 .ASCII  /T K/
015613      116     117     127                 .ASCII  /NOW/
015616      116     045     116                 .ASCII  /N%N/
015621      000                                 .ASCII  <00>
015622      045     101     114         P.AHO:  .ASCII  /%AL/
```

```
015625      117     101     104                .ASCII  /OAD/
015630      040     106     101                .ASCII  / FA/
015633      111     114     125                .ASCII  /ILU/
015636      122     105     045                .ASCII  /RE%/
015641      116     000     000                .ASCII  /N/<00><00>
015644      045     101     123        P.AHP:  .ASCII  /%AS/
015647      124     101     116                .ASCII  /TAN/
015652      104     101     114                .ASCII  /DAL/
015655      117     116     105                .ASCII  /ONE/
015660      045     116     000                .ASCII  /%N/<00>
015663      000                                .ASCII  <00>
015664   015450'                       P.AHI:  .WORD   P.AHJ
015666   015470'                               .WORD   P.AHK
015670   015514'                               .WORD   P.AHL
015672   015544'                               .WORD   P.AHM
015674   015574'                               .WORD   P.AHN
015676   015622'                               .WORD   P.AHO
015700   015644'                               .WORD   P.AHP
015702      045     101     123        P.AHR:  .ASCII  /%AS/
015705      125     103     103                .ASCII  /UCC/
015710      105     123     123                .ASCII  /ESS/
015713      045     116     000                .ASCII  /%N/<00>
015716      045     101     111        P.AHS:  .ASCII  /%AI/
015721      116     126     101                .ASCII  /NVA/
015724      114     111     104                .ASCII  /LID/
015727      040     103     117                .ASCII  / CO/
015732      115     115     101                .ASCII  /MMA/
015735      116     104     045                .ASCII  /ND%/
015740      116     000                        .ASCII  /N/<00>
015742      045     101     103        P.AHT:  .ASCII  /%AC/
015745      117     115     115                .ASCII  /OMM/
015750      101     116     104                .ASCII  /AND/
015753      040     101     102                .ASCII  / AB/
015756      117     122     124                .ASCII  /ORT/
015761      105     104     045                .ASCII  /ED%/
015764      116     000                        .ASCII  /N/<00>
015766      045     101     125        P.AHU:  .ASCII  /%AU/
015771      116     111     124                .ASCII  /NIT/
015774      055     117     106                .ASCII  /-OF/
015777      106     114     111                .ASCII  /FLI/
016002      116     105     045                .ASCII  /NE%/
016005      116     000     000                .ASCII  /N/<00><00>
016010      045     101     125        P.AHV:  .ASCII  /%AU/
016013      116     111     124                .ASCII  /NIT/
016016      055     101     126                .ASCII  /-AV/
016021      101     111     114                .ASCII  /AIL/
016024      101     102     114                .ASCII  /ABL/
016027      105     045     116                .ASCII  /E%N/
016032      000     000                        .ASCII  <00><00>
016034      045     101     115        P.AHW:  .ASCII  /%AM/
016037      105     104     111                .ASCII  /EDI/
016042      101     040     106                .ASCII  /A F/
016045      117     122     115                .ASCII  /ORM/
016050      101     124     040                .ASCII  /AT /
016053      105     122     122                .ASCII  /ERR/
016056      117     122     045                .ASCII  /OR%/
016061      116     000     000                .ASCII  /N/<00><00>
```

```
016064  045  101  127        P.AHX:  .ASCII  /%AW/
016067  122  111  124                .ASCII  /RIT/
016072  105  040  120                .ASCII  /E P/
016075  122  117  124                .ASCII  /ROT/
016100  105  103  124                .ASCII  /ECT/
016103  105  104  045                .ASCII  /ED%/
016106  116  000                    .ASCII  /N/<00>
016110  045  101  103        P.AHY:  .ASCII  /%AC/
016113  117  115  120                .ASCII  /OMP/
016116  101  122  105                .ASCII  /ARE/
016121  040  105  122                .ASCII  / ER/
016124  122  117  122                .ASCII  /ROR/
016127  045  116  000                .ASCII  /%N/<00>
016132  045  101  104        P.AHZ:  .ASCII  /%AD/
016135  101  124  101                .ASCII  /ATA/
016140  040  105  122                .ASCII  / ER/
016143  122  117  122                .ASCII  /ROR/
016146  045  116  000                .ASCII  /%N/<00>
016151  000                        .ASCII  <00>
016152  045  101  110        P.AIA:  .ASCII  /%AH/
016155  117  123  124                .ASCII  /OST/
016160  040  102  125                .ASCII  / BU/
016163  106  106  105                .ASCII  /FFE/
016166  122  040  101                .ASCII  /R A/
016171  103  103  105                .ASCII  /CCE/
016174  123  123  040                .ASCII  /SS /
016177  105  122  122                .ASCII  /ERR/
016202  117  122  045                .ASCII  /OR%/
016205  116  000  000                .ASCII  /N/<00><00>
016210  045  101  103        P.AIB:  .ASCII  /%AC/
016213  117  116  124                .ASCII  /ONT/
016216  122  117  114                .ASCII  /ROL/
016221  114  105  122                .ASCII  /LER/
016224  040  105  122                .ASCII  / ER/
016227  122  117  122                .ASCII  /ROR/
016232  045  116  000                .ASCII  /%N/<00>
016235  000                        .ASCII  <00>
016236  045  101  104        P.AIC:  .ASCII  /%AD/
016241  122  111  126                .ASCII  /RIV/
016244  105  040  105                .ASCII  /E E/
016247  122  122  117                .ASCII  /RRO/
016252  122  045  116                .ASCII  /R%N/
016255  000                        .ASCII  <00>
016256  045  101  115        P.AID:  .ASCII  /%AM/
016261  105  123  123                .ASCII  /ESS/
016264  101  107  105                .ASCII  /AGE/
016267  040  106  122                .ASCII  / FR/
016272  117  115  040                .ASCII  /OM /
016275  101  116  040                .ASCII  /AN /
016300  111  116  124                .ASCII  /INT/
016303  105  122  116                .ASCII  /ERN/
016306  101  114  040                .ASCII  /AL /
016311  104  111  101                .ASCII  /DIA/
016314  107  116  117                .ASCII  /GNO/
016317  123  124  111                .ASCII  /STI/
016322  103  045  116                .ASCII  /C%N/
016325  000                        .ASCII  <00>
```

```
016326  015702'                 P.AHQ:  .WORD   P.AHR
016330  015716'                         .WORD   P.AHS
016332  015742'                         .WORD   P.AHT
016334  015766'                         .WORD   P.AHU
016336  016010'                         .WORD   P.AHV
016340  016034'                         .WORD   P.AHW
016342  016064'                         .WORD   P.AHX
016344  016110'                         .WORD   P.AHY
016346  016132'                         .WORD   P.AHZ
016350  016152'                         .WORD   P.AIA
016352  016210'                         .WORD   P.AIB
016354  016236'                         .WORD   P.AIC
016356  016256'                         .WORD   P.AID


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
002024                          CLK.HERTZ::
                                        .BLKW   1
002026                          CLK.START::
                                        .BLKW   1
002030                          UNIT::  .BLKW   1
002032                          LOG.UNIT::
                                        .BLKW   1
002034                          VEC.AD::.BLKB   1
                                        .EVEN
002036                          RC25.ADDR::
                                        .BLKW   1
002040                          RC25.DATA::
                                        .BLKW   2
002044                          COM.AREA::
                                        .BLKW   104
002254                          HEAD.AREA::
                                        .BLKW   1
002256                          RECEIVE.RING::
                                        .BLKW   1
002260                          SEND.RING::
                                        .BLKW   1
002262                          REC.ENVELOPE::
                                        .BLKW   1000
004262                          SND.ENVELOPE::
                                        .BLKW   540
```

```
005562                          BUF.DESCRPTR::
                                        .BLKW   1
005564                          CMD.REF::
                                        .BLKW   1
005566                          BYTE.COUNT::
                                        .BLKW   1
005570  000001                  TICKS:: .WORD   1
005572  000000                  SECONDS::
                                        .WORD   0
005574  000000                  MINUTES::
                                        .WORD   0
005576                          TIP::   .BLKW   1
005600                          DATA1:: .BLKW   1
005602                          DATA2:: .BLKW   1
005604                          DATA3:: .BLKW   1
005606                          DATA4:: .BLKW   1
005610  000000                  I.AM.NEX::
                                        .WORD   0
005612                          MSGADR::.BLKW   1
005614  003071                  END.LBN::
                                        .WORD   3071
005616                          P.MASK::.BLKB   1
005617                          B.MASK::.BLKB   1
005620                          MANU.SW::
                                        .BLKW   1
005622                          SWITCH2::
                                        .BLKW   1
005624                          RET.UNIT.FLAG::
                                        .BLKW   1
005626                          P1::    .BLKW   1
005630                          P2::    .BLKW   1
005632                          P3::    .BLKW   1
005634                          P4::    .BLKW   1
005636                          P5::    .BLKW   1
005640                          P6::    .BLKW   1
005642                          RET.STATUS::
                                        .BLKW   1
005644                          CANCEL.TIMER::
                                        .BLKW   1
005646                          CMD.SLOT::
                                        .BLKW   1
005650                          RES.SLOT::
                                        .BLKW   1
005652                          LBN::   .BLKW   1
005654                          LBN.ST::.BLKW   1
005656                          LBN.ED::.BLKW   1
005660                          LBN.SZ::.BLKW   1
005662                          FREE.MEM.ADDR::
                                        .BLKW   1
005664                          MEM.SIZE::
                                        .BLKW   1
005666                          H.SADD::.BLKW   1
005670                          H.EADD::.BLKW   1
005672                          BUF.LENGTH::
                                        .BLKW   1
005674                          NUM.RETRIES::
                                        .BLKW   1
```

```
005676  000000                          RETRIES::
                                                .WORD   0
005700  000001                          FAL.CODE::
                                                .WORD   1
005702                                  DMC.TEST::
                                                .BLKW   1
005704                                  BYT.CNT::
                                                .BLKW   1
005706                                  DM.REC::.BLKW   1
005710                                  DM.XMT::.BLKW   1
005712                                  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

        000154'                         L$ERRTBL==              ERRTYP
        000202'                         L$SW==                  L$SWLEN+2
        000166'                         L$HW==                  L$HWLEN+2
        000011'                         L$DEPO==                L$REV+1
        000166'                         DFPTBL==                L$HWLEN+2
        000202'                         SFPTBL==                L$SWLEN+2
        002054'                         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
        000432'                         QST8==                  P.AAM
        000452'                         QST9==                  P.AAN
        000470'                         QST10==                 P.AAO
        000550'                         QS10.1==                P.AAP
        000600'                         QS10.2==                P.AAQ
        000660'                         QST11==                 P.AAR
        000734'                         QST12==                 P.AAS
        000762'                         QST13==                 P.AAT
        001006'                         QST14==                 P.AAU
        001060'                         QST15==                 P.AAV
        001132'                         DBM1==                  P.AAW
        001232'                         DBM2==                  P.AAX
        001246'                         DBM3==                  P.AAY
        001260'                         DBM4==                  P.AAZ
        001274'                         DBM5==                  P.ABA
        001310'                         DBM6==                  P.ABB
        001322'                         DBM7==                  P.ABC
        001366'                         DBM8==                  P.ABD
```

```
001450'                 DBM9==          P.ABE
001532'                 DBM10==         P.ABF
001574'                 DBM11==         P.ABG
001642'                 DBM12==         P.ABH
001702'                 DBM13==         P.ABI
001736'                 DBM14==         P.ABJ
001772'                 DBM15==         P.ABK
002034'                 DBM16==         P.ABL
002100'                 DBM17==         P.ABM
002150'                 DBM18==         P.ABN
002220'                 DBM19==         P.ABO
002260'                 DBM20==         P.ABP
002330'                 DBM21==         P.ABQ
002400'                 DBM22==         P.ABR
002446'                 DBM23==         P.ABS
002530'                 DBM24==         P.ABT
002574'                 DBM25==         P.ABU
002632'                 DBM26==         P.ABV
002676'                 DBM27==         P.ABW
002736'                 DBM28==         P.ABX
003006'                 DBM29==         P.ABY
003046'                 DBM30==         P.ABZ
003104'                 DBM31==         P.ACA
003146'                 DBM32==         P.ACB
003202'                 DBM33==         P.ACC
003222'                 DBM34==         P.ACD
003244'                 DBM35==         P.ACE
003270'                 DBM36==         P.ACF
003332'                 DBM37==         P.ACG
003376'                 DBM38==         P.ACH
003436'                 DBM39==         P.ACI
003514'                 MSG.01==        P.ACJ
003546'                 ERR.01==        P.ACK
003572'                 ERR.02==        P.ACL
003640'                 FMT$C==         P.ACM
003646'                 FMT1==          P.ACN
003732'                 FMT2==          P.ACO
004012'                 FMT3==          P.ACP
004106'                 FMT4==          P.ACQ
004144'                 FMT5==          P.ACR
004224'                 FMT6==          P.ACS
004304'                 FMT$A==         P.ACT
004342'                 MSG.PWR==       P.ACU
004376'                 MSG.1==         P.ACV
004426'                 MSG.2==         P.ACW
004456'                 MSG.7==         P.ACX
004530'                 MSG.8==         P.ACY
004572'                 MSG.9==         P.ACZ
004630'                 MSG.10==        P.ADA
004676'                 MSG.11==        P.ADB
004734'                 MSG.13==        P.ADC
004756'                 MSG.14==        P.ADD
005004'                 BUFF.ERR==      P.ADE
005062'                 DMC.ERR==       P.ADF
005120'                 INI.MSG==       P.ADG
005202'                 END.MSG==       P.ADH
005254'                 BREAK==         P.ADI
```

```
005332'                 MSG.17==            P.ADJ
005400'                 MSG.18==            P.ADK
005456'                 MSG.19==            P.ADL
005514'                 MSG.20==            P.ADM
005562'                 MSG.21==            P.ADN
005626'                 MSG.28==            P.ADO
005654'                 MSG.29==            P.ADP
005714'                 MSG.30==            P.ADQ
005754'                 CTO.ERR==           P.ADR
005776'                 PFE.ERR==           P.ADS
006016'                 AHEAD.MSG==         P.ADT
006052'                 BHEAD.MSG==         P.ADU
006106'                 CHEAD.MSG==         P.ADV
006142'                 DHEAD.MSG==         P.ADW
006176'                 MSG.TK.DSP==        P.ADX
006256'                 MSG.LBN.DSP==       P.ADY
006362'                 MSG.STATUS.ERR==    P.ADZ
006442'                 MSG.BUSA.ERR==      P.AEA
006506'                 MSG.ADDR.ERR==      P.AEB
006554'                 MSG.DATA.ERR==      P.AEC
006614'                 MSG.SEEK.ERR==      P.AED
006636'                 MSG.ERR.CONT==      P.AEE
006670'                 MSG.HSWICH.ERR==    P.AEF
006720'                 MSG.SURFACE.ERR==   P.AEG
006770'                 MSG.READ.ERR==      P.AEH
007014'                 MSG.SAC.ERR==       P.AEI
007070'                 MSG.COM.WPT==       P.AEJ
007122'                 MSG.PT.ERR1==       P.AEK
007204'                 MSG.WRP.ERR2==      P.AEL
007270'                 MSG.AVE.TIME==      P.AEM
007340'                 AZT.READY.ERR==     P.AEN
007400'                 EXE.SUP.ERR==       P.AEO
007440'                 SND.DATA.ERR==      P.AEP
007476'                 RE.DATA.ERR==       P.AEQ
011520'                 PFE.STRUCT==        P.AER
012044'                 EMSG.STRUCT==       P.AFP
015332'                 RC.STRUCTURE==      P.AFU
015664'                 SDUP.STRUCT==       P.AHI
016326'                 SMSCP.STRUCT==      P.AHQ


                        PSECT SUMMARY

Psect Name              Words     Attributes
 AA$CODE                   77       RO  ,  I  ,  LCL,  REL,  CON
 $GLOB$                  1510       RO  ,  D  ,  GBL,  REL,  CON
 $PLIT$                  3704       RO  ,  D  ,  GBL,  REL,  CON


                    LIBRARY STATISTICS

                            -------- Symbols --------    Blocks
File                        Total    Loaded   Percent      Read
```

```
;  SPIDER$USERS:[LAKSHMANA.11REL.REAL]AZTECO.L16;1
;                                                  523        154        29        47



;                                          COMMAND QUALIFIERS

;       BLISS /PDP11/LIST ZRCFA1.B16/EN:NOEIS

; Size:          0 code + 5291 data words
; Run Time:         00:29.7
; Elapsed Time:     01:29.0
; Memory Used:   274 pages
; Compilation Complete
```

```
0001  MODULE ZRCFA2 (%TITLE 'CZRCFA0 RC25 FR END TEST'
0002                 IDENT = 'V01.0',
0003                 OPTLEVEL = 0,
0004                 ADDRESSING_MODE (RELATIVE)
0005                 ) =
0006  BEGIN
0007  !
0008  !<BLF/LOWERCASE_KEY>
0009  !
0010
0011  library 'AZTEC0';
0012
0013  require 'BLSMAC.REQ';
1502
1503  !
1504  !
1505  structure
1506      RC25 [O, P, S, E] =                          ! DEFINE ACCESS ALGORITHM
1507           begin                                   ! TO ALLOW FIELD REFERENCE
1508
1509           local                                   ! TO THE AZTEC
1510               RC_REG;
1511
1512           RC_REG = .(RC25 + %upval*0)<0, %bpval, 0>;
1513           RC_REG
1514           end
1515           <P, S, E>;
1516
1517  psect
1518      code = AA$CODE;
1519
1520  forward routine
1521      FIND_CLOCK : novalue,
1522      CLOCK_INIT : novalue,
1523      RC25$ERR_RPT : novalue,
1524      AZT_INIT,
1525      AZP_INIT,
1526      PRT$FRU_CALLOUT : novalue,
1527      INIT_COM_AREA,
1528      NXMI : L$ISR novalue,
1529      CLK_INT_SERV : L$ISR novalue,
1530      SET_INT_VECTOR : novalue,
1531      REC_STATUS,
1532      SET_CNTLR_CHAR,
1533      AVAILABLE,
1534      ON_LINE,
1535      READ_CMD,
1536      READ_FILL_RING : novalue,
1537      GET_UNIT_STATUS,
1538      RANDOM_NUM : novalue,
1539      GET_CMD_SLOT : novalue,
1540      GET_RES_SLOT : novalue,
1541      EXAM_DATA,
1542      DM_ADDR_SETUP : novalue,
1543      DATA_XMT_REC,
1544      WRT_PROTECT_TST : novalue,
1545      AZTEC_READY,
```

```
;        1546      DO_RETRIES : novalue,
;        1547      decode : novalue;
;        1548
;        1549  external
;        1550      ADAPTO,
;        1551      CONTRO,
;        1552      COM_AREA : blockvector [REC_ALLOCATE + SND_ALLOCATE + HDR_SIZ, 2, word],
;        1553      HEAD_AREA : ref block [4, word] field (HDR_FIELD),
;        1554      RECEIVE_RING : ref blockvector [REC_ALLOCATE, 2, word] field (DSC_FIELD),
;        1555      SEND_RING : ref blockvector [SND_ALLOCATE, 2, word] field (DSC_FIELD),
;        1556      REC_ENVELOPE : blockvector [REC_ALLOCATE, RB_SIZE + 2, word] field (ENV_FIELD),
;        1557      SND_ENVELOPE : blockvector [SND_ALLOCATE, SB_SIZE + 2, word] field (ENV_FIELD),
;        1558      BUF_DESCRPTR : word volatile,                ! BUFFER DESCRIPTOR AREA
;        1559      BYTE_COUNT : word volatile,                  ! BYTE COUNT BUFFER
;        1560      CLK_ADR : word,                              !LOCATION TO RETURN CLOCK ADDRESS
;        1561      CLK_TYPE : word,                             !TYPE OF CLOCK ON SYSTEM
;        1562                                                   !(0=NO CLOCK, -1= L-CLOCK, 1=P-CLOCK)
;        1563      CLK_CSR : word,                              !STORE CSR ADDRESS FOR CLOCK HERE
;        1564      CLK_HERTZ : word,                            ! CLOCK RATE
;        1565      CLK_START : word,                            ! STORE CLOCK START VALUE
;        1566      TICKS : word volatile,                       ! STORE NUMBERS OF CLOCK INT. OCCURED
;        1567      SECONDS : word volatile,                     ! STORE SECONDS
;        1568      MINUTES : word volatile,                     ! STORE MINUTES
;        1569      MSGADR : word volatile,                      ! STORE MESSAGE ADDRESS
;        1570      DATA1 : word,                                ! STEP 1 WRITE DATA TO AZTEC_INIT
;        1571      DATA2 : word volatile,                       ! STEP 2 WRITE DATA TO AZTEC_INIT
;        1572      DATA3 : word volatile,                       ! STEP 3 WRITE DATA TO AZTEC_INIT
;        1573      DATA4 : word volatile,                       ! STEP 4 WRITE DATA TO AZTEC_INIT
;        1574      B_MASK : byte volatile,                      ! MASK FOR WITCH STEP TO DO
;        1575                                                   ! IN AZTEC_INIT.
;        1576      LBN : word volatile,                         ! LOGICAL BLOCK NUMBER BUFFER
;        1577      LBN_ST : word volatile,                      ! START LOGICAL BLOCK NUMBER
;        1578      LBN_ED : word volatile,                      ! ENDING LOGICAL BLOCK NUMBER
;        1579      CMD_REF : word volatile,                     ! COMMAND REFERENCE
;        1580      RES_SLOT : word volatile,                    ! RECEIVING RING SLOT
;        1581      CMD_SLOT : word volatile,                    ! SENDING RING SLOT
;        1582      VEC_AD : byte,                               ! INIT INTERRUPT VECTOR
;        1583      !P_VECTOR : word volatile,                   ! INTERRUPT VECTOR
;        1584      !P_UNIT_NUMBER : word volatile,              ! UNIT NUMBER
;        1585      RET_STATUS : word volatile,                  ! RETURN STATUS
;        1586      TEMP : word volatile,
;        1587      FREE_MEM_ADDR,                               ! STARING FREE MEMORY ADDRESS
;        1588      MEM_SIZE,                                    ! FREE MEMORY SIZE
;        1589      RINGBASE,
;        1590      DRIVE_,
;        1591      DBM1,
;        1592      DBM2,
;        1593      DBM3,
;        1594      DBM4,
;        1595      DBM5,
;        1596      DBM6,
;        1597      DBM33,
;        1598      DBM34,
;        1599      DBM35,
;        1600      ERR_01,
;        1601      ERR_02,
;        1602      FMT$C,
```

```
;          1603      FRU,
;          1604      FMT2,
;          1605      FMT3,
;          1606      DMC_TEST,
;          1607      BYT_CNT,
;          1608      DM_XMT,
;          1609      DM_REC,
;          1610      H_SADD,
;          1611      H_EADD,
;          1612      BUF_LENGTH,
;          1613      MAN0_SW,
;          1614      SWITCH2,
;          1615      TIP,
;          1616      SWP_CONTINUE,
;          1617      FMT$A,
;          1618      QST15,
;          1619      QST14,
;          1620      ! RUN TIME TABLE STORAGE
;          1621      HWP_TABLE : ref block [WORD2_IN_HWP_TAB, word] field (HWP_FIELDS),
;          1622      RT_TABLE : ref block [WORD1_IN_RT_TAB, word] field (RT_FIELDS),
;          1623      RT : vector [WORD1_IN_RT_TAB, word],
;          1624      I_AM_NEX : word volatile,
;          1625      CANCEL_TIMER : word volatile,
;          1626      RETRIES,
;          1627      SWP_RETRIES,
;          1628      NUM_RETRIES,
;          1629      SWP_TRACE,
;          1630      L$UNIT,
;          1631      MECHAN,
;          1632      MSG_PWR,
;          1633      MSG_14,
;          1634      CTO_ERR,
;          1635      PFE_ERR,
;          1636      FAL_CODE,
;          1637      MSG_STATUS_ERR,
;          1638      END_LBN : word volatile,
;          1639      P_MASK : byte volatile,
;          1640      RET_UNIT_FLAG : word,
;          1641      P1 : word volatile,
;          1642      P2 : word volatile,
;          1643      P3 : word volatile,
;          1644      P4 : word volatile,
;          1645      P5 : word volatile,
;          1646      P6 : word volatile,
;          1647      QST1,
;          1648      QST2,
;          1649      QST3,
;          1650      QST4,
;          1651      QST6,
;          1652      QST7,
;          1653      QST8,
;          1654      QST9,
;          1655      QST10,
;          1656      QS10_1,
;          1657      QS10_2,
;          1658      QST11,
;          1659      RC25_ADDR : ref RC25 field (RC_REG),
```

```
;          1660      RC25_DATA : block [2, word] field (RC_REG),
;          1661      EMSG_STRUCT : vector [4],
;          1662      PFE_STRUCT : vector [23],
;          1663      RC_STRUCTURE : vector [39],
;          1664      SDUP_STRUCT : vector [7],
;          1665      SMSCP_STRUCT : vector [13],
;          1666      XMT_DATA_BUF : vector [256, word],
;          1667      RCV_DATA_BUF : vector [256, word],
;          1668      UNIT : word,
;          1669      LOG_UNIT : word;
;          1670
```

```
;         1671  %title 'MISCELLANEOUS SECTIONS'
;         1672  %sbttl 'TYPE AND DESCRIPTION'
;         1673  !; NAMES OF DEVICES SUPPORTED BY PROGRAM
;         1674  DEVTYP (%asciz'AZTEC RC25 PLATTER');
;         1675  !; TEST DESCRIPTION
;       C 1676  DESCRIPT (%asciz'RC25 FRONT END/HOST DIAGNOSTIC');%(
;       C 1677  ;++
;       C 1678  ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
;       C 1679  ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
;       C 1680  ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
;       C 1681  ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
;       C 1682  ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
;       C 1683  ; WITH THE OPERATOR.
;       C 1684  ;--
;         1685  )%
;         1686  BGNHRD;
;         1687  GPRMA (QST1, %o'0', 0, %o'00000', %o'177777', YES, 1);  !IP ADDRESS?
;         1688  GPRMA (QST2, %o'2', 0, %o'4', %o'774', YES, 1); !VECTOR?
;         1689  GPRMD (QST3, %o'4', 0, %o'177777', %o'4', %o'7', YES, 1);        !BR LEVEL
;         1690  GPRMD (QST4, %o'6', D, %o'377', %o'0', %decimal'253', NO, 1);   !UNIT NUMBER(S)
;         1691  ENDHRD;
```

```
;        1692  %sbttl 'SOFTWARE PARAMETER CODING SECTION'
;      C 1693  %(
;      C 1694  ;++
;      C 1695  ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
;      C 1696  ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
;      C 1697  ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
;      C 1698  ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
;      C 1699  ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
;      C 1700  ; WITH THE OPERATOR.
;      C 1701  ;--
;        1702  )%
;        1703  BGNSFT;
;        1704  !GPRML (QST6, %o'0', %o'177777', YES, 1);        !USE TOP SURFACE FOR SINGLE SURFACE TESTS?
;        1705  !GPRML (QST7, %o'2', %o'177777', YES, 1);        !DO YOU WISH TO LIMIT THE AREA TESTED
;        1706                                                    !IN TESTS #13 - #15?
;        1707  !XFERF (MANINT);                                  !IF NO, SKIP NEXT TWO QUESTIONS
;        1708  !GPRMD (QST8, %o'4', D, %o'1777', %o'0', %decimal'800', YES, 1);!STARTING TRACK?
;        1709  !GPRMD (QST9, %o'6', D, %o'1777', %o'0', %decimal'800', YES, 1) !ENDING TRACK?
;        1710  !$L (MANINT);                                    !LABEL THIS QUESTION
;        1711  GPRMD (QST11, %o'10', D, %o'1777', %o'0', %o'1777', YES, 1);    !NUMBER OF RETRIES FOR TEST
;        1712  GPRML (QS10_2, %o'12', 1, YES, 1);               !DO YOU WISH TO CONTINUE TESTING?
;        1713  !GPRML (QST10, %o'14', 1, YES, 1);               !DO YOU WANT TO DO THE MANUAL
;        1714                                                    !INTERVENSION TEST?
;        1715  GPRML (QS10_1, %o'16', 1, YES, 1);               !DO YOU NEED TRACE MODE?
;        1716  ENDSFT;
```

```
;        1717  %sbttl 'REPORT CODING SECTION'
;        1718  !+
;        1719  ! THE REPORT CODING SECTION CONTAINS THE
;        1720  ! 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
;        1721  !-
;        1722  BGNRPT;
;        1723  PRINTF (DBM2);                                  !'REPORT'
;        1724  return;
;        1725  ENDRPT;


                                        .TITLE  ZRCFA2 MISCELLANEOUS SECTIONS
                                        .IDENT  /V01.0/


                                        .PSECT  AA$CODE,  RO
000000                                  
000000   101     132     124    L$DVTYP::
                                        .ASCII  /AZT/
000003   105     103     040            .ASCII  /EC /
000006   122     103     062            .ASCII  /RC2/
000011   065     040     120            .ASCII  /5 P/
000014   114     101     124            .ASCII  /LAT/
000017   124     105     122            .ASCII  /TER/
000022   000     000                    .ASCII  <00><00>
000024   122     103     062    L$DESC::.ASCII  /RC2/
000027   065     040     106            .ASCII  /5 F/
000032   122     117     116            .ASCII  /RON/
000035   124     040     105            .ASCII  /T E/
000040   116     104     057            .ASCII  /ND/<57>
000043   110     117     123            .ASCII  /HOS/
000046   124     040     104            .ASCII  /T D/
000051   111     101     107            .ASCII  /IAG/
000054   116     117     123            .ASCII  /NOS/
000057   124     111     103            .ASCII  /TIC/
000062   000     000                    .ASCII  <00><00>
000064   000000C                L$HRDLN::
                                        .WORD   <<<L$NDHRD-L$HRDLN>/2>-1>
000066   000031                 GP$1::  .WORD   31
000070   000000G                        .WORD   QST1
000072   000000                         .WORD   0
000074   177777                         .WORD   -1
000076   001031                 GP$2::  .WORD   1031
000100   000000G                        .WORD   QST2
000102   000004                         .WORD   4
000104   000774                         .WORD   774
000106   002032                 GP$3::  .WORD   2032
000110   000000G                        .WORD   QST3
000112   177777                         .WORD   -1
000114   000004                         .WORD   4
000116   000007                         .WORD   7
000120   003042                 GP$4::  .WORD   3042
000122   000000G                        .WORD   QST4
000124   000377                         .WORD   377
000126   000000                         .WORD   0
000130   000375                         .WORD   375
000132                         L$NDHRD::
                                        .BLKW   1
```

```
000134  000000C                         L$SFTLN::
                                                .WORD   <<<L$NDSFT-L$SFTLN>/2>-1>
000136  004052                          GP$5::  .WORD   4052
000140  000000G                                 .WORD   QST11
000142  001777                                  .WORD   1777
000144  000000                                  .WORD   0
000146  001777                                  .WORD   1777
000150  005130                          GP$6::  .WORD   5130
000152  000000G                                 .WORD   QS10.2
000154  000001                                  .WORD   1
000156  007130                          GP$7::  .WORD   7130
000160  000000G                                 .WORD   QS10.1
000162  000001                                  .WORD   1
000164                                  L$NDSFT::
                                                .BLKW   1


                                                .GLOBL  ADAPTO, CONTRO, COM.AREA, HEAD.AREA
                                                .GLOBL  RECEIVE.RING, SEND.RING, REC.ENVELOPE
                                                .GLOBL  SND.ENVELOPE, BUF.DESCRPTR, BYTE.COUNT
                                                .GLOBL  CLK.ADR, CLK.TYPE, CLK.CSR, CLK.HERTZ
                                                .GLOBL  CLK.START, TICKS, SECONDS, MINUTES
                                                .GLOBL  MSGADR, DATA1, DATA2, DATA3, DATA4
                                                .GLOBL  B.MASK, LBN, LBN.ST, LBN.ED, CMD.REF
                                                .GLOBL  RES.SLOT, CMD.SLOT, VEC.AD, RET.STATUS
                                                .GLOBL  TEMP, FREE.MEM.ADDR, MEM.SIZE
                                                .GLOBL  RINGBASE, DRIVE., DBM1, DBM2, DBM3
                                                .GLOBL  DBM4, DBM5, DBM6, DBM33, DBM34
                                                .GLOBL  DBM35, ERR.01, ERR.02, FMT$C, FRU
                                                .GLOBL  FMT2, FMT3, DMC.TEST, BYT.CNT
                                                .GLOBL  DM.XMT, DM.REC, H.SADD, H.EADD
                                                .GLOBL  BUF.LENGTH, MANU.SW, SWITCH2, TIP
                                                .GLOBL  SWP.CONTINUE, FMT$A, QST15, QST14
                                                .GLOBL  HWP.TABLE, RT.TABLE, RT, I.AM.NEX
                                                .GLOBL  CANCEL.TIMER, RETRIES, SWP.RETRIES
                                                .GLOBL  NUM.RETRIES, SWP.TRACE, L$UNIT
                                                .GLOBL  MECHAN, MSG.PWR, MSG.14, CTO.ERR
                                                .GLOBL  PFE.ERR, FAL.CODE, MSG.STATUS.ERR
                                                .GLOBL  END.LBN, P.MASK, RET.UNIT.FLAG
                                                .GLOBL  P1, P2, P3, P4, P5, P6, QST1, QST2
                                                .GLOBL  QST3, QST4, QST6, QST7, QST8, QST9
                                                .GLOBL  QST10, QS10.1, QS10.2, QST11, RC25.ADDR
                                                .GLOBL  RC25.DATA, EMSG.STRUCT, PFE.STRUCT
                                                .GLOBL  RC.STRUCTURE, SDUP.STRUCT, SMSCP.STRUCT
                                                .GLOBL  XMT.DATA.BUF, RCV.DATA.BUF, UNIT
                                                .GLOBL  LOG.UNIT

        000066'                 L$HARD==                L$HRDLN+2
        000136'                 L$SOFT==                L$SFTLN+2


                                                .SBTTL  LRPT REPORT CODING SECTION
000000  012746  000000G          LRPT:   MOV     #DBM2,-(SP)                     ;                       1723
000004  012746  000001                  MOV     #1,-(SP)
000010  010600                          MOV     SP,R0                           ; SP,*
000012  104417                          TRAP    17
```

```
000014  022626                          CMP     (SP)+,(SP)+                     ;                                       1716
000016  000207                          RTS     PC

; Routine Size:  8 words,       Routine Base:  AA$CODE + 0166
; Maximum stack depth per invocation:  4 words



                                        .SBTTL  L$RPT REPORT CODING SECTION
000000  004767  177754          L$RPT:: JSR     PC,LRPT                         ;                                       1724
000004  104425                          TRAP    25
000006  000207                          RTS     PC

; Routine Size:  4 words,       Routine Base:  AA$CODE + 0206
; Maximum stack depth per invocation:  2 words
```

```
;       1726  %sbttl 'INITIALIZE SECTION'
;       1727  !+
;       1728  ! THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
;       1729  ! AT THE BEGINNING OF EACH PASS.
;       1730  !-
;       1731  BGNINIT;
;       1732
;       1733  local
;       1734      DELAY_MULT;                             !CONTAINS DELAY FACTOR
;       1735
;       1736  SETPRI (PRI00);                             !PRIORITY 0
;       1737
;       1738  if READEF (EF_PWR)                          !ARE WE HERE BECAUSE OF POWER FAIL?
;       1739  then
;       1740      begin
;       1741      PRINTF (MSG_PWR);                       !"POWER DELAY - WAITING"
;       1742
;       1743      incru COUNT from 0 to 60 do             ! WAIT APPROX. 60 SECONDS
;       1744          begin
;       1745          DELAY_MULT = 10000;
;       1746          DELAY (.DELAY_MULT);
;       1747          BREAK;                              ! BREAK FOR ACT
;       1748          end;
;       1749
;       1750      DOCLN;
;       1751      end;
;       1752
;       1753  ! +
;       1754  !       MAKE SURE NOT MORE THAN 16 UNITS (PLATTERS) HAVE BEEN SPECIFIED.
;       1755  !       IF THERE ARE TOO MANY, NOTIFY USER AND RETURN TO SUPERVISOR.
;       1756  ! -
;       1757
;       1758  if .L$UNIT gequ 16
;       1759  then                                        !MORE THAN 16 UNITS?
;       1760      begin
;       1761      PRINTF (ERR_01);                        !ERROR - TOO MANY UNITS
;       1762      DOCLN;                                  !RETURN TO SUPERVISOR AND CLEAN UP
;       1763      end;
;       1764
;       1765  if READEF (EF_CONTINUE) then return;       !IF CONTINUE GETS YOU HERE SKIP INIT.
;       1766
;       1767  if READEF (EF_START) or READEF (EF_RESTART) or READEF (EF_NEW)
;       1768  then
;       1769      begin
;       1770      LOG_UNIT = -1;
;       1771      NUM_RETRIES = ZERO;
;       1772      RETRIES = FALSE;
;       1773      FIND_CLOCK ();
;       1774
;       1775      if CLK_TYPE eqlu NO_CLOCK               ! IF THERE IS NO CLOCK
;       1776      then                                    ! IN THE SYSTEM TELL THE
;       1777          begin                               ! OPERATOR
;       1778          PRINTF (ERR_02);
;       1779          DOCLN;
;       1780          end
;       1781      else
;       1782          .CLK_CSR = ZERO;                    ! STOP THE CLOCK
```

```
;          1783
;          1784  !+
;          1785  !       DETERMINE THE FREE MEMORY STARTING ADDRESS AND IT SIZE
;          1786  !-
;          1787      MEMORY (FREE_MEM_ADDR);                     !FIND THE STARTING ADDR
;          1788      MEM_SIZE = ..FREE_MEM_ADDR;                 !DETERMINE THE SIZE
;          1789  !+
;          1790  !-
;          1791      end;
;          1792
;          1793  do                                              !OTHERWISE, INCREMENT LOGICAL UNIT
;          1794      begin                                       !AND CHECK FOR HIGH LIMIT.
;          1795      LOG_UNIT = .LOG_UNIT + 1;
;          1796
;          1797      if .LOG_UNIT gequ .L$UNIT then DOCLN;       !IF SO QUIT INIT AND DO CLEANUP.
;          1798
;          1799      end
;          1800  until (GPHARD (.LOG_UNIT, HWP_TABLE)) neqa 0;   !GET HARDWARE P_TABLE POINTER
;          1801
;          1802  RT_TABLE = RT [0];                             !AND LOAD RT_TABLE WITH THE
;          1803  RT_TABLE [RT_IP_ADDRESS] = .HWP_TABLE [HWP_IP_ADDRESS]; !HARDWARE P_TABLE INFO.
;          1804  RT_TABLE [RT_VECTOR] = .HWP_TABLE [HWP_VECTOR];
;          1805  RT_TABLE [RT_BR_LEVEL] = .HWP_TABLE [HWP_BR_LEVEL];
;          1806  RT_TABLE [RT_UNIT_1] = .HWP_TABLE [HWP_UNIT_NUMBER];    !PLATTER #
;          1807  RC25_ADDR = .RT_TABLE [RT_IP_ADDRESS];          !IP ADDRESS FOR THE CONTROLLER
;          1808  UNIT = .RT_TABLE [RT_UNIT_1];                  !AND PLATTER # UNDER TEST
;          1809  SETVEC (.RT_TABLE [RT_VECTOR], NXMI, PRI07);   !SET UP INTERRUPT ROUTINE
;          1810  PRINTF (DBMT, .LOG_UNIT, .RC25_ADDR, .UNIT);   !GIVE THIS INFO TO OPERATOR.
;          1811  !-
;          1812  ENDINIT;

                                .GLOBL  L$DLY


                                .SBTTL  LINIT INITIALIZE SECTION
000000 004167 000000G   LINIT:  JSR     R1,$SAVE3               ;                                1725
000004 005746                   TST     -(SP)
000006 005000                   CLR     R0                      ;                                1736
000010 104441                   TRAP    41
000012 012700 000034            MOV     #34,R0                  ;                                1738
000016 104447                   TRAP    47
000020 103033                   BHIS    6$
000022 012746 000000G           MOV     #MSG.PWR,-(SP)          ;                                1741
000026 012746 000001            MOV     #1,-(SP)
000032 010600                   MOV     SP,R0                   ; SP,*
000034 104417                   TRAP    17
000036 005002                   CLR     R2                      ; COUNT                          1743
000040 012703 023420    1$:     MOV     #23420,R3               ; *,DELAY.MULT                   1745
000044 010301                   MOV     R3,R1                   ; DELAY.MULT,$$TMP2              1746
000046 001411           2$:     BEQ     5$
000050 016700 000000G           MOV     L$DLY,R0                ; *,$$TMP1
000054 001404                   BEQ     4$
000056 005066 000004    3$:     CLR     4(SP)                   ; $$TMP
000062 005300                   DEC     R0                      ; $$TMP1
000064 001374                   BNE     3$
000066 005301           4$:     DEC     R1                      ; $$TMP2
```

```
000070  000766                          BR      2$
000072  104422                  5$:     TRAP    22
000074  005202                          INC     R2                      ; COUNT                 1743
000076  020227  000074                  CMP     R2,#74                  ; COUNT,*
000102  101756                          BLOS    1$
000104  104444                          TRAP    44                      ;                       1748
000106  022626                          CMP     (SP)+,(SP)+             ;                       1740
000110  026727  000000G 000020  6$:     CMP     L$UNIT,#20              ;                       1758
000116  103410                          BLO     7$
000120  012746  000000G                 MOV     #ERR.01,-(SP)           ;                       1761
000124  012746  000001                  MOV     #1,-(SP)
000130  010600                          MOV     SP,R0                   ; SP,*
000132  104417                          TRAP    17
000134  104444                          TRAP    44
000136  022626                          CMP     (SP)+,(SP)+             ;                       1760
000140  012700  000036          7$:     MOV     #36,R0                  ;                       1765
000144  104447                          TRAP    47
000146  103542                          BLO     13$
000150  012700  000040                  MOV     #40,R0                  ;                       1767
000154  104447                          TRAP    47
000156  103410                          BCS     8$
000160  012700  000037                  MOV     #37,R0
000164  104447                          TRAP    47
000166  103404                          BCS     8$
000170  012700  000035                  MOV     #35,R0
000174  104447                          TRAP    47
000176  103034                          BHIS    11$
000200  012767  177777  000000G  8$:     MOV     #-1,LOG.UNIT            ;                       1770
000206  005067  000000G                 CLR     NUM.RETRIES             ;                       1771
000212  005067  000000G                 CLR     RETRIES                 ;                       1772
000216  004767  000000V                 JSR     PC,FIND.CLOCK           ;                       1773
000222  005727  000000G                 TST     #CLK.TYPE               ;                       1775
000226  001011                          BNE     9$
000230  012746  000000G                 MOV     #ERR.02,-(SP)           ;                       1778
000234  012746  000001                  MOV     #1,-(SP)
000240  010600                          MOV     SP,R0                   ; SP,*
000242  104417                          TRAP    17
000244  104444                          TRAP    44
000246  022626                          CMP     (SP)+,(SP)+             ;                       1777
000250  000402                          BR      10$                     ;                       1775
000252  005077  000000G          9$:     CLR     @CLK.CSR                ;                       1782
000256  104431                  10$:    TRAP    31                      ;                       1787
000260  010067  000000G                 MOV     R0,FREE.MEM.ADDR
000264  011067  000000G                 MOV     (R0),MEM.SIZE           ; FREE.MEM.ADDR,*       1788
000270  005267  000000G          11$:    INC     LOG.UNIT                ;                       1795
000274  026767  000000G 000000G         CMP     LOG.UNIT,L$UNIT         ;                       1797
000302  103401                          BLO     12$
000304  104444                          TRAP    44
000306  016700  000000G          12$:    MOV     LOG.UNIT,R0             ;                       1800
000312  104442                          TRAP    42
000314  010067  000000G                 MOV     R0,HWP.TABLE
000320  001763                          BEQ     11$
000322  012767  000000G 000000G         MOV     #RT,RT.TABLE            ;                       1802
000330  011067  000000G                 MOV     (R0),RT                 ; HWP.TABLE,*           1803
000334  012701  000000G                 MOV     #RT,R1                  ;                       1804
000340  016061  000002  000002         MOV     2(R0),2(R1)
000346  016061  000004  000004         MOV     4(R0),4(R1)             ;                       1805
```

```
000354  016061  000006  000006                MOV     6(R0),6(R1)                     ;                                       1806
000362  011067  000000G                       MOV     (R0),RC25.ADDR                  ; RT,*                                  1807
000366  010100                                MOV     R1,R0                           ; RT,*                                  1808
000370  016067  000006  000000G               MOV     6(R0),UNIT
000376  012746  000340                        MOV     #340,-(SP)                      ;                                       1809
000402  012746  000000V                       MOV     #NXMI,-(SP)
000406  016046  000002                        MOV     2(R0),-(SP)
000412  012746  000003                        MOV     #3,-(SP)
000416  104437                                TRAP    37
000420  016716  000000G                       MOV     UNIT,(SP)                       ;                                       1810
000424  016746  000000G                       MOV     RC25.ADDR,-(SP)
000430  016746  000000G                       MOV     LOG.UNIT,-(SP)
000434  012746  000000G                       MOV     #DBM1,-(SP)
000440  012746  000004                        MOV     #4,-(SP)
000444  010600                                MOV     SP,R0                           ; SP,*
000446  104417                                TRAP    17
000450  062706  000020                        ADD     #20,SP                          ;                                       1725
000454  005726                        13$:    TST     (SP)+
000456  000207                                RTS     PC

; Routine Size:  152 words,     Routine Base:  AA$CODE + 0216
; Maximum stack depth per invocation:  15 words




                                              .SBTTL  L$INIT INITIALIZE SECTION
000000  004767  177314                L$INIT::JSR     PC,LINIT                        ;                                       1810
000004  104411                                TRAP    11
000006  000207                                RTS     PC

; Routine Size:  4 words,       Routine Base:  AA$CODE + 0676
; Maximum stack depth per invocation:  2 words
```

```
;          1813  %sbttl 'AUTODROP SECTION'
;          1814  !+
;          1815  ! THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
;          1816  ! THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
;          1817  ! SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
;          1818  ! DROPPED FROM TESTING.
;          1819  !-
;          1820  BGNAUTO;
;          1821  !if .SWP_TRACE then PRINTF (DBM3);
;          1822  return;
;          1823  ENDAUTO;


                                        .SBTTL  LAUTO AUTODROP SECTION
000000  000207                  LAUTO:  RTS     PC                              ;                       1812

; Routine Size:  1 word,        Routine Base:  AA$CODE + 0706
; Maximum stack depth per invocation:  0 words



                                        .SBTTL  L$AUTO AUTODROP SECTION
000000  004767  177772          L$AUTO::JSR     PC,LAUTO                        ;                       1822
000004  104461                          TRAP    61
000006  000207                          RTS     PC

; Routine Size:  4 words,       Routine Base:  AA$CODE + 0710
; Maximum stack depth per invocation:  2 words
```

```
;         1824  %sbttl 'CLEANUP CODING SECTION'
;         1825  !+
;         1826  ! THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
;         1827  ! AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
;         1828  !-
;         1829  BGNCLN;
;         1830  !if .SWP_TRACE then PRINTF (DBM4);
;         1831  .CLK_CSR = ZERO;                          ! TURN OFF THE CLOCK
;         1832  P1 = ZERO;                                ! CLEAR ERROR ROUTINE
;         1833  P2 = ZERO;                                ! PARAMETERS P1 - P6
;         1834  P3 = ZERO;
;         1835  P4 = ZERO;
;         1836  P5 = ZERO;
;         1837  P6 = ZERO;
;         1838  RET_STATUS = ZERO;                       ! CLEAR STATUS AND
;         1839  NUM_RETRIES = ZERO;                     ! FLAGS
;         1840  RETRIES = FALSE;
;         1841  return;
;         1842  ENDCLN;


                                        .SBTTL  LCLEAN CLEANUP CODING SECTION
000000  005077  000000G            LCLEAN: CLR     @CLK.CSR                        ;                       1831
000004  005067  000000G                    CLR     P1                              ;                       1832
000010  005067  000000G                    CLR     P2                              ;                       1833
000014  005067  000000G                    CLR     P3                              ;                       1834
000020  005067  000000G                    CLR     P4                              ;                       1835
000024  005067  000000G                    CLR     P5                              ;                       1836
000030  005067  000000G                    CLR     P6                              ;                       1837
000034  005067  000000G                    CLR     RET.STATUS                      ;                       1838
000040  005067  000000G                    CLR     NUM.RETRIES                     ;                       1839
000044  005067  000000G                    CLR     RETRIES                         ;                       1840
000050  000207                            RTS     PC                              ;                       1823

; Routine Size:  21 words,      Routine Base:  AA$CODE + 0720
; Maximum stack depth per invocation:  0 words



                                        .SBTTL  L$CLEAN CLEANUP CODING SECTION
000000  004767  177722          L$CLEAN::
                                        JSR     PC,LCLEAN                       ;                       1841
000004  104412                          TRAP    12
000006  000207                          RTS     PC

; Routine Size:  4 words,       Routine Base:  AA$CODE + 0772
; Maximum stack depth per invocation:  2 words
```

```
;        1843  %sbttl 'DROP UNIT SECTION'
;        1844  !+
;        1845  ! THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
;        1846  ! TO NO LONGER BE TESTED.
;        1847  !-
;        1848  BGNDU;
;        1849  !if .SWP_TRACE then PRINTF (DBM5);
;        1850  return;
;        1851  ENDDU;


                                               .SBTTL  LDU DROP UNIT SECTION
000000  000207                       LDU:      RTS     PC                          ;          1842

; Routine Size:  1 word,      Routine Base:  AA$CODE + 1002
; Maximum stack depth per invocation:  0 words


                                               .SBTTL  L$DU DROP UNIT SECTION
000000  004767  177772               L$DU::    JSR     PC,LDU                      ;          1850
000004  104453                                 TRAP    53
000006  000207                                 RTS     PC

; Routine Size:  4 words,     Routine Base:  AA$CODE + 1004
; Maximum stack depth per invocation:  2 words
```

```
;          1852  %sbttl 'ADD UNIT SECTION'
;          1853  !+
;          1854  ! THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
;          1855  !  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
;          1856  !  TO THE TEST CYCLE.
;          1857  !-
;          1858  BGNAU;
;          1859  !if .SWP_TRACE then PRINTF (DBM6);
;          1860  return;
;          1861  ENDAU;


                                        .SBTTL  LAU ADD UNIT SECTION
000000  000207                  LAU:    RTS     PC                      ;                       1851

; Routine Size:  1 word,        Routine Base:  AA$CODE + 1014
; Maximum stack depth per invocation:  0 words



                                        .SBTTL  L$AU ADD UNIT SECTION
000000  004767  177772          L$AU::  JSR     PC,LAU                  ;                       1860
000004  104452                          TRAP    52
000006  000207                          RTS     PC

; Routine Size:  4 words,       Routine Base:  AA$CODE + 1016
; Maximum stack depth per invocation:  2 words

;          1862  !<BLF/PAGE>
```

```
;          1863  psect
;          1864      code = ABS$CODE;
;          1865
;          1866  !++
;          1867  ! GLOBAL LOCATION "I_AM_NEX" IS SET TO TRUE WHICH INDICATES
;          1868  ! THE INITIALIZATION SEQUENCE INTERRUPT OCCURED.
;          1869  !--
;          1870
;          1871  BGNSRV (NXMI);
;          1872  I_AM_NEX = %o'177777';
;          1873  CANCEL_TIMER = %o'177777';
;          1874  ENDSRV;


                                          .SBTTL  NXMI ADD UNIT SECTION
000000                                    .PSECT  ABS$CODE,  RO

000000  012767  177777  000000G     NXMI::  MOV     #-1,I.AM.NEX                    ;                1872
000006  012767  177777  000000G             MOV     #-1,CANCEL.TIMER                ;                1873
000014  000002                               RTI                                     ;                1871

; Routine Size:  7 words,       Routine Base:  ABS$CODE + 0000
; Maximum stack depth per invocation:  0 words


;          1875
;          1876  !++
;          1877  ! THE CLOCK INTERRUPT SERVICE ROUTINE IS ENTERED AT THE CLOCK RATE
;          1878  !--
;          1879
;          1880  BGNSRV (CLK_INT_SERV);
;          1881  TICKS = .TICKS + 1;                          ! INCREMENT THE NUMBER OF TICK
;          1882
;          1883  if .TICKS eqlu .CLK_HERTZ                    ! IF TOTAL NUMBER OF TICK = 60
;          1884  then                                         ! THEN
;          1885      begin
;          1886      TICKS = 0;                               ! RESET TICK TO ZERO
;          1887      SECONDS = .SECONDS + 1;                  ! INCREMENT THEN SECOND
;          1888
;          1889      if .SECONDS eqlu 60                      ! IF SECOND = 60
;          1890      then                                     ! THEN
;          1891          begin
;          1892          SECONDS = 0;                         ! RESET SECOND TO ZERO
;          1893          MINUTES = .MINUTES + 1;              ! INCREMENT THE MINUTES
;          1894          end;
;          1895
;          1896      end;
;          1897
;          1898  ENDSRV;


                                          .SBTTL  CLK.INT.SERV ADD UNIT SECTION
000000  005267  000000G           CLK.INT.SERV::
                                  INC     TICKS                           ;                1881
000004  026767  000000G 000000G           CMP     TICKS,CLK.HERTZ         ;                1883
000012  001014                            BNE     1$
000014  005067  000000G                   CLR     TICKS                   ;                1886
```

```
000020  005267  000000G                        INC     SECONDS                         ;                       1887
000024  026727  000000G 000074                 CMP     SECONDS,#74                     ;                       1889
000032  001004                                 BNE     1$
000034  005067  000000G                        CLR     SECONDS                         ;                       1892
000040  005267  000000G                        INC     MINUTES                         ;                       1893
000044  000002                          1$:    RTI                                     ;                       1880

; Routine Size:  19 words,      Routine Base:  AB$CODE + 0016
; Maximum stack depth per invocation:  0 words
```

```
;         1899  %sbttl 'FIND CLOCK ROUTINE'
;         1900
;         1901  global routine FIND_CLOCK : novalue =
;         1902  !+
;         1903  !CHECK TO MAKE SURE THERE IS A CLOCK ON THE SYSTEM.  IF NO_CLOCK, ABORT TO
;         1904  !SUPERVISOR.
;         1905  !OTHERWISE, DETERMINE WHETHER CLOCK IS AN L OR P CLOCK, GET ITS PARAMETERS.
;         1906  !-
;         1907      begin
;         1908      CLK_TYPE = NO_CLOCK;                        !SET FLAG FOR NO CLOCK
;         1909
;         1910      if CLOCK (P, CLK_ADR)                       !IS THERE A P_CLOCK?
;         1911      then
;         1912          begin
;         1913          CLK_TYPE = P_CLOCK;                     !SET THE FLAGE FOR P_CLOCK
;         1914          CLK_CSR = ..CLK_ADR;                    !SAVE THE CSR ADDRESS
;         1915          CLK_HERTZ = .(.CLK_ADR + 6);            !GET THE CLOCK RATE
;         1916          CLK_START = %o'105';                    !SAVE THE STARTTING CLOCK VALUE
;         1917          end
;         1918      else
;         1919          begin
;         1920
;         1921          if CLOCK (L, CLK_ADR)                   !IS THERE AN L_CLOCK?
;         1922          then
;         1923              begin
;         1924              CLK_TYPE = L_CLOCK;                 !SET THE FLAG FOR L_CLOCK
;         1925              CLK_CSR = ..CLK_ADR;                !SAVE THE CSR ADDRESS
;         1926              CLK_HERTZ = .(.CLK_ADR + 6);        !GET THE CLOCK RATE
;         1927              CLK_START = %o'100';                !SAVE THE STARTING CLOCK VALUE
;         1928              end;
;         1929
;         1930          end;
;         1931
;         1932      if .CLK_TYPE nequ NO_CLOCK                  !IF  CLOCK WAS FOUND THEN
;         1933      then
;         1934          begin
;         1935          VEC_AD = .(.CLK_ADR + 4);               !GET CLOCK VECTOR ADDRESS
;         1936          SETVEC (.VEC_AD, CLK_INT_SERV, PRI05);  !SET VECTOR & SERVICE ADDR.
;         1937          end;
;         1938
;         1939      end;


                                        .SBTTL  FIND.CLOCK FIND CLOCK ROUTINE
000000  005067  000000G          FIND.CLOCK::
                                        CLR     CLK.TYPE                        ;                       1908
000004  012700  000120                  MOV     #120,R0                         ;                       1910
000010  104462                          TRAP    62
000012  103016                          BHIS    1$
000014  010067  000000G                 MOV     R0,CLK.ADR                      ; R0,*
000020  012767  000001  000000G         MOV     #1,CLK.TYPE                     ;                       1913
000026  011067  000000G                 MOV     (R0),CLK.CSR                    ; CLK.ADR,*             1914
000032  016067  000006  000000G         MOV     6(R0),CLK.HERTZ                 ;                       1915
000040  012767  000105  000000G         MOV     #105,CLK.START                  ;                       1916
000046  000421                          BR      2$                              ;                       1910
000050  012700  000114          1$:     MOV     #114,R0                         ;                       1921
000054  104462                          TRAP    62
```

```
000056  103015                          BHIS    2$                              ; RO,*
000060  010067  000000G                 MOV     RO,CLK.ADR                      ;                       1924
000064  012767  177777  000000G         MOV     #-1,CLK.TYPE                    ; CLK.ADR,*             1925
000072  011067  000000G                 MOV     (RO),CLK.CSR                    ;                       1926
000076  016067  000006  000000G         MOV     6(RO),CLK.HERTZ                 ;                       1927
000104  012767  000100  000000G         MOV     #100,CLK.START                  ;                       1932
000112  005767  000000G           2$:   TST     CLK.TYPE                        ;
000116  001421                          BEQ     3$                                                      1935
000120  016700  000000G                 MOV     CLK.ADR,RO                      ;
000124  116067  000004  000000G         MOVB    4(RO),VEC.AD                                            1936
000132  012746  000240                  MOV     #240,-(SP)                      ;
000136  012746  000016'                 MOV     #CLK.INT.SERV,-(SP)
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                          ;                       1934
000162  000207                    3$:   RTS     PC                              ;                       1901

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


;         1940
```

```
;          1941  %sbttl 'CLOCK INIT ROUTINE'
;          1942
;          1943  global routine CLOCK_INIT : novalue =
;          1944  ! ++
;          1945  !INIT CLOCK
;          1946  !
;          1947  !
;          1948      begin
;          1949      TICKS = 0;
;          1950      SECONDS = 0;
;          1951      MINUTES = 0;
;          1952  !START THE CLOCK
;          1953      .CLK_CSR = .CLK_START;
;          1954      end;


                                                .SBTTL  CLOCK.INIT CLOCK INIT ROUTINE
000000  005067  000000G                  CLOCK.INIT::
                                                CLR     TICKS                           ;1949
000004  005067  000000G                         CLR     SECONDS                         ;1950
000010  005067  000000G                         CLR     MINUTES                         ;1951
000014  016777  000000G 000000G                 MOV     CLK.START,@CLK.CSR              ;1953
000022  000207                                  RTS     PC                              ;1943

; Routine Size:  10 words,      Routine Base:  AB$CODE + 0250
; Maximum stack depth per invocation:  0 words


;          1955
```

```
;        1956  %sbttl 'RC25 CONTROLLER ERROR REPORTING'
;        1957  BGNMSG (RC25$ERR_RPT);


                                        .SBTTL  RC25$ERR.RPT RC25 CONTROLLER ERROR REPORTING
000000  004767  000000V              RC25$ERR.RPT::
                                        JSR     PC,M$RC25$ERR.RPT               ;                       1957
000004  104423                              TRAP    23
000006  000207                              RTS     PC

; Routine Size:  4 words,      Routine Base:  AB$CODE + 0274
; Maximum stack depth per invocation:  2 words


;        1958  ! ++
;        1959  ! FUNCTIONAL DESCRIPTION:
;        1960  !
;        1961  !       THIS ROUTINE IS CALLED BY THE DIAGNOSTIC SUPERVISOR VIA
;        1962  !       THE "PRLINK" ARGUMENT SPECIFIED IN THE $DS_ERRXXX MACRO
;        1963  !       TO REPORT DETAILED RC 25 CONTROLLER ERRORS.
;        1964  !
;        1965  ! FORMAL PARAMETERS:
;        1966  !
;        1967  !       P1              -  POINTER TO FORMATED ERROR MESSAGE.
;        1968  !       P2              -  FIELD REPLACEABLE UNIT CALL-OUT MASK.
;        1969  !       P3              -  RC 25 CONTROLLER REGISTER PRINT-OUT MASK.
;        1970  !       P4              -  DATA.
;        1971  !       P5              -  DATA.
;        1972  !       P6              -  DATA.
;        1973  !
;        1974  ! IMPLICIT INPUTS:
;        1975  !
;        1976  !       RET_STATUS
;        1977  !
;        1978  ! IMPLICIT OUTPUTS:
;        1979  !
;        1980  !       - NONE -
;        1981  !
;        1982  ! COMPLETION CODES:
;        1983  !
;        1984  !       - NONE -
;        1985  !
;        1986  ! SIDE EFFECTS:
;        1987  !
;        1988  !       - NONE -
;        1989  ! --
;        1990  !
;        1991  ! PRINT SUPPLEMENTAL ERROR INFO
;        1992  !
;        1993
;        1994  if .P1 neqa 0                                    ! IF ERROR MESSAGE POINTER
;        1995  then                                             ! ISN'T 0, THEN PRINT-OUT
;        1996      begin
;        1997
;        1998      if .P_MASK eqlu 3 then PRINTB (.P1, .P4, .P5, .P6); ! SUPPLEMENTAL ERROR INFO.
;        1999
;        2000      if .P_MASK eqlu 2 then PRINTB (.P1, .P6, .P4, .P5);
```

```
;
;        if .P_MASK eqlu 1 then PRINTB (.P1, .P4);
;
;        end;
;
;    if .P3 nequ 0                                    ! IF ELIGIBLE  REGISTER(S)
;    then                                             ! PRESENT, THEN PRINT-OUT
;        PRINTB (FMT3, .P6, .P3);                     ! SELECTED  CONTROLLER
;
;                                                     ! REGISTER(S).
;    !
;    ! PERFORM FIELD REPLACEABLE UNIT CALL-OUT
;    !
;    !if .P2 nequ 0                                   ! IF ELIGIBLE FRU CALL-OUT(S)
;    !then                                            ! PRESENT, THEN PRINT-OUT
;    PRT$FRU_CALLOUT (.P2);                           ! FEILD REPLACEABLE UNITS.
;    ! CLEAR ALL PARAMETERS
;    P1 = ZERO;
;    P2 = ZERO;
;    P3 = ZERO;
;    P4 = ZERO;
;    P5 = ZERO;
;    P6 = ZERO;
;    ENDMSG;                                          ! END OF ROUTINE:

                                         .SBTTL  M$RC25$ERR.RPT RC25 CONTROLLER ERROR REPORTING
000000  005767  000000G          M$RC25$ERR.RPT:
                                         TST     P1                          ;            1994
000004  001462                           BEQ     3$
000006  126727  000000G 000003           CMPB    P.MASK,#3                   ;            1998
000014  001016                           BNE     1$
000016  016746  000000G                  MOV     P6,-(SP)
000022  016746  000000G                  MOV     P5,-(SP)
000026  016746  000000G                  MOV     P4,-(SP)
000032  016746  000000G                  MOV     P1,-(SP)
000036  012746  000004                   MOV     #4,-(SP)
000042  010600                           MOV     SP,R0                       ; SP,*
000044  104414                           TRAP    14
000046  062706  000012                   ADD     #12,SP
000052  126727  000000G 000002   1$:     CMPB    P.MASK,#2                   ;            2000
000060  001016                           BNE     2$
000062  016746  000000G                  MOV     P5,-(SP)
000066  016746  000000G                  MOV     P4,-(SP)
000072  016746  000000G                  MOV     P6,-(SP)
000076  016746  000000G                  MOV     P1,-(SP)
000102  012746  000004                   MOV     #4,-(SP)
000106  010600                           MOV     SP,R0                       ; SP,*
000110  104414                           TRAP    14
000112  062706  000012                   ADD     #12,SP
000116  126727  000000G 000001   2$:     CMPB    P.MASK,#1                   ;            2002
000124  001012                           BNE     3$
000126  016746  000000G                  MOV     P4,-(SP)
000132  016746  000000G                  MOV     P1,-(SP)
000136  012746  000002                   MOV     #2,-(SP)
000142  010600                           MOV     SP,R0                       ; SP,*
000144  104414                           TRAP    14
```

```
000146  062706  000006                      ADD     #6,SP
000152  005767  000000G             3$:     TST     P3                      ;                       2006
000156  001414                              BEQ     4$
000160  016746  000000G                     MOV     P3,-(SP)                ;                       2008
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$:     MOV     P2,-(SP)                ;                       2016
000214  004767  000000V                     JSR     PC,PRT$FRU.CALLOUT
000220  005067  000000G                     CLR     P1                      ;                       2018
000224  005067  000000G                     CLR     P2                      ;                       2019
000230  005067  000000G                     CLR     P3                      ;                       2020
000234  005067  000000G                     CLR     P4                      ;                       2021
000240  005067  000000G                     CLR     P5                      ;                       2022
000244  005067  000000G                     CLR     P6                      ;                       2023
000250  005726                              TST     (SP)+                   ;                       1957
000252  000207                              RTS     PC

; Routine Size:  86 words,      Routine Base:  AB$CODE + 0304
; Maximum stack depth per invocation:  7 words
```

```
;       2025  %sbttl 'FIELD REPLACEABLE UNIT REPORTING'
;       2026
;       2027  global routine PRT$FRU_CALLOUT (FRU$MASK) : novalue =
;       2028  !++
;       2029  ! FUNCTIONAL DESCRIPTION:
;       2030  !
;       2031  !       THIS ROUTINE REPORTS FIELD REPLACEABLE UNITS WHICH ARE
;       2032  !       DEEMED ELIGIBLE FOR PRINT-OUT BY THE FAILING TEST.
;       2033  !
;       2034  ! FORMAL PARAMETERS:
;       2035  !
;       2036  !       FRU$MASK        -  FIELD REPLACEABLE UNIT CALL-OUT MASK.
;       2037  !
;       2038  ! IMPLICIT INPUTS:
;       2039  !
;       2040  !       - NONE -
;       2041  !
;       2042  ! IMPLICIT OUTPUTS:
;       2043  !
;       2044  !       - NONE -
;       2045  !
;       2046  ! COMPLETION CODES:
;       2047  !
;       2048  !       - NONE -
;       2049  !
;       2050  ! SIDE EFFECTS:
;       2051  !
;       2052  !       - NONE -
;       2053  !--
;       2054    begin
;       2055
;       2056    local
;       2057        FRU$MSG;                                ! ALLOCATE STORAGE FOR
;       2058                                                ! POINTER TO FRU MESSAGE.
;       2059
;       2060    !
;       2061    ! PERFORM FIELD REPLACEABLE UNIT CALL-OUT
;       2062    !
;       2063
;       2064    incru FRU_SELECT from 0 to 3 do             ! CHECK EACH FRU FOR
;       2065                                                ! POSSIBLE CALL-OUT.
;       2066
;       2067        if BIT_TST (.FRU$MASK, 1^.FRU_SELECT)   ! IF CURRENT FRU ELIGIBLE
;       2068        then                                    ! FOR PRINT-OUT THEN GET
;       2069            begin                               ! POINTER TO FRU MESSAGE.
;       2070
;       2071            selectu 1^.FRU_SELECT of            ! SELECT FRU FROM ONE OF
;       2072                set                             ! THE FOLLOWING:
;       2073
;       2074                [ADAPT] :
;       2075                    FRU$MSG = ADAPTO;           ! GET ASYNC FRU MESSAGE.
;       2076
;       2077                [CONTR] :
;       2078                    FRU$MSG = CONTRO;           ! GET SYNC FRU MESSAGE
;       2079
;       2080                [DRIVE] :
;       2081                    FRU$MSG = DRIVE_;           ! GET ARR_DAT FRU MESSAGE
```

```
;          2082
;          2083                    [MECH] :
;          2084                        FRU$MSG = MECHAN;           ! GET MEM ARR FRU MESSAGE
;          2085                    tes;                            ! PRINT FRU CALL-OUT.
;          2086
;          2087                PRINTX (FRU, .FRU$MSG, .UNIT);
;          2088                end;                                ! END OF ROUTINE:
;          2089
;          2090        end;                                        ! 'PRT$FRU_CALLOUT'.


                                        .SBTTL  PRT$FRU.CALLOUT FIELD REPLACEABLE UNIT REPORTING
000000  004167  000000G          PRT$FRU.CALLOUT::
                                        JSR     R1,$SAVE3                       ;                       2027
000004  005003                          CLR     R3                              ; FRU.SELECT            2064
000006  012746  000001          1$:     MOV     #1,-(SP)                        ;                       2067
000012  010346                          MOV     R3,-(SP)                        ; FRU.SELECT,*
000014  004767  000000G                 JSR     PC,BL$SHF
000020  010001                          MOV     R0,R1
000022  005726                          TST     (SP)+
000024  016600  000014                  MOV     14(SP),R0                       ; FRU$MASK,*
000030  005100                          COM     R0
000032  040001                          BIC     R0,R1
000034  012716  000001                  MOV     #1,(SP)
000040  010346                          MOV     R3,-(SP)                        ; FRU.SELECT,*
000042  004767  000000G                 JSR     PC,BL$SHF
000046  022626                          CMP     (SP)+,(SP)+
000050  020100                          CMP     R1,R0
000052  001044                          BNE     6$
000054  012746  000001                  MOV     #1,-(SP)                        ;                       2071
000060  010346                          MOV     R3,-(SP)                        ; FRU.SELECT,*
000062  004767  000000G                 JSR     PC,BL$SHF
000066  020027  000001                  CMP     R0,#1
000072  001002                          BNE     2$
000074  012702  000000G                 MOV     #ADAPTO,R2                      ; *,FRU$MSG             2075
000100  020027  000002          2$:     CMP     R0,#2                           ;                       2071
000104  001002                          BNE     3$
000106  012702  000000G                 MOV     #CONTRO,R2                      ; *,FRU$MSG             2078
000112  020027  000004          3$:     CMP     R0,#4                           ;                       2071
000116  001002                          BNE     4$
000120  012702  000000G                 MOV     #DRIVE.,R2                      ; *,FRU$MSG             2081
000124  020027  000010          4$:     CMP     R0,#10                          ;                       2071
000130  001002                          BNE     5$
000132  012702  000000G                 MOV     #MECHAN,R2                      ; *,FRU$MSG             2084
000136  016716  000000G          5$:     MOV     UNIT,(SP)                       ;                       2087
000142  010246                          MOV     R2,-(SP)                        ; FRU$MSG,*
000144  012746  000000G                 MOV     #FRU,-(SP)
000150  012746  000003                  MOV     #3,-(SP)
000154  010600                          MOV     SP,R0                           ; SP,*
000156  104415                          TRAP    15
000160  062706  000012                  ADD     #12,SP                          ;                       2069
000164  005203                  6$:     INC     R3                              ; FRU.SELECT            2064
000166  020327  000003                  CMP     R3,#3                           ; FRU.SELECT,*
000172  101705                          BLOS    1$
000174  000207                          RTS     PC                              ;                       2027

; Routine Size:  63 words,      Routine Base:  AB$CODE + 0560
```

```
; Maximum stack depth per invocation:  11 words

;        2091
```

```
;       2092  %sbttl 'AZTEC INITIALIZATION'
;       2093
;       2094  global routine AZT_INIT =
;       2095  ! ++
;       2096  ! FUNCTIONAL DESCRIPTION:
;       2097  !
;       2098  !       THIS ROUTINE WILL DO STEP 1 THROUGH STEP 3 CHECK FOR ANY ERRORS
;       2099  !       IN EACH STEP AND RETURN TRUE OR FALSE.
;       2100  !
;       2101  ! FORMAL PARAMETERS:
;       2102  !
;       2103  !       - NONE -
;       2104  !
;       2105  ! IMPLICIT INPUTS:
;       2106  !
;       2107  !       DATA1 = STEP 1 WRITE DATA
;       2108  !       DATA2 = STEP 2 WRITE DATA
;       2109  !       DATA3 = STEP 3 WRITE DATA
;       2110  !       DATA4 = STEP 4 WRITE DATA
;       2111  !
;       2112  !       B_MASK = WITCH STEPS WILL BE DONE
;       2113  !                %O 1 = STEP 1
;       2114  !                %O 3 = STEP 1,2
;       2115  !                %O 7 = STEP 1,2,3
;       2116  !                %O17 = STEP 1,2,3,4
;       2117  !
;       2118  ! IMPLICIT OUTPUTS:
;       2119  !
;       2120  !       IF ERROR OR NO STEP IT WILL RETURN
;       2121  !       P1-P5,P_MASK
;       2122  !       RET_STATUS
;       2123  ! COMPLETION CODES:
;       2124  !
;       2125  !       TRUE OR FALSE
;       2126  !
;       2127  ! SIDE EFFECTS:
;       2128  !
;       2129  !       - NONE -
;       2130  ! --
;       2131    begin
;       2132
;       2133    local
;       2134        N,                                  !STEP NUMBER
;       2135        MASK,                               !STEP MASK
;       2136        COUNT,                              !TIME OUT COUNT
;       2137        DATA;                               !WRITE DATA FOR THE STEP
;       2138
;       2139    !
;       2140    ! INIT THE AZTEC
;       2141    !
;       2142    I_AM_NEX = ALL_ONES;                    ! INIT INTERRUPT FLAG
;       2143    !
;       2144
;       2145    ! THE FOLLOWING LOOP WILL DO STEP 1 THRU 4 AS GIVEN BY B_MASK
;       2146    ! INPUT SELECTING APPROPRIATE DATA INPUT FOR STEP WRITES. IF
;       2147    ! ERROR IN SA REGISTER P1 - P4 AND P_MASK WILL BE SUPPLIED FOR
;       2148    ! ERROR REPORT. ONLY SA DATA FOR THE FINAL WRITE STEP IS PRESERVED.
```

```
;          2149      MASK = %b'0001';                                ! STEP MASK
;          2150      WRT_RC25 (RCIP, ALL_ONES);                      ! START INIT
;          2151      DELAY (2);                                      ! WAIT FOR COMPLETION
;          2152
;          2153      incru N from 0 to 4 do
;          2154          begin
;          2155
;          2156          if (.N eqlu 0 or BIT_TST (.B_MASK, .MASK))      ! TEST FOR STEP NUMBER
;          2157          then
;          2158              begin
;          2159              !
;          2160
;          2161              selectoneu .N of                            ! SELECT CORRECT WRITE
;          2162                  set                                     !
;          2163
;          2164                  [0] :
;          2165                      DATA = ALL_ONES;                    !
;          2166
;          2167                  [1] :
;          2168                      DATA = .DATA1;                      ! DATA FOR STEP WRITES
;          2169
;          2170                  [2] :
;          2171                      DATA = .DATA2;
;          2172
;          2173                  [3] :
;          2174                      DATA = .DATA3;
;          2175
;          2176                  [4] :
;          2177                      DATA = .DATA4;
;          2178                  tes;
;          2179
;          2180              !
;          2181
;          2182              if .N gequ 1 then WRT_RC25 (RCSA, .DATA);   ! STEP N WRITE DATA TO SA
;          2183
;          2184              incru COUNT from 0 to 20 do                 ! TIME OUT WAIT LOOP
;          2185                  begin
;          2186                  DELAY (333);                            ! DELAY 1 SEC. APPROX.
;          2187
;          2188                  if .I_AM_NEX eqlu ALL_ONES then exitloop;
;          2189
;          2190                  end;
;          2191
;          2192              if .I_AM_NEX eqlu ALL_ONES
;          2193              then
;          2194                  begin
;          2195                  DELAY (2);
;          2196                  RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];   ! STEP N   READ
;          2197
;          2198                  if .N nequ 0 then MASK = .MASK^1;       ! INCREMENT STEP
;          2199
;          2200                  I_AM_NEX = ZERO;
;          2201
;          2202                  if (.RC25_DATA [RCSA, RCSA_ER] nequ ZERO)       ! IF SA REGISTER CONTAINS
;          2203                  then
;          2204                      begin
;          2205                      RET_STATUS = PFE_CODE;
```

```
;        2206                        exitloop;
;        2207                        end
;        2208                    else
;        2209                        begin
;        2210
;        2211                        if (.RC25_DATA [RCSA, RCSA_STEP] nequ .MASK)        ! ERROR OR INCORRECT STEP
;        2212                        then
;        2213                            begin                               ! SUPPLY P1 THRU P5 AND
;        2214                            P_MASK = 2;
;        2215                            P1 = FMT3;
;        2216                            P2 = ADAPT;
;        2217                            P4 = (.RC25_ADDR) + 2;
;        2218                            P5 = .RC25_DATA [RCSA, RC_ALL];
;        2219                            P6 = .MASK;
;        2220                            MSGADR = MSG_14;
;        2221                            RET_STATUS = TRUE;
;        2222                            return .RET_STATUS;     ! TRUE STATUS.
;        2223                            end;
;        2224
;        2225                        end;
;        2226
;        2227                    end
;        2228                else
;        2229                    begin
;        2230                    RET_STATUS = CTO_CODE;
;        2231                    exitloop;
;        2232                    end;
;        2233
;        2234                end;
;        2235
;        2236            end;
;        2237
;        2238        if .RET_STATUS
;        2239        then
;        2240            begin
;        2241            P_MASK = 2;
;        2242            P1 = FMT3;
;        2243            P2 = ADAPT;
;        2244            P4 = (.RC25_ADDR) + 2;
;        2245            P5 = .RC25_DATA [RCSA, RC_ALL];
;        2246            P6 = .MASK;
;        2247            MSGADR = MSG_14;
;        2248            return .RET_STATUS;
;        2249            end
;        2250        else
;        2251            return RET_STATUS = PAS_CODE;
;        2252            end;


                                .SBTTL  AZT.INIT AZTEC INITIALIZATION
000000  004167  000000G    AZT.INIT::
                            JSR     R1,$SAVE5                       ;                       2094
000004  024646              CMP     -(SP),-(SP)
000006  012767  177777  000000G  MOV     #-1,I.AM.NEX                    ;                       2142
000014  012704  000001      MOV     #1,R4                           ; *,MASK                2149
000020  012700  177777      MOV     #-1,R0                          ; *,RCM.REG             2150
000024  010077  000000G      MOV     R0,@RC25.ADDR                   ; RCM.REG,*
```

```
000030  012701  000002                    MOV     #2,R1                           ; *,$$TMP2                      2151
000034  001411                    1$:     BEQ     4$
000036  016700  000000G                   MOV     L$DLY,R0                        ; *,$$TMP1
000042  001404                            BEQ     3$
000044  005066  000002            2$:     CLR     2(SP)                           ; $$TMP
000050  005300                            DEC     R0                              ; $$TMP1
000052  001374                            BNE     2$
000054  005301                    3$:     DEC     R1                              ; $$TMP2
000056  000766                            BR      1$
000060  005005                    4$:     CLR     R5                              ; N                             2153
000062  005705                    5$:     TST     R5                              ; N                             2156
000064  001412                            BEQ     6$
000066  010401                            MOV     R4,R1                           ; MASK,*
000070  005000                            CLR     R0
000072  156700  000000G                   BISB    B.MASK,R0
000076  005101                            COM     R1
000100  040100                            BIC     R1,R0
000102  020004                            CMP     R0,R4                           ; *,MASK
000104  001402                            BEQ     6$
000106  000167  000416                    JMP     26$
000112  010500                    6$:     MOV     R5,R0                           ; N,*                           2161
000114  001003                            BNE     7$
000116  012702  177777                    MOV     #-1,R2                          ; *,DATA                        2165
000122  000427                            BR      11$                             ;                               2161
000124  020027  000001            7$:     CMP     R0,#1
000130  001003                            BNE     8$
000132  016702  000000G                   MOV     DATA1,R2                        ; *,DATA                        2168
000136  000421                            BR      11$                             ;                               2161
000140  020027  000002            8$:     CMP     R0,#2
000144  001003                            BNE     9$
000146  016702  000000G                   MOV     DATA2,R2                        ; *,DATA                        2171
000152  000413                            BR      11$                             ;                               2161
000154  020027  000003            9$:     CMP     R0,#3
000160  001003                            BNE     10$
000162  016702  000000G                   MOV     DATA3,R2                        ; *,DATA                        2174
000166  000405                            BR      11$                             ;                               2161
000170  020027  000004            10$:    CMP     R0,#4
000174  001002                            BNE     11$
000176  016702  000000G                   MOV     DATA4,R2                        ; *,DATA                        2177
000202  005705                    11$:    TST     R5                              ; N                             2182
000204  001405                            BEQ     12$
000206  010201                            MOV     R2,R1                           ; DATA,RCM.REG
000210  016700  000000G                   MOV     RC25.ADDR,R0
000214  010160  000002                    MOV     R1,2(R0)                        ; RCM.REG,*
000220  005003                    12$:    CLR     R3                              ; COUNT                         2184
000222  012701  000515            13$:    MOV     #515,R1                         ; *,$$TMP2                      2186
000226  001411                    14$:    BEQ     17$
000230  016700  000000G                   MOV     L$DLY,R0                        ; *,$$TMP1
000234  001404                            BEQ     16$
000236  005066  000002            15$:    CLR     2(SP)                           ; $$TMP
000242  005300                            DEC     R0                              ; $$TMP1
000244  001374                            BNE     15$
000246  005301                    16$:    DEC     R1                              ; $$TMP2
000250  000766                            BR      14$
000252  026727  000000G 177777    17$:    CMP     I.AM.NEX,#-1                    ;                               2188
000260  001404                            BEQ     18$
000262  005203                            INC     R3                              ; COUNT                         2184
```

```
000264  020327  000024                          CMP     R3,#24                  ;; COUNT,*
000270  101754                                  BLOS    13$
000272  026727  000000G 177777        18$:     CMP     I.AM.NEX,#-1            ;                               2192
000300  001107                                  BNE     25$
000302  012703  000002                          MOV     #2,R3                   ; *,$$TMP2                      2195
000306  001411                        19$:     BEQ     22$
000310  016700  000000G                         MOV     L$DLY,R0                ; *,$$TMP1
000314  001404                                  BEQ     21$
000316  005066  000002                20$:     CLR     2(SP)                   ; $$TMP
000322  005300                                  DEC     R0                      ; $$TMP1
000324  001374                                  BNE     20$
000326  005303                        21$:     DEC     R3                      ; $$TMP2
000330  000766                                  BR      19$
000332  016700  000000G               22$:     MOV     RC25.ADDR,R0            ;                               2196
000336  016016  000002                          MOV     2(R0),(SP)              ; *,RC.REG
000342  011667  000002G                         MOV     (SP),RC25.DATA+2        ; RC.REG,*
000346  005705                                  TST     R5                      ; N                             2198
000350  001401                                  BEQ     23$
000352  006304                                  ASL     R4                      ; MASK
000354  005067  000000G               23$:     CLR     I.AM.NEX                ;                               2200
000360  032767  100000  000002G               BIT     #100000,RC25.DATA+2     ;                               2202
000366  001404                                  BEQ     24$
000370  012767  000021  000000G               MOV     #21,RET.STATUS          ;                               2205
000376  000462                                  BR      27$                     ;                               2204
000400  010401                        24$:     MOV     R4,R1                   ; MASK,*                        2211
000402  016700  000002G                         MOV     RC25.DATA+2,R0
000406  006200                                  ASR     R0
000410  006200                                  ASR     R0
000412  006200                                  ASR     R0
000414  000300                                  SWAB    R0
000416  042700  177760                          BIC     #177760,R0
000422  020001                                  CMP     R0,R1
000424  001441                                  BEQ     26$
000426  112767  000002  000000G               MOVB    #2,P.MASK               ;                               2214
000434  012767  000000G 000000G               MOV     #FMT3,P1                ;                               2215
000442  012767  000001  000000G               MOV     #1,P2                   ;                               2216
000450  016700  000000G                       MOV     RC25.ADDR,R0            ;                               2217
000454  062700  000002                        ADD     #2,R0
000460  010067  000000G                       MOV     R0,P4
000464  016767  000002G 000000G               MOV     RC25.DATA+2,P5          ;                               2218
000472  010467  000000G                       MOV     R4,P6                   ; MASK,*                        2219
000476  012767  000000G 000000G               MOV     #MSG.14,MSGADR          ;                               2220
000504  012767  000001  000000G               MOV     #1,RET.STATUS           ;                               2221
000512  016700  000000G                       MOV     RET.STATUS,R0           ;                               2213
000516  000453                                BR      29$
000520  012767  000011  000000G        25$:     MOV     #11,RET.STATUS          ;                               2230
000526  000406                                  BR      27$                     ;                               2229
000530  005205                        26$:     INC     R5                      ; N                             2153
000532  020527  000004                          CMP     R5,#4                   ; N,*
000536  101002                                  BHI     27$
000540  000167  177316                          JMP     5$
000544  032767  000001  000000G        27$:     BIT     #1,RET.STATUS           :                               2238
000552  001432                                  BEQ     28$
000554  112767  000002  000000G               MOVB    #2,P.MASK               ;                               2241
000562  012767  000000G 000000G               MOV     #FMT3,P1                ;                               2242
000570  012767  000001  000000G               MOV     #1,P2                   ;                               2243
000576  016700  000000G                       MOV     RC25.ADDR,R0            ;                               2244
```

```
000602  062700  000002                          ADD     #2,R0
000606  010067  000000G                         MOV     R0,P4
000612  016767  000002G 000000G                 MOV     RC25.DATA+2,P5          ;                       2245
000620  010467  000000G                         MOV     R4,P6                   ; MASK,*                2246
000624  012767  000000G 000000G                 MOV     #MSG.14,MSGADR          ;                       2247
000632  016700  000000G                         MOV     RET.STATUS,R0           ;                       2131
000636  000403                                  BR      29$
000640  005067  000000G                 28$:    CLR     RET.STATUS              ;                       2251
000644  005000                                  CLR     R0                      ;                       2131
000646  022626                          29$:    CMP     (SP)+,(SP)+             ;                       2094
000650  000207                                  RTS     PC

; Routine Size:  213 words,     Routine Base:  AB$CODE + 0756
; Maximum stack depth per invocation:  9 words
```

```
;       2253  %sbttl 'AZTEC INITIALIZATION BY POLING'
;       2254
;       2255  global routine AZP_INIT =
;       2256  !++
;       2257  ! FUNCTIONAL DESCRIPTION:
;       2258  !
;       2259  !       THIS ROUTINE WILL DO STEP 1 THROUGH STEP 4 CHECK FOR ANY ERRORS
;       2260  !       IN EACH STEP AND RETURN TRUE OR FALSE.
;       2261  !
;       2262  ! FORMAL PARAMETERS:
;       2263  !
;       2264  !       - NONE -
;       2265  !
;       2266  ! IMPLICIT INPUTS:
;       2267  !
;       2268  !       DATA1 = STEP 1 WRITE DATA
;       2269  !       DATA2 = STEP 2 WRITE DATA
;       2270  !       DATA3 = STEP 3 WRITE DATA
;       2271  !       DATA4 = STEP 4 WRITE DATA
;       2272  !
;       2273  !       B_MASK = WITCH STEPS WILL BE DONE
;       2274  !                %O 1 = STEP 1
;       2275  !                %O 3 = STEP 1,2
;       2276  !                %O 7 = STEP 1,2,3
;       2277  !                %O17 = STEP 1,2,3,4
;       2278  !
;       2279  ! IMPLICIT OUTPUTS:
;       2280  !
;       2281  !       IF ERROR OR NO STEP IT WILL RETURN
;       2282  !       P1-P5,P_MASK
;       2283  !       RET_STATUS
;       2284  ! COMPLETION CODES:
;       2285  !
;       2286  !       TRUE OR FALSE
;       2287  !
;       2288  ! SIDE EFFECTS:
;       2289  !
;       2290  !       - NONE -
;       2291  !--
;       2292          begin
;       2293
;       2294          local
;       2295              N,                                  !STEP NUMBER
;       2296              MASK,                               !STEP MASK
;       2297              COUNT,                              !TIME OUT COUNT
;       2298              DATA;                               !WRITE DATA FOR THE STEP
;       2299
;       2300          !
;       2301          ! INIT THE AZTEC
;       2302          !
;       2303          !
;       2304          ! THE FOLLOWING LOOP WILL DO STEP 1 THRU 4 AS GIVEN BY B_MASK
;       2305          ! INPUT SELECTING APPROPRIATE DATA INPUT FOR STEP WRITES. IF
;       2306          ! ERROR IN SA REGISTER P1 - P4 AND P_MASK WILL BE SUPPLIED FOR
;       2307          ! ERROR REPORT. ONLY SA DATA FOR THE FINAL WRITE STEP IS PRESERVED.
;       2308          MASK = %b'0001';                        ! STEP MASK
;       2309          WRT_RC25 (RCIP, ALL_ONES);              ! START INIT
```

```
;        2310          DELAY (2);                                  ! WAIT FOR COMPLETION
;        2311
;        2312          incru N from 0 to 4 do
;        2313              begin
;        2314
;        2315              if (.N eqlu 0 or BIT_TST (.B_MASK, .MASK))  ! TEST FOR STEP NUMBER
;        2316              then
;        2317                  begin
;        2318                  !
;        2319
;        2320                  selectoneu .N of                    ! SELECT CORRECT WRITE
;        2321                      set                             !
;        2322
;        2323                      [0] :
;        2324                          DATA = ALL_ONES;            !
;        2325
;        2326                      [1] :
;        2327                          DATA = .DATA1;              ! DATA FOR STEP WRITES
;        2328
;        2329                      [2] :
;        2330                          DATA = .DATA2;
;        2331
;        2332                      [3] :
;        2333                          DATA = .DATA3;
;        2334
;        2335                      [4] :
;        2336                          DATA = .DATA4;
;        2337                      tes;
;        2338
;        2339                  !
;        2340
;        2341                  incru COUNT from 0 to 20 do         ! TIME OUT WAIT LOOP
;        2342                      begin
;        2343                      DELAY (333);                    ! DELAY 1 SEC. APPROX.
;        2344                      RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];
;        2345
;        2346                      if .RC25_DATA [RCSA, RCSA_STEP] eqlu .MASK then exitloop;
;        2347
;        2348                      RET_STATUS = CTO_CODE;
;        2349                      end;
;        2350
;        2351                  if (.RC25_DATA [RCSA, RCSA_ER] nequ ZERO)
;        2352                  then
;        2353                      begin
;        2354                      RET_STATUS = PFE_CODE;
;        2355                      exitloop;
;        2356                      end
;        2357                  else
;        2358                      begin
;        2359
;        2360                      if (.RC25_DATA [RCSA, RCSA_STEP] nequ .MASK)
;        2361                      then
;        2362                          begin
;        2363                          P_MASK = 2;
;        2364                          PT = FMT3;
;        2365                          P2 = ADAPT;
;        2366                          P4 = (.RC25_ADDR) + 2;
```

```
;          2367                         P5 = .RC25_DATA [RCSA, RC_ALL];
;          2368                         P6 = .MASK;
;          2369                         MSGADR = MSG_14;
;          2370                         return .RET_STATUS;     ! TRUE STATUS.
;          2371                         end
;          2372                     else
;          2373                         begin
;          2374                         RET_STATUS = PAS_CODE;
;          2375                         end;
;          2376
;          2377                     end;
;          2378
;          2379                 if .N nequ ZERO
;          2380                 then
;          2381                     begin
;          2382                     MASK = .MASK^1;
;          2383                     WRT_RC25 (RCSA, .DATA);     ! STEP N WRITE DATA TO SA
;          2384                     end;
;          2385
;          2386                 end;
;          2387
;          2388             end;
;          2389
;          2390         if .RET_STATUS
;          2391         then
;          2392             begin
;          2393             P_MASK = 2;
;          2394             P1 = FMT3;
;          2395             P2 = ADAPT;
;          2396             P4 = (.RC25_ADDR) + 2;
;          2397             P5 = .RC25_DATA [RCSA, RC_ALL];
;          2398             P6 = .MASK;
;          2399             MSGADR = MSG_14;
;          2400             return .RET_STATUS;
;          2401             end
;          2402         else
;          2403             return .RET_STATUS;
;          2404
;          2405         end;


                                .SBTTL  AZP.INIT AZTEC INITIALIZATION BY POLING
000000  004167  000000G   AZP.INIT::
                            JSR     R1,$SAVE5                       ;                       2255
000004  024646              CMP     -(SP),-(SP)
000006  012705  000001      MOV     #1,R5                           ; *,MASK                2308
000012  012700  177777      MOV     #-1,R0                          ; *,RCM.REG             2309
000016  010077  000000G     MOV     R0,@RC25.ADDR                   ; RCM.REG,*
000022  012701  000002      MOV     #2,R1                           ; *,$$TMP2              2310
000026  001411          1$: BEQ     4$
000030  016700  000000G     MOV     L$DLY,R0                        ; *,$$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$:     CLR     R3                              ; N                             2312
000054  005703                          5$:     TST     R3                              ; N                             2315
000056  001410                                  BEQ     6$
000060  010501                                  MOV     R5,R1                           ; MASK,*
000062  005000                                  CLR     R0
000064  156700  000000G                         BISB    B.MASK,R0
000070  005101                                  COM     R1
000072  040100                                  BIC     R1,R0
000074  020005                                  CMP     R0,R5                           ; *,MASK
000076  001167                                  BNE     20$
000100  010300                          6$:     MOV     R3,R0                           ; N,*                           2320
000102  001003                                  BNE     7$
000104  012702  177777                          MOV     #-1,R2                          ; *,DATA                        2324
000110  000427                                  BR      11$                             ;                               2320
000112  020027  000001                  7$:     CMP     R0,#1
000116  001003                                  BNE     8$
000120  016702  000000G                         MOV     DATA1,R2                        ; *,DATA                        2327
000124  000421                                  BR      11$                             ;                               2320
000126  020027  000002                  8$:     CMP     R0,#2
000132  001003                                  BNE     9$
000134  016702  000000G                         MOV     DATA2,R2                        ; *,DATA                        2330
000140  000413                                  BR      11$                             ;                               2320
000142  020027  000003                  9$:     CMP     R0,#3
000146  001003                                  BNE     10$
000150  016702  000000G                         MOV     DATA3,R2                        ; *,DATA                        2333
000154  000405                                  BR      11$                             ;                               2320
000156  020027  000004                  10$:    CMP     R0,#4
000162  001002                                  BNE     11$
000164  016702  000000G                         MOV     DATA4,R2                        ; *,DATA                        2336
000170  005004                          11$:    CLR     R4                              ; COUNT                         2341
000172  012701  000515                  12$:    MOV     #515,R1                         ; *,$$TMP2                      2343
000176  001411                          13$:    BEQ     16$
000200  016700  000000G                         MOV     L$DLY,R0                        ; *,$$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$:    MOV     RC25.ADDR,R0                    ;                               2344
000226  016016  000002                          MOV     2(R0),(SP)                      ; *,RC.REG
000232  011667  000002G                         MOV     (SP),RC25.DATA+2                ; RC.REG,*
000236  010501                                  MOV     R5,R1                           ; MASK,*                        2346
000240  011600                                  MOV     (SP),R0                         ; RC25.DATA+2,*
000242  006200                                  ASR     R0
000244  006200                                  ASR     R0
000246  006200                                  ASR     R0
000250  000300                                  SWAB    R0
000252  042700  177760                         BIC     #177776G,R0
000256  020001                                  CMP     R0,R1
000260  001407                                  BEQ     17$
000262  012767  000011  000000G                 MOV     #11,RET.STATUS                  ;                               2348
000270  005204                                  INC     R4                              ; COUNT                         2341
000272  020427  000024                          CMP     R4,#24                          ; COUNT,*
000276  101735                                  BLOS    12$
000300  032767  100000  000002G         17$:    BIT     #100000,RC25.DATA+2             ;                               2351
000306  001404                                  BEQ     18$
```

```
000310  012767  000021  000000G             MOV     #21,RET.STATUS                  ;                               2354
000316  000465                              BR      21$                             ;                               2353
000320  010501                      18$:    MOV     R5,R1                           ; MASK,*                        2360
000322  016700  000002G                     MOV     RC25.DATA+2,R0
000326  006200                              ASR     R0
000330  006200                              ASR     R0
000332  006200                              ASR     R0
000334  000300                              SWAB    R0
000336  042700  177760                      BIC     #177760,R0
000342  020001                              CMP     R0,R1
000344  001432                              BEQ     19$
000346  112767  000002  000000G             MOVB    #2,P.MASK                       ;                               2363
000354  012767  000000G 000000G             MOV     #FMT3,P1                        ;                               2364
000362  012767  000001  000000G             MOV     #1,P2                           ;                               2365
000370  016700  000000G                     MOV     RC25.ADDR,R0                    ;                               2366
000374  062700  000002                      ADD     #2,R0
000400  010067  000000G                     MOV     R0,P4
000404  016767  000002G 000000G             MOV     RC25.DATA+2,P5                  ;                               2367
000412  010567  000000G                     MOV     R5,P6                           ; MASK,*                        2368
000416  012767  000000G 000000G             MOV     #MSG.14,MSGADR                  ;                               2369
000424  016700  000000G                     MOV     RET.STATUS,R0                   ;                               2362
000430  000460                              BR      23$
000432  005067  000000G             19$:    CLR     RET.STATUS                      ;                               2374
000436  005703                              TST     R3                              ; N                             2379
000440  001406                              BEQ     20$
000442  006305                              ASL     R5                              ; MASK                          2382
000444  010200                              MOV     R2,R0                           ; DATA,RCM.REG                  2383
000446  016704  000000G                     MOV     RC25.ADDR,R4
000452  010064  000002                      MOV     R0,2(R4)                        ; RCM.REG,*
000456  005203                      20$:    INC     R3                              ; N                             2312
000460  020327  000004                      CMP     R3,#4                           ; N,*
000464  101002                              BHI     21$
000466  000167  177362                      JMP     5$
000472  032767  000001  000000G     21$:    BIT     #1,RET.STATUS                   ;                               2390
000500  001432                              BEQ     22$
000502  112767  000002  000000G             MOVB    #2,P.MASK                       ;                               2393
000510  012767  000000G 000000G             MOV     #FMT3,P1                        ;                               2394
000516  012767  000001  000000G             MOV     #1,P2                           ;                               2395
000524  016700  000000G                     MOV     RC25.ADDR,R0                    ;                               2396
000530  062700  000002                      ADD     #2,R0
000534  010067  000000G                     MOV     R0,P4
000540  016767  000002G 000000G             MOV     RC25.DATA+2,P5                  ;                               2397
000546  010567  000000G                     MOV     R5,P6                           ; MASK,*                        2398
000552  012767  000000G 000000G             MOV     #MSG.14,MSGADR                  ;                               2399
000560  016700  000000G                     MOV     RET.STATUS,R0                   ;                               2292
000564  000402                              BR      23$
000566  016700  000000G             22$:    MOV     RET.STATUS,R0
000572  022626                      23$:    CMP     (SP)+,(SP)+                     ;                               2255
000574  000207                              RTS     PC

; Routine Size:  191 words,    Routine Base:  AB$CODE + 1630
; Maximum stack depth per invocation:  9 words


;        2406
```

```
;          2407  %sbttl 'COMMUNICATION RING INITIALIZATION'
;          2408  !
;          2409
;          2410      global routine INIT_COM_AREA =
;          2411
;          2412  !++
;          2413  ! FUNCTIONAL DESCRIPTIONS:
;          2414  !       THIS ROUTINE FIRST MAKES SURE THAT THE COMMUNICATION AREA'S
;          2415  !       RING BUFFERS ARE CLEARED, THEN THE COMMUNICATIONS AREA IS
;          2416  !       INITIALIZED AS FOLLOWS:
;          2417  !
;          2418  !       1. DEFINES FROM THE CONTIGIOUS DATA STORAGE STRUCTURE "COM_AREA"
;          2419  !          THE HEADER AREA ADDRESS, RECEIVE RING ADDRESS AND THE SENDING
;          2420  !          RING ADDRESS.
;          2421  !
;          2422  !       2. CLEARS THE INTERRUPT INDICATORS (RING BASE -1, -2, -3, -4)
;          2423  !          DEFINED AS HEAD_AREA.
;          2424  !
;          2425  !       3. LOADS THE RECEIVE AND SEND DESCRIPTORS WITH THE VALUES:
;          2426  !
;          2427  !          A. ENVELOPE LOW, HIGN AND Q BUS ADDRESS
;          2428  !          B. RESERVED FIELD
;          2429  !          C. FLAG BIT
;          2430  !          D. OWENERSHIP BIT
;          2431  !
;          2432  !       4. LOAD THE RECEIVE ENVELOPE MESSAGE LENGTH WITH THE BUFFER SIZE
;          2433  !          IN BYTES.
;          2434  !
;          2435  ! FORMAL PARAMETERS:
;          2436  !       -NONE -
;          2437  !
;          2438  ! IMPLICIT INPUTS:
;          2439  !       HEAD_AREA, RECEIVE_RING, SENDING_RING, COM_AREA
;          2440  !
;          2441  ! IMPLICIT OUTPUTS:
;          2442  !       AS A RESULT OF THIS ROUTINE THE COMMUNICATION AREA WILL
;          2443  !       BE INITIALIZED.
;          2444  !
;          2445  ! COMPLETTION CODES:
;          2446  !       FAL_CODE : INDICATE AN ERROR HAS OCCURED
;          2447  !       PAS_CODE : INDICATE NO ERROR
;          2448  !
;          2449  ! SIDE EFFECTS:
;          2450  !       - NONE -
;          2451  !--
;          2452
;          2453          begin
;          2454
;          2455          incru I from 0 to RING_SIZE - 1 do      ! TEST RING AREA FOR ZEROS
;          2456
;          2457              incru J from 0 to 1 do              !
;          2458
;          2459                  if .COM_AREA [.I, .J, WORD_REF] nequ 0  ! IF RING AREA IS NOT CLEAR
;          2460                  then                                    ! THEN
;          2461                      begin
;          2462                      RET_STATUS = FAL_CODE;
;          2463                      return .RET_STATUS;                 ! RETURN WITH ERROR CODE SET
```

```
2464                    end;
2465
2466    !
2467    ! DEFINE THE ADDRESS LOCATIONS OF THE HEAD_AREA, RECEIVE_RING
2468    ! AND SEND_RING.
2469    !
2470            HEAD_AREA = COM_AREA;                   ! DEFINE THE HEADER AREA
2471            RECEIVE_RING = COM_AREA [REC_BASE];     ! DEFINE THE RESPONSE RING AREA
2472            SEND_RING = COM_AREA [SND_BASE];        ! DEFINE THE COMMAND RING AREA
2473
2474            incru I from WORD0 to WORD3 do          ! CLEAR THE HEADER AREA
2475                HEAD_AREA [.I, WORD_REF] = ZERO;    !
2476
2477    !+
2478    ! LOAD UP THE COMMAND RING DESCRIPTORS WITH AN ENVELOPE ADDRESS,
2479    ! DEFINE THE "FLAG BIT" TO = 1 (INTERRUPT REQUESTED), DEFINE THE
2480    ! "OWNERSHIP BIT" TO ZERO (OWENED BY HOST) AND LOAD THE RESERVED
2481    ! FIELD WITH ZERO.
2482    !-
2483
2484            incru I from 0 to SND_ALLOCATE - 1 do   !
2485                begin
2486                SEND_RING [.I, LO_EN$AD] = SND_ENVELOPE [.I, CMD_LREF];     ! LO-ORDER SEND ENVELOPE ADDR
2487                SEND_RING [.I, HI_EN$AD] = ZERO;    ! HI-ORDER SEND ENVELOPE ADDR
2488                SEND_RING [.I, QB_EXT] = ZERO;      ! HI-ORDER PORTION OF UNIBUS
2489                SEND_RING [.I, D_RSVD] = ZERO;      ! OR Q-BUS ADDRESS
2490                SEND_RING [.I, FLAG_BIT] = ZERO;    ! FLAG BIT, 1=INT. REQUESTED
2491                SEND_RING [.I, OWN_BIT] = ZERO;     ! OWNERSHIP BIT, 0=OWNED BY HO
2492                end;
2493
2494    !+
2495    ! LOAD UP THE RESPONSE RING DESCRIPTORS WITH AN ENVELOPE ADDRESS,
2496    ! DEFINE THE "OWNERSHIP BIT" = 1 (OWNED BY PORT) DEFINE THE "FLAG
2497    ! BIT" TO = 1 (INTERRUPT REQUESTED) AND THE RESERVED FIELD SET TO
2498    ! ZEROS.
2499    !-
2500
2501            incru I from 0 to REC_ALLOCATE - 1 do
2502                begin
2503                RECEIVE_RING [.I, LO_EN$AD] = REC_ENVELOPE [.I, CMD_LREF];  ! LO-ORDER SEC ENVELOPE ADDR
2504                RECEIVE_RING [.I, HI_EN$AD] = ZERO; ! HI-ORDER COMMAND ENV. ADDR
2505                RECEIVE_RING [.I, QB_EXT] = ZERO;   ! HI-ORDER PORTION OF UQ ADDR
2506                RECEIVE_RING [.I, D_RSVD] = ZERO;   ! RESERVED
2507                RECEIVE_RING [.I, FLAG_BIT] = ZERO; !
2508                RECEIVE_RING [.I, OWN_BIT] = ONE;   ! OWENER BIT, 1=OWNED BY PORT
2509                end;
2510
2511    !
2512    ! SET THE RESPONSE ENVELOPE MESSAGE LENGTH
2513    !
2514
2515            incru I from 0 to REC_ALLOCATE - 1 do
2516                REC_ENVELOPE [.I, MSG_LENGTH] = RB_SIZE*2;  ! CONVERT TO BYTES BEFORE
2517
2518            RET_STATUS = PAS_CODE;                  ! LOADING
2519            return .RET_STATUS;
2520            end;
```

```
                                        .SBTTL  INIT.COM.AREA COMMUNICATION RING INITIALIZATION
000000  004167  000000G        INIT.COM.AREA::
                                        JSR     R1,$SAVE2                       ;                               2410
000004  005002                          CLR     R2                              ; I                             2455
000006  005001                  1$:     CLR     R1                              ; J                             2457
000010  010200                  2$:     MOV     R2,R0                           ; I,*                           2459
000012  006300                          ASL     R0
000014  060100                          ADD     R1,R0                           ; J,*
000016  006300                          ASL     R0
000020  005760  000000G                TST     COM.AREA(R0)
000024  001406                          BEQ     3$
000026  012767  000000G 000000G        MOV     #FAL.CODE,RET.STATUS            ;                               2462
000034  016700  000000G                MOV     RET.STATUS,R0                   ;                               2461
000040  000207                          RTS     PC
000042  005201                  3$:     INC     R1                              ; J                             2457
000044  020127  000001                CMP     R1,#1                           ; J,*
000050  101757                          BLOS    2$
000052  005202                          INC     R2                              ; I                             2455
000054  020227  000037                CMP     R2,#37                          ; I,*
000060  101752                          BLOS    1$
000062  012767  000000G 000000G        MOV     #COM.AREA,HEAD.AREA             ;                               2470
000070  012767  000010G 000000G        MOV     #COM.AREA+10,RECEIVE.RING       ;                               2471
000076  012767  000110G 000000G        MOV     #COM.AREA+110,SEND.RING         ;                               2472
000104  005000                          CLR     R0                              ; I                             2474
000106  010001                  4$:     MOV     R0,R1                           ; I,*                           2475
000110  006301                          ASL     R1
000112  066701  000000G                ADD     HEAD.AREA,R1
000116  005011                          CLR     (R1)
000120  005200                          INC     R0                              ; I                             2474
000122  020027  000003                CMP     R0,#3                           ; I,*
000126  101767                          BLOS    4$
000130  005002                          CLR     R2                              ; I                             2484
000132  010201                  5$:     MOV     R2,R1                           ; I,*                           2486
000134  006301                          ASL     R1
000136  006301                          ASL     R1
000140  066701  000000G                ADD     SEND.RING,R1
000144  010246                          MOV     R2,-(SP)                        ; I,*
000146  012746  000054                MOV     #54,-(SP)
000152  004767  000000G                JSR     PC,BL$MUL
000156  062700  000004G                ADD     #SND.ENVELOPE+4,R0
000162  010011                          MOV     R0,(R1)
000164  010200                          MOV     R2,R0                           ; I,*                           2487
000166  006300                          ASL     R0
000170  006300                          ASL     R0
000172  066700  000000G                ADD     SEND.RING,R0
000176  142760  000003  000002        BICB    #3,2(R0)
000204  010200                          MOV     R2,R0                           ; I,*                           2488
000206  006300                          ASL     R0
000210  006300                          ASL     R0
000212  066700  000000G                ADD     SEND.RING,R0
000216  142760  000074  000002        BICB    #74,2(R0)
000224  010200                          MOV     R2,R0                           ; I,*                           2489
000226  006300                          ASL     R0
000230  006300                          ASL     R0
000232  066700  000000G                ADD     SEND.RING,R0
```

```
000236  042760  037700  000002          BIC     #37700,2(R0)
000244  010200                          MOV     R2,R0                   ; I,*                   2490
000246  006300                          ASL     R0
000250  006300                          ASL     R0
000252  066700  000000G                 ADD     SEND.RING,R0
000256  042760  040000  000002          BIC     #40000,2(R0)
000264  010200                          MOV     R2,R0                   ; I,*                   2491
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)+             ;                       2485
000306  005202                          INC     R2                      ; I                     2484
000310  020227  000017                  CMP     R2,#17                  ; I,*
000314  101706                          BLOS    5$
000316  005002                          CLR     R2                      ; I                     2501
000320  010201                  6$:     MOV     R2,R1                   ; I,*                   2503
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,*                   2504
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,*                   2505
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,*                   2506
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,*                   2507
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,*                   2508
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                     2501
000500  020227  000017                  CMP     R2,#17                  ; I,*
000504  101705                          BLOS    6$
000506  005001                          CLR     R1                      ; I                     2515
```

```
000510  010100                          7$:     MOV     R1,R0                           ; I,*                           2516
000512  000300                                  SWAB    R0
000514  106000                                  RORB    R0
000516  006000                                  ROR     R0
000520  006000                                  ROR     R0
000522  142700  000077                          BICB    #77,R0
000526  012760  000074  000000G                MOV     #74,REC.ENVELOPE(R0)
000534  005201                                  INC     R1                              ; I                             2515
000536  020127  000017                          CMP     R1,#17                          ; I,*
000542  101762                                  BLOS    7$
000544  005067  000000G                        CLR     RET.STATUS                      ;                               2518
000550  016700  000000G                        MOV     RET.STATUS,R0                   ;                               2453
000554  000207                                  RTS     PC                              ;                               2410

; Routine Size:  183 words,    Routine Base:  AB$CODE + 2426
; Maximum stack depth per invocation:  6 words


;         2521
```

```
;       2522  %sbttl 'AZTEC GLOBAL ROUTINE'
;       2523  !
;       2524
;       2525      global routine EX_SUP_PRG =
;       2526
;       2527  !++
;       2528  ! FUNCTIONAL DESCRIPTION :
;       2529  !
;       2530  !       THIS CAMMNAD CAUSES THE SERVER TO TRANSFER THE PROGRAM FROM HOST
;       2531  !       MEMORY TO AN AREA IN THE CONTROLLER AND START IT EXECUTION.
;       2532  !
;       2533  ! FORMAL PARAMETERS :
;       2534  !
;       2535  ! IMPLICIT INPUTS :   BUF_DESCRPTR
;       2536  !
;       2537  ! IMPLICIT OUTPUTS : RET_STATUS
;       2538  ! SIDE EFFECTS :
;       2539  !
;       2540  !--
;       2541
;       2542          begin
;       2543
;       2544          local
;       2545              TEMP;
;       2546
;       2547          !
;       2548          ! THE INTERRUPT ROUTINE WILL SET THE FLAG CANCEL_TIMER WHEN CALLED.
;       2549          ! CLEAR THE FLAG HERE TO INSURE THE DETECTION OF THE INTERRUPT.
;       2550          !
;       2551          I_AM_NEX = ZERO;
;       2552          !
;       2553
;       2554          ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
;       2555          !
;       2556          SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_ESP;
;       2557          SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;
;       2558          SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0;
;       2559          SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 2;
;       2560          !
;       2561          ! DUP COMMAND ENVELOPE FIELD DEFINITION
;       2562          !
;       2563          SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;
;       2564          SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;
;       2565          SND_ENVELOPE [.CMD_SLOT, UN_[USED] = ZERO;
;       2566          SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;
;       2567          SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_ESP;
;       2568          SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;
;       2569          SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;
;       2570          SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT;        ! BYTE COUNT LOW WORD
;       2571          SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO;       ! BYTE COUNT HIGH WORD
;       2572          SND_ENVELOPE [.CMD_SLOT, BD_0] = .BUF_DESCRPTR; ! BUFFER DESCRIPTOR WORD 0
;       2573          SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO;  ! BUFFER DESCRIPTOR WORD 1
;       2574          SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO;  ! BUFFER DESCRIPTOR WORD 2
;       2575          SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO;  ! BUFFER DESCRIPTOR WORD 3
;       2576          SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO;  ! BUFFER DESCRIPTOR WORD 4
;       2577          SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO;  ! BUFFER DESCRIPTOR WORD 5
;       2578          SND_ENVELOPE [.CMD_SLOT, OBD_0] = ZERO; ! BUFFER DESCRIPTOR WORD 0
```

```
;        2579          SND_ENVELOPE [.CMD_SLOT, OBD_1] = ZERO; ! BUFFER DESCRIPTOR WORD 1
;        2580          SND_ENVELOPE [.CMD_SLOT, OBD_2] = ZERO; ! BUFFER DESCRIPTOR WORD 2
;        2581          SND_ENVELOPE [.CMD_SLOT, OBD_3] = ZERO; ! BUFFER DESCRIPTOR WORD 3
;        2582          SND_ENVELOPE [.CMD_SLOT, OBD_4] = ZERO; ! BUFFER DESCRIPTOR WORD 4
;        2583          SND_ENVELOPE [.CMD_SLOT, OBD_5] = ZERO; ! BUFFER DESCRIPTOR WORD 5
;        2584          !
;        2585          ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;        2586          !
;        2587          SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;        2588          !
;        2589          ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
;        2590          !
;        2591          TEMP = .RC25_ADDR [RCIP, RC_ALL];
;        2592          !
;        2593          ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
;        2594          !
;        2595          GET_CMD_SLOT ();
;        2596          !
;        2597          !
;        2598          DELAY (1);
;        2599          !
;        2600          ! CHECK THE END PACKET FOR GOOD STATUS
;        2601          !
;        2602          return REC_STATUS ();                   !RETURN THE STATUS
;        2603          end;

                                        .SBTTL  EX.SUP.PRG AZTEC GLOBAL ROUTINE
000000  010146                  EX.SUP.PRG::
                                        MOV     R1,-(SP)                        ;                       2525
000002  024646                          CMP     -(SP),-(SP)
000004  005067  000000G                 CLR     I.AM.NEX                        ;                       2551
000010  016746  000000G                 MOV     CMD.SLOT,-(SP)                  ;                       2556
000014  012746  000054                  MOV     #54,-(SP)
000020  004767  000000G                 JSR     PC,BL$MUL
000024  012760  000050  000000G         MOV     #50,SND.ENVELOPE(R0)
000032  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2557
000036  012746  000054                  MOV     #54,-(SP)
000042  004767  000000G                 JSR     PC,BL$MUL
000046  142760  000017  000002G         BICB    #17,SND.ENVELOPE+2(R0)
000054  152760  000001  000002G         BISB    #1,SND.ENVELOPE+2(R0)
000062  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2558
000066  012746  000054                  MOV     #54,-(SP)
000072  004767  000000G                 JSR     PC,BL$MUL
000076  142760  000360  000002G         BICB    #360,SND.ENVELOPE+2(R0)
000104  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2559
000110  012746  000054                  MOV     #54,-(SP)
000114  004767  000000G                 JSR     PC,BL$MUL
000120  112760  000002  000003G         MOVB    #2,SND.ENVELOPE+3(R0)
000126  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2563
000132  012746  000054                  MOV     #54,-(SP)
000136  004767  000000G                 JSR     PC,BL$MUL
000142  016760  000000G  000004G        MOV     CMD.REF,SND.ENVELOPE+4(R0)
000150  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2564
000154  012746  000054                  MOV     #54,-(SP)
000160  004767  000000G                 JSR     PC,BL$MUL
000164  005060  000006G                 CLR     SND.ENVELOPE+6(R0)
```

```
000170  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2565
000174  012746  000054                   MOV     #54,-(SP)
000200  004767  000000G                 JSR     PC,BL$MUL
000204  005060  000010G                 CLR     SND.ENVELOPE+10(R0)
000210  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2566
000214  012746  000054                   MOV     #54,-(SP)
000220  004767  000000G                 JSR     PC,BL$MUL
000224  005060  000012G                 CLR     SND.ENVELOPE+12(R0)
000230  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2567
000234  012746  000054                   MOV     #54,-(SP)
000240  004767  000000G                 JSR     PC,BL$MUL
000244  112760  000002  000014G         MOVB    #2,SND.ENVELOPE+14(R0)
000252  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2568
000256  012746  000054                   MOV     #54,-(SP)
000262  004767  000000G                 JSR     PC,BL$MUL
000266  105060  000015G                 CLRB    SND.ENVELOPE+15(R0)
000272  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2569
000276  012746  000054                   MOV     #54,-(SP)
000302  004767  000000G                 JSR     PC,BL$MUL
000306  005060  000016G                 CLR     SND.ENVELOPE+16(R0)
000312  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2570
000316  012746  000054                   MOV     #54,-(SP)
000322  004767  000000G                 JSR     PC,BL$MUL
000326  016760  000000G  000020G         MOV     BYTE.COUNT,SND.ENVELOPE+20(R0)
000334  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2571
000340  012746  000054                   MOV     #54,-(SP)
000344  004767  000000G                 JSR     PC,BL$MUL
000350  005060  000022G                 CLR     SND.ENVELOPE+22(R0)
000354  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2572
000360  012746  000054                   MOV     #54,-(SP)
000364  004767  000000G                 JSR     PC,BL$MUL
000370  016760  000000G  000024G         MOV     BUF.DESCRPTR,SND.ENVELOPE+24(R0)
000376  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2573
000402  012746  000054                   MOV     #54,-(SP)
000406  004767  000000G                 JSR     PC,BL$MUL
000412  005060  000026G                 CLR     SND.ENVELOPE+26(R0)
000416  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2574
000422  012746  000054                   MOV     #54,-(SP)
000426  004767  000000G                 JSR     PC,BL$MUL
000432  005060  000030G                 CLR     SND.ENVELOPE+30(R0)
000436  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2575
000442  012746  000054                   MOV     #54,-(SP)
000446  004767  000000G                 JSR     PC,BL$MUL
000452  005060  000032G                 CLR     SND.ENVELOPE+32(R0)
000456  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2576
000462  012746  000054                   MOV     #54,-(SP)
000466  004767  000000G                 JSR     PC,BL$MUL
000472  005060  000034G                 CLR     SND.ENVELOPE+34(R0)
000476  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2577
000502  012746  000054                   MOV     #54,-(SP)
000506  004767  000000G                 JSR     PC,BL$MUL
000512  005060  000036G                 CLR     SND.ENVELOPE+36(R0)
000516  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2578
000522  012746  000054                   MOV     #54,-(SP)
000526  004767  000000G                 JSR     PC,BL$MUL
000532  005060  000040G                 CLR     SND.ENVELOPE+40(R0)
000536  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       2579
```

```
000542  012746  000054                          MOV     #54,-(SP)
000546  004767  000000G                         JSR     PC,BL$MUL
000552  005060  000042G                         CLR     SND.ENVELOPE+42(R0)
000556  016716  000000G                         MOV     CMD.SLOT,(SP)           ;                                       2580
000562  012746  000054                          MOV     #54,-(SP)
000566  004767  000000G                         JSR     PC,BL$MUL
000572  005060  000044G                         CLR     SND.ENVELOPE+44(R0)
000576  016716  000000G                         MOV     CMD.SLOT,(SP)           ;                                       2581
000602  012746  000054                          MOV     #54,-(SP)
000606  004767  000000G                         JSR     PC,BL$MUL
000612  005060  000046G                         CLR     SND.ENVELOPE+46(R0)
000616  016716  000000G                         MOV     CMD.SLOT,(SP)           ;                                       2582
000622  012746  000054                          MOV     #54,-(SP)
000626  004767  000000G                         JSR     PC,BL$MUL
000632  005060  000050G                         CLR     SND.ENVELOPE+50(R0)
000636  016716  000000G                         MOV     CMD.SLOT,(SP)           ;                                       2583
000642  012746  000054                          MOV     #54,-(SP)
000646  004767  000000G                         JSR     PC,BL$MUL
000652  005060  000052G                         CLR     SND.ENVELOPE+52(R0)
000656  016700  000000G                         MOV     CMD.SLOT,R0             ;                                       2587
000662  006300                                  ASL     R0
000664  006300                                  ASL     R0
000666  066700  000000G                         ADD     SEND.RING,R0
000672  052760  100000  000002                  BIS     #100000,2(R0)
000700  017766  000000G 000064                  MOV     @RC25.ADDR,64(SP)       ; *,RC.REG                              2591
000706  016600  000064                          MOV     64(SP),R0               ; RC.REG,TEMP
000712  004767  000000V                         JSR     PC,GET.CMD.SLOT         ;                                       2595
000716  012701  000001                          MOV     #1,R1                   ; *,$$TMP2                              2598
000722  001411                          1$:     BEQ     4$
000724  016700  000000G                         MOV     L$DLY,R0                ; *,$$TMP1
000730  001404                                  BEQ     3$
000732  005066  000066                  2$:     CLR     66(SP)                  ; $$TMP
000736  005300                                  DEC     R0                      ; $$TMP1
000740  001374                                  BNE     2$
000742  005301                          3$:     DEC     R1                      ; $$TMP2
000744  000766                                  BR      1$
000746  004767  000000V                 4$:     JSR     PC,REC.STATUS           ;                                       2602
000752  062706  000070                          ADD     #70,SP                  ;                                       2525
000756  012601                                  MOV     (SP)+,R1
000760  000207                                  RTS     PC

; Routine Size:  249 words,     Routine Base:  AB$CODE + 3204
; Maximum stack depth per invocation:  30 words


;          2604
;          2605  !
;          2606
```

```
;        2607
;        2608      global routine SEND_DATA =
;        2609
;        2610  !++
;        2611  ! FUNCTIONAL DESCRIPTION:
;        2612  !
;        2613  !       SEND_DATA PROVIDES BUF_ADDRESS AND SIZE TO THE DM CODE
;        2614  !       PROGRAM FOR POSSIBLE READ, WRITE TO MEMORY. PARAMETERS
;        2615  !       NEEDED ARE BUF_LENGTH, H_SADD, E_SADD
;        2616  !
;        2617  ! FORMAL PARAMETERS :
;        2618  !
;        2619  ! IMPLICIT INPUTS : H_SADD, E_SADD, BUF_LENGTH
;        2620  !
;        2621  ! IMPLICIT OUTPUTS : RET_STATUS
;        2622  !
;        2623  ! SIDE EFFECTS :
;        2624  !
;        2625  !--
;        2626
;        2627          begin
;        2628
;        2629          local
;        2630              TEMP;
;        2631
;        2632          !
;        2633          ! THE INTERRUPT ROUTINE WILL SET THE FLAG CANCEL_TIMER WHEN CALLED.
;        2634          ! CLEAR THE FLAG HERE TO INSURE THE DETECTION OF THE INTERRUPT.
;        2635          !
;        2636          I_AM_NEX = ZERO;
;        2637          !
;        2638
;        2639          ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
;        2640          !
;        2641          SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_SED;
;        2642          SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;
;        2643          SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0;
;        2644          SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 2;
;        2645          !
;        2646          ! DUP COMMAND ENVELOPE FIELD DEFINITION
;        2647          !
;        2648          SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;
;        2649          SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;
;        2650          SND_ENVELOPE [.CMD_SLOT, UN_[USED] = ZERO;
;        2651          SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;
;        2652          SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_SED;
;        2653          SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;
;        2654          SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;
;        2655          SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT;        ! BYTE COUNT LOW WORD
;        2656          SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO;       ! BYTE COUNT HIGH WORD
;        2657          SND_ENVELOPE [.CMD_SLOT, BD_0] = .BUF_DESCRPTR; ! BUFFER DESCRIPTOR WORD 0
;        2658          SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO;  ! BUFFER DESCRIPTOR WORD 1
;        2659          SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO;  ! BUFFER DESCRIPTOR WORD 2
;        2660          SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO;  ! BUFFER DESCRIPTOR WORD 3
;        2661          SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO;  ! BUFFER DESCRIPTOR WORD 4
;        2662          SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO;  ! BUFFER DESCRIPTOR WORD 5
;        2663          !
```

```
;          2664          ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;          2665          !
;          2666          SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;          2667          !
;          2668          ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
;          2669          !
;          2670          TEMP = .RC25_ADDR [RCIP, RC_ALL];
;          2671          !
;          2672          ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
;          2673          !
;          2674          GET_CMD_SLOT ();
;          2675          !
;          2676          DELAY (1);
;          2677          !
;          2678          ! CHECK THE END PACKET FOR GOOD STATUS
;          2679          !
;          2680          return REC_STATUS ();                    ! RETURN THE STATUS
;          2681          end;


                                        .SBTTL  SEND.DATA AZTEC GLOBAL ROUTINE
000000  010146                    SEND.DATA::
                                        MOV     R1,-(SP)                        ;                       2608
000002  024646                          CMP     -(SP),-(SP)
000004  005067  000000G                 CLR     I.AM.NEX                        ;                       2636
000010  016746  000000G                 MOV     CMD.SLOT,-(SP)                  ;                       2641
000014  012746  000054                  MOV     #54,-(SP)
000020  004767  000000G                 JSR     PC,BL$MUL
000024  012760  000034  000000G         MOV     #34,SND.ENVELOPE(R0)
000032  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2642
000036  012746  000054                  MOV     #54,-(SP)
000042  004767  000000G                 JSR     PC,BL$MUL
000046  142760  000017  000002G         BICB    #17,SND.ENVELOPE+2(R0)
000054  152760  000001  000002G         BISB    #1,SND.ENVELOPE+2(R0)
000062  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2643
000066  012746  000054                  MOV     #54,-(SP)
000072  004767  000000G                 JSR     PC,BL$MUL
000076  142760  000360  000002G         BICB    #360,SND.ENVELOPE+2(R0)
000104  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2644
000110  012746  000054                  MOV     #54,-(SP)
000114  004767  000000G                 JSR     PC,BL$MUL
000120  112760  000002  000003G         MOVB    #2,SND.ENVELOPE+3(R0)
000126  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2648
000132  012746  000054                  MOV     #54,-(SP)
000136  004767  000000G                 JSR     PC,BL$MUL
000142  016760  000000G  000004G       MOV     CMD.REF,SND.ENVELOPE+4(R0)
000150  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2649
000154  012746  000054                  MOV     #54,-(SP)
000160  004767  000000G                 JSR     PC,BL$MUL
000164  005060  000006G                 CLR     SND.ENVELOPE+6(R0)
000170  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2650
000174  012746  000054                  MOV     #54,-(SP)
000200  004767  000000G                 JSR     PC,BL$MUL
000204  005060  000010G                 CLR     SND.ENVELOPE+10(R0)
000210  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2651
000214  012746  000054                  MOV     #54,-(SP)
000220  004767  000000G                 JSR     PC,BL$MUL
```

```
000224  005060  000012G                        CLR     SND.ENVELOPE+12(R0)
000230  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2652
000234  012746  000054                         MOV     #54,-(SP)
000240  004767  000000G                        JSR     PC,BL$MUL
000244  112760  000004  000014G                MOVB    #4,SND.ENVELOPE+14(R0)
000252  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2653
000256  012746  000054                         MOV     #54,-(SP)
000262  004767  000000G                        JSR     PC,BL$MUL
000266  105060  000015G                        CLRB    SND.ENVELOPE+15(R0)
000272  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2654
000276  012746  000054                         MOV     #54,-(SP)
000302  004767  000000G                        JSR     PC,BL$MUL
000306  005060  000016G                        CLR     SND.ENVELOPE+16(R0)
000312  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2655
000316  012746  000054                         MOV     #54,-(SP)
000322  004767  000000G                        JSR     PC,BL$MUL
000326  016760  000000G 000020G                MOV     BYTE.COUNT,SND.ENVELOPE+20(R0)
000334  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2656
000340  012746  000054                         MOV     #54,-(SP)
000344  004767  000000G                        JSR     PC,BL$MUL
000350  005060  000022G                        CLR     SND.ENVELOPE+22(R0)
000354  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2657
000360  012746  000054                         MOV     #54,-(SP)
000364  004767  000000G                        JSR     PC,BL$MUL
000370  016760  000000G 000024G                MOV     BUF.DESCRPTR,SND.ENVELOPE+24(R0)
000376  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2658
000402  012746  000054                         MOV     #54,-(SP)
000406  004767  000000G                        JSR     PC,BL$MUL
000412  005060  000026G                        CLR     SND.ENVELOPE+26(R0)
000416  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2659
000422  012746  000054                         MOV     #54,-(SP)
000426  004767  000000G                        JSR     PC,BL$MUL
000432  005060  000030G                        CLR     SND.ENVELOPE+30(R0)
000436  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2660
000442  012746  000054                         MOV     #54,-(SP)
000446  004767  000000G                        JSR     PC,BL$MUL
000452  005060  000032G                        CLR     SND.ENVELOPE+32(R0)
000456  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2661
000462  012746  000054                         MOV     #54,-(SP)
000466  004767  000000G                        JSR     PC,BL$MUL
000472  005060  000034G                        CLR     SND.ENVELOPE+34(R0)
000476  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2662
000502  012746  000054                         MOV     #54,-(SP)
000506  004767  000000G                        JSR     PC,BL$MUL
000512  005060  000036G                        CLR     SND.ENVELOPE+36(R0)
000516  016700  000000G                        MOV     CMD.SLOT,R0                     ;                                       2666
000522  006300                                 ASL     R0
000524  006300                                 ASL     R0
000526  066700  000000G                        ADD     SEND.RING,R0
000532  052760  100000  000002                BIS     #100000,2(R0)
000540  017766  000000G 000050                MOV     @RC25.ADDR,50(SP)               ; *,RC.REG                              2670
000546  016600  000050                         MOV     50(SP),R0                       ; RC.REG,TEMP
000552  004767  000000V                        JSR     PC,GET.CMD.SLOT                 ;                                       2674
000556  012701  000001                         MOV     #1,R1                           ; *,$$TMP2                              2676
000562  001411                         1$:     BEQ     4$
000564  016700  000000G                        MOV     L$DLY,R0                        ; *,$$TMP1
000570  001404                                 BEQ     3$
```

```
000572  005066  000052                  2$:     CLR     52(SP)                          ; $$TMP
000576  005300                                  DEC     R0                              ; $$TMP1
000600  001374                                  BNE     2$
000602  005301                          3$:     DEC     R1                              ; $$TMP2
000604  000766                                  BR      1$
000606  004767  000000V                 4$:     JSR     PC,REC.STATUS                   ;                                       2680
000612  062706  000054                          ADD     #54,SP                          ;                                       2608
000616  012601                                  MOV     (SP)+,R1
000620  000207                                  RTS     PC

; Routine Size:  201 words,     Routine Base:  AB$CODE + 4166
; Maximum stack depth per invocation:  24 words


;         2682
;         2683  !
```

```
;        2684      global routine REC_DATA =
;        2685
;        2686  !++
;        2687  ! FUNCTION DESCRIPTION :
;        2688  !
;        2689  !       THE REMOTE PROGRAM WRITES DATA INTO THE BUFFER UP TO THE AMOUNT
;        2690  !       SPECIFIED BY THE BYTE COUNT AND RETURNS STATUS. THE STATUS IS
;        2691  !       RECEIVED BY THE USE OF THIS COMMAND. BUF_DESCRPTR POINTS TO
;        2692  !       THE WORD RECEIVED DM CODE.
;        2693  !
;        2694  ! FORMAL PARAMETERS :
;        2695  !
;        2696  ! IMPLICIT INPUTS :  BUF_DESCRPTR
;        2697  !
;        2698  ! IMPLICIT OUTPUTS : RET_STATUS
;        2699  !
;        2700  ! SIDE EFFECTS :
;        2701  !
;        2702  !--
;        2703
;        2704          begin
;        2705
;        2706          local
;        2707              TEMP;
;        2708
;        2709          !
;        2710          ! THE INTERRUPT ROUTINE WILL SET THE FLAG CANCEL_TIMER WHEN CALLED.
;        2711          ! CLEAR THE FLAG HERE TO INSURE THE DETECTION OF THE INTERRUPT.
;        2712          !
;        2713          I_AM_NEX = ZERO;
;        2714          !
;        2715
;        2716          ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
;        2717          !
;        2718          SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_RED;
;        2719          SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;
;        2720          SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0;
;        2721          SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 2;
;        2722          !
;        2723          ! DUP COMMAND ENVELOPE FIELD DEFINITION
;        2724          !
;        2725          SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;
;        2726          SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;
;        2727          SND_ENVELOPE [.CMD_SLOT, UN_[USED] = ZERO;
;        2728          SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;
;        2729          SND_ENVELOPE [.CMD_SLOT, OP[ODE] = OP_RED;
;        2730          SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;
;        2731          SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;
;        2732          SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT;        ! BYTE COUNT LOW WORD
;        2733          SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO;       ! BYTE COUNT HIGH WORD
;        2734          SND_ENVELOPE [.CMD_SLOT, BD_0] = .BUF_DESCRPTR; ! BUFFER DESCRIPTOR WORD 0
;        2735          SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO;  ! BUFFER DESCRIPTOR WORD 1
;        2736          SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO;  ! BUFFER DESCRIPTOR WORD 2
;        2737          SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO;  ! BUFFER DESCRIPTOR WORD 3
;        2738          SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO;  ! BUFFER DESCRIPTOR WORD 4
;        2739          SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO;  ! BUFFER DESCRIPTOR WORD 5
;        2740          !
```

```
;         2741          ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;         2742          !
;         2743          SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;         2744          !
;         2745          ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
;         2746          !
;         2747          TEMP = .RC25_ADDR [RCIP, RC_ALL];
;         2748          !
;         2749          ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
;         2750          !
;         2751          GET_CMD_SLOT ();
;         2752          !
;         2753          DELAY (1);
;         2754          !
;         2755          ! CHECK THE END PACKET FOR GOOD STATUS
;         2756          !
;         2757          return REC_STATUS ();                    ! RETURN THE STATUS
;         2758          end;


                                        .SBTTL  REC.DATA AZTEC GLOBAL ROUTINE
000000  010146                    REC.DATA::
                                        MOV     R1,-(SP)                        ;                       2684
000002  024646                          CMP     -(SP),-(SP)
000004  005067  000000G                 CLR     I.AM.NEX                        ;                       2713
000010  016746  000000G                 MOV     CMD.SLOT,-(SP)                  ;                       2718
000014  012746  000054                  MOV     #54,-(SP)
000020  004767  000000G                 JSR     PC,BL$MUL
000024  012760  000034  000000G         MOV     #34,SND.ENVELOPE(R0)
000032  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2719
000036  012746  000054                  MOV     #54,-(SP)
000042  004767  000000G                 JSR     PC,BL$MUL
000046  142760  000017  000002G         BICB    #17,SND.ENVELOPE+2(R0)
000054  152760  000001  000002G         BISB    #1,SND.ENVELOPE+2(R0)
000062  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2720
000066  012746  000054                  MOV     #54,-(SP)
000072  004767  000000G                 JSR     PC,BL$MUL
000076  142760  000360  000002G         BICB    #360,SND.ENVELOPE+2(R0)
000104  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2721
000110  012746  000054                  MOV     #54,-(SP)
000114  004767  000000G                 JSR     PC,BL$MUL
000120  112760  000002  000003G         MOVB    #2,SND.ENVELOPE+3(R0)
000126  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2725
000132  012746  000054                  MOV     #54,-(SP)
000136  004767  000000G                 JSR     PC,BL$MUL
000142  016760  000000G  000004G        MOV     CMD.REF,SND.ENVELOPE+4(R0)
000150  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2726
000154  012746  000054                  MOV     #54,-(SP)
000160  004767  000000G                 JSR     PC,BL$MUL
000164  005060  000006G                 CLR     SND.ENVELOPE+6(R0)
000170  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2727
000174  012746  000054                  MOV     #54,-(SP)
000200  004767  000000G                 JSR     PC,BL$MUL
000204  005060  000010G                 CLR     SND.ENVELOPE+10(R0)
000210  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2728
000214  012746  000054                  MOV     #54,-(SP)
000220  004767  000000G                 JSR     PC,BL$MUL
```

```
000224  005060  000012G                        CLR     SND.ENVELOPE+12(R0)
000230  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2729
000234  012746  000054                         MOV     #54,-(SP)
000240  004767  000000G                        JSR     PC,BL$MUL
000244  112760  000005  000014G                MOVB    #5,SND.ENVELOPE+14(R0)
000252  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2730
000256  012746  000054                         MOV     #54,-(SP)
000262  004767  000000G                        JSR     PC,BL$MUL
000266  105060  000015G                        CLRB    SND.ENVELOPE+15(R0)
000272  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2731
000276  012746  000054                         MOV     #54,-(SP)
000302  004767  000000G                        JSR     PC,BL$MUL
000306  005060  000016G                        CLR     SND.ENVELOPE+16(R0)
000312  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2732
000316  012746  000054                         MOV     #54,-(SP)
000322  004767  000000G                        JSR     PC,BL$MUL
000326  016760  000000G  000020G                MOV     BYTE.COUNT,SND.ENVELOPE+20(R0)
000334  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2733
000340  012746  000054                         MOV     #54,-(SP)
000344  004767  000000G                        JSR     PC,BL$MUL
000350  005060  000022G                        CLR     SND.ENVELOPE+22(R0)
000354  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2734
000360  012746  000054                         MOV     #54,-(SP)
000364  004767  000000G                        JSR     PC,BL$MUL
000370  016760  000000G  000024G                MOV     BUF.DESCRPTR,SND.ENVELOPE+24(R0)
000376  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2735
000402  012746  000054                         MOV     #54,-(SP)
000406  004767  000000G                        JSR     PC,BL$MUL
000412  005060  000026G                        CLR     SND.ENVELOPE+26(R0)
000416  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2736
000422  012746  000054                         MOV     #54,-(SP)
000426  004767  000000G                        JSR     PC,BL$MUL
000432  005060  000030G                        CLR     SND.ENVELOPE+30(R0)
000436  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2737
000442  012746  000054                         MOV     #54,-(SP)
000446  004767  000000G                        JSR     PC,BL$MUL
000452  005060  000032G                        CLR     SND.ENVELOPE+32(R0)
000456  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2738
000462  012746  000054                         MOV     #54,-(SP)
000466  004767  000000G                        JSR     PC,BL$MUL
000472  005060  000034G                        CLR     SND.ENVELOPE+34(R0)
000476  016716  000000G                        MOV     CMD.SLOT,(SP)                   ;                                       2739
000502  012746  000054                         MOV     #54,-(SP)
000506  004767  000000G                        JSR     PC,BL$MUL
000512  005060  000036G                        CLR     SND.ENVELOPE+36(R0)
000516  016700  000000G                        MOV     CMD.SLOT,R0                     ;                                       2743
000522  006300                                 ASL     R0
000524  006300                                 ASL     R0
000526  066700  000000G                        ADD     SEND.RING,R0
000532  052760  100000  000002                BIS     #100000,2(R0)
000540  017766  000000G  000050                MOV     @RC25.ADDR,50(SP)               ; *,RC.REG                              2747
000546  016600  000050                         MOV     50(SP),R0                       ; RC.REG,TEMP
000552  004767  000000V                        JSR     PC,GET.CMD.SLOT                 ;                                       2751
000556  012701  000001                         MOV     #1,R1                           ; *,$$TMP2                              2753
000562  001411                         1$:     BEQ     4$
000564  016700  000000G                        MOV     L$DLY,R0                        ; *,$$TMP1
000570  001404                                 BEQ     3$
```

```
000572  005066  000052                  2$:     CLR     52(SP)                          ; $$TMP
000576  005300                                  DEC     R0                              ; $$TMP1
000600  001374                                  BNE     2$
000602  005301                          3$:     DEC     R1                              ; $$TMP2
000604  000766                                  BR      1$
000606  004767  000000V                 4$:     JSR     PC,REC.STATUS                   ;                                               2757
000612  062706  000054                          ADD     #54,SP                          ;                                               2684
000616  012601                                  MOV     (SP)+,R1
000620  000207                                  RTS     PC

; Routine Size:  201 words,     Routine Base:  AB$CODE + 5010
; Maximum stack depth per invocation:  24 words


;         2759
;         2760  !
```

```
;        2761      global routine SET_CNTLR_CHAR =
;        2762
;        2763  !++
;        2764  ! FUNCTION DESCRIPTION :
;        2765  !       THE SET CONTROLLER CHARACTER COMMAND IS USED TO SET HOST SETTABLE
;        2766  !       UNIT CHARACTERISTICS AND OBTAIN THOSE UNIT CHARACTERISTICS THAT
;        2767  !       ARE ESSENTIAL FOR PROPER CLASS DRIVER OPERATION.   THIS COMMAND
;        2768  !       NEVER ALTERS THE UNIT'S STATE ('UNIT-ONLINE', 'UNIT-AVAILABLE',
;        2769  !       'UNIT-OFFLINE').
;        2770  !
;        2771  ! FORMAL PARAMETERS :
;        2772  !       - NONE -
;        2773  !
;        2774  ! IMPLICIT INPUTS :
;        2775  !
;        2776  ! INPLICIT OUTPUTS :
;        2777  !       - NONE -
;        2778  !
;        2779  ! COMPLETEDTION CODES :
;        2780  !       RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
;        2781  !
;        2782  !
;        2783  ! SIDE EFFECTS :
;        2784  !       ANY PREVIOUSLY DEFINED CONTROLLER CHARACTERISTICS WILL POSSIBLY
;        2785  !       BE ALTERED AFTERE EXECUTION OF THEIS COMMAND.
;        2786  !--
;        2787
;        2788          begin
;        2789
;        2790          local
;        2791              TEMP;
;        2792
;        2793  !
;        2794  ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
;        2795  !
;        2796          SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_SCC;  ! LOAD MESSAGE LENGTH
;        2797          SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;        ! LOAD CREDIT SIZE
;        2798          SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE 'SEQUENTIAL'
;        2799          SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0;  ! DEFINE CONNECTION ID 'DUP'
;        2800  !
;        2801  ! MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
;        2802  !
;        2803          SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;  !LOAD COMMAND REFERENCE #
;        2804          SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;      ! ZERO HI ORDER CMD REF #
;        2805          SND_ENVELOPE [.CMD_SLOT, UN_[USED] = ZERO;      ! NOT USED IN DUP IMPLIMENT.
;        2806          SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;      ! NOT USED IN DUP IMPLIMENT.
;        2807          SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_SCC;      ! DEFINE COMMAND OPCODE
;        2808          SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;        ! NOT USED
;        2809          SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;      ! DEFINE CMD MODIFIERS
;        2810  !
;        2811  ! COMMAND SPECIFIC COMMAND ENVELOPE FIELD DEFINITION
;        2812  !
;        2813          SND_ENVELOPE [.CMD_SLOT, MSCP_VER] = ZERO;      ! MSCP VERSION
;        2814          SND_ENVELOPE [.CMD_SLOT, CTL_FLAGS] = ZERO;     ! CONTROLLER GLAGS
;        2815          SND_ENVELOPE [.CMD_SLOT, HOST_TOU] = ZERO;      ! HOST TIMEOUT VALUE
;        2816          SND_ENVELOPE [.CMD_SLOT, RS$VD] = ZERO; ! RESERVED
;        2817          SND_ENVELOPE [.CMD_SLOT, T$D_0] = ZERO; ! TIME AND DATE WORD 0
```

```
;        2818          SND_ENVELOPE [.CMD_SLOT, T$D_1] = ZERO; ! TIME AND DATE WORD 1
;        2819          SND_ENVELOPE [.CMD_SLOT, T$D_2] = ZERO; ! TIME AND DATE WORD 2
;        2820          SND_ENVELOPE [.CMD_SLOT, T$D_3] = ZERO; ! TIME AND DATE WORD 3
;        2821          SND_ENVELOPE [.CMD_SLOT, CDP_LO] = ZERO;        ! CNTL DEP PARAMETER LO WORD
;        2822          SND_ENVELOPE [.CMD_SLOT, CDP_HI] = ZERO;        ! CNTL DEP PARAMETER HI WORD
;        2823          !
;        2824          ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;        2825          !
;        2826          SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;        2827          !
;        2828          ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
;        2829          !
;        2830          TEMP = .RC25_ADDR [RCIP, RC_ALL];
;        2831          !
;        2832          ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
;        2833          !
;        2834          GET_CMD_SLOT ();
;        2835          !
;        2836          DELAY (1);
;        2837          !
;        2838          ! CHECK THE END PACKET FOR GOOD STATUS
;        2839          !
;        2840          return REC_STATUS ();                   ! RETURN THE STATUS
;        2841          end;


                                        .SBTTL  SET.CNTLR.CHAR AZTEC GLOBAL ROUTINE
000000  010146                        SET.CNTLR.CHAR::
                                        MOV     R1,-(SP)                        ;                       2761
000002  024646                          CMP     -(SP),-(SP)
000004  016746  000000G                 MOV     CMD.SLOT,-(SP)                  ;                       2796
000010  012746  000054                  MOV     #54,-(SP)
000014  004767  000000G                 JSR     PC,BL$MUL
000020  012760  000040  000000G         MOV     #40,SND.ENVELOPE(R0)
000026  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2797
000032  012746  000054                  MOV     #54,-(SP)
000036  004767  000000G                 JSR     PC,BL$MUL
000042  142760  000017  000002G         BICB    #17,SND.ENVELOPE+2(R0)
000050  152760  000001  000002G         BISB    #1,SND.ENVELOPE+2(R0)
000056  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2798
000062  012746  000054                  MOV     #54,-(SP)
000066  004767  000000G                 JSR     PC,BL$MUL
000072  142760  000360  000002G         BICB    #360,SND.ENVELOPE+2(R0)
000100  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2799
000104  012746  000054                  MOV     #54,-(SP)
000110  004767  000000G                 JSR     PC,BL$MUL
000114  105060  000003G                 CLRB    SND.ENVELOPE+3(R0)
000120  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2803
000124  012746  000054                  MOV     #54,-(SP)
000130  004767  000000G                 JSR     PC,BL$MUL
000134  016760  000000G  000004G        MOV     CMD.REF,SND.ENVELOPE+4(R0)
000142  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2804
000146  012746  000054                  MOV     #54,-(SP)
000152  004767  000000G                 JSR     PC,BL$MUL
000156  005060  000006G                 CLR     SND.ENVELOPE+6(R0)
000162  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2805
000166  012746  000054                  MOV     #54,-(SP)
```

```
000172  004767  000000G                        JSR     PC,BL$MUL
000176  005060  000010G                        CLR     SND.ENVELOPE+10(R0)
000202  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2806
000206  012746  000054                         MOV     #54,-(SP)
000212  004767  000000G                        JSR     PC,BL$MUL
000216  005060  000012G                        CLR     SND.ENVELOPE+12(R0)
000222  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2807
000226  012746  000054                         MOV     #54,-(SP)
000232  004767  000000G                        JSR     PC,BL$MUL
000236  112760  000004  000014G               MOVB    #4,SND.ENVELOPE+14(R0)
000244  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2808
000250  012746  000054                         MOV     #54,-(SP)
000254  004767  000000G                        JSR     PC,BL$MUL
000260  105060  000015G                        CLRB    SND.ENVELOPE+15(R0)
000264  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2809
000270  012746  000054                         MOV     #54,-(SP)
000274  004767  000000G                        JSR     PC,BL$MUL
000300  005060  000016G                        CLR     SND.ENVELOPE+16(R0)
000304  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2813
000310  012746  000054                         MOV     #54,-(SP)
000314  004767  000000G                        JSR     PC,BL$MUL
000320  005060  000020G                        CLR     SND.ENVELOPE+20(R0)
000324  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2814
000330  012746  000054                         MOV     #54,-(SP)
000334  004767  000000G                        JSR     PC,BL$MUL
000340  005060  000022G                        CLR     SND.ENVELOPE+22(R0)
000344  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2815
000350  012746  000054                         MOV     #54,-(SP)
000354  004767  000000G                        JSR     PC,BL$MUL
000360  005060  000024G                        CLR     SND.ENVELOPE+24(R0)
000364  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2816
000370  012746  000054                         MOV     #54,-(SP)
000374  004767  000000G                        JSR     PC,BL$MUL
000400  005060  000026G                        CLR     SND.ENVELOPE+26(R0)
000404  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2817
000410  012746  000054                         MOV     #54,-(SP)
000414  004767  000000G                        JSR     PC,BL$MUL
000420  005060  000030G                        CLR     SND.ENVELOPE+30(R0)
000424  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2818
000430  012746  000054                         MOV     #54,-(SP)
000434  004767  000000G                        JSR     PC,BL$MUL
000440  005060  000032G                        CLR     SND.ENVELOPE+32(R0)
000444  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2819
000450  012746  000054                         MOV     #54,-(SP)
000454  004767  000000G                        JSR     PC,BL$MUL
000460  005060  000034G                        CLR     SND.ENVELOPE+34(R0)
000464  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2820
000470  012746  000054                         MOV     #54,-(SP)
000474  004767  000000G                        JSR     PC,BL$MUL
000500  005060  000036G                        CLR     SND.ENVELOPE+36(R0)
000504  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2821
000510  012746  000054                         MOV     #54,-(SP)
000514  004767  000000G                        JSR     PC,BL$MUL
000520  005060  000040G                        CLR     SND.ENVELOPE+40(R0)
000524  016716  000000G                        MOV     CMD.SLOT,(SP)           ;                                       2822
000530  012746  000054                         MOV     #54,-(SP)
000534  004767  000000G                        JSR     PC,BL$MUL
```

```
000540  005060  000042G                        CLR     SND.ENVELOPE+42(R0)
000544  016700  000000G                        MOV     CMD.SLOT,R0             ;                                       2826
000550  006300                                  ASL     R0
000552  006300                                  ASL     R0
000554  066700  000000G                        ADD     SEND.RING,R0
000560  052760  100000  000002                 BIS     #100000,2(R0)
000566  017766  000000G 000054                 MOV     @RC25.ADDR,54(SP)       ; *,RC.REG                              2830
000574  016600  000054                         MOV     54(SP),R0               ; RC.REG,TEMP
000600  004767  000000V                        JSR     PC,GET.CMD.SLOT         ;                                       2834
000604  012701  000001                         MOV     #1,R1                   ; *,$$TMP2                              2836
000610  001411                          1$:    BEQ     4$
000612  016700  000000G                        MOV     L$DLY,R0                ; *,$$TMP1
000616  001404                                BEQ     3$
000620  005066  000056                  2$:    CLR     56(SP)                  ; $$TMP
000624  005300                                DEC     R0                      ; $$TMP1
000626  001374                                BNE     2$
000630  005301                          3$:    DEC     R1                      ; $$TMP2
000632  000766                                BR      1$
000634  004767  000000V                  4$:    JSR     PC,REC.STATUS           ;                                       2840
000640  062706  000060                        ADD     #60,SP                  ;                                       2761
000644  012601                                MOV     (SP)+,R1
000646  000207                                RTS     PC

; Routine Size:  212 words,     Routine Base:  AB$CODE + 5632
; Maximum stack depth per invocation:  26 words


;         2842
;         2843  !
```

```
    global routine AVAILABLE =

!++
! FUNCTIONAL DESCRIPTION :
!       THE AVAILABLE COMMAND IS USED TO SET THE UNIT-ABAILABLE WHEN
!       ALL OUTSTANDING COMMANDS FOR THE SPECIFIED UNIT ARE COMPLETED.
!       IF THE "SPIN-DOWN" MODIFIER IS SPECIFIED, THE DISK SPINS DOWN
!       AND ITS HEADS ARE UNLOADED.
!
! FORMAL PARAMETERS :
! IMPLICIT INPUTS :
! IMPLICIT OUTPUTS :
! SIDE EFFECTS :
!
!--

        begin

        local
            TEMP;

!
! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
!
        SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_AVL;  ! LOAD MESSAGE LENGTH
        SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;        ! LOAD CREDIT SIZE
        SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE 'SEQUENTIAL'
        SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0;  ! DEFINE CONNECTION ID 'DUP'
!
! MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
!
        SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;  !LOAD COMMAND REFERENCE #
        SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;      ! ZERO HI ORDER CMD REF #
        SND_ENVELOPE [.CMD_SLOT, UN_[USED] = .UNIT;     ! SELECTED UNIT
        SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;      ! NOT USED IN DUP IMPLIMENT.
        SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_AVL;      ! DEFINE COMMAND OPCODE
        SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;        ! NOT USED
        SND_ENVELOPE [.CMD_SLOT, MODIFIER] = MD_SPD;    ! DEFINE CMD MODIFIERS
!
! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
!
        SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
!
! READ THE IP REGISTER TO STIMULATE PORT POLLING.
!
        TEMP = .RC25_ADDR [RCIP, RC_ALL];
!
! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
!
        GET_CMD_SLOT ();
!
!
        DELAY (1);
!
! CHECK THE END PACKET FOR GOOD STATUS
!
        return REC_STATUS ();                   ! RETURN THE STATUS
```

```
;       2901            end;


                                .SBTTL  AVAILABLE AZTEC GLOBAL ROUTINE
000000  010146                AVAILABLE::
                                MOV     R1,-(SP)                        ;                       2844
000002  024646                         CMP     -(SP),-(SP)
000004  016746  000000G                MOV     CMD.SLOT,-(SP)          ;                       2868
000010  012746  000054                 MOV     #54,-(SP)
000014  004767  000000G                JSR     PC,BL$MUL
000020  012760  000014  000000G        MOV     #14,SND.ENVELOPE(R0)
000026  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2869
000032  012746  000054                 MOV     #54,-(SP)
000036  004767  000000G                JSR     PC,BL$MUL
000042  142760  000017  000002G        BICB    #17,SND.ENVELOPE+2(R0)
000050  152760  000001  000002G        BISB    #1,SND.ENVELOPE+2(R0)
000056  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2870
000062  012746  000054                 MOV     #54,-(SP)
000066  004767  000000G                JSR     PC,BL$MUL
000072  142760  000360  000002G        BICB    #360,SND.ENVELOPE+2(R0)
000100  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2871
000104  012746  000054                 MOV     #54,-(SP)
000110  004767  000000G                JSR     PC,BL$MUL
000114  105060  000003G                CLRB    SND.ENVELOPE+3(R0)
000120  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2875
000124  012746  000054                 MOV     #54,-(SP)
000130  004767  000000G                JSR     PC,BL$MUL
000134  016760  000000G  000004G       MOV     CMD.REF,SND.ENVELOPE+4(R0)
000142  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2876
000146  012746  000054                 MOV     #54,-(SP)
000152  004767  000000G                JSR     PC,BL$MUL
000156  005060  000006G                CLR     SND.ENVELOPE+6(R0)
000162  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2877
000166  012746  000054                 MOV     #54,-(SP)
000172  004767  000000G                JSR     PC,BL$MUL
000176  016760  000000G  000010G       MOV     UNIT,SND.ENVELOPE+10(R0)
000204  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2878
000210  012746  000054                 MOV     #54,-(SP)
000214  004767  000000G                JSR     PC,BL$MUL
000220  005060  000012G                CLR     SND.ENVELOPE+12(R0)
000224  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2879
000230  012746  000054                 MOV     #54,-(SP)
000234  004767  000000G                JSR     PC,BL$MUL
000240  112760  000010  000014G       MOVB    #10,SND.ENVELOPE+14(R0)
000246  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2880
000252  012746  000054                 MOV     #54,-(SP)
000256  004767  000000G                JSR     PC,BL$MUL
000262  105060  000015G                CLRB    SND.ENVELOPE+15(R0)
000266  016716  000000G                MOV     CMD.SLOT,(SP)           ;                       2881
000272  012746  000054                 MOV     #54,-(SP)
000276  004767  000000G                JSR     PC,BL$MUL
000302  012760  000001  000016G       MOV     #1,SND.ENVELOPE+16(R0)
000310  016700  000000G                MOV     CMD.SLOT,R0             ;                       2885
000314  006300                         ASL     R0
000316  006300                         ASL     R0
000320  066700  000000G                ADD     SEND.RING,R0
000324  052760  100000  000002        BIS     #100000,2(R0)
```

```
000332  017766  000000G 000030          MOV     @RC25.ADDR,30(SP)       ; *,RC.REG                      2889
000340  016600  000030                  MOV     30(SP),R0               ; RC.REG,TEMP
000344  004767  000000V                JSR     PC,GET.CMD.SLOT         ;                               2893
000350  012701  000001                  MOV     #1,R1                   ; *,$$TMP2                      2896
000354  001411                  1$:     BEQ     4$
000356  016700  000000G                MOV     L$DLY,R0                ; *,$$TMP1
000362  001404                          BEQ     3$
000364  005066  000032          2$:     CLR     32(SP)                  ; $$TMP
000370  005300                          DEC     R0                      ; $$TMP1
000372  001374                          BNE     2$
000374  005301                  3$:     DEC     R1                      ; $$TMP2
000376  000766                          BR      1$
000400  004767  000000V          4$:     JSR     PC,REC.STATUS           ;                               2900
000404  062706  000034                  ADD     #34,SP                  ;                               2844
000410  012601                          MOV     (SP)+,R1
000412  000207                          RTS     PC

; Routine Size:  134 words,     Routine Base:  AB$CODE + 6502
; Maximum stack depth per invocation:  16 words


;         2902
;         2903  !
```

```
;          2904      global routine ON_LINE =
;          2905
;          2906  !++
;          2907  ! FUNCTIONAL DESCRIPTION :
;          2908  !       THE ONLINE COMMAND IS USED TO BRING A UNIT 'UNIT-ONLINE, SET
;          2909  !       HOST SETTABLE UNIT CHARACTERISTICS AND OBTAIN THOSE UNIT
;          2910  !       CHARACTERISTICS THAT ARE ESSENTIAL FOR PROPER CLASS DRIVER
;          2911  !       OPERATION.  THE UNIT IS SPUN-UP,  IF NECESSARY, AND IS HEADS
;          2912  !       ARE LOADED PRIOR TO RETURNING THE ONLINE COMMAND'S END MESSAGE.
;          2913  !       HOST SETTABLE CHARACTERISTICS COMMAND WERE ISSUED.    HOST
;          2914  !       SETTABLE  CHARACTERISTICS  ARE  SET AFTER  THE UNIT HAS BEEN
;          2915  !       SUCCESSFULLY SPUN-UP AND ANY OTHER VALIDITY CHECKS HAVE SUCCEDED.
;          2916  !
;          2917  ! FORMAL PARAMETERS :
;          2918  !       - NONE -
;          2919  !
;          2920  ! IMPLICIT INPUTS :
;          2921  !
;          2922  ! INPLICIT OUTPUTS :
;          2923  !       - NONE -
;          2924  !
;          2925  ! COMPLETEDTION CODES :
;          2926  !       RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
;          2927  !
;          2928  !
;          2929  ! SIDE EFFECTS :
;          2930  !       ANY PREVIOUSLY DEFINED CONTROLLER CHARACTERISTICS WILL POSSIBLY
;          2931  !       BE ALTERED AFTERE EXECUTION OF THEIS COMMAND.
;          2932  !--
;          2933
;          2934          begin
;          2935
;          2936          local
;          2937              TEMP;
;          2938
;          2939  !
;          2940  ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
;          2941  !
;          2942          SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_ONL;   ! LOAD MESSAGE LENGTH
;          2943          SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;         ! LOAD CREDIT SIZE
;          2944          SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE 'SEQUENTIAL'
;          2945          SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0;  ! DEFINE CONNECTION ID 'DUP'
;          2946  !
;          2947  ! MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
;          2948  !
;          2949          SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;   !LOAD COMMAND REFERENCE #
;          2950          SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;       ! ZERO HI ORDER CMD REF #
;          2951          SND_ENVELOPE [.CMD_SLOT, UN_[USED] = .UNIT;      ! SELECTED UNIT
;          2952          SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;       ! NOT USED IN DUP IMPLIMENT.
;          2953          SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_ONL;       ! DEFINE COMMAND OPCODE
;          2954          SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;         ! NOT USED
;          2955          SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;       ! DEFINE CMD MODIFIERS
;          2956  !
;          2957  ! COMMAND SPECIFIC COMMAND ENVELOPE FIELD DEFINITION
;          2958  !
;          2959          SND_ENVELOPE [.CMD_SLOT, RSV$D] = ZERO; ! RESERVED
;          2960          SND_ENVELOPE [.CMD_SLOT, UNT_FLAGS] = ZERO;      ! UNIT FLAG FIELD
```

```
;        2961          SND_ENVELOPE [.CMD_SLOT, RSVD$0] = ZERO;        ! RESERVED FIELD
;        2962          SND_ENVELOPE [.CMD_SLOT, RSVD$1] = ZERO;        ! RESERVED FIELD
;        2963          SND_ENVELOPE [.CMD_SLOT, RSVD$2] = ZERO;        ! RESERVED FIELD
;        2964          SND_ENVELOPE [.CMD_SLOT, RSVD$3] = ZERO;        ! RESERVED FIELD
;        2965          SND_ENVELOPE [.CMD_SLOT, RSVD$4] = ZERO;        ! RESERVED FIELD
;        2966          SND_ENVELOPE [.CMD_SLOT, RSVD$5] = ZERO;        ! RESERVED FIELD
;        2967          SND_ENVELOPE [.CMD_SLOT, DDP_LO] = ZERO;        ! DEVICE DEPENDENT PARAMETER
;        2968          SND_ENVELOPE [.CMD_SLOT, DDP_HI] = ZERO;        ! DEVICE DEPENDENT PARAMETER
;        2969          SND_ENVELOPE [.CMD_SLOT, SHADOW_UNIT] = ZERO;   ! SHADOW UNIT
;        2970          SND_ENVELOPE [.CMD_SLOT, COPY_SPEED] = ZERO;    ! COPY SPEED
;        2971          !
;        2972          ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;        2973          !
;        2974          SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;        2975          !
;        2976          ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
;        2977          !
;        2978          TEMP = .RC25_ADDR [RCIP, RC_ALL];
;        2979          !
;        2980          ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
;        2981          !
;        2982          GET_CMD_SLOT ();
;        2983          !
;        2984          !
;        2985          DELAY (1);
;        2986          !
;        2987          ! CHECK THE END PACKET FOR GOOD STATUS
;        2988          !
;        2989          return REC_STATUS ();                           ! RETURN THE STATUS
;        2990          end;


                                        .SBTTL  ON.LINE AZTEC GLOBAL ROUTINE
000000  010146                 ON.LINE::
                                        MOV     R1,-(SP)                        ;                       2904
000002  024646                          CMP     -(SP),-(SP)
000004  016746  000000G                 MOV     CMD.SLOT,-(SP)                  ;                       2942
000010  012746  000054                  MOV     #54,-(SP)
000014  004767  000000G                 JSR     PC,BL$MUL
000020  012760  000044  000000G         MOV     #44,SND.ENVELOPE(R0)
000026  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2943
000032  012746  000054                  MOV     #54,-(SP)
000036  004767  000000G                 JSR     PC,BL$MUL
000042  142760  000017  000002G         BICB    #17,SND.ENVELOPE+2(R0)
000050  152760  000001  000002G         BISB    #1,SND.ENVELOPE+2(R0)
000056  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2944
000062  012746  000054                  MOV     #54,-(SP)
000066  004767  000000G                 JSR     PC,BL$MUL
000072  142760  000360  000002G         BICB    #360,SND.ENVELOPE+2(R0)
000100  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2945
000104  012746  000054                  MOV     #54,-(SP)
000110  004767  000000G                 JSR     PC,BL$MUL
000114  105060  000003G                 CLRB    SND.ENVELOPE+3(R0)
000120  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       2949
000124  012746  000054                  MOV     #54,-(SP)
000130  004767  000000G                 JSR     PC,BL$MUL
000134  016760  000000G 000004G         MOV     CMD.REF,SND.ENVELOPE+4(R0)
```

```
000142  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2950
000146  012746  000054                 MOV     #54,-(SP)
000152  004767  000000G                JSR     PC,BL$MUL
000156  005060  000006G                CLR     SND.ENVELOPE+6(R0)
000162  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2951
000166  012746  000054                 MOV     #54,-(SP)
000172  004767  000000G                JSR     PC,BL$MUL
000176  016760  000000G  000010G       MOV     UNIT,SND.ENVELOPE+10(R0)
000204  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2952
000210  012746  000054                 MOV     #54,-(SP)
000214  004767  000000G                JSR     PC,BL$MUL
000220  005060  000012G                CLR     SND.ENVELOPE+12(R0)
000224  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2953
000230  012746  000054                 MOV     #54,-(SP)
000234  004767  000000G                JSR     PC,BL$MUL
000240  112760  000011   000014G       MOVB    #11,SND.ENVELOPE+14(R0)
000246  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2954
000252  012746  000054                 MOV     #54,-(SP)
000256  004767  000000G                JSR     PC,BL$MUL
000262  105060  000015G                CLRB    SND.ENVELOPE+15(R0)
000266  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2955
000272  012746  000054                 MOV     #54,-(SP)
000276  004767  000000G                JSR     PC,BL$MUL
000302  005060  000016G                CLR     SND.ENVELOPE+16(R0)
000306  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2959
000312  012746  000054                 MOV     #54,-(SP)
000316  004767  000000G                JSR     PC,BL$MUL
000322  005060  000020G                CLR     SND.ENVELOPE+20(R0)
000326  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2960
000332  012746  000054                 MOV     #54,-(SP)
000336  004767  000000G                JSR     PC,BL$MUL
000342  005060  000022G                CLR     SND.ENVELOPE+22(R0)
000346  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2961
000352  012746  000054                 MOV     #54,-(SP)
000356  004767  000000G                JSR     PC,BL$MUL
000362  005060  000024G                CLR     SND.ENVELOPE+24(R0)
000366  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2962
000372  012746  000054                 MOV     #54,-(SP)
000376  004767  000000G                JSR     PC,BL$MUL
000402  005060  000026G                CLR     SND.ENVELOPE+26(R0)
000406  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2963
000412  012746  000054                 MOV     #54,-(SP)
000416  004767  000000G                JSR     PC,BL$MUL
000422  005060  000030G                CLR     SND.ENVELOPE+30(R0)
000426  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2964
000432  012746  000054                 MOV     #54,-(SP)
000436  004767  000000G                JSR     PC,BL$MUL
000442  005060  000032G                CLR     SND.ENVELOPE+32(R0)
000446  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2965
000452  012746  000054                 MOV     #54,-(SP)
000456  004767  000000G                JSR     PC,BL$MUL
000462  005060  000034G                CLR     SND.ENVELOPE+34(R0)
000466  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2966
000472  012746  000054                 MOV     #54,-(SP)
000476  004767  000000G                JSR     PC,BL$MUL
000502  005060  000036G                CLR     SND.ENVELOPE+36(R0)
000506  016716  000000G                MOV     CMD.SLOT,(SP)                   ;                                       2967
```

```
000512  012746  000054                          MOV     #54,-(SP)
000516  004767  000000G                         JSR     PC,BL$MUL
000522  005060  000040G                         CLR     SND.ENVELOPE+40(R0)
000526  016716  000000G                         MOV     CMD.SLOT,(SP)           ;                                       2968
000532  012746  000054                          MOV     #54,-(SP)
000536  004767  000000G                         JSR     PC,BL$MUL
000542  005060  000042G                         CLR     SND.ENVELOPE+42(R0)
000546  016716  000000G                         MOV     CMD.SLOT,(SP)           ;                                       2969
000552  012746  000054                          MOV     #54,-(SP)
000556  004767  000000G                         JSR     PC,BL$MUL
000562  005060  000044G                         CLR     SND.ENVELOPE+44(R0)
000566  016716  000000G                         MOV     CMD.SLOT,(SP)           ;                                       2970
000572  012746  000054                          MOV     #54,-(SP)
000576  004767  000000G                         JSR     PC,BL$MUL
000602  005060  000046G                         CLR     SND.ENVELOPE+46(R0)
000606  016700  000000G                         MOV     CMD.SLOT,R0             ;                                       2974
000612  006300                                  ASL     R0
000614  006300                                  ASL     R0
000616  066700  000000G                         ADD     SEND.RING,R0
000622  052760  100000  000002                  BIS     #100000,2(R0)
000630  017766  000000G 000060                  MOV     @RC25.ADDR,60(SP)       ; *,RC.REG                              2978
000636  016600  000060                          MOV     60(SP),R0               ; RC.REG,TEMP
000642  004767  000000V                         JSR     PC,GET.CMD.SLOT         ;                                       2982
000646  012701  000001                          MOV     #1,R1                   ; *,$$TMP2                              2985
000652  001411                          1$:     BEQ     4$
000654  016700  000000G                         MOV     L$DLY,R0                ; *,$$TMP1
000660  001404                                  BEQ     3$
000662  005066  000062                  2$:     CLR     62(SP)                  ; $$TMP
000666  005300                                  DEC     R0                      ; $$TMP1
000670  001374                                  BNE     2$
000672  005301                          3$:     DEC     R1                      ; $$TMP2
000674  000766                                  BR      1$
000676  004767  000000V                 4$:     JSR     PC,REC.STATUS           ;                                       2989
000702  062706  000064                          ADD     #64,SP                  ;                                       2904
000706  012601                                  MOV     (SP)+,R1
000710  000207                                  RTS     PC

; Routine Size:  229 words,     Routine Base:  AB$CODE + 7116
; Maximum stack depth per invocation:  28 words


;         2991
;         2992  !
```

```
;        2993      global routine READ_CMD =
;        2994  ! ++
;        2995  ! FUNCTIONAL DESCRIPTION :
;        2996  !       THE READ COMMAND IS USED TO READ FROM THE UNIT AND TRANSFERRED
;        2997  !       TO THE HOST BUFFER.
;        2998  !
;        2999  ! FORMAL PARAMETERS :
;        3000  !       - NONE -
;        3001  !
;        3002  ! IMPLICIT INPUTS :
;        3003  !
;        3004  ! INPLICIT OUTPUTS :
;        3005  !       - NONE -
;        3006  !
;        3007  ! COMPLETEDTION CODES :
;        3008  !       RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
;        3009  !
;        3010  !
;        3011  ! SIDE EFFECTS :
;        3012  !       - NONE -
;        3013  !--
;        3014
;        3015          begin
;        3016
;        3017          local
;        3018              TEMP;
;        3019
;        3020  !
;        3021  ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
;        3022  !
;        3023          SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_RD;    ! LOAD MESSAGE LENGTH
;        3024          SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;         ! LOAD CREDIT SIZE
;        3025          SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE
;        3026          SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0;  ! DEFINE CONNECTION ID
;        3027  !
;        3028  ! MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
;        3029  !
;        3030          SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;   !LOAD COMMAND REFERENCE #
;        3031          SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;       ! ZERO HI ORDER CMD REF #
;        3032          SND_ENVELOPE [.CMD_SLOT, UN_[USED] = .UNIT;      ! SELECTED UNIT
;        3033          SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;       ! NOT USED IN DUP IMPLIMENT.
;        3034          SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_RD;        ! DEFINE COMMAND OPCODE
;        3035          SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;         ! NOT USED
;        3036          SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;       ! DEFINE CMD MODIFIERS
;        3037  !
;        3038  ! COMMAND SPECIFIC COMMAND ENVELOPE FIELD DEFINITION
;        3039  !
;        3040          SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT;        ! BYTE COUNT LOW WORD
;        3041          SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO;       ! BYTE COUNT HIGH WORD
;        3042          SND_ENVELOPE [.CMD_SLOT, BD_0] = ZERO;   ! BUFFER DESCRIBTOR FIELD
;        3043          SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO;   ! BUFFER DESCRIBTOR FIELD
;        3044          SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO;   ! BUFFER DESCRIBTOR FIELD
;        3045          SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO;   ! BUFFER DESCRIBTOR FIELD
;        3046          SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO;   ! BUFFER DESCRIBTOR FIELD
;        3047          SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO;   ! BUFFER DESCRIBTOR FIELD
;        3048          SND_ENVELOPE [.CMD_SLOT, LBN_LO] = .LBN_ST;     ! LOGICAL BLOCK NUMBER
;        3049          SND_ENVELOPE [.CMD_SLOT, LBN_HI] = ZERO;        ! LOGICAL BLOCK NUMBER
```

```
;          3050          !
;          3051          ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;          3052          !
;          3053          SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;          3054          !
;          3055          ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
;          3056          !
;          3057          TEMP = .RC25_ADDR [RCIP, RC_ALL];
;          3058          !
;          3059          ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
;          3060          !
;          3061          GET_CMD_SLOT ();
;          3062          !
;          3063          DELAY (1);
;          3064          !
;          3065          ! CHECK THE END PACKET FOR GOOD STATUS
;          3066          !
;          3067          return REC_STATUS ();                    ! RETURN THE STATUS
;          3068          end;


                                        .SBTTL  READ.CMD AZTEC GLOBAL ROUTINE
000000  010146                  READ.CMD::
                                        MOV     R1,-(SP)                        ;                       2993
000002  024646                          CMP     -(SP),-(SP)
000004  016746  000000G                 MOV     CMD.SLOT,-(SP)                  ;                       3023
000010  012746  000054                  MOV     #54,-(SP)
000014  004767  000000G                 JSR     PC,BL$MUL
000020  012760  000040  000000G         MOV     #40,SND.ENVELOPE(R0)
000026  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3024
000032  012746  000054                  MOV     #54,-(SP)
000036  004767  000000G                 JSR     PC,BL$MUL
000042  142760  000017  000002G         BICB    #17,SND.ENVELOPE+2(R0)
000050  152760  000001  000002G         BISB    #1,SND.ENVELOPE+2(R0)
000056  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3025
000062  012746  000054                  MOV     #54,-(SP)
000066  004767  000000G                 JSR     PC,BL$MUL
000072  142760  000360  000002G         BICB    #360,SND.ENVELOPE+2(R0)
000100  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3026
000104  012746  000054                  MOV     #54,-(SP)
000110  004767  000000G                 JSR     PC,BL$MUL
000114  105060  000003G                 CLRB    SND.ENVELOPE+3(R0)
000120  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3030
000124  012746  000054                  MOV     #54,-(SP)
000130  004767  000000G                 JSR     PC,BL$MUL
000134  016760  000000G  000004G        MOV     CMD.REF,SND.ENVELOPE+4(R0)
000142  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3031
000146  012746  000054                  MOV     #54,-(SP)
000152  004767  000000G                 JSR     PC,BL$MUL
000156  005060  000006G                 CLR     SND.ENVELOPE+6(R0)
000162  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3032
000166  012746  000054                  MOV     #54,-(SP)
000172  004767  000000G                 JSR     PC,BL$MUL
000176  016760  000000G  000010G        MOV     UNIT,SND.ENVELOPE+10(R0)
000204  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3033
000210  012746  000054                  MOV     #54,-(SP)
000214  004767  000000G                 JSR     PC,BL$MUL
```

```
000220  005060  000012G                 CLR     SND.ENVELOPE+12(R0)
000224  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3034
000230  012746  000054                   MOV     #54,-(SP)
000234  004767  000000G                 JSR     PC,BL$MUL
000240  112760  000041  000014G         MOVB    #41,SND.ENVELOPE+14(R0)
000246  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3035
000252  012746  000054                   MOV     #54,-(SP)
000256  004767  000000G                 JSR     PC,BL$MUL
000262  105060  000015G                 CLRB    SND.ENVELOPE+15(R0)
000266  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3036
000272  012746  000054                   MOV     #54,-(SP)
000276  004767  000000G                 JSR     PC,BL$MUL
000302  005060  000016G                 CLR     SND.ENVELOPE+16(R0)
000306  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3040
000312  012746  000054                   MOV     #54,-(SP)
000316  004767  000000G                 JSR     PC,BL$MUL
000322  016760  000000G  000020G        MOV     BYTE.COUNT,SND.ENVELOPE+20(R0)
000330  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3041
000334  012746  000054                   MOV     #54,-(SP)
000340  004767  000000G                 JSR     PC,BL$MUL
000344  005060  000022G                 CLR     SND.ENVELOPE+22(R0)
000350  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3042
000354  012746  000054                   MOV     #54,-(SP)
000360  004767  000000G                 JSR     PC,BL$MUL
000364  005060  000024G                 CLR     SND.ENVELOPE+24(R0)
000370  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3043
000374  012746  000054                   MOV     #54,-(SP)
000400  004767  000000G                 JSR     PC,BL$MUL
000404  005060  000026G                 CLR     SND.ENVELOPE+26(R0)
000410  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3044
000414  012746  000054                   MOV     #54,-(SP)
000420  004767  000000G                 JSR     PC,BL$MUL
000424  005060  000030G                 CLR     SND.ENVELOPE+30(R0)
000430  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3045
000434  012746  000054                   MOV     #54,-(SP)
000440  004767  000000G                 JSR     PC,BL$MUL
000444  005060  000032G                 CLR     SND.ENVELOPE+32(R0)
000450  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3046
000454  012746  000054                   MOV     #54,-(SP)
000460  004767  000000G                 JSR     PC,BL$MUL
000464  005060  000034G                 CLR     SND.ENVELOPE+34(R0)
000470  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3047
000474  012746  000054                   MOV     #54,-(SP)
000500  004767  000000G                 JSR     PC,BL$MUL
000504  005060  000036G                 CLR     SND.ENVELOPE+36(R0)
000510  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3048
000514  012746  000054                   MOV     #54,-(SP)
000520  004767  000000G                 JSR     PC,BL$MUL
000524  016760  000000G  000040G        MOV     LBN.ST,SND.ENVELOPE+40(R0)
000532  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                                       3049
000536  012746  000054                   MOV     #54,-(SP)
000542  004767  000000G                 JSR     PC,BL$MUL
000546  005060  000042G                 CLR     SND.ENVELOPE+42(R0)
000552  016700  000000G                 MOV     CMD.SLOT,R0                     ;                                       3053
000556  006300                           ASL     R0
000560  006300                           ASL     R0
000562  066700  000000G                 ADD     SEND.RING,R0
```

```
000566  052760  100000  000002                  BIS     #100000,2(R0)
000574  017766  000000G 000054                  MOV     @RC25.ADDR,54(SP)       ; *,RC.REG                      3057
000602  016600  000054                          MOV     54(SP),R0               ; RC.REG,TEMP
000606  004767  000000V                         JSR     PC,GET.CMD.SLOT         ;                               3061
000612  012701  000001                          MOV     #1,R1                   ; *,$$TMP2                      3063
000616  001411                          1$:     BEQ     4$
000620  016700  000000G                         MOV     L$DLY,R0                ; *,$$TMP1
000624  001404                                  BEQ     3$
000626  005066  000056                  2$:     CLR     56(SP)                  ; $$TMP
000632  005300                                  DEC     R0                      ; $$TMP1
000634  001374                                  BNE     2$
000636  005301                          3$:     DEC     R1                      ; $$TMP2
000640  000766                                  BR      1$
000642  004767  000000V                 4$:     JSR     PC,REC.STATUS           ;                               3067
000646  062706  000060                          ADD     #60,SP                  ;                               2993
000652  012601                                  MOV     (SP)+,R1
000654  000207                                  RTS     PC

; Routine Size:  215 words,     Routine Base:  AB$CODE + 10030
; Maximum stack depth per invocation:  26 words


;         3069
;         3070  !
```

```
3071        global routine READ_FILL_RING : novalue =
3072    !++
3073    ! FUNCTIONAL DESCRIPTION :
3074    !       THE READ COMMAND IS USED TO READ THE DATA FROM THE UNIT AND
3075    !       TRANSFERED TO THE HOST BUFFER.
3076    !
3077    ! FORMAL PARAMETERS :
3078    !       - NONE -
3079    !
3080    ! IMPLICIT INPUTS :
3081    !
3082    ! INPLICIT OUTPUTS :
3083    !       - NONE -
3084    !
3085    ! COMPLETEDTION CODES :
3086    !       RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
3087    !
3088    !
3089    ! SIDE EFFECTS :
3090    !       - NONE -
3091    !--
3092
3093            begin
3094
3095            local
3096                TEMP;
3097
3098    !
3099    ! MSCP PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
3100    !
3101            SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_RD;    ! LOAD MESSAGE LENGTH
3102            SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;         ! LOAD CREDIT SIZE
3103            SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE
3104            SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0;  ! DEFINE CONNECTION ID
3105    !
3106    ! MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
3107    !
3108            SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;  !LOAD COMMAND REFERENCE #
3109            SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;      ! ZERO HI ORDER CMD REF #
3110            SND_ENVELOPE [.CMD_SLOT, UN_[USED] = .UNIT;     ! SELECTED UNIT
3111            SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;      ! NOT USED IN DUP IMPLIMENT.
3112            SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_RD;       ! DEFINE COMMAND OPCODE
3113            SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;        ! NOT USED
3114            SND_ENVELOPE [.CMD_SLOT, MODIFIER] = MD_EXP;    ! DEFINE CMD MODIFIERS
3115    !
3116    ! COMMAND SPECIFIC COMMAND ENVELOPE FIELD DEFINITION
3117    !
3118            SND_ENVELOPE [.CMD_SLOT, BLO_CNT] = .BYTE_COUNT;        ! BYTE COUNT LOW WORD
3119            SND_ENVELOPE [.CMD_SLOT, BHI_CNT] = ZERO;       ! BYTE COUNT HIGH WORD
3120            SND_ENVELOPE [.CMD_SLOT, BD_0] = .BUF_DESCRPTR; ! BUFFER DESCRIBTOR FIELD
3121            SND_ENVELOPE [.CMD_SLOT, BD_1] = ZERO;  ! BUFFER DESCRIBTOR FIELD
3122            SND_ENVELOPE [.CMD_SLOT, BD_2] = ZERO;  ! BUFFER DESCRIBTOR FIELD
3123            SND_ENVELOPE [.CMD_SLOT, BD_3] = ZERO;  ! BUFFER DESCRIBTOR FIELD
3124            SND_ENVELOPE [.CMD_SLOT, BD_4] = ZERO;  ! BUFFER DESCRIBTOR FIELD
3125            SND_ENVELOPE [.CMD_SLOT, BD_5] = ZERO;  ! BUFFER DESCRIBTOR FIELD
3126            SND_ENVELOPE [.CMD_SLOT, LBN_LO] = .LBN_ST;     ! LOGICAL BLOCK NUMBER
3127            SND_ENVELOPE [.CMD_SLOT, LBN_HI] = ZERO;        ! LOGICAL BLOCK NUMBER
```

```
;         3128          !
;         3129          ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;         3130          !
;         3131          SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;         3132          !
;         3133          end;


                                        .SBTTL  READ.FILL.RING AZTEC GLOBAL ROUTINE
000000  016746  000000G            READ.FILL.RING::
                                        MOV     CMD.SLOT,-(SP)                  ;                       3101
000004  012746  000054                  MOV     #54,-(SP)
000010  004767  000000G                 JSR     PC,BL$MUL
000014  012760  000040  000000G        MOV     #40,SND.ENVELOPE(R0)
000022  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3102
000026  012746  000054                  MOV     #54,-(SP)
000032  004767  000000G                 JSR     PC,BL$MUL
000036  142760  000017  000002G        BICB    #17,SND.ENVELOPE+2(R0)
000044  152760  000001  000002G        BISB    #1,SND.ENVELOPE+2(R0)
000052  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3103
000056  012746  000054                  MOV     #54,-(SP)
000062  004767  000000G                 JSR     PC,BL$MUL
000066  142760  000360  000002G        BICB    #360,SND.ENVELOPE+2(R0)
000074  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3104
000100  012746  000054                  MOV     #54,-(SP)
000104  004767  000000G                 JSR     PC,BL$MUL
000110  105060  000003G                 CLRB    SND.ENVELOPE+3(R0)
000114  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3108
000120  012746  000054                  MOV     #54,-(SP)
000124  004767  000000G                 JSR     PC,BL$MUL
000130  016760  000000G  000004G        MOV     CMD.REF,SND.ENVELOPE+4(R0)
000136  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3109
000142  012746  000054                  MOV     #54,-(SP)
000146  004767  000000G                 JSR     PC,BL$MUL
000152  005060  000006G                 CLR     SND.ENVELOPE+6(R0)
000156  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3110
000162  012746  000054                  MOV     #54,-(SP)
000166  004767  000000G                 JSR     PC,BL$MUL
000172  016760  000000G  000010G        MOV     UNIT,SND.ENVELOPE+10(R0)
000200  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3111
000204  012746  000054                  MOV     #54,-(SP)
000210  004767  000000G                 JSR     PC,BL$MUL
000214  005060  000012G                 CLR     SND.ENVELOPE+12(R0)
000220  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3112
000224  012746  000054                  MOV     #54,-(SP)
000230  004767  000000G                 JSR     PC,BL$MUL
000234  112760  000041  000014G        MOVB    #41,SND.ENVELOPE+14(R0)
000242  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3113
000246  012746  000054                  MOV     #54,-(SP)
000252  004767  000000G                 JSR     PC,BL$MUL
000256  105060  000015G                 CLRB    SND.ENVELOPE+15(R0)
000262  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3114
000266  012746  000054                  MOV     #54,-(SP)
000272  004767  000000G                 JSR     PC,BL$MUL
000276  012760  100000  000016G        MOV     #-100000,SND.ENVELOPE+16(R0)
000304  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3118
000310  012746  000054                  MOV     #54,-(SP)
```

```
000314  004767  000000G                 JSR     PC,BL$MUL
000320  016760  000000G 000020G         MOV     BYTE.COUNT,SND.ENVELOPE+20(R0)
000326  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3119
000332  012746  000054                  MOV     #54,-(SP)
000336  004767  000000G                 JSR     PC,BL$MUL
000342  005060  000022G                 CLR     SND.ENVELOPE+22(R0)
000346  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3120
000352  012746  000054                  MOV     #54,-(SP)
000356  004767  000000G                 JSR     PC,BL$MUL
000362  016760  000000G 000024G         MOV     BUF.DESCRPTR,SND.ENVELOPE+24(R0)
000370  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3121
000374  012746  000054                  MOV     #54,-(SP)
000400  004767  000000G                 JSR     PC,BL$MUL
000404  005060  000026G                 CLR     SND.ENVELOPE+26(R0)
000410  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3122
000414  012746  000054                  MOV     #54,-(SP)
000420  004767  000000G                 JSR     PC,BL$MUL
000424  005060  000030G                 CLR     SND.ENVELOPE+30(R0)
000430  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3123
000434  012746  000054                  MOV     #54,-(SP)
000440  004767  000000G                 JSR     PC,BL$MUL
000444  005060  000032G                 CLR     SND.ENVELOPE+32(R0)
000450  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3124
000454  012746  000054                  MOV     #54,-(SP)
000460  004767  000000G                 JSR     PC,BL$MUL
000464  005060  000034G                 CLR     SND.ENVELOPE+34(R0)
000470  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3125
000474  012746  000054                  MOV     #54,-(SP)
000500  004767  000000G                 JSR     PC,BL$MUL
000504  005060  000036G                 CLR     SND.ENVELOPE+36(R0)
000510  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3126
000514  012746  000054                  MOV     #54,-(SP)
000520  004767  000000G                 JSR     PC,BL$MUL
000524  016760  000000G 000040G         MOV     LBN.ST,SND.ENVELOPE+40(R0)
000532  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3127
000536  012746  000054                  MOV     #54,-(SP)
000542  004767  000000G                 JSR     PC,BL$MUL
000546  005060  000042G                 CLR     SND.ENVELOPE+42(R0)
000552  016700  000000G                 MOV     CMD.SLOT,R0                     ;                       3131
000556  006300                          ASL     R0
000560  006300                          ASL     R0
000562  066700  000000G                 ADD     SEND.RING,R0
000566  052760  100000  000002         BIS     #100000,2(R0)
000574  062706  000054                  ADD     #54,SP                          ;                       3093
000600  000207                          RTS     PC                              ;                       3071

; Routine Size:  193 words,     Routine Base:  AB$CODE + 10706
; Maximum stack depth per invocation:  23 words


;        3134
;        3135  !
```

```
;          3136      global routine GET_UNIT_STATUS =
;          3137      !++
;          3138      ! FUNCTIONAL DESCRIPTION :
;          3139      !       THE GET UNIT STATUS COMMAND IS USED TO READ THE CURRENT
;          3140      !       STATE OF THE UNIT, PLUS CERTAIN UNIT CHARACTERISTIACS.
;          3141      !
;          3142      ! FORMAL PARAMETERS :
;          3143      !       - NONE -
;          3144      !
;          3145      ! IMPLICIT INPUTS :
;          3146      !
;          3147      ! INPLICIT OUTPUTS :
;          3148      !       - NONE -
;          3149      !
;          3150      ! COMPLETEDTION CODES :
;          3151      !       RET_STATUS : RETURN STATUS PASSES BACK TO THE CALLING ROUTINE
;          3152      !
;          3153      !
;          3154      ! SIDE EFFECTS :
;          3155      !       - NONE -
;          3156      !--
;          3157
;          3158              begin
;          3159
;          3160              local
;          3161                  TEMP;
;          3162
;          3163      !
;          3164      ! UQ PORT COMMAND ENVELOPE HEADER FIELD DEFINITION
;          3165      !
;          3166              SND_ENVELOPE [.CMD_SLOT, MSG_LENGTH] = SZ_GUS;  ! LOAD MESSAGE LENGTH
;          3167              SND_ENVELOPE [.CMD_SLOT, CREDITS] = ONE;        ! LOAD CREDIT SIZE
;          3168              SND_ENVELOPE [.CMD_SLOT, MSG_TYPE] = 0; ! MESSAGE TYPE
;          3169              SND_ENVELOPE [.CMD_SLOT, CONN_ID] = 0;  ! DEFINE CONNECTION ID
;          3170      !
;          3171      ! MSCP GENERIC COMMAND ENVELOPE FIELD DEFINITION
;          3172      !
;          3173              SND_ENVELOPE [.CMD_SLOT, CMD_LREF] = .CMD_REF;  !LOAD COMMAND REFERENCE #
;          3174              SND_ENVELOPE [.CMD_SLOT, CMD_HREF] = ZERO;      ! ZERO HI ORDER CMD REF #
;          3175              SND_ENVELOPE [.CMD_SLOT, UN_[USED] = .UNIT;     ! SELECTED UNIT
;          3176              SND_ENVELOPE [.CMD_SLOT, UN_HUSED] = ZERO;      ! NOT USED IN DUP IMPLIMENT.
;          3177              SND_ENVELOPE [.CMD_SLOT, OPCODE] = OP_GUS;      ! DEFINE COMMAND OPCODE
;          3178              SND_ENVELOPE [.CMD_SLOT, UQRSVD] = ZERO;        ! NOT USED
;          3179              SND_ENVELOPE [.CMD_SLOT, MODIFIER] = ZERO;      ! DEFINE CMD MODIFIERS
;          3180              !
;          3181              ! SET THE OWNERSHIP BIT TO 1 WHICH GIVE THIS SLOT TO THE PORT.
;          3182              !
;          3183              SEND_RING [.CMD_SLOT, OWN_BIT] = PORT_OWNED;
;          3184              !
;          3185              ! READ THE IP REGISTER TO STIMULATE PORT POLLING.
;          3186              !
;          3187              TEMP = .RC25_ADDR [RCIP, RC_ALL];
;          3188              !
;          3189              ! GET THE COMMAND SLOT NUMBER FOR NEXT COMMAND
;          3190              !
;          3191              GET_CMD_SLOT ();
;          3192              !
```

```
;          3193          DELAY (1);
;          3194          !
;          3195          ! CHECK THE END PACKET FOR GOOD STATUS
;          3196          !
;          3197
;          3198          if REC_STATUS ()                        ! READ THE STATUS
;          3199          then
;          3200              begin
;          3201              return .RET_STATUS;                 ! RETURN WITH A STATUS ERR
;          3202              end
;          3203          else
;          3204              RES_SLOT = .RES_SLOT - 1;           ! GET THE CURRENT RES. SLOT
;          3205
;          3206          RET_UNIT_FLAG = .REC_ENVELOPE [.RES_SLOT, UNIT_FLAG];   ! READ UNIT FLAG
;          3207          GET_RES_SLOT ();                        ! GET NEXT RES. SLOT
;          3208          return .RET_STATUS;                     ! RETURN WITH A PASS CODE
;          3209          end;

                                        .SBTTL  GET.UNIT.STATUS AZTEC GLOBAL ROUTINE
000000  010146                    GET.UNIT.STATUS::
                                        MOV     R1,-(SP)                        ;                       3136
000002  024646                          CMP     -(SP),-(SP)
000004  016746  000000G                 MOV     CMD.SLOT,-(SP)                  ;                       3166
000010  012746  000054                  MOV     #54,-(SP)
000014  004767  000000G                 JSR     PC,BL$MUL
000020  012760  000014  000000G         MOV     #14,SND.ENVELOPE(R0)
000026  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3167
000032  012746  000054                  MOV     #54,-(SP)
000036  004767  000000G                 JSR     PC,BL$MUL
000042  142760  000017  000002G         BICB    #17,SND.ENVELOPE+2(R0)
000050  152760  000001  000002G         BISB    #1,SND.ENVELOPE+2(R0)
000056  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3168
000062  012746  000054                  MOV     #54,-(SP)
000066  004767  000000G                 JSR     PC,BL$MUL
000072  142760  000360  000002G         BICB    #360,SND.ENVELOPE+2(R0)
000100  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3169
000104  012746  000054                  MOV     #54,-(SP)
000110  004767  000000G                 JSR     PC,BL$MUL
000114  105060  000003G                 CLRB    SND.ENVELOPE+3(R0)
000120  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3173
000124  012746  000054                  MOV     #54,-(SP)
000130  004767  000000G                 JSR     PC,BL$MUL
000134  016760  000000G  000004G        MOV     CMD.REF,SND.ENVELOPE+4(R0)
000142  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3174
000146  012746  000054                  MOV     #54,-(SP)
000152  004767  000000G                 JSR     PC,BL$MUL
000156  005060  000006G                 CLR     SND.ENVELOPE+6(R0)
000162  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3175
000166  012746  000054                  MOV     #54,-(SP)
000172  004767  000000G                 JSR     PC,BL$MUL
000176  016760  000000G  000010G        MOV     UNIT,SND.ENVELOPE+10(R0)
000204  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3176
000210  012746  000054                  MOV     #54,-(SP)
000214  004767  000000G                 JSR     PC,BL$MUL
000220  005060  000012G                 CLR     SND.ENVELOPE+12(R0)
000224  016716  000000G                 MOV     CMD.SLOT,(SP)                   ;                       3177
```

```
000230  012746  000054                          MOV     #54,-(SP)
000234  004767  000000G                         JSR     PC,BL$MUL
000240  112760  000003  000014G                 MOVB    #3,SND.ENVELOPE+14(R0)
000246  016716  000000G                         MOV     CMD.SLOT,(SP)                   ;                               3178
000252  012746  000054                          MOV     #54,-(SP)
000256  004767  000000G                         JSR     PC,BL$MUL
000262  105060  000015G                         CLRB    SND.ENVELOPE+15(R0)
000266  016716  000000G                         MOV     CMD.SLOT,(SP)                   ;                               3179
000272  012746  000054                          MOV     #54,-(SP)
000276  004767  000000G                         JSR     PC,BL$MUL
000302  005060  000016G                         CLR     SND.ENVELOPE+16(R0)
000306  016700  000000G                         MOV     CMD.SLOT,R0                     ;                               3183
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                      3187
000336  016600  000030                          MOV     30(SP),R0                       ; RC.REG,TEMP
000342  004767  000000V                         JSR     PC,GET.CMD.SLOT                 ;                               3191
000346  012701  000001                          MOV     #1,R1                           ; *,$$TMP2                      3193
000352  001411                          1$:     BEQ     4$
000354  016700  000000G                         MOV     L$DLY,R0                        ; *,$$TMP1
000360  001404                                  BEQ     3$
000362  005066  000032                  2$:     CLR     32(SP)                          ; $$TMP
000366  005300                                  DEC     R0                              ; $$TMP1
000370  001374                                  BNE     2$
000372  005301                          3$:     DEC     R1                              ; $$TMP2
000374  000766                                  BR      1$
000376  004767  000000V                 4$:     JSR     PC,REC.STATUS                   ;                               3198
000402  006000                                  ROR     R0
000404  103005                                  BCC     5$
000406  062706  000030                          ADD     #30,SP
000412  016700  000000G                         MOV     RET.STATUS,R0                   ;                               3200
000416  000423                                  BR      6$
000420  005367  000000G                 5$:     DEC     RES.SLOT                        ;                               3204
000424  016700  000000G                         MOV     RES.SLOT,R0                     ;                               3206
000430  000300                                  SWAB    R0
000432  106000                                  RORB    R0
000434  006000                                  ROR     R0
000436  006000                                  ROR     R0
000440  142700  000077                          BICB    #77,R0
000444  016067  000022G 000000G                 MOV     REC.ENVELOPE+22(R0),RET.UNIT.FLAG ;
000452  004767  000000V                         JSR     PC,GET.RES.SLOT                 ;                               3207
000456  062706  000030                          ADD     #30,SP                          ;                               3136
000462  016700  000000G                         MOV     RET.STATUS,R0                   ;                               3158
000466  022626                          6$:     CMP     (SP)+,(SP)+                     ;                               3136
000470  012601                                  MOV     (SP)+,R1
000472  000207                                  RTS     PC

; Routine Size:  158 words,     Routine Base:  AB$CODE + 11510
; Maximum stack depth per invocation:  16 words


;         3210
;         3211  !
;         3212
```

```
;          3213      global routine GET_CMD_SLOT : novalue =
;          3214  ! ++
;          3215  ! FUNCTIONAL DESCRIPTION:
;          3216  !
;          3217  !       THIS ROUTINE ASSIGNS A COMMAND SLOT NUMBER FOR THE COMMUNICATION
;          3218  !       RING, IT WILL WRAP AROUND, AS THE SLOT NUMBER REACHED TO THE BOTTOM.
;          3219  ! --
;          3220      begin
;          3221      begin
;          3222
;          3223      if .CMD_SLOT eqlu SND_ALLOCATE - 1      ! IS SLOT # REACHED TO THE END
;          3224      then                                    ! YES
;          3225          CMD_SLOT = ZERO                     ! WRAP AROUND THE COMMAND RING
;          3226      else                                    ! ELSE
;          3227          CMD_SLOT = .CMD_SLOT + 1;           ! INCREMENT THE CMD SLOT NUMBER
;          3228
;          3229      end;
;          3230      SEND_RING [.CMD_SLOT, FLAG_BIT] = ZERO; ! CLEAR CMD_RING FLAG BIT
;          3231      return;
;          3232      end;


                                        .SBTTL  GET.CMD.SLOT AZTEC GLOBAL ROUTINE
000000  026727  000000G 000017    GET.CMD.SLOT::
                                        CMP     CMD.SLOT,#17                    ;                       3223
000006  001003                          BNE     1$
000010  005067  000000G                 CLR     CMD.SLOT                        ;                       3225
000014  000402                          BR      2$                              ;                       3223
000016  005267  000000G         1$:     INC     CMD.SLOT                        ;                       3227
000022  016700  000000G         2$:     MOV     CMD.SLOT,R0                     ;                       3230
000026  006300                          ASL     R0
000030  006300                          ASL     R0
000032  066700  000000G                 ADD     SEND.RING,R0
000036  042760  040000  000002         BIC     #40000,2(R0)
000044  000207                          RTS     PC                              ;                       3213

; Routine Size:  19 words,      Routine Base:  AB$CODE + 12204
; Maximum stack depth per invocation:  0 words


;          3233
;          3234  !
```

```
;          3235      global routine GET_RES_SLOT : novalue =
;          3236  ! ++
;          3237  ! FUNCTIONAL DESCRIPTION:
;          3238  !
;          3239  !       THIS ROUTINE ASSIGNS A RESPONSE SLOT NUMBER FOR THE COMMUNICATION
;          3240  !       RING, IT WILL WRAP AROUND, AS THE SLOT NUMBER REACHED TO THE BOTTOM.
;          3241  ! --
;          3242        begin
;          3243        begin
;          3244
;          3245        if .RES_SLOT eqlu REC_ALLOCATE - 1      ! IS SLOT # REACHED TO THE END?
;          3246        then                                    ! YES. THEN
;          3247            RES_SLOT = ZERO                     ! WRAP AROUND THE RESPONSE RING
;          3248        else                                    ! ELSE
;          3249            RES_SLOT = .RES_SLOT + 1;           ! INCREMENT THE RES SLOT NUMBER
;          3250
;          3251        end;
;          3252        RECEIVE_RING [.RES_SLOT, FLAG_BIT] = ZERO;      ! CLEAR RECEIVE RING FLAG BIT
;          3253        return;
;          3254        end;


                                        .SBTTL  GET.RES.SLOT AZTEC GLOBAL ROUTINE
000000  026727  000000G 000017  GET.RES.SLOT::
                                        CMP     RES.SLOT,#17                    ;                       3245
000006  001003                          BNE     1$
000010  005067  000000G                 CLR     RES.SLOT                        ;                       3247
000014  000402                          BR      2$                              ;                       3245
000016  005267  000000G         1$:     INC     RES.SLOT                        ;                       3249
000022  016700  000000G         2$:     MOV     RES.SLOT,R0                     ;                       3252
000026  006300                          ASL     R0
000030  006300                          ASL     R0
000032  066700  000000G                 ADD     RECEIVE.RING,R0
000036  042760  040000  000002         BIC     #40000,2(R0)
000044  000207                          RTS     PC                              ;                       3235

; Routine Size:  19 words,      Routine Base:  AB$CODE + 12252
; Maximum stack depth per invocation:  0 words


;          3255
;          3256  !
```

```
;          3257      global routine DUP_MSCP_INTS : INT_LNK$TYP =
;          3258
;          3259  !++
;          3260  ! FUNCTIONAL DESCRIPTION :
;          3261  !
;          3262  !       THIS ROUTINE SERVICE THE DUP AND MSCP INTERRUPT
;          3263  !
;          3264  ! FORMAL PARAMETERS :
;          3265  ! IMPLICIT INPUTS :
;          3266  ! IMPLICIT OUTPUTS :
;          3267  ! COMPLETION CODES :
;          3268  ! SIDE EFFECTS :
;          3269  !--
;          3270
;          3271          begin
;          3272          RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];   ! GET RCSA DATA
;          3273
;          3274          if .RC25_DATA [RCSA, RCSA_ER]           ! CHECK SA REG.ERROR BIT
;          3275          then
;          3276              begin
;          3277              I_AM_NEX = ALL_ONES;                ! INDICATE THE INT. HAPPENED
;          3278              RET_STATUS = PFE_CODE;              ! SAVE THE PORT/CTLER FAILURE
;          3279              return .RET_STATUS;
;          3280              end
;          3281          else
;          3282              begin
;          3283              HEAD_AREA [RSP_INT] = ZERO;         ! CLEAR INT. FLAG IN THE HEADER WORD
;          3284              HEAD_AREA [CMD_INT] = ZERO;         !
;          3285              RET_STATUS = PAS_CODE;              ! RETURN A NON-ERROR CODE
;          3286              I_AM_NEX = ALL_ONES;                ! INDICATE THE INTERRUPT OCCURED
;          3287              end;
;          3288
;          3289          end;


                                        .SBTTL  DUP.MSCP.INTS AZTEC GLOBAL ROUTINE
000000  010046                  DUP.MSCP.INTS::
                                MOV     R0,-(SP)                ;                       3257
000002  016700  000000G         MOV     RC25.ADDR,R0            ;                       3272
000006  016046  000002          MOV     2(R0),-(SP)             ; *,RC.REG
000012  011667  000002G         MOV     (SP),RC25.DATA+2        ; RC.REG,*
000016  100007                  BPL     1$                      ;                       3274
000020  012767  177777  000000G MOV     #-1,I.AM.NEX            ;                       3277
000026  012767  000021  000000G MOV     #21,RET.STATUS          ;                       3278
000034  000413                  BR      2$                      ;                       3276
000036  016700  000000G 1$:     MOV     HEAD.AREA,R0            ;                       3283
000042  005060  000006          CLR     6(R0)
000046  005060  000004          CLR     4(R0)                   ;                       3284
000052  005067  000000G         CLR     RET.STATUS              ;                       3285
000056  012767  177777  000000G MOV     #-1,I.AM.NEX            ;                       3286
000064  005726          2$:     TST     (SP)+                   ;                       3257
000066  012600                  MOV     (SP)+,R0
000070  000002                  RTI

; Routine Size:  29 words,      Routine Base:  AB$CODE + 12320
; Maximum stack depth per invocation:  3 words
;          3290
```

```
;        3291  !
```

```
;          3292      global routine SET_INT_VECTOR : novalue =
;          3293
;          3294  !++
;          3295  ! FUNCTIONAL DESCRIPTION :
;          3296  !
;          3297  !       THIS ROUTINE SET UP THE INTERRUPT VECTOR
;          3298  !
;          3299  ! FORMAL PARAMETERS :
;          3300  ! IMPLICIT INPUTS :
;          3301  ! IMPLICIT OUTPUTS :
;          3302  ! COMPLETION CODES :
;          3303  ! SIDE EFFECTS :
;          3304  !--
;          3305
;          3306          begin
;          3307          CLRVEC (.RT_TABLE [RT_VECTOR]);          ! CLEAR VECTOR ADDRESS
;          3308          SETVEC (.RT_TABLE [RT_VECTOR], DUP_MSCP_INTS, .RT_TABLE [RT_BR_LEVEL]); ! SET VECTOR ADDR. SERVICE
;          3309                                                   ! RPUTINE ADDR. & PRIORITY
;          3310  !   WRT_RC25 (RCSA, TRUE);                       ! LET CONTROLLER GO
;          3311          return;
;          3312          end;


                                        .SBTTL  SET.INT.VECTOR AZTEC GLOBAL ROUTINE
000000  010146                    SET.INT.VECTOR::
                                        MOV     R1,-(SP)                        ;                       3292
000002  016701  000000G                 MOV     RT.TABLE,R1                     ;                       3307
000006  016100  000002                  MOV     2(R1),R0
000012  104436                          TRAP    36
000014  016700  000000G                 MOV     RT.TABLE,R0                     ;                       3308
000020  016046  000004                  MOV     4(R0),-(SP)
000024  012746  012320'                 MOV     #DUP.MSCP.INTS,-(SP)
000030  016046  000002                  MOV     2(R0),-(SP)
000034  012746  000003                  MOV     #3,-(SP)
000040  104437                          TRAP    37
000042  062706  000010                  ADD     #10,SP                          ;                       3292
000046  012601                          MOV     (SP)+,R1
000050  000207                          RTS     PC

; Routine Size:  21 words,      Routine Base:  AB$CODE + 12412
; Maximum stack depth per invocation:  7 words


;          3313
;          3314  !
```

```
;       3315      global routine REC_STATUS =
;       3316
;       3317  !++
;       3318  ! FUNCTIONAL DESCRIPTION :
;       3319  !
;       3320  !       THIS ROUTINE READ THE END MESSAGE PACKET AND RETURN THE PORT
;       3321  !       TO THE CONTROLLER, AND A STATUS FLAG IS SEND TO THE CALLER.
;       3322  !
;       3323  !       IF STATUS BIT INDICATES UNSUCESS, THEN A ERROR MESSAGE WILL
;       3324  !       BE REPORTED.
;       3325  !
;       3326  ! FORMAL PARAMETERS :
;       3327  ! IMPLICIT INPUTS :
;       3328  ! IMPLICIT OUTPUTS :
;       3329  !       ERROR : TRUE
;       3330  !       NO ERROR : FALSE
;       3331  !
;       3332  ! COMPLETION CODES :
;       3333  ! SIDE EFFECTS :
;       3334  !--
;       3335
;       3336          begin
;       3337  !
;       3338  ! WAITING FOR THE CONTROLLER TO FILLED THE DESCRIPTOR AND RELEASING
;       3339  ! IT TO THE HOST, IF WAITING TIME EXPIRED THEN AN ERROR WILL BE REPORTED.
;       3340  !
;       3341
;       3342          incru COUNT from 0 to 30000 do          ! SET TIME OUT RANGE
;       3343              begin
;       3344              DELAY (5);                          ! DELAY
;       3345
;       3346              if .RECEIVE_RING [.RES_SLOT, OWN_BIT] eqlu 0        ! IF HOST OWN THE SLOT
;       3347              then                                ! THEN
;       3348                  begin
;       3349                  DELAY (25);                     ! DELAY
;       3350
;       3351                  if (.REC_ENVELOPE [.RES_SLOT, STATUS]   ! READ THE STATUS BITS
;       3352                      nequ ZERO)
;       3353                  then                            ! IF ERROR
;       3354                      begin                       ! THEN FLAG THE ERROR
;       3355                      RECEIVE_RING [.RES_SLOT, OWN_BIT] = ONE;    ! PORT OWN THE RING
;       3356                      RET_STATUS = RSE_CODE;      ! REPORT THE ERROR & SET STATUS
;       3357                      return .RET_STATUS;         ! SET ERROR FLAG
;       3358                      end
;       3359                  else
;       3360                      begin
;       3361                      RECEIVE_RING [.RES_SLOT, OWN_BIT] = ONE;    ! PORT OWN THE RING
;       3362                      GET_RES_SLOT ();            ! GET NEXT RESPONSE SLOT #
;       3363                      RET_STATUS = PAS_CODE;      ! CLEAR STATUS
;       3364                      return .RET_STATUS;         ! RETURN WITH PASS FLAG
;       3365                      end;
;       3366
;       3367                  end;
;       3368
;       3369              end;
;       3370
;       3371          RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];   ! GET RCSA DATA
```

```
;          3372
;          3373          if .RC25_DATA [RCSA, RCSA_ER]          ! CHECK SA REG.ERROR BIT
;          3374          then
;          3375              begin
;          3376              RET_STATUS = PFE_CODE;             ! SAVE THE PORT/CTLER FAILURE
;          3377              return .RET_STATUS;
;          3378              end
;          3379          else
;          3380              begin
;          3381              RET_STATUS = CTO_CODE;             ! SET TIME EXPIRED IN STATUS BUF
;          3382              return .RET_STATUS;                ! RETURN WITH A TIME EXPIRED FLAG
;          3383              end;
;          3384
;          3385          end;


                                        .SBTTL  REC.STATUS AZTEC GLOBAL ROUTINE
000000  004167  000000G          REC.STATUS::
                                        JSR     R1,$SAVE2                       ;                       3315
000004  024646                          CMP     -(SP),-(SP)
000006  005002                          CLR     R2                              ; COUNT                 3342
000010  012701  000005          1$:     MOV     #5,R1                           ; *,$$TMP2              3344
000014  001411                  2$:     BEQ     5$
000016  016700  000000G                 MOV     L$DLY,R0                        ; *,$$TMP1
000022  001404                          BEQ     4$
000024  005066  000002          3$:     CLR     2(SP)                           ; $$TMP
000030  005300                          DEC     R0                              ; $$TMP1
000032  001374                          BNE     3$
000034  005301                  4$:     DEC     R1                              ; $$TMP2
000036  000766                          BR      2$
000040  016700  000000G          5$:     MOV     RES.SLOT,R0                     ;                       3346
000044  006300                          ASL     R0
000046  006300                          ASL     R0
000050  066700  000000G                 ADD     RECEIVE.RING,R0
000054  032760  100000  000002          BIT     #100000,2(R0)
000062  001066                          BNE     11$
000064  012701  000031                  MOV     #31,R1                          ; *,$$TMP2              3349
000070  001411                  6$:     BEQ     9$
000072  016700  000000G                 MOV     L$DLY,R0                        ; *,$$TMP1
000076  001404                          BEQ     8$
000100  005066  000002          7$:     CLR     2(SP)                           ; $$TMP
000104  005300                          DEC     R0                              ; $$TMP1
000106  001374                          BNE     7$
000110  005301                  8$:     DEC     R1                              ; $$TMP2
000112  000766                          BR      6$
000114  016700  000000G          9$:     MOV     RES.SLOT,R0                     ;                       3351
000120  000300                          SWAB    R0
000122  106000                          RORB    R0
000124  006000                          ROR     R0
000126  006000                          ROR     R0
000130  142700  000077                  BICB    #77,R0
000134  005760  000016G                 TST     REC.ENVELOPE+16(R0)
000140  001417                          BEQ     10$
000142  016700  000000G                 MOV     RES.SLOT,R0                     ;                       3355
000146  006300                          ASL     R0
000150  006300                          ASL     R0
000152  066700  000000G                 ADD     RECEIVE.RING,R0
```

```
000156  052760  100000  000002          BIS     #100000,2(R0)
000164  012767  000031  000000G         MOV     #31,RET.STATUS          ;                       3356
000172  016700  000000G                 MOV     RET.STATUS,R0           ;                       3348
000176  000446                          BR      13$
000200  016700  000000G         10$:    MOV     RES.SLOT,R0             ;                       3361
000204  006300                          ASL     R0
000206  006300                          ASL     R0
000210  066700  000000G                 ADD     RECEIVE.RING,R0
000214  052760  100000  000002          BIS     #100000,2(R0)
000222  004767  177340                  JSR     PC,GET.RES.SLOT         ;                       3362
000226  005067  000000G                 CLR     RET.STATUS              ;                       3363
000232  016700  000000G                 MOV     RET.STATUS,R0           ;                       3348
000236  000426                          BR      13$
000240  005202                  11$:    INC     R2                      ; COUNT                 3342
000242  020227  072460                  CMP     R2,#72460               ; COUNT,*
000246  101660                          BLOS    1$
000250  016700  000000G                 MOV     RC25.ADDR,R0            ;                       3371
000254  016016  000002                  MOV     2(R0),(SP)              ; *,RC.REG
000260  011667  000002G                 MOV     (SP),RC25.DATA+2        ; RC.REG,*
000264  100006                          BPL     12$                     ;                       3373
000266  012767  000021  000000G         MOV     #21,RET.STATUS          ;                       3376
000274  016700  000000G                 MOV     RET.STATUS,R0           ;                       3336
000300  000405                          BR      13$
000302  012767  000011  000000G 12$:    MOV     #11,RET.STATUS          ;                       3381
000310  016700  000000G                 MOV     RET.STATUS,R0           ;                       3336
000314  022626                  13$:    CMP     (SP)+,(SP)+             ;                       3315
000316  000207                          RTS     PC

; Routine Size:  104 words,     Routine Base:  AB$CODE + 12464
; Maximum stack depth per invocation:  6 words

;               3386
;               3387  !
```

```
;          3388      global routine RANDOM_NUM : novalue =
;          3389  ! ++
;          3390  ! FUNCTIONAL DESCRIPTION:
;          3391  !
;          3392  !       THIS ROUTINE RECEIVED A SEEK FROM CALLER AND GENERAT A RANDOM
;          3393  !       NUMBER
;          3394  ! --
;          3395          begin
;          3396          P3 = 14657;                             ! CONSTANT NUMBER
;          3397          P6 = 34176;                             ! CONSTANT NUMBER
;          3398          P2 = .P3*(.P2 + .P6) mod .END_LBN;      ! RANDOM LBN NUMBER
;          3399  !   P1 = .TICKS mod 2;                         ! UNIT NUMBER
;          3400          return;
;          3401          end;


                                        .SBTTL  RANDOM.NUM AZTEC GLOBAL ROUTINE
000000  012767  034501  000000G   RANDOM.NUM::
                                        MOV     #34501,P3                       ;                       3396
000006  012767  102600  000000G                 MOV     #-75200,P6              ;                       3397
000014  016746  000000G                         MOV     P3,-(SP)                ;                       3398
000020  016746  000000G                         MOV     P2,-(SP)
000024  066716  000000G                         ADD     P6,(SP)
000030  004767  000000G                         JSR     PC,BL$MUL
000034  010016                                  MOV     R0,(SP)
000036  016746  000000G                         MOV     END.LBN,-(SP)
000042  004767  000000G                         JSR     PC,BL$MOD
000046  010067  000000G                         MOV     R0,P2
000052  062706  000006                          ADD     #6,SP                   ;                       3388
000056  000207                                  RTS     PC

; Routine Size:  24 words,     Routine Base:  AB$CODE + 13004
; Maximum stack depth per invocation:  4 words


;          3402
;          3403  !
;          3404
;          3405      global routine AVERAGE_TIME : novalue =
;          3406  ! ++
;          3407  ! FUNCTIONAL DESCRIPTION:
;          3408  !
;          3409  !       THIS ROUTINE CACULATE THE AVERAGE SEEK TIME FOR
;          3410  !       AZTEC MACHINE.
;          3411  ! --
;          3412          begin
;          3413          P4 = .TICKS + .SECONDS*60 + .MINUTES*60*60;     ! CONVERTED IT TO TOTAL TICKS
;          3414          DATA4 = (.P4*16)/.P6;                           ! GET THE AVERAGE TIME
;          3415          DATA2 = .P4 mod .P6;                            ! GET THE AVERAGE TIME FRACTION
;          3416          DATA3 = .P4/2;                                  ! TIME .5
;          3417          DATA3 = .DATA3 + .DATA2;                        ! GET THE TOTAL FRACTION
;          3418          return;
;          3419          end;


                                        .SBTTL  AVERAGE.TIME AZTEC GLOBAL ROUTINE
000000  010146                      AVERAGE.TIME::
```

```
                                                        MOV     R1,-(SP)                                ;                       3405
                                                        MOV     SECONDS,-(SP)                           ;                       3413
000002  016746  000000G                                MOV     #74,-(SP)
000006  012746  000074                                  JSR     PC,BL$MUL
000012  004767  000000G                                 MOV     R0,R1
000016  010001                                          ADD     TICKS,R1
000020  066701  000000G                                 MOV     MINUTES,(SP)
000024  016716  000000G                                 MOV     #7020,-(SP)
000030  012746  007020                                  JSR     PC,BL$MUL
000034  004767  000000G                                 ADD     R0,R1
000040  060001                                          MOV     R1,P4
000042  010167  000000G                                 MOV     P4,R0                           ;                       3414
000046  016700  000000G                                 ASL     R0
000052  006300                                          ASL     R0
000054  006300                                          ASL     R0
000056  006300                                          ASL     R0
000060  006300                                          MOV     R0,(SP)
000062  010016                                          MOV     P6,-(SP)
000064  016746  000000G                                 JSR     PC,BL$DIV
000070  004767  000000G                                 MOV     R0,DATA4
000074  010067  000000G                                 MOV     P4,(SP)                         ;                       3415
000100  016716  000000G                                 MOV     P6,-(SP)
000104  016746  000000G                                 JSR     PC,BL$MOD
000110  004767  000000G                                 MOV     R0,DATA2
000114  010067  000000G                                 MOV     P4,(SP)                         ;                       3416
000120  016716  000000G                                 MOV     #2,-(SP)
000124  012746  000002                                  JSR     PC,BL$DIV
000130  004767  000000G                                 MOV     R0,DATA3
000134  010067  000000G                                 ADD     DATA2,DATA3                     ;                       3417
000140  066767  000000G 000000G                        ADD     #14,SP                          ;                       3405
000146  062706  000014                                  MOV     (SP)+,R1
000152  012601                                          RTS     PC
000154  000207

; Routine Size:  55 words,      Routine Base:  AB$CODE + 13064
; Maximum stack depth per invocation:  8 words
```

```
;        3420     global routine EXAM_DATA : =
;        3421     ! ++
;        3422     ! FUNCTIONAL DESCRIPTION:
;        3423     !
;        3424     !   THE FUNCTION OF THIS ROUTINE IS TO EXAMINE THE
;        3425     !   FREE MEMORY FOR EXPECTED DATA.
;        3426     !
;        3427     ! IMPLICIT INPUTS:
;        3428     !       H_SADD
;        3429     !       BUF_LENGTH
;        3430     !       TIP
;        3431     ! IMPLICIT OUTPUTS:
;        3432     !       RETURN STATUS
;        3433     !
;        3434     ! SIDE EFFECTS:
;        3435     !       - NONE -
;        3436     ! --
;        3437           begin
;        3438
;        3439           local
;        3440               PATTERN,
;        3441               FLAG;
;        3442
;        3443           FLAG = ZERO;                                  ! INIT ERROR FLAG
;        3444           TEMP = .H_SADD;                               ! SAVE ADDR. IN TEMP. BUFFER
;        3445           H_EADD = .H_SADD - 2 + (.BUF_LENGTH*2);       ! END OF FREE HOST MEMORY
;        3446           PATTERN = .TIP;                               ! PUT PATTERN FOR COMPARE
;        3447
;        3448           incru COUNT from .H_SADD to .H_EADD by 2 do    ! EXAMINE CONTENTS OF MEMORY
;        3449               begin
;        3450
;        3451               if .TIP eqlu 1 then PATTERN = ( not .TEMP);
;        3452
;        3453               if .TIP eqlu 2 then PATTERN = .TEMP;
;        3454
;        3455               if ..TEMP nequ .PATTERN
;        3456               then
;        3457                   begin
;        3458                   FLAG = TRUE;
;        3459                   TIP = .PATTERN;
;        3460                   exitloop;
;        3461                   end;
;        3462
;        3463               TEMP = .TEMP + 2;
;        3464               end;
;        3465
;        3466           if .FLAG
;        3467           then
;        3468               begin                                     ! GET ERROR DATA
;        3469               P_MASK = 2;                               ! FOR TEST MODULE
;        3470               P1 = FMT2;
;        3471               P2 = ZERO;
;        3472               P3 = ZERO;
;        3473               P4 = .TIP;
;        3474               P5 = ..TEMP;
;        3475               P6 = .TEMP;
;        3476               return RET_STATUS = TRUE;
```

```
;         3477              end
;         3478          else
;         3479              return RET_STATUS = FALSE;
;         3480
;         3481          end;


                                        .SBTTL  EXAM.DATA AZTEC GLOBAL ROUTINE
000000  004167  000000G                EXAM.DATA::
                                        JSR     R1,$SAVE3                       ;                               3420
000004  005003                          CLR     R3                              ; FLAG                          3443
000006  016767  000000G 000000G        MOV     H.SADD,TEMP                     ;                               3444
000014  016700  000000G                MOV     BUF.LENGTH,R0                   ;                               3445
000020  006300                          ASL     R0
000022  066700  000000G                ADD     H.SADD,R0
000026  010067  000000G                MOV     R0,H.EADD
000032  162767  000002  000000G        SUB     #2,H.EADD
000040  016700  000000G                MOV     TIP,R0                          ; *,PATTERN                     3446
000044  016702  000000G                MOV     H.EADD,R2                       ;                               3448
000050  016701  000000G                MOV     H.SADD,R1                       ; *,COUNT
000054  000432                          BR      5$
000056  026727  000000G 000001  1$:    CMP     TIP,#1                          ;                               3451
000064  001003                          BNE     2$
000066  016700  000000G                MOV     TEMP,R0                         ; *,PATTERN
000072  005100                          COM     R0                              ; PATTERN
000074  026727  000000G 000002  2$:    CMP     TIP,#2                          ;                               3453
000102  001002                          BNE     3$
000104  016700  000000G                MOV     TEMP,R0                         ; *,PATTERN
000110  027700  000000G        3$:     CMP     @TEMP,R0                        ; *,PATTERN                     3455
000114  001405                          BEQ     4$
000116  012703  000001                 MOV     #1,R3                           ; *,FLAG                        3458
000122  010067  000000G                MOV     R0,TIP                          ; PATTERN,*                     3459
000126  000407                          BR      6$                              ;                               3457
000130  062767  000002  000000G  4$:    ADD     #2,TEMP                         ;                               3463
000136  062701  000002                 ADD     #2,R1                           ; *,COUNT                       3448
000142  020102                  5$:     CMP     R1,R2                           ; COUNT,*
000144  101744                          BLOS    1$
000146  006003                  6$:     ROR     R3                              ; FLAG                          3466
000150  103030                          BCC     7$
000152  112767  000002  000000G        MOVB    #2,P.MASK                       ;                               3469
000160  012767  000000G 000000G        MOV     #FMT2,P1                        ;                               3470
000166  005067  000000G                CLR     P2                              ;                               3471
000172  005067  000000G                CLR     P3                              ;                               3472
000176  016767  000000G 000000G        MOV     TIP,P4                          ;                               3473
000204  017767  000000G 000000G        MOV     @TEMP,P5                        ;                               3474
000212  016767  000000G 000000G        MOV     TEMP,P6                         ;                               3475
000220  012700  000001                 MOV     #1,R0                           ;                               3476
000224  010067  000000G                MOV     R0,RET.STATUS
000230  000207                          RTS     PC                              ;                               3437
000232  005067  000000G        7$:     CLR     RET.STATUS                      ;                               3479
000236  005000                          CLR     R0                              ;                               3437
000240  000207                          RTS     PC                              ;                               3420

; Routine Size:  81 words,      Routine Base:  AB$CODE + 13242
; Maximum stack depth per invocation:  5 words

;         3482
```

```
;        3483  !<BLF/PAGE>
```

```
;          3484
;          3485      global routine DATA_XMT_REC =
;          3486  ! ++
;          3487  !  THIS ROUTINE DOES THE FOLLOWING :
;          3488  !
;          3489  !  A. INITIALIZE COMMUNICATION AREA
;          3490  !  B. SEND EXECUTE AND SUPPLY COMMAND
;          3491  !  C. EXAMINE END RESPONSE PACKET. IF TIME EXPIRED,
;          3492  !     THEN SEND DUST_STATUS COMMAND.
;          3493  !  D. COMPARE TRANSMITTING DATA WITH RECEIVING DATA
;          3494  !  E. REPORT ERROR, IF THERE IS ONE
;          3495  !
;          3496  !  IMPLICIT INPUTS:
;          3497  !  DMC_TEST
;          3498  !  BYT_CNT
;          3499  ! --
;          3500        begin
;          3501
;          3502        local
;          3503            FLAG;
;          3504
;          3505        FLAG = ZERO;                                ! INIT ERROR FLAG
;          3506
;          3507        if AZTEC_READY () then return .RET_STATUS;      ! GET AZTEC READY FOR OPERATION
;          3508
;          3509        CMD_REF = 3;                                ! COMMAND REFERENCE NUMBER
;          3510        BUF_DESCRPTR = .DMC_TEST;                   ! DMCODE STARTING ADDRESS
;          3511        BYTE_COUNT = .BYT_CNT;                      ! BYTE COUNTS
;          3512
;          3513        if EX_SUP_PRG () then return .RET_STATUS;       ! ISSUE AN EXECUTE SUPPLIED -
;          3514
;          3515        CMD_REF = 4;                                ! COMMAND REFERENCE #
;          3516        BUF_DESCRPTR = TIP;                         ! CLEAN THE BUFFER
;          3517        BYTE_COUNT = 02;                            ! SET BYTE COUNTS = 2
;          3518
;          3519        if REC_DATA () then return .RET_STATUS; ! SEND A RECEIVE DATA COMMAND
;          3520
;          3521  ! ++
;          3522  !  COMPARE TRANSMITTING DATAS AND RECEIVING DATAS
;          3523  !  IF ERROR, REPORT BLOCK LENGTH, BAD DATA AND GOOD DATA.
;          3524  ! --
;          3525
;          3526        incru J from 0 to 256 do
;          3527            begin
;          3528
;          3529            if .XMT_DATA_BUF [.J] nequ .RCV_DATA_BUF [.J]
;          3530            then
;          3531                begin
;          3532                FLAG = TRUE;
;          3533                exitloop;
;          3534                end;
;          3535
;          3536            end;
;          3537
;          3538        if .FLAG
;          3539        then
;          3540            begin
```

```
;       3541            P_MASK = 2;                         ! GET ERROR INFO
;       3542            PT = FMT2;                          ! FOR TESTMODULE
;       3543            P2 = ZERO;
;       3544            P3 = ZERO;
;       3545            P4 = .TIP;
;       3546            P5 = ..TEMP;
;       3547            P6 = .TEMP;
;       3548            return RET_STATUS = TRUE;
;       3549            end
;       3550        else
;       3551            return RET_STATUS = FALSE;
;       3552
;       3553        end;


                                        .SBTTL  DATA.XMT.REC AZTEC GLOBAL ROUTINE
000000  004167  000000G                DATA.XMT.REC::
                                        JSR     R1,$SAVE3                       ;                       3485
000004  005003                          CLR     R3                              ; FLAG                  3505
000006  004767  000000V                JSR     PC,AZTEC.READY                  ;                       3507
000012  006000                          ROR     R0
000014  103003                          BCC     1$
000016  016700  000000G                MOV     RET.STATUS,R0
000022  000207                          RTS     PC
000024  012767  000003  000000G  1$:    MOV     #3,CMD.REF                      ;                       3509
000032  016767  000000G 000000G        MOV     DMC.TEST,BUF.DESCRPTR           ;                       3510
000040  016767  000000G 000000G        MOV     BYT.CNT,BYTE.COUNT              ;                       3511
000046  004767  167426                 JSR     PC,EX.SUP.PRG                   ;                       3513
000052  006000                          ROR     R0
000054  103003                          BCC     2$
000056  016700  000000G                MOV     RET.STATUS,R0
000062  000207                          RTS     PC
000064  012767  000004  000000G  2$:    MOV     #4,CMD.REF                      ;                       3515
000072  012767  000000G 000000G        MOV     #TIP,BUF.DESCRPTR               ;                       3516
000100  012767  000002  000000G        MOV     #2,BYTE.COUNT                   ;                       3517
000106  004767  171172                 JSR     PC,REC.DATA                     ;                       3519
000112  006000                          ROR     R0
000114  103003                          BCC     3$
000116  016700  000000G                MOV     RET.STATUS,R0
000122  000207                          RTS     PC
000124  005002                  3$:     CLR     R2                              ; J                     3526
000126  010201                  4$:     MOV     R2,R1                           ; J,*                   3529
000130  006301                          ASL     R1
000132  010200                          MOV     R2,R0                           ; J,*
000134  006300                          ASL     R0
000136  026160  000000G 000000G        CMP     XMT.DATA.BUF(R1),RCV.DATA.BUF(R0) ;
000144  001403                          BEQ     5$
000146  012703  000001                 MOV     #1,R3                           ; *,FLAG                3532
000152  000404                          BR      6$                              ;                       3531
000154  005202                  5$:     INC     R2                              ; J                     3526
000156  020227  000400                 CMP     R2,#400                         ; J,*
000162  101761                          BLOS    4$
000164  006003                  6$:     ROR     R3                              ; FLAG                  3538
000166  103030                          BCC     7$
000170  112767  000002  000000G        MOVB    #2,P.MASK                       ;                       3541
000176  012767  000000G 000000G        MOV     #FMT2,P1                        ;                       3542
000204  005067  000000G                CLR     P2                              ;                       3543
```

```
000210  005067  000000G                        CLR     P3                              ;                       3544
000214  016767  000000G 000000G                MOV     TIP,P4                          ;                       3545
000222  017767  000000G 000000G                MOV     @TEMP,P5                        ;                       3546
000230  016767  000000G 000000G                MOV     TEMP,P6                         ;                       3547
000236  012700  000001                         MOV     #1,R0                           ;                       3548
000242  010067  000000G                        MOV     R0,RET.STATUS
000246  000207                                 RTS     PC                              ;                       3500
000250  005067  000000G                7$:     CLR     RET.STATUS                      ;                       3551
000254  005000                                 CLR     R0                              ;                       3500
000256  000207                                 RTS     PC                              ;                       3485

; Routine Size:  88 words,      Routine Base:  AB$CODE + 13504
; Maximum stack depth per invocation:  5 words
```

```
;          3554      global routine DM_ADDR_SETUP : novalue =
;          3555    ! ++
;          3556    !  THIS ROUTINE PASS TRANSMITTING AND RECEIVING BUFFERS STARTING
;          3557    !  ADDRESS TO DM CODE
;          3558    !
;          3559    !  IMPLICIT INPUTS
;          3560    !  DM_XMT
;          3561    !  DM_REC
;          3562    !
;          3563    !
;          3564    ! --
;          3565    !
;          3566          begin
;          3567          DM_XMT = XMT_DATA_BUF [0];              ! XMT BUFFER 1 STARTING ADDR.
;          3568          DM_REC = RCV_DATA_BUF [0];              ! REC BUFFER 1 STARTING ADDR.
;          3569          return RET_STATUS = FALSE;
;          3570          end;


                                        .SBTTL  DM.ADDR.SETUP AZTEC GLOBAL ROUTINE
000000  012767  000000G 000000G  DM.ADDR.SETUP::
                                        MOV     #XMT.DATA.BUF,DM.XMT            ;       3567
000006  012767  000000G 000000G         MOV     #RCV.DATA.BUF,DM.REC            ;       3568
000014  005067  000000G                 CLR     RET.STATUS                      ;       3569
000020  000207                          RTS     PC                              ;       3554

; Routine Size:  9 words,       Routine Base:  AB$CODE + 13764
; Maximum stack depth per invocation:  0 words

;          3571
;          3572  !<BLF/PAGE>
```

```
;       3573
;       3574        global routine WRT_PROTECT_TST : novalue =
;       3575
;       3576    !++
;       3577    ! FUNCTIONAL DESCRIPTIONS:
;       3578    !
;       3579    !       THIS ROUTINE WILL ASK THE OPERATOR TO MAKE SURE THE WRITE
;       3580    !       PROTECT SWITCH FOR THE UNIT IS IN THE OFF POSITION.   IT
;       3581    !       WILL GET UNIT STATUS TO VERIFY THAT THE CONTROLLER KNOW IT
;       3582    !       IS NOT WRITE PROTECTED.  THEN THE OPERATOR WILL BE ASKED TO
;       3583    !       PUT THE WRITE PROTECT SWITCH IN THE ON POSTION AND A GUS
;       3584    !       WILL BE DONE TO MAKE SURE THE CONTROLLER RECOGNIZES THA THE
;       3585    !       THE UNIT IS WRITE PROTECTED.
;       3586    !
;       3587            begin
;       3588    !
;       3589    ! DISPLAY MESSAGE "TURN OFF WRITE PROTECT SWITCH"  THEN ISSUE
;       3590    ! A  GET UNIT STATUS COMMAND AND EXAMINE THE UNIT FLAG.
;       3591    !
;       3592            MANU_SW = ONE;                          ! MANUAL SWITCH IS SET TO 'YES'
;       3593            GMANIL (QST14, MANU_SW, 1, YES, 0);     ! DISPLAY MESSAGE TURN OFF WRT
;       3594                                                    ! PROTECT SWITCH
;       3595    !
;       3596    ! PROGRAM WAITING FOR GO (CR) SIGNAL
;       3597    !
;       3598
;       3599            if (.MANU_SW eqlu YES)
;       3600            then                                    ! WAITING FOR CR SIGNAL
;       3601                begin
;       3602                CMD_REF = 3;                        ! COMMAND REFERENCE NUMBER
;       3603
;       3604                if GET_UNIT_STATUS ()               ! ISSUE A GET UNIT STATUS COMMAND
;       3605                then                                ! IF RESPONSE STATUS BIT ERROR, THEN
;       3606                    begin
;       3607                    RET_UNIT_FLAG = .RET_UNIT_FLAG and %o'020000';  ! MASKED OUT OTHER BITS
;       3608
;       3609                    if .RET_UNIT_FLAG eqlu UF_WPH   ! IF WRT PROT. FLAG SET
;       3610                    then                            ! ERROR
;       3611                        begin
;       3612                        RETRIES = TRUE;
;       3613                        end;
;       3614
;       3615                    end;
;       3616
;       3617                end
;       3618            else
;       3619                RETRIES = TRUE;
;       3620
;       3621    !
;       3622    ! DISPLAY MESSAGE "TURN ON THE WRITE PROTECT SWITCH" THEN
;       3623    ! ISSUE A GET UNIT SATUS COMMAND AND EXAMINE THE RESPONSE
;       3624    ! UNIT FLAGS.
;       3625    !
;       3626            SWITCH2 = ONE;                          ! SET MANUAL SWITCH
;       3627            GMANIL (QST15, SWITCH2, 1, YES, 0);     ! DISPLAY MESSAGE TURN OFF WRT
;       3628                                                    ! PROTECT SWITCH
;       3629    !
```

```
;        3630  ! PROGRAM WAITING FOR GO (CR) SIGNAL
;        3631  !
;        3632
;        3633          if (.SWITCH2 eqlu YES)
;        3634          then                                    ! WAITING FOR CR SIGNAL
;        3635              begin
;        3636              CMD_REF = 4;                        ! COMMAND REFERENCE NUMBER
;        3637
;        3638              if GET_UNIT_STATUS ()               ! ISSUE A GET UNIT STATUS COMMAND
;        3639              then                                ! IF RESPONSE STATUS BIT ERROR, THEN
;        3640                  begin
;        3641                  RET_UNIT_FLAG = .RET_UNIT_FLAG and %o'020000';  ! MASKED OUT OTHER BITS
;        3642
;        3643                  if .RET_UNIT_FLAG nequ UF_WPH   ! IF WRT PROT. FLAG CLEAR
;        3644                  then                            ! ERROR
;        3645                      begin
;        3646  !                   PRINTF (MSG_WRP_ERR2, .LOG_UNIT);       !
;        3647  !                   ERRDF (73, MSG_COM_WPT, 0);     ! REPORT ERROR
;        3648                          RETRIES = TRUE;
;        3649                          end;
;        3650
;        3651                      end
;        3652                  else
;        3653                      RETRIES = TRUE;
;        3654
;        3655                  end;
;        3656
;        3657          return;
;        3658          end;


                                        .SBTTL  WRT.PROTECT.TST AZTEC GLOBAL ROUTINE
000000  012767  000001  000000G  WRT.PROTECT.TST::
                                        MOV     #1,MANU.SW                      ;       3592
000006  104443                          TRAP    43                              ;       3593
000010  000404                          .WORD   404
000012  000000G                         .WORD   MANU.SW
000014  000130                          .WORD   130
000016  000000G                         .WORD   QST14
000020  000001                          .WORD   1
000022  026727  000000G 000001          CMP     MANU.SW,#1                      ;       3599
000030  001022                          BNE     1$
000032  012767  000003  000000G         MOV     #3,CMD.REF                      ;       3602
000040  004767  175436                  JSR     PC,GET.UNIT.STATUS              ;       3604
000044  006000                          ROR     R0
000046  103016                          BCC     2$
000050  042767  157777  000000G         BIC     #157777,RET.UNIT.FLAG           ;       3607
000056  026727  000000G 020000          CMP     RET.UNIT.FLAG,#20000            ;       3609
000064  001007                          BNE     2$
000066  012767  000001  000000G         MOV     #1,RETRIES                      ;       3612
000074  000403                          BR      2$                              ;       3599
000076  012767  000001  000000G  1$:     MOV     #1,RETRIES                      ;       3619
000104  012767  000001  000000G  2$:     MOV     #1,SWITCH2                      ;       3626
000112  104443                          TRAP    43                              ;       3627
000114  000404                          .WORD   404
000116  000000G                         .WORD   SWITCH2
000120  000130                          .WORD   130
```

```
000122  000000G                                 .WORD   QST15
000124  000001                                  .WORD   1
000126  026727  000000G 000001                  CMP     SWITCH2,#1                      ;                       3633
000134  001025                                  BNE     4$
000136  012767  000004  000000G                 MOV     #4,CMD.REF                      ;                       3636
000144  004767  175332                          JSR     PC,GET.UNIT.STATUS              ;                       3638
000150  006000                                  ROR     R0
000152  103013                                  BCC     3$
000154  042767  157777  000000G                 BIC     #157777,RET.UNIT.FLAG           ;                       3641
000162  026727  000000G 020000                  CMP     RET.UNIT.FLAG,#20000            ;                       3643
000170  001407                                  BEQ     4$
000172  012767  000001  000000G                 MOV     #1,RETRIES                      ;                       3648
000200  000207                                  RTS     PC                              ;                       3638
000202  012767  000001  000000G         3$:     MOV     #1,RETRIES                      ;                       3653
000210  000207                          4$:     RTS     PC                              ;                       3574

; Routine Size:  69 words,      Routine Base:  AB$CODE + 14006
; Maximum stack depth per invocation:  2 words
```

```
;        3659     global routine AZTEC_READY =
;        3660
;        3661 !++
;        3662 ! FUNCTIONAL DESCRIPTIONS:
;        3663 !       THIS ROUTINE CALL OTHER ROUTINES TO GET THE AZTEC READY
;        3664 !       TO DO THE DM OR READ/WRITE OPERATION.
;        3665 !
;        3666 !       PERFORM OPERATIONS AS FOLLOWING ORDER:
;        3667 !
;        3668 !       1. DEFINED INITIALIZATION CONSTANTS.
;        3669 !
;        3670 !       2. DO STEP 1 THROUGH STEP 3 CHECK FOR ANY ERRORS
;        3671 !          IN EACH STEP.
;        3672 !
;        3673 !       3. SET UP COMMUNICATION AREA'S.
;        3674 !
;        3675 !       4. SET HOST SETTABLE UNIT CHARACTERISTICS AND OBTAIN THOSE
;        3676 !          UNIT CHARACTERISTICS THAT ARE ESSENTIAL FOR PROPER CLASS
;        3677 !          DRIVER OPERATION.
;        3678 !
;        3679 !       5. BRING A UNIT 'UNIT-ONLINE.  THE UNIT IS SPUN-UP, IF NECESSARY,
;        3680 !          AND ITS HEADS ARE LOADED PRIOR TO RETURNING THE ONLINE COMMANDS'S
;        3681 !          END MESSAGE.
;        3682 !
;        3683 ! FORMAL PARAMETERS:
;        3684 !       -NONE -
;        3685 !
;        3686 ! IMPLICIT INPUTS:
;        3687 !
;        3688 ! IMPLICIT OUTPUTS:
;        3689 !       AS A RESULT OF THIS ROUTINE THE COMMUNICATION AREA WILL
;        3690 !       BE INITIALIZED AND UNIT IS SPUN-UP.
;        3691 !
;        3692 ! COMPLETTION CODES:
;        3693 !
;        3694 ! SIDE EFFECTS:
;        3695 !       - NONE -
;        3696         begin
;        3697         B_MASK = %o'17';                    ! SET MASK BIT FOR COMPLETE INIT.
;        3698         DATA1<15, 1> = TRUE;                ! SET BIT 15 FOR STEP-1 WRITE
;        3699         DATA1<14, 1> = 0;                   ! NO DIAGNOSTIC WRAP MODE
;        3700         DATA1<11, 3> = SND_SIZ;             ! SET UP COMMAND RINGS LENGTH
;        3701         DATA1<8, 3> = REC_SIZ;              ! SET RESPONSE RING LENGTH
;        3702         DATA1<7, 1> = 0;                    ! DISABLE INTERRUPT
;        3703         DATA1<0, 7> = 0;                    ! LOAD NO  VECTOR ADDRESS
;        3704         DATA2 = RINGBASE;                   ! LOAD COMMUNICATIONS AREA ADDRESS
;        3705         DATA3 = ZERO;                       ! HI-ORDER ADDR = ZERO
;        3706         DATA4 = %o'177403';                 ! "LAST FAIL" PACKET RESPONSE BIT SET
;        3707         CMD_SLOT = 0;                       ! CLEAR COMMAND RING SLOT POINTER
;        3708         RES_SLOT = 0;                       ! CLEAR RESPONSE RING SLOT POINTER
;        3709         TICKS = 0;                          ! CLEAR TICK AREA
;        3710         SECONDS = 0;                        ! CLEAR SECOND AREA
;        3711         MINUTES = 0;                        ! CLEAR MINUTES AREA
;        3712
;        3713         if AZP_INIT ()                      ! DO STEP INIT AND CHECK FOR ERROR
;        3714         then
;        3715             return .RET_STATUS;
```

```
;         3716
;         3717          if INIT_COM_AREA ()                    ! INIT THE COMMUNICATION AREA
;         3718          then                                   ! ERROR ?
;         3719              return .RET_STATUS;
;         3720
;         3721          CMD_REF = 01;                          ! SET COMMAND REFERENCE TO 1
;         3722
;         3723          if SET_CNTLR_CHAR ()                   ! ISSUE SET CONTROLLER CHAR CMD
;         3724          then
;         3725              return .RET_STATUS;                ! IF COMMAND FAILED
;         3726
;         3727          CMD_REF = 2;                           ! SET COMMAND REFERENCE TO 2
;         3728
;         3729          if ON_LINE ()                          ! ISSUE ON LINE CMOOAND
;         3730          then                                   ! IF COMMAND FAILED
;         3731              return .RET_STATUS;
;         3732
;         3733          return RET_STATUS = FALSE;
;         3734          end;

                                        .SBTTL  AZTEC.READY AZTEC GLOBAL ROUTINE
000000  112767  000017  000000G  AZTEC.READY::
                                        MOVB    #17,B.MASK                      ;3697
000006  012767  122000  000000G         MOV     #122000,DATA1                   ;3703
000014  012767  000000G 000000G         MOV     #RINGBASE,DATA2                 ;3704
000022  005067  000000G                 CLR     DATA3                           ;3705
000026  012767  177403  000000G         MOV     #-375,DATA4                     ;3706
000034  005067  000000G                 CLR     CMD.SLOT                        ;3707
000040  005067  000000G                 CLR     RES.SLOT                        ;3708
000044  005067  000000G                 CLR     TICKS                           ;3709
000050  005067  000000G                 CLR     SECONDS                         ;3710
000054  005067  000000G                 CLR     MINUTES                         ;3711
000060  004767  165324                  JSR     PC,AZP.INIT                     ;3713
000064  006000                          ROR     R0
000066  103003                          BCC     1$
000070  016700  000000G                 MOV     RET.STATUS,R0                   ;3715
000074  000207                          RTS     PC
000076  004767  166104          1$:     JSR     PC,INIT.COM.AREA                ;3717
000102  006000                          ROR     R0
000104  103003                          BCC     2$
000106  016700  000000G                 MOV     RET.STATUS,R0                   ;3719
000112  000207                          RTS     PC
000114  012767  000001  000000G 2$:     MOV     #1,CMD.REF                      ;3721
000122  004767  171264                  JSR     PC,SET.CNTLR.CHAR               ;3723
000126  006000                          ROR     R0
000130  103003                          BCC     3$
000132  016700  000000G                 MOV     RET.STATUS,R0                   ;3725
000136  000207                          RTS     PC
000140  012767  000002  000000G 3$:     MOV     #2,CMD.REF                      ;3727
000146  004767  172524                  JSR     PC,ON.LINE                      ;3729
000152  006000                          ROR     R0
000154  103003                          BCC     4$
000156  016700  000000G                 MOV     RET.STATUS,R0                   ;3731
000162  000207                          RTS     PC
000164  005067  000000G 4$:     CLR     RET.STATUS                      ;3733
000170  005000                          CLR     R0                              ;3696
```

```
000172  000207                  RTS     PC                              ;                       3659

; Routine Size:  62 words,      Routine Base:  AB$CODE + 14220
; Maximum stack depth per invocation:  1 word
```

```
;          3735      global routine DO_RETRIES : novalue =
;          3736      !+
;          3737      ! COME HERE ON AN ERROR AND KEEP TRACK OF RETRIES.
;          3738      ! IF NECESSARY DROP UNIT UNDER TEST.
;          3739      !
;          3740            begin
;          3741            NUM_RETRIES = .NUM_RETRIES + 1;
;          3742
;          3743            if (.NUM_RETRIES lequ .SWP_RETRIES)
;          3744            then
;          3745                begin
;          3746                PRINTB (FMT$A, .NUM_RETRIES);
;          3747                end
;          3748            else
;          3749                begin
;          3750                RETRIES = FALSE;
;          3751
;          3752                if not .SWP_CONTINUE
;          3753                then
;          3754                    begin
;          3755                    DODU (.LOG_UNIT);
;          3756                    DOCLN;
;          3757                    end;
;          3758
;          3759                end;
;          3760
;          3761            end;


                                        .SBTTL  DO.RETRIES AZTEC GLOBAL ROUTINE
000000  005267  000000G                DO.RETRIES::
                                        INC     NUM.RETRIES                     ;                       3741
000004  026767  000000G 000000G         CMP     NUM.RETRIES,SWP.RETRIES         ;                       3743
000012  101013                          BHI     1$
000014  016746  000000G                 MOV     NUM.RETRIES,-(SP)               ;                       3746
000020  012746  000000G                 MOV     #FMT$A,-(SP)
000024  012746  000002                  MOV     #2,-(SP)
000030  010600                          MOV     SP,R0                           ; SP,*
000032  104414                          TRAP    14
000034  062706  000006                  ADD     #6,SP                           ;                       3745
000040  000207                          RTS     PC                              ;                       3743
000042  005067  000000G         1$:     CLR     RETRIES                         ;                       3750
000046  032767  000001  000000G         BIT     #1,SWP.CONTINUE                 ;                       3752
000054  001004                          BNE     2$
000056  016700  000000G                 MOV     LOG.UNIT,R0                     ;                       3755
000062  104451                          TRAP    51
000064  104444                          TRAP    44
000066  000207                  2$:     RTS     PC                              ;                       3735

; Routine Size:  28 words,      Routine Base:  AB$CODE + 14414
; Maximum stack depth per invocation:  5 words
```

```
;       3762      global routine DECODE : novalue =          !Decodes failing SA reg data
;       3763
;       3764  !++
;       3765  ! Functional Description :
;       3766  !       Due  to the implimentation of the DUP  and UQ Port protocol there
;       3767  !       are  two levels  at which an issued command to  a port/controller
;       3768  !       can fail and they are:
;       3769  !
;       3770  !         1. The issued command can time out.
;       3771  !
;       3772  !         2. An  error  can be posted in SA register bit 15 by the port  to
;       3773  !            report an error.
;       3774  !
;       3775  !         3. The  issued  command  to  the port/controller can  be executed
;       3776  !            correctly without  any  errors but the response  packet status
;       3777  !            field could have an error or  status other than success posted.
;       3778  !
;       3779  !       This  routine  will  then be called when the return from a queued
;       3780  !       command  comes  back with an error code or non successfull status
;       3781  !       code.  This is  by  definition  when bit 0 in the returned status
;       3782  !       is equal to 1.
;       3783  !
;       3784  !
;       3785  ! Formal Parameters :
;       3786  !       none
;       3787  !
;       3788  ! Implicit Inputs :
;       3789  !       RET_STATUS:     Stored in this global storage is the returned error
;       3790  !                       code or  non-successful  status  code from a queued
;       3791  !                       command.
;       3792  !
;       3793  ! Implicit Outputs :
;       3794  !       none
;       3795  !
;       3796  ! Completion Codes :
;       3797  !       none
;       3798  !
;       3799  ! Side Effects :
;       3800  !       after execution of  this  routine  the  RC25  controller
;       3801  !       is initialized  aborting any DM code running in the controller.
;       3802  !--
;       3803
;       3804          begin
;       3805
;       3806          !+
;       3807          ! Use the contents of "RET_STATUS" to select what
;       3808          ! type error or non-successful status code is to
;       3809          ! be processed.
;       3810          !-
;       3811
;       3812          if .RET_STATUS eqlu ONE then return RET_STATUS = ZERO;  ! NO ACTION IF RET_STATUS IS ONE
;       3813
;       3814          selectoneu .RET_STATUS of
;       3815              set
;       3816              !
;       3817              ! "Port/Controller time out" error code
;       3818              !
```

```
;       3819                ! Port/Controller timed out after the specified
;       3820                ! time out interval.
;       3821                !
;       3822
;       3823                [CTO_CODE] :                            !Code equals %o'11'
;       3824                    begin
;       3825                    PRINTF (.EMSG_STRUCT [MSG3]);
;       3826                    end;
;       3827                !
;       3828                ! "Port fatal error" code
;       3829                !
;       3830                ! The error bit in the SA Register was set when
;       3831                ! examined.  This error indicates a Port fatal error code.
;       3832                !
;       3833
;       3834                [PFE_CODE] :                            !Code equals %o'21'
;       3835                    begin
;       3836                    TEMP = .RC25_DATA [RCSA, RCSA_ERC];
;       3837
;       3838                    if .TEMP gequ 200
;       3839                    then
;       3840                        begin
;       3841                        PRINTF (.RC_STRUCTURE [.TEMP - 200]);       !print RCSA error code
;       3842                        end
;       3843                    else
;       3844                        begin
;       3845                        PRINTF (.PFE_STRUCT [.TEMP]);
;       3846                        end;
;       3847
;       3848                    end;
;       3849                !
;       3850                ! "Return status error" code
;       3851                !
;       3852                ! This indicates that a non-successful return status
;       3853                ! code was returned  from an issued command.
;       3854                !
;       3855
;       3856                [RSE_CODE] :                            !Code equals %o'31'
;       3857                    begin
;       3858                    PRINTF (.EMSG_STRUCT [MSG0]);
;       3859                    !
;       3860                    ! Look at UQPORT connection ID field to determine the type
;       3861                    ! of response
;       3862                    !
;       3863
;       3864                    if .REC_ENVELOPE [.RES_SLOT, CONN_ID] eqlu 2 ! CONN_ID = DUP
;       3865                    then
;       3866                        begin
;       3867                        PRINTF (.SDUP_STRUCT [.REC_ENVELOPE [.RES_SLOT, STATUS]]);
;       3868                        end
;       3869                    else
;       3870                        begin
;       3871                        PRINTF (.SMSCP_STRUCT [.REC_ENVELOPE [.RES_SLOT, STA_CODE]]);
;       3872                        end;
;       3873
;       3874                    end;
;       3875                !
```

```
;        3876                ! "SERIOUS EXCEPTION" error code
;        3877                !
;        3878
;        3879                [SEX_CODE] :                        !Code equals %o'601'
;        3880                    begin
;        3881                    PRINTF (.EMSG_STRUCT [MSG2]);
;        3882                    end;
;        3883                !
;        3884                ! This is here to trap any unknown  return status codes
;        3885                ! sent to this routine.
;        3886                !
;        3887
;        3888                [otherwise] :                       !Code equals non of the above
;        3889                    begin
;        3890                    PRINTF (.EMSG_STRUCT [MSG3]);
;        3891                    end;
;        3892                tes;
;        3893
;        3894            !
;        3895            ! All errors are fatal so init the RC25
;        3896            !
;        3897            WRT_RC25 (RCIP, ALL_ONES);              !Init the controller
;        3898            RET_STATUS = ZERO;
;        3899            return;
;        3900            end;

                                        .SBTTL  DECODE AZTEC GLOBAL ROUTINE
000000  010146                  DECODE::MOV     R1,-(SP)                    ;                   3762
000002  026727  000000G 000001          CMP     RET.STATUS,#1               ;                   3812
000010  001004                          BNE     1$
000012  005067  000000G                 CLR     RET.STATUS
000016  000167  000400                  JMP     10$
000022  016701  000000G         1$:     MOV     RET.STATUS,R1               ;                   3814
000026  020127  000011                  CMP     R1,#11
000032  001007                          BNE     2$
000034  016746  000006G                 MOV     EMSG.STRUCT+6,-(SP)         ;                   3825
000040  012746  000001                  MOV     #1,-(SP)
000044  010600                          MOV     SP,R0                       ; SP,*
000046  104417                          TRAP    17
000050  000555                          BR      9$                          ;                   3814
000052  020127  000021          2$:     CMP     R1,#21
000056  001036                          BNE     4$
000060  016767  000002G 000000G         MOV     RC25.DATA+2,TEMP            ;                   3836
000066  042767  174000  000000G         BIC     #174000,TEMP
000074  026727  000000G 000310          CMP     TEMP,#310                   ;                   3838
000102  103412                          BLO     3$
000104  016700  000000G                 MOV     TEMP,R0                     ;                   3841
000110  006300                          ASL     R0
000112  016046  177160G                 MOV     RC.STRUCTURE-620(R0),-(SP)
000116  012746  000001                  MOV     #1,-(SP)
000122  010600                          MOV     SP,R0                       : SP,*
000124  104417                          TRAP    17
000126  000526                          BR      9$                          ;                   3838
000130  016700  000000G         3$:     MOV     TEMP,R0                     ;                   3845
000134  006300                          ASL     R0
000136  016046  000000G                 MOV     PFE.STRUCT(R0),-(SP)
```

```
000142  012746  000001                          MOV     #1,-(SP)
000146  010600                                  MOV     SP,R0                           ; SP,*
000150  104417                                  TRAP    17
000152  000514                                  BR      9$                              ;                                       3814
000154  020127  000031                  4$:     CMP     R1,#31
000160  001071                                  BNE     7$
000162  016746  000000G                         MOV     EMSG.STRUCT,-(SP)               ;                                       3858
000166  012746  000001                          MOV     #1,-(SP)
000172  010600                                  MOV     SP,R0                           ; SP,*
000174  104417                                  TRAP    17
000176  016700  000000G                         MOV     RES.SLOT,R0                     ;                                       3864
000202  000300                                  SWAB    R0
000204  106000                                  RORB    R0
000206  006000                                  ROR     R0
000210  006000                                  ROR     R0
000212  142700  000077                          BICB    #77,R0
000216  126027  000003G 000002                  CMPB    REC.ENVELOPE+3(R0),#2
000224  001022                                  BNE     5$
000226  016700  000000G                         MOV     RES.SLOT,R0                     ;                                       3867
000232  000300                                  SWAB    R0
000234  106000                                  RORB    R0
000236  006000                                  ROR     R0
000240  006000                                  ROR     R0
000242  142700  000077                          BICB    #77,R0
000246  016000  000016G                         MOV     REC.ENVELOPE+16(R0),R0
000252  006300                                  ASL     R0
000254  016016  000000G                         MOV     SDUP.STRUCT(R0),(SP)
000260  012746  000001                          MOV     #1,-(SP)
000264  010600                                  MOV     SP,R0                           ; SP,*
000266  104417                                  TRAP    17
000270  000423                                  BR      6$                              ;                                       3864
000272  016700  000000G                 5$:     MOV     RES.SLOT,R0                     ;                                       3871
000276  000300                                  SWAB    R0
000300  106000                                  RORB    R0
000302  006000                                  ROR     R0
000304  006000                                  ROR     R0
000306  142700  000077                          BICB    #77,R0
000312  116000  000016G                         MOVB    REC.ENVELOPE+16(R0),R0
000316  042700  177740                          BIC     #177740,R0
000322  006300                                  ASL     R0
000324  016016  000000G                         MOV     SMSCP.STRUCT(R0),(SP)
000330  012746  000001                          MOV     #1,-(SP)
000334  010600                                  MOV     SP,R0                           ; SP,*
000336  104417                                  TRAP    17
000340  005726                          6$:     TST     (SP)+                           ;                                       3857
000342  000420                                  BR      9$                              ;                                       3814
000344  020127  000601                  7$:     CMP     R1,#601
000350  001007                                  BNE     8$
000352  016746  000004G                         MOV     EMSG.STRUCT+4,-(SP)             ;                                       3881
000356  012746  000001                          MOV     #1,-(SP)
000362  010600                                  MOV     SP,R0                           ; SP,*
000364  104417                                  TRAP    17
000366  000406                                  BR      9$                              ;                                       3814
000370  016746  000006G                 8$:     MOV     EMSG.STRUCT+6,-(SP)             ;                                       3890
000374  012746  000001                          MOV     #1,-(SP)
000400  010600                                  MOV     SP,R0                           ; SP,*
000402  104417                                  TRAP    17
```

```
000404  012700  177777                9$:     MOV     #-1,R0                  ; *,RCM.REG                             3897
000410  010077  000000G                        MOV     R0,@RC25.ADDR           ; RCM.REG,*
000414  005067  000000G                        CLR     RET.STATUS              ;                                       3898
000420  022626                               CMP     (SP)+,(SP)+             ;                                       3762
000422  012601                       10$:    MOV     (SP)+,R1
000424  000207                               RTS     PC

; Routine Size:  139 words,     Routine Base:  AB$CODE + 14504
; Maximum stack depth per invocation:  6 words


;         3901
;         3902
;         3903  end
;         3904
;         3905  eludom


;                                      OTS external references
                                               .GLOBL  $SAVE5, $SAVE3, $SAVE2, BL$SHF
                                               .GLOBL  BL$DIV, BL$MOD, BL$MUL


;                                      PSECT SUMMARY
;
;       Psect Name                     Words     Attributes
;        AA$CODE                         267       RO ,  I  ,  LCL,  REL,  CON
;        AB$CODE                        3373       RO ,  I  ,  LCL,  REL,  CON


;                                  LIBRARY STATISTICS
;
;                                              -------- Symbols --------     Blocks
;       File                                    Total    Loaded   Percent      Read
;
;  SPIDER$USERS:[LAKSHMANA.11REL.REAL]AZTEC0.L16;1
;                                                523       212        40        77




;                                          COMMAND QUALIFIERS

;       BLISS /PDP11/LIST ZRCFA2.B16/EN:NOEIS

; Size:          3581 code + 59 data words
; Run Time:        01:28.7
; Elapsed Time:    07:29.4
; Memory Used:  253 pages
; Compilation Complete
```

```
0001  MODULE ZRCFA3 (%TITLE 'CZRCFA0 RC25 FR END TEST'
0002                  IDENT = 'V01.0',
0003                  OPTLEVEL = 0,
0004                  ADDRESSING_MODE (RELATIVE)
0005                  ) =
0006  BEGIN
0007  !<BLF/LOWERCASE_KEY>
0008  !
0009  %sbttl 'TEST SECTION'
0010
0011  library 'AZTEC0';                              ! AZTEC LIBRARY
0012
0013  require 'BLSMAC.REQ';                          ! DIAGNOSTIC SUPERVISOR LIBRARY
1502
1503  structure                                      ! DEFINE ACCESS ALGORITHM TO
1504      RC25 [O, P, S, E] =                        ! ALLOW FIELD REFERANCES TO
1505           begin                                 ! THE RC25
1506                                                 !
1507
1508           local
1509               RC_REG;
1510
1511           RC_REG = .(RC25 + %upval*0)<0, %bpval, 0>;
1512           RC_REG
1513           end
1514           <P, S, E>;
1515
1516  !<BLF/PAGE>
```

```
;        1517  psect
;        1518      plit = $plit$( global),
;        1519      global = $GLOB$(nowrite, noexecute, global, concatenate),
;        1520      code = AC$CODE;
;        1521
;        1522  own
;        1523      CMDBF1 : block [16, word] field (PACKET_FIELDS),    ! COMMAND BUFFER 1
;        1524      ENDBF1 : block [16, word] field (PACKET_FIELDS),    ! END MESSAGE BUFFER 1
;        1525      RING_B : vector [32, word],                         ! COMMAND BUFFER=16 WORDS
;        1526                                                          ! WITH 16 WORDS BELOW FOR
;        1527                                                          ! END MESSAGES.
;        1528      DATA_PAT1 : vector [3, word] preset (
;        1529            [0]=   %o'111111',                            ! DATA PATTERN 1
;        1530            [1]=   %o'044444',                            ! DATA PATTERN 1
;        1531            [2]=   %o'022222'),                           ! DATA PATTERN 1
;        1532  DATA_PAT2 : vector [3, word] preset (
;        1533            [0]=   %o'177400',                            ! DATA PATTERN 2
;        1534            [1]=   %o'007760',                            ! DATA PATTERN 2
;        1535            [2]=   %o'000377'),                           ! DATA PATTERN 2
;        1536  DATA_PAT3 : vector [3, word] preset (
;        1537            [0]=   %o'155555',                            ! DATA PATTERN 3
;        1538            [1]=   %o'133333',                            ! DATA PATTERN 3
;        1539            [2]=   %o'066666'),                           ! DATA PATTERN 3
;        1540  DATA_PAT4 : vector [3, word] preset (
;        1541            [0]=   %o'000377',                            ! DATA PATTERN 4
;        1542            [1]=   %o'170017',                            ! DATA PATTERN 4
;        1543            [2]=   %o'177400'),                           ! DATA PATTERN 4
;        1544  HOST_BUF : vector [260, word];                          ! HOST BUFFER AREA
;        1545
;        1546  external
;        1547      !
;        1548      ! HARDWARE P TABLE DATA IS STORED HERE
;        1549      !
;        1550      RT_TABLE : ref block [WORD1_IN_RT_TAB, word] field (RT_FIELDS),
;        1551      RC25_ADDR : ref RC25 field (RC_REG),                ! READ REGISTER  ALGORITHM
;        1552      RC25_DATA : block [2, word] field (RC_REG),         ! RCSA DATA
;        1553      UNIT : word,                                        ! UNIT UNDER TEST
;        1554      LOG_UNIT : word,
;        1555      RETRIES : word,
;        1556      NUM_RETRIES : word volatile,
;        1557      SWP_TRACE : word volatile,
;        1558      SWP_RETRIES : word volatile,
;        1559      I_AM_NEX : word volatile,                           ! INTERRUPT FLAG
;        1560      CANCEL_TIMER : word volatile,                       ! INTERRUPT FLAG
;        1561      COM_AREA : blockvector [REC_ALLOCATE + SND_ALLOCATE + HDR_SIZ, 2, word],
;        1562      HEAD_AREA : ref block [4, word] field (HDR_FIELD),
;        1563      RECEIVE_RING : ref blockvector [REC_ALLOCATE, 2, word] field (DSC_FIELD),
;        1564      SEND_RING : ref blockvector [SND_ALLOCATE, 2, word] field (DSC_FIELD),
;        1565      REC_ENVELOPE : blockvector [REC_ALLOCATE, RB_SIZE + 2, word] field (ENV_FIELD),
;        1566      SND_ENVELOPE : blockvector [SND_ALLOCATE, SB_SIZE + 2, word] field (ENV_FIELD),
;        1567      XMT_DATA_BUF : vector [256, word],
;        1568      RCV_DATA_BUF : vector [256, word],
;        1569      RINGBASE,                                           ! RING BASE ADDRESS
;        1570      BUF_DESCRPTR : word volatile,                       ! BUFFER DESCRIPTOR AREA
;        1571      CMD_REF : word volatile,                            ! COMMAND REFERENCE BUFFER
;        1572      CMD_SLOT : word volatile,                           ! COMMAND RING SLOT
;        1573      RES_SLOT : word volatile,                           ! RECEIVE RING SLOT
```

```
;        1574        DM_09 : vector [93, word],          ! DM PROGRAM 09
;        1575        DM_10 : vector [58, word],          ! DM PROGRAM 10
;        1576        DM_11 : vector [100, word],         ! DM PROGRAM 11
;        1577        DM_12 : vector [202, word],         ! DM PROGRAM 12
;        1578    !   DM_13 : vector [110, word],         !       DM PROGRAM 13
;        1579    !   DM_19 : vector [113, word],         !       DM PROGRAM 19
;        1580    !   DM_21 : vector [132, word],         !       DM PROGRAM 21
;        1581    !   DM_26 : vector [200, word],         !       DM PROGRAM 26
;        1582    !   DM_27 : vector [260, word],         !       DM PROGRAM 27
;        1583        BYTE_COUNT : word volatile,         ! BYTE COUNT BUFFER
;        1584        MSGADR : word volatile,             ! ERROR MESSAGE ADDRES
;        1585        VEC_AD : byte volatile,             ! RC25 VECTOR ADDRESS
;        1586        MEM_SIZ : word,                     ! FREE MEMORY SIZE
;        1587        P_MASK : byte volatile,             ! PRINT MASK FOR NUMBER OF AURGMENTS
;        1588        B_MASK : byte volatile,             ! INIT MASK FOR WHAT STEP TO DO
;        1589        DATA1 : word,                       ! DATA FOR STEP 1 WRITE
;        1590        DATA2 : word volatile,              ! DATA FOR STEP 2 WRITE
;        1591        DATA3 : word volatile,              ! DATA FOR STEP 3 WRITE
;        1592        DATA4 : word volatile,              ! DATA FOR STEP 4 WRITE
;        1593        END_LBN : word volatile,            ! ENDING LOGICAL BLOCK #
;        1594        SWP_CONTINUE : word volatile,
;        1595        SWP_MANUAL : word volatile,         ! SOFTWARE P-TAB MANUAL SWITCH
;        1596        MANU_SW : word volatile,            ! MENUAL INTERVENTION SWITCH1
;        1597        SWITCH2 : word volatile,            ! MENUAL INTERVENTION SWITCH2
;        1598        RET_UNIT_FLAG : word volatile,      ! RETURN UNIT STATUS BUFFER
;        1599        P1 : word volatile,                 ! FORMAT ADDRESS FOR ERROR REPOT
;        1600        P2 : word volatile,                 ! FAILING FRU
;        1601        P3 : word volatile,                 ! FAILING REGISTER
;        1602        P4 : word volatile,                 ! DATA FOR ERROR REPORT
;        1603        P5 : word volatile,                 ! DATA FOR ERROR REPORT
;        1604        P6 : word volatile,                 ! DATA FOR ERROR REPORT
;        1605        LBN : word volatile,                ! DATA FOR LBN
;        1606        LBN_ST : word volatile,             ! DATA FOR STARTING LBN
;        1607        LBN_ED : word volatile,             ! DATA FOR ENDING LBN
;        1608        LBN_SZ : word volatile,             ! LBN INCREMENTING SIZE
;        1609        CLK_ADR : word,                     ! LOC. TO RETURN CLOCK ADDR.
;        1610        CLK_CSR : word,                     ! STORE CLOCK STARTING ADDR.
;        1611        CLK_START : word,                   ! THE CLOCK STARTING VALUE
;        1612        TICKS : word volatile,              ! THE # OF CLOCK INT. BUFFER
;        1613        SECONDS : word,                     ! THE NUMBERS OF SECONDS BUFFER
;        1614        MINUTES : word,                     ! THE NUMBERS OF MINUTES BUFFER
;        1615        SWP_START,                          ! STARTING TRACK BUF
;        1616        SWP_END,                            ! ENDING TRACK BUF
;        1617        BUF_LENGTH,                         ! BUFFER LENGTH
;        1618        TEMP,                               ! TEMP. BUFFER
;        1619        FREE_MEM_ADDR,                      ! STARTING FREE MEMORY ADDR.
;        1620        MEM_SIZE,                           ! FREE MEMORY SIZE
;        1621        H_SADD,                             ! FREE HOST MEMORY START AD.
;        1622        H_EADD,                             ! FREE HOST MEMORY END AD.
;        1623        INI_MSG,                            ! INIT ERROR MESSAGE
;        1624        P_VECTOR,                           ! VECTOR BUFFER
;        1625        P_IP_ADDRESS,                       ! RC25 ADDRESS
;        1626        RET_STATUS,                         ! COMMAND STATUS BUFFER
;        1627        ADAPTO,                             ! ADAPTOR FRU MESSAGE
;        1628        TIME,                               ! PLIT LOCATION TO STORE DATA
;        1629        MSG_1,                              ! ERROR MESSAGE 1 IN MOD 1
;        1630        MSG_2,                              ! ERROR MESSAGE IN TEST
```

```
;		MSG_7,			! ERROR MESSAGE IN TEST
;		MSG_8,			! ERROR MESSAGE IN TEST
;		MSG_9,			! ERROR MESSAGE IN TEST
;		MSG_10,			! ERROR MESSAGE IN TEST
;		MSG_11,			! ERROR MESSAGE IN TEST
;		MSG_13,			! ERROR MESSAGE IN TEST
;		MSG_14,			! ERROR MESSAGE IN TEST
;		MSG_17,			! ERROR MESSAGE IN TEST
;		MSG_18,			! ERROR MESSAGE IN TEST
;		MSG_19,			! ERROR MESSAGE IN TEST
;		MSG_20,			! ERROR MESSAGE IN TEST
;		MSG_21,			! ERROR MESSAGE IN TEST
;		MSG_28,			! ERROR MESSAGE IN TEST
;		MSG_29,			! ERROR MESSAGE IN TEST
;		MSG_30,			! ERROR MESSAGE IN TEST
;		QSTT2,			! MESSAGE
;		QST13,			! MESSAGE
;		QST14,			! MESSAGE
;		QST15,			! MESSAGE
;		END_MSG,		! ERROR MESSAGE IN TEST
;		FMT1,			! FORMATTED MESSAGE
;		FMT2,			! FROMATTED MESSAGE
;		FMT3,			! FORMATTED MESSAGE
;		FMT4,			! FORMATTED MESSAGE
;		FMT5,			! FORMATTED MESSAGE
;		FMT6,			! FORMATTED MESSAGE
;		FRU,			! FRU = MESSAGE
;		FMT$A,
;		DBM7,			! TEST HEADER MESSAGES
;		DBM8,
;		DBM9,
;		DBM10,
;		DBM11,
;		DBM12,
;		DBM13,
;		DBM14,
;		DBM15,
;		DBM16,
;		DBM17,
;		DBM18,
;		DBM19,
;		DBM20,
;		DBM21,
;		DBM22,
;		DBM23,
;		DBM24,
;		DBM25,
;		DBM26,
;		DBM27,
;		DBM28,
;		DBM29,
;		DBM30,
;		DBM31,
;		DBM32,
;		DBM36,
;		DBM37,
;		DBM38,
```

```
;          1688      DBM39,
;          1689  ! ERROR MESSAGES
;          1690      CTO_ERR,
;          1691      MSG_STATUS_ERR,
;          1692      AHEAD_MSG,
;          1693      BHEAD_MSG,
;          1694      CHEAD_MSG,
;          1695      DHEAD_MSG,
;          1696      MSG_BUSA_ERR,
;          1697      MSG_ADDR_ERR,
;          1698      MSG_DATA_ERR,
;          1699      MSG_ERR_CONT,
;          1700      MSG_SEEK_ERR,
;          1701      MSG_TK_DSP,
;          1702      MSG_LBN_DSP,
;          1703      MSG_HSWICH_ERR,
;          1704      MSG_SURFACE_ERR,
;          1705      MSG_READ_ERR,
;          1706      MSG_SAC_ERR,
;          1707      MSG_AVE_TIME,
;          1708      MSG_PT_ERR1,
;          1709      MSG_WRP_ERR2,
;          1710      MSG_COM_WPT,
;          1711      AZT_READY_ERR,
;          1712      EXE_SUP_ERR,
;          1713      SND_DATA_ERR,
;          1714      RE_DATA_ERR,
;          1715      BUFF_ERR,
;          1716      DMC_ERR,
;          1717      BRERR,
;          1718      TIP;                                    !TEST IN PROGRESS
;          1719
;          1720  external routine
;          1721      NXMI : novalue,
;          1722      AZT_INIT,
;          1723      AZP_INIT,
;          1724      FIND_CLOCK : novalue,
;          1725      CLOCK_INIT : novalue,
;          1726      RC25$ERR_RPT : novalue,
;          1727      INIT_COM_AREA,
;          1728      SET_INT_VECTOR : novalue,
;          1729      REC_STATUS,
;          1730      EX_SUP_PRG,
;          1731      RANDOM_NUM,
;          1732      REC_DATA,
;          1733      SEND_DATA,
;          1734      SET_CNTLR_CHAR,
;          1735      AVAILABLE,
;          1736      READ_CMD,
;          1737      READ_FILL_RING,
;          1738      ON_LINE,
;          1739      GET_UNIT_STATUS,
;          1740      GET_CMD_SLOT,
;          1741      DECODE,
;          1742      AVERAGE_TIME,
;          1743      EXAM_DATA,                              ! EXAMINE THE FREE MEMORY DATA
;          1744      DM_ADDR_SETUP : novalue,                ! PASS ADDR. TO DM PROGRAM
```

```
;        1745      DATA_XMT_REC : novalue,                    !
;        1746      WRT_PROTECT_TST : novalue,                 ! WRITE PROTECT ROUTINE
;        1747      AZTEC_READY,                               ! GET AZTEC READY
;        1748      DO_RETRIES : novalue;
;        1749
;        1750  !<BLF/PAGE>
```

```
;       1751  !
;       1752  BGNTST;
;       1753
;       1754  !++
;       1755  ! TEST 1: REGISTER EXISTENCE TEST
;       1756  ! DESCRIPTION:
;       1757  !       THIS TEST WILL FIRST CHECK FOR THE EXISTENCE OF THE ADDRESS OF THE IP
;       1758  !       AND SA REGISTERS FOR THE DEVICE UNDER TEST.
;       1759  !       IF THESE MEMORY ADDRESSES ARE NON-EXISTENT, THE ERROR WILL BE
;       1760  !       REPORTED.
;       1761  !       IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM THE
;       1762  !       BEGINNING OF SUB TEST.
;       1763  !--
;       1764
;       1765  local
;       1766      DUMMY;
;       1767
;       1768  if .SWP_TRACE then PRINTF (DBM7);                 ! TEST 1
;       1769
;       1770  BGNSUB;
;       1771  NUM_RETRIES = ZERO;                               ! CLEAR RETRY COUNTER
;       1772
;       1773  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;       1774      begin
;       1775      I_AM_NEX = FALSE;                             ! CLEAR OUT NEX FLAG
;       1776      SETVEC (4, NXMI, PRI07);                      ! SET UP FOR AN NEX TRAP
;       1777
;       1778      if .(.RT_TABLE [RT_IP_ADDRESS] + 2)           ! READ THE SA REGISTER
;       1779      then                                          ! THIS IS SO THAT IF THERE
;       1780          begin                                     ! IS AN NEX THERE WILL BE
;       1781          DUMMY = 1;                                ! A SINGLE OPPERAND INST.
;       1782          end;                                      ! SO THAT IT WILL TRAP
;       1783
;       1784      CLRVEC (4);                                   ! CORRECTLY.
;       1785
;       1786      if .I_AM_NEX eqlu ALL_ONES                    ! SEE IF WE GOT AN NEX
;       1787      then
;       1788          begin                                     ! ADDRESS NOT THERE
;       1789          P_MASK = 1;
;       1790          P1 = FMT1;
;       1791          P2 = ADAPT;
;       1792          P3 = 0;
;       1793          P4 = (.RC25_ADDR) + 2;
;       1794          ERRDF (1, MSG_1, RC25$ERR_RPT);           ! PRINT ERROR MESSAGE
;       1795          CKLOOP;
;       1796          DO_RETRIES ();
;       1797          end;
;       1798
;       1799      if (.NUM_RETRIES eqlu ZERO) then exitloop;
;       1800
;       1801      end;
;       1802
;       1803  NUM_RETRIES = ZERO;                               ! CLEAR RETRY COUNTER
;       1804  ENDSUB;
;       1805  BGNSUB;
;       1806
;       1807  while (.NUM_RETRIES lequ .SWP_RETRIES) do
```

```
;         1808      begin
;         1809      I_AM_NEX = FALSE;                       ! CLEAR TRAP FLAG
;         1810      SETVEC (4, NXMI, PRI07);                ! SET UP TRAP VECTOR IF NEX
;         1811
;         1812      if .(.RT_TABLE [RT_IP_ADDRESS])         ! READ IP REGISTER
;         1813      then
;         1814          begin
;         1815          DUMMY = 1;
;         1816          end;
;         1817
;         1818      CLRVEC (4);                             ! CLEAR THE VRCTOR
;         1819
;         1820      if .I_AM_NEX eqlu ALL_ONES              ! CHECK FOR TRAPS
;         1821      then
;         1822          begin
;         1823          P_MASK = 1;
;         1824          PT = FMT1;
;         1825          P2 = ADAPT;
;         1826          P4 = .RC25_ADDR;
;         1827          ERRDF (2, MSG_2, RC25$ERR_RPT);     ! PRINT OUT ERRO MESSAGE
;         1828          CKLOOP;
;         1829          DO_RETRIES ();
;         1830          end;
;         1831
;         1832      if (.NUM_RETRIES eqlu ZERO) then exitloop;
;         1833
;         1834          end;
;         1835
;         1836  if .I_AM_NEX eqlu ALL_ONES                  ! IF REGISTERS ARE NON-EXISTENT
;         1837  then
;         1838      begin                                   ! THEN DROP THE UNIT FROM TESTING
;         1839      DODU (.LOG_UNIT);
;         1840      DOCLN;
;         1841      end;
;         1842
;         1843  ENDSUB;
;         1844  ENDTST;


                                        .TITLE  ZRCFA3 CZRCFA0 RC25 FR END TEST
                                        .IDENT  /V01.0/


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
```

```
000216  133333                          .WORD   -44445
000220  066666                          .WORD   66666
000222  000377                  DATA.PAT4:
                                        .WORD   377
000224  170017                          .WORD   -7761
000226  177400                          .WORD   -400
000230                          HOST.BUF:
                                        .BLKW   404

                                .GLOBL  RT.TABLE, RC25.ADDR, RC25.DATA
                                .GLOBL  UNIT, LOG.UNIT, RETRIES, NUM.RETRIES
                                .GLOBL  SWP.TRACE, SWP.RETRIES, I.AM.NEX
                                .GLOBL  CANCEL.TIMER, COM.AREA, HEAD.AREA
                                .GLOBL  RECEIVE.RING, SEND.RING, REC.ENVELOPE
                                .GLOBL  SND.ENVELOPE, XMT.DATA.BUF, RCV.DATA.BUF
                                .GLOBL  RINGBASE, BUF.DESCRPTR, CMD.REF
                                .GLOBL  CMD.SLOT, RES.SLOT, DM.09, DM.10
                                .GLOBL  DM.11, DM.12, BYTE.COUNT, MSGADR
                                .GLOBL  VEC.AD, 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, CLK.ADR, CLK.CSR, CLK.START
                                .GLOBL  TICKS, SECONDS, MINUTES, SWP.START
                                .GLOBL  SWP.END, BUF.LENGTH, TEMP, FREE.MEM.ADDR
                                .GLOBL  MEM.SIZE, H.SADD, H.EADD, INI.MSG
                                .GLOBL  P.VECTOR, P.IP.ADDRESS, RET.STATUS
                                .GLOBL  ADAPTO, TIME, MSG.1, MSG.2, MSG.7
                                .GLOBL  MSG.8, MSG.9, MSG.10, MSG.11, MSG.13
                                .GLOBL  MSG.14, MSG.17, MSG.18, MSG.19
                                .GLOBL  MSG.20, MSG.21, MSG.28, MSG.29
                                .GLOBL  MSG.30, QST12, QST13, QST14, QST15
                                .GLOBL  END.MSG, FMT1, FMT2, FMT3, FMT4
                                .GLOBL  FMT5, FMT6, FRU, FMT$A, DBM7, DBM8
                                .GLOBL  DBM9, DBM10, DBM11, DBM12, DBM13
                                .GLOBL  DBM14, DBM15, DBM16, DBM17, DBM18
                                .GLOBL  DBM19, DBM20, DBM21, DBM22, DBM23
                                .GLOBL  DBM24, DBM25, DBM26, DBM27, DBM28
                                .GLOBL  DBM29, DBM30, DBM31, DBM32, DBM36
                                .GLOBL  DBM37, DBM38, DBM39, CTO.ERR, MSG.STATUS.ERR
                                .GLOBL  AHEAD.MSG, BHEAD.MSG, CHEAD.MSG
                                .GLOBL  DHEAD.MSG, MSG.BUSA.ERR, MSG.ADDR.ERR
                                .GLOBL  MSG.DATA.ERR, MSG.ERR.CONT, MSG.SEEK.ERR
                                .GLOBL  MSG.TK.DSP, MSG.LBN.DSP, MSG.HSWICH.ERR
                                .GLOBL  MSG.SURFACE.ERR, MSG.READ.ERR
                                .GLOBL  MSG.SAC.ERR, MSG.AVE.TIME, MSG.PT.ERR1
                                .GLOBL  MSG.WRP.ERR2, MSG.COM.WPT, AZT.READY.ERR
                                .GLOBL  EXE.SUP.ERR, SND.DATA.ERR, RE.DATA.ERR
                                .GLOBL  BUFF.ERR, DMC.ERR, BRERR, TIP
                                .GLOBL  NXMI, AZT.INIT, AZP.INIT, FIND.CLOCK
                                .GLOBL  CLOCK.INIT, RC25$ERR.RPT, INIT.COM.AREA
                                .GLOBL  SET.INT.VECTOR, REC.STATUS, EX.SUP.PRG
                                .GLOBL  RANDOM.NUM, REC.DATA, SEND.DATA
                                .GLOBL  SET.CNTLR.CHAR, AVAILABLE, READ.CMD
                                .GLOBL  READ.FILL.RING, ON.LINE, GET.UNIT.STATUS
```

```
                                .GLOBL  GET.CMD.SLOT, DECODE, AVERAGE.TIME
                                .GLOBL  EXAM.DATA, DM.ADDR.SETUP, DATA.XMT.REC
                                .GLOBL  WRT.PROTECT.TST, AZTEC.READY, DO.RETRIES


                                .SBTTL  $T1 TEST SECTION
000000                          .PSECT  AC$CODE,  RO

000000  010146                  $T1:    MOV     R1,-(SP)                        ;                       1748
000002  032767  000001  000000G         BIT     #1,SWP.TRACE                    ;                       1768
000010  001407                          BEQ     1$
000012  012746  000000G                 MOV     #DBM7,-(SP)
000016  012746  000001                  MOV     #1,-(SP)
000022  010600                          MOV     SP,R0                           ; SP,*
000024  104417                          TRAP    17
000026  022626                          CMP     (SP)+,(SP)+
000030  104402                  1$:     TRAP    2
000032  005067  000000G                 CLR     NUM.RETRIES                     ;                       1771
000036  026767  000000G 000000G  2$:     CMP     NUM.RETRIES,SWP.RETRIES         ;                       1773
000044  101100                          BHI     7$
000046  005067  000000G                 CLR     I.AM.NEX                        ;                       1775
000052  012746  000340                  MOV     #340,-(SP)                      ;                       1776
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,R0                    ;                       1778
000100  032760  000001  000002         BIT     #1,2(R0)
000106  001402                          BEQ     3$
000110  012701  000001                  MOV     #1,R1                           ; *,DUMMY               1781
000114  012700  000004          3$:     MOV     #4,R0                           ;                       1784
000120  104436                          TRAP    36
000122  026727  000000G 177777         CMP     I.AM.NEX,#-1                    ;                       1786
000130  001035                          BNE     5$
000132  112767  000001  000000G         MOVB    #1,P.MASK                       ;                       1789
000140  012767  000000G 000000G         MOV     #FMT1,P1                        ;                       1790
000146  012767  000001  000000G         MOV     #1,P2                           ;                       1791
000154  005067  000000G                 CLR     P3                              ;                       1792
000160  016700  000000G                 MOV     RC25.ADDR,R0                    ;                       1793
000164  062700  000002                  ADD     #2,R0
000170  010067  000000G                 MOV     R0,P4
000174  104455                          TRAP    55                              ;                       1794
000176  000001                          .WORD   1
000200  000000G                         .WORD   MSG.1
000202  000000G                         .WORD   RC25$ERR.RPT
000204  104465                          TRAP    65
000206  006000                          ROR     R0
000210  103003                          BCC     4$
000212  062706  000010                  ADD     #10,SP
000216  000415                          BR      8$
000220  004767  000000G          4$:     JSR     PC,DO.RETRIES                   ;                       1796
000224  005767  000000G          5$:     TST     NUM.RETRIES                     ;                       1799
000230  001003                          BNE     6$
000232  062706  000010                  ADD     #10,SP
000236  000403                          BR      7$
000240  062706  000010          6$:     ADD     #10,SP                          ;                       1774
000244  000674                          BR      2$                              ;                       1773
```

```
000246  005067  000000G                7$:     CLR     NUM.RETRIES                     ;                               1803
000252  104467                          8$:     TRAP    67
000254  006000                                  ROR     R0
000256  103664                                  BLO     1$
000260  104402                          9$:     TRAP    2                               ;                               1804
000262  026767  000000G 000000G         10$:    CMP     NUM.RETRIES,SWP.RETRIES         ;                               1807
000270  101072                                  BHI     15$
000272  005067  000000G                        CLR     I.AM.NEX                        ;                               1809
000276  012746  000340                        MOV     #340,-(SP)                      ;                               1810
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                    ;                               1812
000324  032710  000001                        BIT     #1,(R0)
000330  001402                                BEQ     11$
000332  012701  000001                        MOV     #1,R1                           ; *,DUMMY                       1815
000336  012700  000004                11$:    MOV     #4,R0                           ;                               1818
000342  104436                                TRAP    36
000344  026727  000000G 177777               CMP     I.AM.NEX,#-1                    ;                               1820
000352  001030                                BNE     13$
000354  112767  000001  000000G             MOVB    #1,P.MASK                       ;                               1823
000362  012767  000000G 000000G             MOV     #FMT1,P1                        ;                               1824
000370  012767  000001  000000G             MOV     #1,P2                           ;                               1825
000376  016767  000000G 000000G             MOV     RC25.ADDR,P4                    ;                               1826
000404  104455                                TRAP    55                              ;                               1827
000406  000002                                .WORD   2
000410  000000G                               .WORD   MSG.2
000412  000000G                               .WORD   RC25$ERR.RPT
000414  104465                                TRAP    65
000416  006000                                ROR     R0
000420  103003                                BCC     12$
000422  062706  000010                        ADD     #10,SP
000426  000423                                BR      16$
000430  004767  000000G                12$:    JSR     PC,DO.RETRIES                   ;                               1829
000434  005767  000000G                13$:    TST     NUM.RETRIES                     ;                               1832
000440  001003                                BNE     14$
000442  062706  000010                        ADD     #10,SP
000446  000403                                BR      15$
000450  062706  000010                14$:    ADD     #10,SP                          ;                               1808
000454  000702                                BR      10$                             ;                               1807
000456  026727  000000G 177777        15$:    CMP     I.AM.NEX,#-1                    ;                               1836
000464  001004                                BNE     16$
000466  016700  000000G                        MOV     LOG.UNIT,R0                     ;                               1839
000472  104451                                TRAP    51
000474  104444                                TRAP    44
000476  104467                          16$:    TRAP    67                              ;                               1841
000500  006000                                ROR     R0
000502  103666                                BLO     9$
000504  012601                                MOV     (SP)+,R1                        ;                               1748
000506  000207                                RTS     PC

; Routine Size:  164 words,     Routine Base:  AC$CODE + 0000
; Maximum stack depth per invocation:  7 words
```

```
                                        .SBTTL  T1 TEST SECTION
000000  004767  177264                 T1::
000000                                   1$:    JSR     PC,$T1                  ;                       1843
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

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

;         1845  !<BLF/PAGE>
```

```
;          1846  !
;          1847  BGNTST;
;          1848
;          1849  !++
;          1850  ! TEST 2: INITALZATION TEST (POWER UP DIAGNOSTICS)
;          1851  ! DESCRIPTION:
;          1852  !       THIS TEST INIT' THE AZTEC AND RUNS THE POWER UP DIAGNOSTICS BY
;          1853  !       WRITING WITH STEP1 DATA. THEN IT WILL CHECK FOR ERRORS AND
;          1854  !       REPORT IF AZTEC DOES NOT COME UPTO STEP2 READ
;          1855  !
;          1856  !
;          1857  NUM_RETRIES = ZERO;                          ! CLEAR RETRY COUNTER
;          1858
;          1859  if .SWP_TRACE then PRINTF (DBM8);            ! TEST 2
;          1860
;          1861  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;          1862      begin
;          1863      ! STEP 1 WRITE WITH STEP 2 READ
;          1864      B_MASK = 1;                               ! SELECT B_MASK FOR STEP 1 WRITE
;          1865      DATA1 = %o'137600' + .RT_TABLE [RT_VECTOR]/4;       ! SELECT STEP1 WRITE DATA WITH
;          1866                                                ! MAX RING SIZES ,IE AND VECTOR
;          1867                                                ! ADDRESS
;          1868
;          1869      if AZT_INIT ()                            ! PORT SHOULD NOW GET TO STEP2
;          1870                                                ! AFTER FINISHING INTEGRITY CHECK
;          1871                                                ! DIAG. IF NOT REPORT ERROR
;          1872      then
;          1873          begin
;          1874          ERRDF (3, MSG_14, RC25$ERR_RPT);
;          1875
;          1876          if .RET_STATUS then DECODE ();        ! DECODE STATUS
;          1877
;          1878          CKLOOP;
;          1879          RETRIES = TRUE;
;          1880          end;
;          1881
;          1882      if (.RETRIES) then DO_RETRIES ();         ! RETRY IF ERROR
;          1883
;          1884      if (.NUM_RETRIES eqlu ZERO) then exitloop;
;          1885
;          1886      end;
;          1887
;          1888  return;
;          1889  ENDTST;


                                        .SBTTL  $T2 TEST SECTION
000000  005067  000000G                $T2:    CLR     NUM.RETRIES                     ;                       1857
000004  032767  000001  000000G               BIT     #1,SWP.TRACE                    ;                       1859
000012  001407                                BEQ     1$
000014  012746  000000G                       MOV     #DBM8,-(SP)
000020  012746  000001                        MOV     #1,-(SP)
000024  010600                                MOV     SP,R0                           ; SP,*
000026  104417                                TRAP    17
000030  022626                                CMP     (SP)+,(SP)+
000032  026767  000000G 000000G         1$:    CMP     NUM.RETRIES,SWP.RETRIES         ;                       1861
000040  101060                                BHI     6$
```

```
000042  112767  000001  000000G             MOVB    #1,B.MASK                       ;                       1864
000050  016700  000000G                     MOV     RT.TABLE,R0                     ;                       1865
000054  016046  000002                      MOV     2(R0),-(SP)
000060  012746  000004                      MOV     #4,-(SP)
000064  004767  000000G                     JSR     PC,BL$DIV
000070  010067  000000G                     MOV     R0,DATA1
000074  162767  040200  000000G             SUB     #40200,DATA1
000102  004767  000000G                     JSR     PC,AZT.INIT                     ;                       1869
000106  006000                              ROR     R0
000110  103022                              BCC     4$
000112  104455                              TRAP    55                              ;                       1874
000114  000003                              .WORD   3
000116  000000G                             .WORD   MSG.14
000120  000000G                             .WORD   RC25$ERR.RPT
000122  032767  000001  000000G             BIT     #1,RET.STATUS                   ;                       1876
000130  001402                              BEQ     2$
000132  004767  000000G                     JSR     PC,DECODE
000136  104465                      2$:     TRAP    65
000140  006000                              ROR     R0
000142  103002                              BCC     3$
000144  022626                              CMP     (SP)+,(SP)+
000146  000207                              RTS     PC
000150  012767  000001  000000G     3$:     MOV     #1,RETRIES                      ;                       1879
000156  032767  000001  000000G     4$:     BIT     #1,RETRIES                      ;                       1882
000164  001402                              BEQ     5$
000166  004767  000000G                     JSR     PC,DO.RETRIES
000172  005767  000000G             5$:     TST     NUM.RETRIES                     ;                       1884
000176  001002                              BNE     7$
000200  022626                              CMP     (SP)+,(SP)+
000202  000207                      6$:     RTS     PC
000204  022626                      7$:     CMP     (SP)+,(SP)+                     ;                       1862
000206  000711                              BR      1$                              ;                       1861

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



                                            .SBTTL  T2 TEST SECTION
000000  004767  177564              T2::
000000                              1$:     JSR     PC,$T2                          ;                       1888
000004  104466                              TRAP    66
000006  006000                              ROR     R0
000010  103773                              BLO     1$
000012  000207                              RTS     PC

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

;       1890  !<BLF/PAGE>
```

```
;        1891  BGNTST;
;        1892
;        1893  !++
;        1894  ! TEST #3 - DIAGNOSTIC WRAP TEST
;        1895  !
;        1896  ! DESCRIPTION:
;        1897  !
;        1898  !       THE AZTEC WILL BE INITIALIZED IN DIAGNOSTIC WRAP MODE AND A ONE BIT
;        1899  !       AND ALSO ZERO BIT FLOATED THROUGH THE SA REGISTER TO SEE THAT IT
;        1900  !       ECHOES PROPERLY.
;        1901  !
;        1902  !       A FAILURE TO ECHO WHAT WAS WRITTEN WILL RESULT IN A CALLOUT TO THE
;        1903  !       ADAPTER CARD FRU.
;        1904  !
;        1905  !       IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, THE PROGRAM WILL LOOP ON
;        1906  !       THE FAILING WRITE AND READ.
;        1907  !
;        1908  !--
;        1909
;        1910  local
;        1911      TST_PAT;
;        1912
;        1913  if .SWP_TRACE then PRINTF (DBM10);                ! TEST 3
;        1914
;        1915  NUM_RETRIES = ZERO;
;        1916
;        1917  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;        1918      begin
;        1919      TIP = 4;
;        1920  !
;        1921  ! STEP1 WRITE
;        1922  !
;        1923      B_MASK = 0;                                   ! MASK FOR STEP1 READ
;        1924      DATA1 = %o'140000';                           ! STEP1 WRITE WITH WRAP MODE BIT SET
;        1925      DATA2 = %o'10';                               ! TIME OUT COUNTER
;        1926      DATA3 = ZERO;                                 ! TEMP STORAGE FOR RCSA DATA
;        1927
;        1928      if AZT_INIT ()                                ! CALL STEP 1 ROUTINE
;        1929      then
;        1930          begin
;        1931          ERRDF (4, MSG_14, RC25$ERR_RPT);                ! PRINT OUT ERROR REPORT
;        1932          CKLOOP;
;        1933          RETRIES = TRUE;
;        1934          end
;        1935      else
;        1936          begin
;        1937          WRT_RC25 (RCSA, .DATA1);                  ! DO STEP1 WRITE WITH DWM.
;        1938
;        1939          while ((.DATA3 nequ .DATA1) and (.DATA2 nequ ZERO)) do
;        1940              begin
;        1941              DELAY (333);
;        1942              DATA2 = .DATA2 - 1;
;        1943              DATA3 = .RC25_ADDR [RCSA, RC_ALL];  !
;        1944              end;
;        1945
;        1946          TST_PAT = %o'000001';                     ! START TEST PATTERN TO
;        1947
```

```
;        1948          incru FLOAT from 0 to 15 do             ! NOW FLOAT TEST PAT
;        1949              begin                               !
;        1950
;        1951              incru COUNT from 0 to 1 do          ! FLOAT ZEROES AND THEN ONES
;        1952                  begin
;        1953
;        1954                  if .COUNT eqlu 1 then TST_PAT = not .TST_PAT;
;        1955
;        1956                  BGNSUB;
;        1957                  WRT_RC25 (RCSA, .TST_PAT);      ! WRITE TEST PATTERN TO SA
;        1958                  DELAY (10);                     ! WAIT FOR IT TO ECHO.
;        1959                  RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];   ! GET RCSA DATA
;        1960
;        1961                  if .RC25_DATA [RCSA, RC_ALL] nequ .TST_PAT      ! TEST SA FOR TEST PATTERN
;        1962                  then                            ! IF NOT EQU THEN
;        1963                      begin                       ! PRINT OUT ERROR REPORT
;        1964                      P_MASK = 2;
;        1965                      P1 = FMT2;                  ! MESSAGE ADDRESS
;        1966                      P2 = ADAPT;                 ! FAILING FRU
;        1967                      P6 = (.RC25_ADDR) + 2;      ! FAILING ADDRESS
;        1968                      P4 = .TST_PAT;              ! GOOD DATA
;        1969                      P5 = .RC25_DATA [RCSA, RC_ALL];     ! BAD DATA
;        1970                      ERRDF (5, MSG_7, RC25$ERR_RPT);     !
;        1971                      CKLOOP;
;        1972                      RETRIES = TRUE;
;        1973                      end;
;        1974
;        1975                  ENDSUB;
;        1976                  end;
;        1977
;        1978              TST_PAT = not .TST_PAT;
;        1979              TST_PAT = .TST_PAT^1;               ! SHIFT THE BIT DOWN 1
;        1980              end;
;        1981
;        1982          end;
;        1983
;        1984      if (.RETRIES) then DO_RETRIES ();           ! DO RETRIES IF IN ERROR
;        1985
;        1986      if (.NUM_RETRIES eqlu ZERO) then exitloop;
;        1987
;        1988      end;
;        1989
;        1990  WRT_RC25 (RCIP, ALL_ONES);                      !REINITIALIZE THE PORT
;        1991  ENDTST;


                                        .GLOBL  L$DLY


                                        .SBTTL  $T3 TEST SECTION
000000  004167  000000G          $T3:  JSR     R1,$SAVE4                 ;                 1889
000004  162706  000006                  SUB     #6,SP
000010  032767  000001  000000G        BIT     #1,SWP.TRACE              ;                 1913
000016  001407                          BEQ     1$
000020  012746  000000G                 MOV     #DBM10,-(SP)
000024  012746  000001                  MOV     #1,-(SP)
000030  010600                          MOV     SP,R0                     ; SP,*
```

```
000032  104417                          TRAP    17
000034  022626                          CMP     (SP)+,(SP)+
000036  005067  000000G                1$:     CLR     NUM.RETRIES                     ;                               1915
000042  026767  000000G 000000G        2$:     CMP     NUM.RETRIES,SWP.RETRIES         ;                               1917
000050  101402                          BLOS    3$
000052  000167  000466                  JMP     22$
000056  012767  000004  000000G        3$:     MOV     #4,TIP                          ;                               1919
000064  105067  000000G                CLRB    B.MASK                          ;                               1923
000070  012767  140000  000000G        MOV     #-40000,DATA1                   ;                               1924
000076  012767  000010  000000G        MOV     #10,DATA2                       ;                               1925
000104  005067  000000G                CLR     DATA3                           ;                               1926
000110  004767  000000G                JSR     PC,AZT.INIT                     ;                               1928
000114  006000                          ROR     R0
000116  103015                          BCC     5$
000120  104455                          TRAP    55                              ;                               1931
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  000412                  JMP     23$
000142  012767  000001  000000G        4$:     MOV     #1,RETRIES                      ;                               1933
000150  000562                          BR      20$                             ;                               1928
000152  016701  000000G                5$:     MOV     DATA1,R1                        ; *,RCM.REG                     1937
000156  016700  000000G                MOV     RC25.ADDR,R0
000162  010160  000002                  MOV     R1,2(R0)                        ; RCM.REG,*
000166  026767  000000G 000000G        6$:     CMP     DATA3,DATA1                     ;                               1939
000174  001432                          BEQ     11$
000176  005767  000000G                TST     DATA2
000202  001427                          BEQ     11$
000204  012701  000515                  MOV     #515,R1                         ; *,$$TMP2                      1941
000210  001411                  7$:     BEQ     10$
000212  016700  000000G                MOV     L$DLY,R0                        ; *,$$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                              ; $$TMP2
000232  000766                          BR      7$
000234  005367  000000G          10$:    DEC     DATA2                           ;                               1942
000240  016700  000000G                MOV     RC25.ADDR,R0                    ;                               1943
000244  016066  000002  000002        MOV     2(R0),2(SP)                     ; *,RC.REG
000252  016667  000002  000000G        MOV     2(SP),DATA3                     ; RC.REG,*
000260  000742                          BR      6$                              ;                               1939
000262  012702  000001          11$:    MOV     #1,R2                           ; *,TST.PAT                     1946
000266  005004                          CLR     R4                              ; FLOAT                         1948
000270  005003                  12$:    CLR     R3                              ; COUNT                         1951
000272  022727  000000  000001        CMP     #0,#1                           ;                               1954
000300  001001                  13$:    BNE     14$
000302  005102                          COM     R2                              ; TST.PAT
000304  104402                  14$:    TRAP    2
000306  010201                          MOV     R2,R1                           ; TST.PAT,RCM.REG               1957
000310  016700  000000G                MOV     RC25.ADDR,R0
000314  010160  000002                  MOV     R1,2(R0)                        ; RCM.REG,*
000320  012701  000012                  MOV     #12,R1                          ; *,$$TMP2                      1958
000324  001411                  15$:    BEQ     18$
```

```
000326  016700  000000G                        MOV     L$DLY,R0                ; *,$$TMP1
000332  001404                                 BEQ     17$
000334  005066  000004              16$:      CLR     4(SP)                   ; $$TMP
000340  005300                                 DEC     R0                      ; $$TMP1
000342  001374                                 BNE     16$
000344  005301                      17$:      DEC     R1                      ; $$TMP2
000346  000766                                 BR      15$
000350  016700  000000G               18$:      MOV     RC25.ADDR,R0            ;                                       1959
000354  016016  000002                         MOV     2(R0),(SP)              ; *,RC.REG
000360  011667  000002G                        MOV     (SP),RC25.DATA+2        ; RC.REG,*
000364  021602                                 CMP     (SP),R2                 ; RC25.DATA+2,TST.PAT                   1961
000366  001436                                 BEQ     19$
000370  112767  000002  000000G               MOVB    #2,P.MASK               ;                                       1964
000376  012767  000000G 000000G               MOV     #FMT2,P1                ;                                       1965
000404  012767  000001  000000G               MOV     #1,P2                   ;                                       1966
000412  016700  000000G                        MOV     RC25.ADDR,R0            ;                                       1967
000416  062700  000002                         ADD     #2,R0
000422  010067  000000G                        MOV     R0,P6
000426  010267  000000G                        MOV     R2,P4                   ; TST.PAT,*                             1968
000432  016767  000002G 000000G               MOV     RC25.DATA+2,P5          ;                                       1969
000440  104455                                 TRAP    55                      ;                                       1970
000442  000005                                 .WORD   5
000444  000000G                                .WORD   MSG.7
000446  000000G                                .WORD   RC25$ERR.RPT
000450  104465                                 TRAP    65
000452  006000                                 ROR     R0
000454  103403                                 BLO     19$
000456  012767  000001  000000G               MOV     #1,RETRIES              ;                                       1972
000464  104467                      19$:      TRAP    67                      ;                                       1973
000466  006000                                 ROR     R0
000470  103705                                 BLO     14$
000472  005203                                 INC     R3                      ; COUNT                                 1951
000474  020327  000001                         CMP     R3,#1                   ; COUNT,*
000500  101677                                 BLOS    13$
000502  005102                                 COM     R2                      ; TST.PAT                               1978
000504  006302                                 ASL     R2                      ; TST.PAT                               1979
000506  005204                                 INC     R4                      ; FLOAT                                 1948
000510  020427  000017                         CMP     R4,#17                  ; FLOAT,*
000514  101665                                 BLOS    12$
000516  032767  000001  000000G       20$:      BIT     #1,RETRIES              ;                                       1984
000524  001402                                 BEQ     21$
000526  004767  000000G                        JSR     PC,DO.RETRIES
000532  005767  000000G               21$:      TST     NUM.RETRIES             ;                                       1986
000536  001402                                 BEQ     22$
000540  000167  177276                         JMP     2$
000544  012700  177777               22$:      MOV     #-1,R0                  ; *,RCM.REG                             1990
000550  010077  000000G                        MOV     R0,@RC25.ADDR           ; RCM.REG,*
000554  062706  000006               23$:      ADD     #6,SP                   ;                                       1889
000560  000207                                 RTS     PC

; Routine Size:  185 words,    Routine Base:  AC$CODE + 0750
; Maximum stack depth per invocation:  12 words


                                .SBTTL  T3 TEST SECTION
```

```
000000  004767  177212                  T3::
000000                                  1$:     JSR     PC,$T3                  ;                       1990
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

; Routine Size:  6 words,      Routine Base:  AC$CODE + 1532
; Maximum stack depth per invocation:  2 words


;         1992  !<BLF/PAGE>
```

```
;       1993  !
;       1994  BGNTST;
;       1995
;       1996  !++
;       1997  ! TEST #4 - VECTOR AND BR LEVEL TEST
;       1998  !
;       1999  ! DESCRIPTION:
;       2000  !
;       2001  !       THE INIT SEQUENCE WILL BE STARTED WITH THE INTERRUPT ENABLE BIT SET TO
;       2002  !       VERIFY THE AZTEC'S VECTOR AND BR LEVEL.
;       2003  !
;       2004  !       THIS TEST ASSUMES THE VECTOR GIVEN BY THE OPERATOR IS CORRECT.
;       2005  !
;       2006  !       THE PRIORITY LEVEL OF THE INTERRUPT REQUEST WILL  BE VERIFIED.
;       2007  !
;       2008  !       FAILURE OF THE AZTEC TO VECTOR PROPERLY WILL NECESSITATE THAT THIS
;       2009  !       PROGRAM BE RESTARTED.  A COMPLETED INTERRUPT AT THE WRONG BR LEVEL
;       2010  !       WILL BE REPORTED.
;       2011  !
;       2012  !       LOOP ON ERROR WILL RESTART THIS TEST IF THE ERROR IS RECOVERABLE.
;       2013  !
;       2014  !
;       2015  !--
;       2016
;       2017  !
;       2018  NUM_RETRIES = ZERO;
;       2019
;       2020  if .SWP_TRACE then PRINTF (DBM11);                ! TEST 4
;       2021
;       2022  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;       2023      begin
;       2024      TIP = 5;
;       2025      TEMP = PRI07;                                 ! START WITH HIGHEST PRIORITY
;       2026      I_AM_NEX = FALSE;                             ! CLEAR INTERUPT FLAG
;       2027      B_MASK = 0;                                   ! STEP 1 READ MASK
;       2028      DATA1 = %o'104600' + .RT_TABLE [RT_VECTOR]/4; !       !INTERRUPT ENABLE BIT SET
;       2029      SETPRI (.TEMP);                               ! SET HOST PRIORITY
;       2030
;       2031      if AZT_INIT ()                                ! BRING UP TO STEP 1 READ
;       2032                                                    ! AND GET STATUS
;       2033      then                                          ! IF ERROR
;       2034          begin                                     ! THEN
;       2035          ERRDF (6, MSG_14, RC25$ERR_RPT);          ! REPORT IT
;       2036
;       2037          if .RET_STATUS then DECODE ();            ! DECODE STATUS
;       2038
;       2039          CKLOOP;
;       2040          RETRIES = TRUE;
;       2041          end
;       2042      else
;       2043          begin
;       2044          WRT_RC25 (RCSA, .DATA1);                  ! WRITE STEP 1 DATA
;       2045          DELAY (1500);                             ! WAIT FOR INTERRUPT
;       2046
;       2047          while (.TEMP gequ %o'140') do
;       2048              begin
;       2049
```

```
;        2050                if .I_AM_NEX eqlu ALL_ONES then exitloop;    !IF INTERRUPT DID NOT
;        2051
;        2052                TEMP = .TEMP - %o'40';                        ! NOT OCCUR
;        2053                SETPRI (.TEMP);                               ! LOWER CPU PRIORITY
;        2054                RETRIES = TRUE;
;        2055                end;
;        2056
;        2057            end;
;        2058
;        2059        if .I_AM_NEX eqlu ALL_ONES                            ! IF INTERRUPT OCCURED
;        2060        then
;        2061            begin                                             !
;        2062            TIP = .TEMP^-5 + 1;                               ! GET PRIORITY
;        2063            SETPRI (PRI00);                                   ! SET HOST PRIORITY TO 0
;        2064            SETVEC (.RT_TABLE [RT_VECTOR], NXMI, .TIP);           ! SET UP SERVICE ROUTINE.
;        2065            PRINTF (INI_MSG, .RT_TABLE [RT_VECTOR], .TIP);
;        2066
;        2067            if .TIP nequ .RT_TABLE [RT_BR_LEVEL] then PRINTF (BRERR);        ! IF RECEIVED BR IS NOT THE
;        2068
;        2069                                                              ! SAME AS TYPED REPORT ERROR
;        2070            RETRIES = FALSE;
;        2071            end
;        2072        else
;        2073            begin
;        2074            RETRIES = TRUE;
;        2075            ERRDF (7, END_MSG, 0);                            ! ERROR
;        2076            CKLOOP;
;        2077            end;
;        2078
;        2079        if .RETRIES then DO_RETRIES ();
;        2080
;        2081        if (.NUM_RETRIES eqlu ZERO) then exitloop;
;        2082
;        2083        end;
;        2084
;        2085    ENDTST;


                                                   .SBTTL  $T4 TEST SECTION
000000  010146                            $T4:     MOV     R1,-(SP)                        ;                1991
000002  005746                                     TST     -(SP)
000004  005067  000000G                            CLR     NUM.RETRIES                     ;                2018
000010  032767  000001  000000G                    BIT     #1,SWP.TRACE                    ;                2020
000016  001407                                     BEQ     1$
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$:      CMP     NUM.RETRIES,SWP.RETRIES         ;                2022
000044  101402                                     BLOS    2$
000046  000167  000520                             JMP     16$
000052  012767  000005  000000G           2$:      MOV     #5,TIP                          ;                2024
000060  012767  000340  000000G                    MOV     #340,TEMP                       ;                2025
000066  005067  000000G                            CLR     I.AM.NEX                        ;                2026
000072  105067  000000G                            CLRB    B.MASK                          ;                2027
000076  016700  000000G                            MOV     RT.TABLE,R0                     ;                2028
```

```
000102  016046  000002                        MOV     2(R0),-(SP)
000106  012746  000004                        MOV     #4,-(SP)
000112  004767  000000G                       JSR     PC,BL$DIV
000116  010067  000000G                       MOV     R0,DATA1
000122  162767  073200  000000G               SUB     #73200,DATA1
000130  016700  000000G                       MOV     TEMP,R0                 ;                               2029
000134  104441                                TRAP    41
000136  004767  000000G                       JSR     PC,AZT.INIT             ;                               2031
000142  006000                                ROR     R0
000144  103023                                BCC     5$
000146  104455                                TRAP    55                      ;                               2035
000150  000006                                .WORD   6
000152  000000G                               .WORD   MSG.14
000154  000000G                               .WORD   RC25$ERR.RPT
000156  032767  000001  000000G               BIT     #1,RET.STATUS           ;                               2037
000164  001402                                BEQ     3$
000166  004767  000000G                       JSR     PC,DECODE
000172  104465                        3$:     TRAP    65
000174  006000                                ROR     R0
000176  103002                                BCC     4$
000200  022626                                CMP     (SP)+,(SP)+
000202  000573                                BR      16$
000204  012767  000001  000000G       4$:     MOV     #1,RETRIES              ;                               2040
000212  000444                                BR      10$                     ;                               2031
000214  016701  000000G               5$:     MOV     DATA1,R1                ; *,RCM.REG                     2044
000220  016700  000000G                       MOV     RC25.ADDR,R0
000224  010160  000002                        MOV     R1,2(R0)                ; RCM.REG,*
000230  012701  002734                        MOV     #2734,R1                ; *,$$TMP2                      2045
000234  001411                        6$:     BEQ     9$
000236  016700  000000G                       MOV     L$DLY,R0                ; *,$$TMP1
000242  001404                                BEQ     8$
000244  005066  000004                7$:     CLR     4(SP)                   ; $$TMP
000250  005300                                DEC     R0                      ; $$TMP1
000252  001374                                BNE     7$
000254  005301                        8$:     DEC     R1                      ; $$TMP2
000256  000766                                BR      6$
000260  026727  000000G 000140        9$:     CMP     TEMP,#140               ;                               2047
000266  103416                                BLO     10$
000270  026727  000000G 177777                CMP     I.AM.NEX,#-1            ;                               2050
000276  001412                                BEQ     10$
000300  162767  000040  000000G               SUB     #40,TEMP                ;                               2052
000306  016700  000000G                       MOV     TEMP,R0                 ;                               2053
000312  104441                                TRAP    41
000314  012767  000001  000000G               MOV     #1,RETRIES              ;                               2054
000322  000756                                BR      9$                      ;                               2047
000324  026727  000000G 177777        10$:    CMP     I.AM.NEX,#-1            ;                               2059
000332  001065                                BNE     12$
000334  016716  000000G                       MOV     TEMP,(SP)               ;                               2062
000340  012746  177773                        MOV     #-5,-(SP)
000344  004767  000000G                       JSR     PC,BL$SHF
000350  010067  000000G                       MOV     R0,TIP
000354  005267  000000G                       INC     TIP
000360  005000                                CLR     R0                      ;                               2063
000362  104441                                TRAP    41
000364  016716  000000G                       MOV     TIP,(SP)                ;                               2064
000370  012746  000000G                       MOV     #NXMI,-(SP)
000374  016700  000000G                       MOV     RT.TABLE,R0
```

```
000400  016046  000002                          MOV     2(R0),-(SP)
000404  012746  000003                          MOV     #3,-(SP)
000410  104437                                  TRAP    37
000412  016716  000000G                         MOV     TIP,(SP)                ;                               2065
000416  016700  000000G                         MOV     RT.TABLE,R0
000422  016046  000002                          MOV     2(R0),-(SP)
000426  012746  000000G                         MOV     #INI.MSG,-(SP)
000432  012746  000003                          MOV     #3,-(SP)
000436  010600                                  MOV     SP,R0                   ; SP,*
000440  104417                                  TRAP    17
000442  016700  000000G                         MOV     RT.TABLE,R0             ;                               2067
000446  026760  000000G 000004                  CMP     TIP,4(R0)
000454  001407                                  BEQ     11$
000456  012716  000000G                         MOV     #BRERR,(SP)
000462  012746  000001                          MOV     #1,-(SP)
000466  010600                                  MOV     SP,R0                   ; SP,*
000470  104417                                  TRAP    17
000472  005726                                  TST     (SP)+
000474  005067  000000G                 11$:    CLR     RETRIES                 ;                               2070
000500  062706  000016                          ADD     #16,SP                  ;                               2061
000504  000414                                  BR      13$                     ;                               2059
000506  012767  000001  000000G         12$:    MOV     #1,RETRIES              ;                               2074
000514  104455                                  TRAP    55                      ;                               2075
000516  000007                                  .WORD   7
000520  000000G                                 .WORD   END.MSG
000522  000000                                  .WORD   0
000524  104465                                  TRAP    65
000526  006000                                  ROR     R0
000530  103002                                  BCC     13$
000532  022626                                  CMP     (SP)+,(SP)+
000534  000416                                  BR      16$
000536  032767  000001  000000G         13$:    BIT     #1,RETRIES                      ;                       2079
000544  001402                                  BEQ     14$
000546  004767  000000G                         JSR     PC,DO.RETRIES
000552  005767  000000G                 14$:    TST     NUM.RETRIES             ;                               2081
000556  001002                                  BNE     15$
000560  022626                                  CMP     (SP)+,(SP)+
000562  000403                                  BR      16$
000564  022626                          15$:    CMP     (SP)+,(SP)+             ;                               2023
000566  000167  177244                          JMP     1$                      ;                               2022
000572  005726                          16$:    TST     (SP)+                   ;                               1991
000574  012601                                  MOV     (SP)+,R1
000576  000207                                  RTS     PC

; Routine Size:  192 words,     Routine Base:  AC$CODE + 1546
; Maximum stack depth per invocation:  14 words



                                                .SBTTL  T4 TEST SECTION
000000  004767  177174                  T4::
000000                                  1$:     JSR     PC,$T4                  ;                               2083
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC
```

```
; Routine Size:  6 words,      Routine Base:  AC$CODE + 2346
; Maximum stack depth per invocation:  2 words

;        2086  !<BLF/PAGE>
```

```
;         2087  BGNTST;
;         2088
;         2089  !++
;         2090  !
;         2091  ! TEST 5: STEP 1 -3 INITIALZATION TEST
;         2092  !
;         2093  ! DESCRIPTION:
;         2094  !
;         2095  ! THIS TEST WILL CHECK FOR INFORMATIONS ECHOED FROM PORT AT
;         2096  ! EACH STEP READ COMING UPTO THAT STEP FROM SCRATCH. IF THERE WAS
;         2097  ! AN ERROR REPORTED OR ECHOED INFORMATIONS WERE INCORRECT
;         2098  ! THE SAME WILL BE REPORTED.
;         2099  ! LOOP ON ERROR WILL BE FROM THE BEGINNING OF SUB TEST.
;         2100  !--
;         2101
;         2102  NUM_RETRIES = ZERO;                          ! CLEAR RETRY COUNTER
;         2103
;         2104  if .SWP_TRACE then PRINTF (DBM9);            ! TEST 5
;         2105
;         2106  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;         2107      begin
;         2108  !
;         2109  !       STEP1 READ
;         2110  !
;         2111      BGNSUB;
;         2112  !
;         2113  !
;         2114  !check if using Q bus and flag
;         2115  !     TEMP = READBUS ();
;         2116  !
;         2117  ! STEP 1 READ
;         2118  !
;         2119      B_MASK = 0;                               ! START PORT INIT WITH MASK = 0
;         2120
;         2121      if AZT_INIT ()                            ! BRING UP TO STEP 1 READ
;         2122                                                ! AND GET STATUS
;         2123      then                                      ! IF ERROR
;         2124          begin                                 ! THEN
;         2125          ERRDF (8, MSG_14, RC25$ERR_RPT);      ! REPORT IT
;         2126
;         2127          if .RET_STATUS then DECODE ();        ! DECODE STATUS
;         2128
;         2129          CKLOOP;
;         2130          RETRIES = TRUE;
;         2131          end;
;         2132
;         2133  ! CHECK FOR CONTROLLER DEPENDENT INFORMATION FROM RCSA AT STEP 1 READ
;         2134
;         2135      if ((.RC25_DATA [RCSA, RCSA_NV])          ! CHECK THAT THE NV BIT DID
;         2136                                                ! NOT SET.
;         2137          or not (.RC25_DATA [RCSA, RCSA_DI]))  ! CHECK IF DI BIT SET
;         2138  !       or (.TEMP) and not (.RC25_DATA [RCSA, RCSA_QB]) ! CHECK THE QB BIT
;         2139  !       or not (.TEMP) and (.RC25_DATA [RCSA, RCSA_QB]))
;         2140      then                                      ! IF NOT SET
;         2141          begin                                 ! THEN
;         2142          P_MASK = 2;
;         2143          PT = FMT3;
```

```
2144            P2 = ADAPT;
2145            P4 = (.RC25_ADDR) + 2;
2146            P5 = .RC25_DATA [RCSA, RC_ALL];
2147            P6 = %o'01';                            ! MASK = STEP 1
2148            ERRDF (9, MSG_14, RC25$ERR_RPT);        ! REPORT ERROR
2149            CKLOOP:
2150            RETRIES = TRUE;
2151            end;
2152
2153        TEMP = .RC25_DATA [RCSA, RC_ALL];
2154        TEMP = .TEMP<6, 5>;                         ! PORT SPECIFIC INFO
2155        PRINTF (FMT5, .TEMP);                       ! GIVE IT TO OPERATOR
2156        ENDSUB;
2157    !
2158    ! STEP1 WRITE WITH STEP 2 READ
2159    !
2160        BGNSUB;
2161        B_MASK = 1;
2162        DATA1 = %o'137600' + .RT_TABLE [RT_VECTOR]/4;       ! STEP1 WRITE DATA FOR MAX
2163                                                    ! RING LENGTHS,IE AND
2164                                                    ! VECTOR ADDRESS
2165
2166        if AZT_INIT ()                              ! DO INIT AND IF ERROR
2167        then
2168            begin
2169            ERRDF (10, MSG_14, RC25$ERR_RPT);       ! REPORT ERROR
2170
2171            if .RET_STATUS then DECODE ();          ! DECODE STATUS
2172
2173            CKLOOP;
2174            RETRIES = TRUE;
2175            end
2176        else
2177            begin
2178    ! CHECK FOR ECHOED INFORMATIONS AT STEP2 READ
2179            TEMP = .DATA1<8, 8>;                    ! SAVE EXPECTED DATA
2180
2181            if (.RC25_DATA [RCSA, RCSA_7_0] nequ .TEMP)
2182            then                                    ! IF ECHOED INFO DOES NOT
2183                begin                               ! MATCH REPORT ERROR
2184                P_MASK = 2;
2185                PT = FMT2;
2186                P2 = ADAPT;
2187                P4 = .TEMP;
2188                P5 = .RC25_DATA [RCSA, RCSA_7_0];
2189                P6 = .RT_TABLE [RT_IP_ADDRESS] + 2;
2190                ERRDF (11, MSG_11, RC25$ERR_RPT);
2191                CKLOOP;
2192                RETRIES = TRUE;
2193                end;
2194
2195            end;
2196
2197        PRINTF (FMT4, .RC25_DATA [RCSA, RCSA_PTN]); ! GIVE PORT TYPE NUMBER
2198        ENDSUB;
2199    !
2200    ! STEP 2 WRITE WITH A STEP 3 READ
```

```
;        2201  !
;        2202      BGNSUB;
;        2203      B_MASK = 3;                                   ! MASK UPTO STEP3 READ
;        2204      DATA2 = COM_AREA;                             ! COM AREA START ADDRESS
;        2205
;        2206      if AZT_INIT ()                                ! DO INIT AND IF ERROR
;        2207      then
;        2208          begin
;        2209          ERRDF (12, MSG_14, RC25$ERR_RPT);         ! PRINT ERROR MESSAGE
;        2210
;        2211          if .RET_STATUS then DECODE ();            ! DECODE STATUS
;        2212
;        2213          CKLOOP;
;        2214          RETRIES = TRUE;
;        2215          end
;        2216      else
;        2217          begin
;        2218  ! CHECK FOR ECHOED VECTOR AND IE BIT
;        2219          TEMP = .DATA1<0, 8>;
;        2220
;        2221          if (.RC25_DATA [RCSA, RCSA_7_0] nequ .TEMP)
;        2222          then                                      ! IF ECHOED INFO NOT CORRECT
;        2223              begin
;        2224              P_MASK = 2;
;        2225              PT = FMT2;
;        2226              P2 = ADAPT;
;        2227              P4 = .TEMP;
;        2228              P5 = .RC25_DATA [RCSA, RCSA_7_0];
;        2229              P6 = .RT_TABLE [RT_IP_ADDRESS] + 2;
;        2230              ERRDF (13, MSG_11, RC25$ERR_RPT);     ! REPORT ERROR
;        2231              CKLOOP;
;        2232              RETRIES = TRUE;
;        2233              end;
;        2234
;        2235          end;
;        2236
;        2237      ENDSUB;
;        2238  !
;        2239  ! STEP 3 WRITE WITH STEP 4 READ
;        2240  !
;        2241      BGNSUB;
;        2242      B_MASK = 7;                                   ! BRING UPTO STEP4 READ
;        2243      DATA3 = 0;                                    ! RING BASE HIGH ADDRESS
;        2244
;        2245      if AZT_INIT ()                                ! INIT AZTEC
;        2246      then                                          ! IF ERROR
;        2247          begin                                     ! THEN
;        2248          ERRDF (14, MSG_14, RC25$ERR_RPT);         ! PRINT OUT ERROR MESSAGE
;        2249
;        2250          if .RET_STATUS then DECODE ();            ! DECODE ERROR
;        2251
;        2252          CKLOOP;
;        2253          RETRIES = TRUE;
;        2254          end;
;        2255
;        2256  ! PRINT MICRO CODE VERSION INFO.
;        2257      PRINTF (FMT6, .RC25_DATA [RCSA, RCSA_MODEL], .RC25_DATA [RCSA, RCSA_U_CODE]);
```

```
;          2258      ENDSUB;
;          2259
;          2260      if (.RETRIES) then DO_RETRIES ();
;          2261
;          2262      if (.NUM_RETRIES eqlu ZERO) then exitloop;
;          2263
;          2264      end;
;          2265
;          2266  ENDTST;


                                          .SBTTL  $T5 TEST SECTION
000000  005067  000000G                $T5:    CLR     NUM.RETRIES                     ;                       2102
000004  032767  000001  000000G               BIT     #1,SWP.TRACE                    ;                       2104
000012  001407                                BEQ     1$
000014  012746  000000G                       MOV     #DBM9,-(SP)
000020  012746  000001                        MOV     #1,-(SP)
000024  010600                                MOV     SP,R0                           ; SP,*
000026  104417                                TRAP    17
000030  022626                                CMP     (SP)+,(SP)+
000032  026767  000000G 000000G          1$:   CMP     NUM.RETRIES,SWP.RETRIES         ;                       2106
000040  101401                                BLOS    2$
000042  000207                                RTS     PC
000044  104402                          2$:   TRAP    2                               ;                       2107
000046  105067  000000G                       CLRB    B.MASK                          ;                       2119
000052  004767  000000G                       JSR     PC,AZT.INIT                     ;                       2121
000056  006000                                ROR     R0
000060  103023                                BCC     5$
000062  104455                                TRAP    55                              ;                       2125
000064  000010                                .WORD   10
000066  000000G                               .WORD   MSG.14
000070  000000G                               .WORD   RC25$ERR.RPT
000072  032767  000001  000000G               BIT     #1,RET.STATUS                   ;                       2127
000100  001402                                BEQ     3$
000102  004767  000000G                       JSR     PC,DECODE
000106  104465                          3$:   TRAP    65
000110  006000                                ROR     R0
000112  103003                                BCC     4$
000114  162706  000006                        SUB     #6,SP
000120  000507                                BR      9$
000122  012767  000001  000000G         4$:   MOV     #1,RETRIES                      ;                       2130
000130  032767  002000  000002G         5$:   BIT     #2000,RC25.DATA+2               ;                       2135
000136  001004                                BNE     6$
000140  032767  000400  000002G               BIT     #400,RC25.DATA+2                ;                       2137
000146  001042                                BNE     8$
000150  112767  000002  000000G         6$:   MOVB    #2,P.MASK                       ;                       2142
000156  012767  000000G 000000G               MOV     #FMT3,P1                        ;                       2143
000164  012767  000001  000000G               MOV     #1,P2                           ;                       2144
000172  016700  000000G                       MOV     RC25.ADDR,R0                    ;                       2145
000176  062700  000002                        ADD     #2,R0
000202  010067  000000G                       MOV     R0,P4
000206  016767  000002G 000000G               MOV     RC25.DATA+2,P5                  ;                       2146
000214  012767  000001  000000G               MOV     #1,P6                           ;                       2147
000222  104455                                TRAP    55                              ;                       2148
000224  000011                                .WORD   11
000226  000000G                               .WORD   MSG.14
000230  000000G                               .WORD   RC25$ERR.RPT
```

```
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$:    MOV     #1,RETRIES                      ;                       2150
000254  016767  000002G 000000G  8$:    MOV     RC25.DATA+2,TEMP                ;                       2153
000262  006267  000000G                  ASR     TEMP                            ;                       2154
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)                      ;                       2155
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$:     ADD     #6,SP                           ;                       2107
000344  104467                          TRAP    67                              ;                       2155
000346  006000                          ROR     R0
000350  103635                          BLO     2$
000352  104402                  10$:    TRAP    2                               ;                       2156
000354  112767  000001  000000G         MOVB    #1,B.MASK                       ;                       2161
000362  016700  000000G                  MOV     RT.TABLE,R0                     ;                       2162
000366  016046  000002                  MOV     2(R0),-(SP)
000372  012746  000004                  MOV     #4,-(SP)
000376  004767  000000G                  JSR     PC,BL$DIV
000402  010067  000000G                  MOV     R0,DATA1
000406  162767  040200  000000G         SUB     #40200,DATA1
000414  004767  000000G                  JSR     PC,AZT.INIT                     ;                       2166
000420  006000                          ROR     R0
000422  103023                          BCC     13$
000424  104455                          TRAP    55                              ;                       2169
000426  000012                          .WORD   12
000430  000000G                         .WORD   MSG.14
000432  000000G                         .WORD   RC25$ERR.RPT
000434  032767  000001  000000G         BIT     #1,RET.STATUS                   ;                       2171
000442  001402                          BEQ     11$
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$:    MOV     #1,RETRIES                      ;                       2174
000470  000456                          BR      15$                             ;                       2166
000472  005067  000000G          13$:    CLR     TEMP                            ;                       2179
000476  116767  000001G 000000G         MOVB    DATA1+1,TEMP
000504  005000                          CLR     R0                              ;                       2181
000506  156700  000002G                  BISB    RC25.DATA+2,R0
000512  020067  000000G                  CMP     R0,TEMP
000516  001443                          BEQ     15$
000520  112767  000002  000000G         MOVB    #2,P.MASK                       ;                       2184
000526  012767  000000G 000000G         MOV     #FMT2,P1                        ;                       2185
000534  012767  000001  000000G         MOV     #1,P2                           ;                       2186
```

```
000542  016767  000000G 000000G          MOV     TEMP,P4                         ;                       2187
000550  005000                           CLR     R0                              ;                       2188
000552  156700  000002G                  BISB    RC25.DATA+2,R0
000556  010067  000000G                  MOV     R0,P5
000562  017700  000000G                  MOV     @RT.TABLE,R0                    ;                       2189
000566  062700  000002                   ADD     #2,R0
000572  010067  000000G                  MOV     R0,P6
000576  104455                           TRAP    55                              ;                       2190
000600  000013                           .WORD   13
000602  000000G                          .WORD   MSG.11
000604  000000G                          .WORD   RC25$ERR.RPT
000606  104465                           TRAP    65
000610  006000                           ROR     R0
000612  103002                           BCC     14$
000614  024646                           CMP     -(SP),-(SP)
000616  000417                           BR      16$
000620  012767  000001  000000G  14$:     MOV     #1,RETRIES                      ;                       2192
000626  016700  000002G          15$:     MOV     RC25.DATA+2,R0                  ;                       2197
000632  000300                           SWAB    R0
000634  042700  177770                   BIC     #177770,R0
000640  010016                           MOV     R0,(SP)
000642  012746  000000G                  MOV     #FMT4,-(SP)
000646  012746  000002                   MOV     #2,-(SP)
000652  010600                           MOV     SP,R0                           ; SP,*
000654  104417                           TRAP    17
000656  062706  000010          16$:     ADD     #10,SP                          ;                       2156
000662  104467                           TRAP    67                              ;                       2197
000664  006000                           ROR     R0
000666  103631                           BLO     10$
000670  104402                  17$:     TRAP    2                               ;                       2198
000672  112767  000003  000000G          MOVB    #3,B.MASK                       ;                       2203
000700  012767  000000G 000000G          MOV     #COM.AREA,DATA2                 ;                       2204
000706  004767  000000G                  JSR     PC,AZT.INIT                     ;                       2206
000712  006000                           ROR     R0
000714  103021                           BCC     19$
000716  104455                           TRAP    55                              ;                       2209
000720  000014                           .WORD   14
000722  000000G                          .WORD   MSG.14
000724  000000G                          .WORD   RC25$ERR.RPT
000726  032767  000001  000000G          BIT     #1,RET.STATUS                   ;                       2211
000734  001402                           BEQ     18$
000736  004767  000000G                  JSR     PC,DECODE
000742  104465                  18$:     TRAP    65
000744  006000                           ROR     R0
000746  103460                           BLO     20$
000750  012767  000001  000000G          MOV     #1,RETRIES                      ;                       2214
000756  000454                           BR      20$                             ;                       2206
000760  005067  000000G          19$:     CLR     TEMP                            ;                       2219
000764  116767  000000G 000000G          MOVB    DATA1,TEMP
000772  005000                           CLR     R0                              ;                       2221
000774  156700  000002G                  BISB    RC25.DATA+2,R0
001000  020067  000000G                  CMP     R0,TEMP
001004  001441                           BEQ     20$
001006  112767  000002  000000G          MOVB    #2,P.MASK                       ;                       2224
001014  012767  000000G 000000G          MOV     #FMT2,P1                        ;                       2225
001022  012767  000001  000000G          MOV     #1,P2                           ;                       2226
001030  016767  000000G 000000G          MOV     TEMP,P4                         ;                       2227
```

```
001036  005000                          CLR     R0                              ;                       2228
001040  156700  000002G                 BISB    RC25.DATA+2,R0
001044  010067  000000G                 MOV     R0,P5
001050  017700  000000G                 MOV     @RT.TABLE,R0                    ;                       2229
001054  062700  000002                  ADD     #2,R0
001060  010067  000000G                 MOV     R0,P6
001064  104455                          TRAP    55                              ;                       2230
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                      ;                       2232
001110  104467                  20$:    TRAP    67                              ;                       2235
001112  006000                          ROR     R0
001114  103665                          BLO     17$
001116  104402                  21$:    TRAP    2                               ;                       2237
001120  112767  000007  000000G        MOVB    #7,B.MASK                       ;                       2242
001126  005067  000000G                 CLR     DATA3                           ;                       2243
001132  004767  000000G                 JSR     PC,AZT.INIT                     ;                       2245
001136  006000                          ROR     R0
001140  103023                          BCC     24$
001142  104455                          TRAP    55                              ;                       2248
001144  000016                          .WORD   16
001146  000000G                         .WORD   MSG.14
001150  000000G                         .WORD   RC25$ERR.RPT
001152  032767  000001  000000G        BIT     #1,RET.STATUS                   ;                       2250
001160  001402                          BEQ     22$
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  23$:   MOV     #1,RETRIES                      ;                       2253
001210  016746  000002G          24$:   MOV     RC25.DATA+2,-(SP)               ;                       2257
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$:    ADD     #10,SP                          ;                       2237
001262  104467                          TRAP    67                              ;                       2257
001264  006000                          ROR     R0
001266  103713                          BLO     21$
001270  032767  000001  000000G        BIT     #1,RETRIES                      ;                       2260
001276  001402                          BEQ     26$
001300  004767  000000G                 JSR     PC,DO.RETRIES
001304  005767  000000G          26$:   TST     NUM.RETRIES                     ;                       2262
```

```
001310  001402                          BEQ     27$
001312  000167  176514                  JMP     1$
001316  000207                  27$:    RTS     PC                      ;                       2085

; Routine Size:  360 words,     Routine Base:  AC$CODE + 2362
; Maximum stack depth per invocation:  6 words



                                        .SBTTL  T5 TEST SECTION
000000  004767  176454          T5::
000000                          1$:     JSR     PC,$T5                  ;                       2264
000004  104466                          TRAP    66
000006  006000                          ROR     R0
000010  103773                          BLO     1$
000012  000207                          RTS     PC

; Routine Size:  6 words,       Routine Base:  AC$CODE + 3702
; Maximum stack depth per invocation:  2 words

;         2267  !<BLF/PAGE>
```

```
2268  !
2269  BGNTST;
2270
2271  !++
2272  ! TEST #6 - PURGE AND POLL TEST
2273  !
2274  ! DESCRIPTION:
2275  !
2276  !       THIS TEST WILL PERFORM THE FIRST THREE STEPS OF THE INIT SEQUENCE.
2277  !       WHEN THE HOST RESPONDS TO THE STEP 3 TRANSITION IT WILL WRITE A ONE
2278  !       BIT TO BIT 15 OF THE SA REGISTER, THERBY REQUESTING THE EXECUTION OF
2279  !       PURGE AND POLL TESTING.  THE HOST THEN WAITS FOR THE SA REGISTER TO
2280  !       TRANSITION TO A ZERO VALUE.  THE HOST THEN WRITES ZEROS TO THE SA
2281  !       REGISTER SIMULATING A "PURGE COMPLETED" HOST ACTION.  THE HOST THEN
2282  !       READS THE IP REGISTER  TO SMULATF A "START POLLING" COMMAND FROM THE
2283  !       HOST TO THE PORT.  THE TEST IS COMPLETE WHEN THE CONTROLLER ANNOUNCES
2284  !       THE TRANSITION TO STEP 4 IN THE SA REGISTER.
2285  !
2286  !       FAILURE TO PROPERLY COMPLETE THIS TEST WILL BE REPORTED.
2287  !
2288  !       LOOP ON ERROR WILL RESTART THE TEST.
2289  !--
2290
2291  if .SWP_TRACE then PRINTF (DBM12);                    ! TEST 6
2292
2293  NUM_RETRIES = ZERO;
2294
2295  while (.NUM_RETRIES lequ .SWP_RETRIES) do
2296      begin
2297      TIP = 6;
2298      B_MASK = 3;
2299      DATA1 = %o'100200' + .RT_TABLE [RT_VECTOR]/4;       ! IE AND VECTOR ADDRESS
2300      DATA2 = RINGBASE;                                   ! RING BASE LOW ADDRESS
2301      DATA3 = %o'100000';                                 ! PURGE AND POLL
2302
2303      if AZT_INIT ()                                      ! DO UPTO STEP 3 READ AND
2304      then                                                ! CHECK FOR ERRORS
2305          begin                                           ! IF ERRORS THEN
2306          ERRDF (15, MSG_14, RC25$ERR_RPT);               ! REPORT THEM
2307
2308          if .RET_STATUS then DECODE ();                  ! DECODE STATUS
2309
2310          CKLOOP;
2311          RETRIES = TRUE;
2312          end
2313      else
2314          begin
2315          WRT_RC25 (RCSA, .DATA3);                        ! WRITE PURGE AND POLL
2316
2317          while (.RC25_ADDR [RCSA, RC_ALL] nequ ZERO) do
2318              DELAY (10);                                 ! WAIT UNTIL SA=0
2319
2320          WRT_RC25 (RCSA, FALSE);                         ! WRITE ALL ZERO'S TO SA
2321          DATA1 = .RC25_ADDR [RCIP, RC_ALL];              ! READ THE IP REGISTER
2322          DATA1 = %o'10';                                 ! INIT THE LOOP COUNT
2323
2324          while (.DATA1 nequ ZERO) do
```

```
                begin
                delay (333);

                if .I_AM_NEX eqlu ALL_ONES then exitloop;

                DATA1 = .DATA1 - 1;
                end;

            if .I_AM_NEX eqlu ALL_ONES
            then
                begin
                RC25_DATA [RCSA, RC_ALL] = .RC25_ADDR [RCSA, RC_ALL];

                if .RC25_DATA [RCSA, RCSA_ER]          ! IF PORT FATAL ERROR
                then
                    begin                              !  THEN REPORT IT.
                    RET_STATUS = PFE_CODE;
                    P1 = FMT3;
                    P2 = ADAPT;
                    P4 = (.RC25_ADDR) + 2;
                    P5 = .RC25_DATA [RCSA, RC_ALL];
                    P6 = %o'04';
                    P_MASK = 2;
                    ERRDF (16, MSG_14, RC25$ERR_RPT);
                    DECODE ();
                    CKLOOP;
                    RETRIES = TRUE;
                    end;

                if (.RC25_DATA [RCSA, RCSA_STEP] nequ %b'1000')      ! CHECK FOR STEP 4 COMPLETE
                then
                    begin
                    P1 = FMT3;
                    P2 = ADAPT;
                    P4 = (.RC25_ADDR) + 2;
                    P5 = .RC25_DATA [RCSA, RC_ALL];
                    P6 = %o'10';                       ! MASK = STEP 4
                    P_MASK = 2;
                    ERRDF (17, MSG_14, RC25$ERR_RPT);
                    CKLOOP;
                    RETRIES = TRUE;
                    end;

                end
            else
                begin
                RET_STATUS = CTO_CODE;
                RETRIES = TRUE;
                ERRDF (18, MSG_9, 0);
                DECODE ();
                end;

            end;

        if (.RETRIES) then DO_RETRIES ();

        if (.NUM_RETRIES eqlu ZERO) then exitloop;
```

```
;         2382
;         2383      end;
;         2384
;         2385  ENDTST;


                                        .SBTTL  $T6 TEST SECTION
000000  010146                  $T6:    MOV     R1,-(SP)                ;                       2266
000002  162706  000010                  SUB     #10,SP
000006  032767  000001  000000G        BIT     #1,SWP.TRACE            ;                       2291
000014  001407                          BEQ     1$
000016  012746  000000G                 MOV     #DBM12,-(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$:     CLR     NUM.RETRIES             ;                       2293
000040  026767  000000G 000000G  2$:     CMP     NUM.RETRIES,SWP.RETRIES ;                       2295
000046  101402                          BLOS    3$
000050  000167  000674                  JMP     25$
000054  012767  000006  000000G  3$:     MOV     #6,TIP                  ;                       2297
000062  112767  000003  000000G        MOVB    #3,B.MASK               ;                       2298
000070  016700  000000G                 MOV     RT.TABLE,R0             ;                       2299
000074  016046  000002                  MOV     2(R0),-(SP)
000100  012746  000004                  MOV     #4,-(SP)
000104  004767  000000G                 JSR     PC,BL$DIV
000110  010067  000000G                 MOV     R0,DATA1
000114  162767  077600  000000G        SUB     #77600,DATA1
000122  012767  000000G 000000G        MOV     #RINGBASE,DATA2         ;                       2300
000130  012767  100000  000000G        MOV     #-100000,DATA3          ;                       2301
000136  004767  000000G                 JSR     PC,AZT.INIT             ;                       2303
000142  006000                          ROR     R0
000144  103025                          BCC     6$
000146  104455                          TRAP    55                      ;                       2306
000150  000017                          .WORD   17
000152  000000G                         .WORD   MSG.14
000154  000000G                         .WORD   RC25$ERR.RPT
000156  032767  000001  000000G        BIT     #1,RET.STATUS           ;                       2308
000164  001402                          BEQ     4$
000166  004767  000000G                 JSR     PC,DECODE
000172  104465                  4$:     TRAP    65
000174  006000                          ROR     R0
000176  103003                          BCC     5$
000200  022626                          CMP     (SP)+,(SP)+
000202  000167  000542                  JMP     25$
000206  012767  000001  000000G  5$:     MOV     #1,RETRIES              ;                       2311
000214  000167  000474                  JMP     22$                     ;                       2303
000220  016701  000000G          6$:     MOV     DATA3,R1                ; *,RCM.REG             2315
000224  016700  000000G                 MOV     RC25.ADDR,R0
000230  010160  000002                  MOV     R1,2(R0)                ; RCM.REG,*
000234  016700  000000G          7$:     MOV     RC25.ADDR,R0            ;                       2317
000240  016066  000002  000010        MOV     2(R0),10(SP)            ; *,RC.REG
000246  001414                          BEQ     11$
000250  012701  000012                  MOV     #12,R1                  ; *,$$TMP2              2318
000254  001767                  8$:     BEQ     7$
000256  016700  000000G                 MOV     L$DLY,R0                ; *,$$TMP1
000262  001404                          BEQ     10$
```

```
000264  005066  000012                  9$:     CLR     12(SP)                  ; $$TMP
000270  005300                                  DEC     R0                      ; $$TMP1
000272  001374                                  BNE     9$
000274  005301                          10$:    DEC     R1                      ; $$TMP2
000276  000766                                  BR      8$
000300  005001                          11$:    CLR     R1                      ; RCM.REG                       2320
000302  016700  000000G                         MOV     RC25.ADDR,R0
000306  005060  000002                          CLR     2(R0)
000312  011066  000006                          MOV     (R0),6(SP)              ; RC25.ADDR,RC.REG              2321
000316  012767  000010  000000G                 MOV     #10,DATA1               ;                               2322
000324  001423                          12$:    BEQ     17$                     ;                               2324
000326  012701  000515                          MOV     #515,R1                 ; *,$$TMP2                      2326
000332  001411                          13$:    BEQ     16$
000334  016700  000000G                         MOV     L$DLY,R0                ; *,$$TMP1
000340  001404                                  BEQ     15$
000342  005066  000012                  14$:    CLR     12(SP)                  ; $$TMP
000346  005300                                  DEC     R0                      ; $$TMP1
000350  001374                                  BNE     14$
000352  005301                          15$:    DEC     R1                      ; $$TMP2
000354  000766                                  BR      13$
000356  026727  000000G 177777          16$:    CMP     I.AM.NEX,#-1            ;                               2328
000364  001403                                  BEQ     17$
000366  005367  000000G                         DEC     DATA1                   ;                               2330
000372  000754                                  BR      12$                     ;                               2324
000374  026727  000000G 177777          17$:    CMP     I.AM.NEX,#-1            ;                               2333
000402  001130                                  BNE     21$
000404  016700  000000G                         MOV     RC25.ADDR,R0            ;                               2336
000410  016066  000002  000004                 MOV     2(R0),4(SP)             ; *,RC.REG
000416  016667  000004  000002G                 MOV     4(SP),RC25.DATA+2       ; RC.REG,*
000424  100046                                  BPL     19$                     ;                               2338
000426  012767  000021  000000G                 MOV     #21,RET.STATUS          ;                               2341
000434  012767  000000G 000000G                 MOV     #FMT3,P1                ;                               2342
000442  012767  000001  000000G                 MOV     #1,P2                   ;                               2343
000450  016700  000000G                         MOV     RC25.ADDR,R0            ;                               2344
000454  062700  000002                          ADD     #2,R0
000460  010067  000000G                         MOV     R0,P4
000464  016767  000002G 000000G                 MOV     RC25.DATA+2,P5          ;                               2345
000472  012767  000004  000000G                 MOV     #4,P6                   ;                               2346
000500  112767  000002  000000G                 MOVB    #2,P.MASK               ;                               2347
000506  104455                                  TRAP    55                      ;                               2348
000510  000020                                  .WORD   20
000512  000000G                                 .WORD   MSG.14
000514  000000G                                 .WORD   RC25$ERR.RPT
000516  004767  000000G                         JSR     PC,DECODE               ;                               2349
000522  104465                                  TRAP    65
000524  006000                                  ROR     R0
000526  103002                                  BCC     18$
000530  022626                                  CMP     (SP)+,(SP)+
000532  000506                                  BR      25$
000534  012767  000001  000000G         18$:    MOV     #1,RETRIES              ;                               2351
000542  016700  000002G                 19$:    MOV     RC25.DATA+2,R0          ;                               2354
000546  042700  103777                          BIC     #103777,R0
000552  020027  040000                          CMP     R0,#40000
000556  001456                                  BEQ     22$
000560  012767  000000G 000000G                 MOV     #FMT3,P1                ;                               2357
000566  012767  000001  000000G                 MOV     #1,P2                   ;                               2358
000574  016700  000000G                         MOV     RC25.ADDR,R0            ;                               2359
```

```
000600  062700  000002                          ADD     #2,R0                                                   2360
000604  010067  000000G                         MOV     R0,P4                                                   2361
000610  016767  000002G 000000G                 MOV     RC25.DATA+2,P5          ;                               2362
000616  012767  000010  000000G                 MOV     #10,P6                  ;                               2363
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$:     MOV     #1,RETRIES              ;                               2365
000662  000414                                  BR      22$                     ;                               2333
000664  012767  000011  000000G        21$:     MOV     #11,RET.STATUS          ;                               2371
000672  012767  000001  000000G                 MOV     #1,RETRIES              ;                               2372
000700  104455                                  TRAP    55                      ;                               2373
000702  000022                                  .WORD   22
000704  000000G                                 .WORD   MSG.9
000706  000000                                  .WORD   0
000710  004767  000000G                         JSR     PC,DECODE               ;                               2374
000714  032767  000001  000000G        22$:     BIT     #1,RETRIES              ;                               2379
000722  001402                                  BEQ     23$
000724  004767  000000G                         JSR     PC,DO.RETRIES
000730  005767  000000G                 23$:    TST     NUM.RETRIES             ;                               2381
000734  001002                                  BNE     24$
000736  022626                                  CMP     (SP)+,(SP)+
000740  000403                                  BR      25$
000742  022626                          24$:    CMP     (SP)+,(SP)+             ;                               2296
000744  000167  177070                         JMP     2$                      ;                               2295
000750  062706  000010                 25$:    ADD     #10,SP                  ;                               2266
000754  012601                                  MOV     (SP)+,R1
000756  000207                                  RTS     PC

; Routine Size:  248 words,     Routine Base:  AC$CODE + 3716
; Maximum stack depth per invocation:  9 words



                                        .SBTTL  T6 TEST SECTION
000000  004767  177014                  T6::
000000                          1$:     JSR     PC,$T6                  ;                               2383
000004  104466                          TRAP    66
000006  006000                          ROR     R0
000010  103773                          BLO     1$
000012  000207                          RTS     PC

; Routine Size:  6 words,       Routine Base:  AC$CODE + 4676
; Maximum stack depth per invocation:  2 words

;        2386  !<BLF/PAGE>
```

```
;        2387  !
;        2388  BGNTST;
;        2389
;        2390  !++
;        2391  ! TEST #7 - SMALL RING BUFFER INIT TEST
;        2392  !
;        2393  ! DESCRIPTION:
;        2394  !
;        2395  !       THE AZTEC WILL BE INITIALIZED WITHOUT INTERRUPTS AND USING THE
;        2396  !       SMALLEST RING BUFFER.  THIS WILL BE THE FIRST TIME THAT THE
;        2397  !       INITIALIZATION SEQUENCE IS CARRIED OUT TO COMPLETION.  INITIALIZING
;        2398  !       WITH THF SMALLEST RING BUFFER MINIMIZES THE HOST MEMORY AREA WITH
;        2399  !       WHICH THE AZTEC CONTROLLER MUST BE ABLE TO COMMUNICATE.
;        2400  !
;        2401  !       FAILURE TO PROPERLY INITIATETHE AZTEC WILL BE REPORTED.
;        2402  !
;        2403  !       IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM THE
;        2404  !       START OF THIS TEST.
;        2405  !
;        2406
;        2407  if .SWP_TRACE then PRINTF (DBM13);                  ! TEST 7
;        2408
;        2409  NUM_RETRIES = ZERO;
;        2410
;        2411  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;        2412      begin
;        2413      TIP = 7;
;        2414      B_MASK = %o'17';                                  ! SELECT ALL STEPS
;        2415      DATA1 = %o'100200';                               ! STEP 1 WRITE WITH MIN. RING SIZES
;        2416      DATA2 = RING_B [0];                               ! SET UP RING BASE ADDRESS
;        2417      DATA3 = 0;
;        2418      DATA4<0, 1> = 1;
;        2419      RING_B [0] = ALL_ONES;                            ! INIT RING_B [0] AND [1]
;        2420      RING_B [1] = ALL_ONES;                            ! WITH ALL ONES (-1)
;        2421
;        2422      if AZP_INIT ()                                    ! DO INIT STEPS
;        2423      then                                              !
;        2424          begin                                         ! IF ERROR THEN
;        2425          ERRDF (19, MSG_14, RC25$ERR_RPT);             ! THEN REPORT THE ERROR
;        2426
;        2427          if .RET_STATUS then DECODE ();                ! DECODE RETURN STATUS
;        2428
;        2429          CKLOOP;
;        2430          RETRIES = TRUE;
;        2431          end;
;        2432
;        2433      if .RING_B [0] nequ 0 and .RING_B [1] nequ 0          ! TEST THAT THE RC25 CLEARED
;        2434      then                                              ! RING BUFFERS
;        2435          begin                                         ! IF NOT THEN ERROR
;        2436          ERRDF (20, MSG_10, 0);                        ! AND REPORT IT
;        2437          CKLOOP;
;        2438          RETRIES = TRUE;
;        2439          end;
;        2440
;        2441      if (.RETRIES) then DO_RETRIES ();
;        2442
;        2443      if (.NUM_RETRIES eqlu ZERO) then exitloop;
```

```
;         2444
;         2445      end;
;         2446
;         2447  return;
;         2448  ENDTST;


                                        .SBTTL  $T7 TEST SECTION
000000  032767  000001  000000G  $T7:   BIT     #1,SWP.TRACE                    ;                       2407
000006  001407                        BEQ     1$
000010  012746  000000G               MOV     #DBM13,-(SP)
000014  012746  000001                MOV     #1,-(SP)
000020  010600                        MOV     SP,R0                           ; SP,*
000022  104417                        TRAP    17
000024  022626                        CMP     (SP)+,(SP)+
000026  005067  000000G        1$:     CLR     NUM.RETRIES                     ;                       2409
000032  026767  000000G 000000G  2$:     CMP     NUM.RETRIES,SWP.RETRIES         ;                       2411
000040  101104                        BHI     7$
000042  012767  000007  000000G       MOV     #7,TIP                          ;                       2413
000050  112767  000017  000000G       MOVB    #17,B.MASK                      ;                       2414
000056  012767  100200  000000G       MOV     #-77600,DATA1                   ;                       2415
000064  012767  000100' 000000G       MOV     #RING.B,DATA2                   ;                       2416
000072  005067  000000G               CLR     DATA3                           ;                       2417
000076  152767  000001  000000G       BISB    #1,DATA4                        ;                       2418
000104  012767  177777  000100'       MOV     #-1,RING.B                      ;                       2419
000112  012767  177777  000102'       MOV     #-1,RING.B+2                    ;                       2420
000120  004767  000000G               JSR     PC,AZP.INIT                     ;                       2422
000124  006000                        ROR     R0
000126  103020                        BCC     4$
000130  104455                        TRAP    55                              ;                       2425
000132  000023                        .WORD   23
000134  000000G                       .WORD   MSG.14
000136  000000G                       .WORD   RC25$ERR.RPT
000140  032767  000001  000000G       BIT     #1,RET.STATUS                   ;                       2427
000146  001402                        BEQ     3$
000150  004767  000000G               JSR     PC,DECODE
000154  104465                3$:     TRAP    65
000156  006000                        ROR     R0
000160  103434                        BLO     7$
000162  012767  000001  000000G       MOV     #1,RETRIES                      ;                       2430
000170  005767  000100'        4$:     TST     RING.B                          ;                       2433
000174  001415                        BEQ     5$
000176  005767  000102'               TST     RING.B+2
000202  001412                        BEQ     5$
000204  104455                        TRAP    55                              ;                       2436
000206  000024                        .WORD   24
000210  000000G                       .WORD   MSG.10
000212  000000                        .WORD   0
000214  104465                        TRAP    65
000216  006000                        ROR     R0
000220  103414                        BLO     7$
000222  012767  000001  000000G       MOV     #1,RETRIES                      ;                       2438
000230  032767  000001  000000G  5$:     BIT     #1,RETRIES                      ;                       2441
000236  001402                        BEQ     6$
000240  004767  000000G               JSR     PC,DO.RETRIES
000244  005767  000000G        6$:     TST     NUM.RETRIES                     ;                       2443
000250  001270                        BNE     2$
```

```
000252  000207                          7$:     RTS     PC                              ;                                       2385

; Routine Size:  86 words,      Routine Base:  AC$CODE + 4712
; Maximum stack depth per invocation:  4 words




                                                .SBTTL  T7 TEST SECTION
000000  004767  177520                  T7::
000000                                  1$:     JSR     PC,$T7                          ;                                       2447
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

; Routine Size:  6 words,       Routine Base:  AC$CODE + 5166
; Maximum stack depth per invocation:  2 words


;          2449  !<BLF/PAGE>
```

```
;          2450  !
;          2451  BGNTST;
;          2452
;          2453  !++
;          2454  ! TEST #8 - LARGE RING BUFFER INIT TEST
;          2455  !
;          2456  ! DESCRIPTION:
;          2457  !
;          2458  !       THE INIT SEQUENCE IS EXECUTED WITHOUT INTERRUPTS WITH A RING BUFFER
;          2459  !       LARGE ENOUGH TO COVER THE NORMAL HOST COMMUNCIATIONS AREA PACKET AND
;          2460  !       BUFFER SPACE ( A 5 IN MESSAGE LENGTH AND A 5 IN COMMAND LENGTH).
;          2461  !
;          2462  !       A FAILURE TO COMLETE THE INITIALIZATION SEQUENCE WITHOUT ERROR WILL BE
;          2463  !       REPORTED.
;          2464  !
;          2465  !       IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE TO THE
;          2466  !       BEGINNING OF THIS TEST.
;          2467  !--
;          2468
;          2469  if .SWP_TRACE then PRINTF (DBM14);                ! TEST 8
;          2470
;          2471  NUM_RETRIES = ZERO;
;          2472
;          2473  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;          2474      begin
;          2475      TIP = 8;
;          2476      B_MASK = %o'17';                               ! SET MASK BIT FOR COMPLETE INIT.
;          2477      DATA1<15, 1> = TRUE;                           ! SET BIT 15 FOR STEP-1 WRITE
;          2478      DATA1<14, 1> = 0;                              ! NO DIAGNOSTIC WRAP MODE
;          2479      DATA1<11, 3> = SND_SIZ;                        ! SET UP 16 COMMAND RINGS LENGTH
;          2480      DATA1<8, 3> = REC_SIZ;                         ! SET UP 16 RESPONSE RINGS LENGTH
;          2481      DATA1<7, 1> = 0;                               ! DISABLE INTERRUPT
;          2482      DATA1<0, 7> = 0;                               ! LOAD INTERRUPT VECTOR ADDRESS
;          2483      DATA2 = COM_AREA;                              ! LOAD COMMUNICATIONS AREA ADDRESS
;          2484      DATA3 = ZERO;                                  ! HI-ORDER ADDR = ZERO
;          2485      DATA4 = %o'177403';                            ! "LAST FAIL" PACKET RESPONSE BIT SET
;          2486  !INITIALIZE COM_AREA WITH ALL_ONES PRIOR TO INIT
;          2487
;          2488      incru I from 0 to RING_SIZE - 1 do
;          2489
;          2490          incru J from 0 to 1 do
;          2491              COM_AREA [.I, .J, WORD_REF] = ALL_ONES;
;          2492
;          2493      if AZP_INIT ()                                 ! DO STEP INIT AND CHECK FOR ERROR
;          2494      then                                           !
;          2495          begin                                      ! IF ERRORS THEN
;          2496          ERRDF (21, MSG_14, RC25$ERR_RPT);          ! REPORT ERROR
;          2497
;          2498          if .RET_STATUS then DECODE ();             ! DECODE STATUS
;          2499
;          2500          CKLOOP;
;          2501          RETRIES = TRUE;
;          2502          end;
;          2503
;          2504      incru I from 0 to RING_SIZE - 1 do             ! TEST RING AREA FOR ZEROES
;          2505
;          2506          incru J from 0 to 1 do
```

```
;          2507
;          2508                if .COM_AREA [.I, .J, WORD_REF] nequ 0      ! IF RING AREA IS NOT CLEAR
;          2509                then                                        ! THEN
;          2510                    begin                                   ! REPORT ERROR
;          2511                    ERRDF (22, MSG_10, 0);                  !
;          2512                    CKLOOP;
;          2513                    RETRIES = TRUE;
;          2514                    end;
;          2515
;          2516            if (.RETRIES) then DO_RETRIES ();
;          2517
;          2518            if (.NUM_RETRIES eqlu ZERO) then exitloop;
;          2519
;          2520            end;
;          2521
;          2522        return;
;          2523        ENDTST;


                                        .SBTTL  $T8 TEST SECTION
000000  004167  000000G                 $T8:    JSR     R1,$SAVE2                       ;                       2448
000004  032767  000001  000000G                 BIT     #1,SWP.TRACE                    ;                       2469
000012  001407                                  BEQ     1$
000014  012746  000000G                         MOV     #DBM14,-(SP)
000020  012746  000001                          MOV     #1,-(SP)
000024  010600                                  MOV     SP,R0                           ; SP,*
000026  104417                                  TRAP    17
000030  022626                                  CMP     (SP)+,(SP)+
000032  005067  000000G                 1$:     CLR     NUM.RETRIES                     ;                       2471
000036  026767  000000G 000000G         2$:     CMP     NUM.RETRIES,SWP.RETRIES         ;                       2473
000044  101132                                  BHI     11$
000046  012767  000010  000000G                 MOV     #10,TIP                         ;                       2475
000054  112767  000017  000000G                 MOVB    #17,B.MASK                      ;                       2476
000062  012767  122000  000000G                 MOV     #122000,DATA1                   ;                       2482
000070  012767  000000G 000000G                 MOV     #COM.AREA,DATA2                 ;                       2483
000076  005067  000000G                         CLR     DATA3                           ;                       2484
000102  012767  177403  000000G                 MOV     #-375,DATA4                     ;                       2485
000110  005001                                  CLR     R1                              ; I                     2488
000112  005002                          3$:     CLR     R2                              ; J                     2490
000114  010100                          4$:     MOV     R1,R0                           ; I,*                   2491
000116  006300                                  ASL     R0
000120  060200                                  ADD     R2,R0                           ; J,*
000122  006300                                  ASL     R0
000124  012760  177777  000000G                 MOV     #-1,COM.AREA(R0)
000132  005202                                  INC     R2                              ; J                     2490
000134  020227  000001                          CMP     R2,#1                           ; J,*
000140  101765                                  BLOS    4$
000142  005201                                  INC     R1                              ; I                     2488
000144  020127  000037                          CMP     R1,#37                          ; I,*
000150  101760                                  BLOS    3$
000152  004767  000000G                         JSR     PC,AZP.INIT                     ;                       2493
000156  006000                                  ROR     R0
000160  103020                                  BCC     6$
000162  104455                                  TRAP    55                              ;                       2496
000164  000025                                  .WORD   25
000166  000000G                                 .WORD   MSG.14
000170  000000G                                 .WORD   RC25$ERR.RPT
```

```
000172  032767  000001  000000G                 BIT     #1,RET.STATUS                   ;                               2498
000200  001402                                  BEQ     5$
000202  004767  000000G                         JSR     PC,DECODE
000206  104465                          5$:     TRAP    65
000210  006000                                  ROR     R0
000212  103447                                  BLO     11$
000214  012767  000001  000000G                 MOV     #1,RETRIES                      ;                               2501
000222  005001                          6$:     CLR     R1                              ; I                             2504
000224  005002                          7$:     CLR     R2                              ; J                             2506
000226  010100                          8$:     MOV     R1,R0                           ; I,*                           2508
000230  006300                                  ASL     R0
000232  060200                                  ADD     R2,R0                           ; J,*
000234  006300                                  ASL     R0
000236  005760  000000G                         TST     COM.AREA(R0)
000242  001412                                  BEQ     9$
000244  104455                                  TRAP    55                              ;                               2511
000246  000026                                  .WORD   26
000250  000000G                                 .WORD   MSG.10
000252  000000                                  .WORD   0
000254  104465                                  TRAP    65
000256  006000                                  ROR     R0
000260  103424                                  BLO     11$
000262  012767  000001  000000G                 MOV     #1,RETRIES                      ;                               2513
000270  005202                          9$:     INC     R2                              ; J                             2506
000272  020227  000001                          CMP     R2,#1                           ; J,*
000276  101753                                  BLOS    8$
000300  005201                                  INC     R1                              ; I                             2504
000302  020127  000037                          CMP     R1,#37                          ; I,*
000306  101746                                  BLOS    7$
000310  032767  000001  000000G                 BIT     #1,RETRIES                      ;                               2516
000316  001402                                  BEQ     10$
000320  004767  000000G                         JSR     PC,DO.RETRIES
000324  005767  000000G                 10$:    TST     NUM.RETRIES                     ;                               2518
000330  001242                                  BNE     2$
000332  000207                          11$:    RTS     PC                              ;                               2448

; Routine Size:  110 words,     Routine Base:  AC$CODE + 5202
; Maximum stack depth per invocation:  7 words



                                                .SBTTL  T8 TEST SECTION
000000  004767  177440                  T8::
000000                                  1$:     JSR     PC,$T8                          ;                               2522
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

; Routine Size:  6 words,       Routine Base:  AC$CODE + 5536
; Maximum stack depth per invocation:  2 words

;       2524  !<BLF/PAGE>
```

```
;       2525  !
;       2526  BGNTST;
;       2527
;       2528  !++
;       2529  ! TEST #9 - "DIAGNOSTIC MACHINE" CODE DOWN LINE LOAD TEST
;       2530  !
;       2531  ! DESCRIPTION:
;       2532  !
;       2533  !       THIS "DIAGNOSTIC MACHINE" PROGRAM WILL ATTEMPT TO TRANSFER A BLOCK
;       2534  !       OF DATA FROM HOST MEMORY TO AN AREA IN THE CONTROLLER AND THEN
;       2535  !       EXAMINE THE TRANSFERED DATA.
;       2536  !
;       2537  !       IF THE TRANSFERED DATA NOT COMPARE CORRECTLY, THEN THE ERROR WILL
;       2538  !       BE REPORTED. THIS TEST ALSO REPORTS ERRORS IF ANY OF THE ROUTINES
;       2539  !       USED RETURNED FAILURE CODE.
;       2540  !
;       2541  !       IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM
;       2542  !       THE START OF THIS TEST.
;       2543  !--
;       2544
;       2545  if .SWP_TRACE then PRINTF (DBM15);                ! TEST 9
;       2546
;       2547  NUM_RETRIES = ZERO;
;       2548
;       2549  while (.NUM_RETRIES lequ .SWP_RETRIES) do
;       2550      begin
;       2551
;       2552      if AZTEC_READY ()                              ! GET AZTEC READY
;       2553      then
;       2554          begin
;       2555          ERRDF (23, AZT_READY_ERR, 0);              ! IF ERROR REPORT ERROR
;       2556
;       2557          if .RET_STATUS then DECODE ();
;       2558
;       2559          CKLOOP;
;       2560          RETRIES = TRUE;
;       2561          end
;       2562      else
;       2563          begin
;       2564          TEMP = .FREE_MEM_ADDR;                     ! SAVE FREE MEMORY STARTING ADDR.
;       2565
;       2566          incru COUNT from 0 to 1024 do              ! FILL NEXT 1024 LOC. WITH DATAS
;       2567              begin                                  !
;       2568              .TEMP = %o'125252';                    ! WRITE DATA O'125252' INTO MEMORY
;       2569              TEMP = .TEMP + 2;                      ! INCREMENT THE POINTER BY 2
;       2570              end;
;       2571
;       2572          CMD_REF = 3;                               ! SET COMMAND REFERENCE #3
;       2573          BUF_DESCRPTR = DM_09;                      ! DM-PROGRAM STARTING ADDRESS
;       2574          BYTE_COUNT = 93*2;                         ! TOTAL DM PROGRAM LENGTH BYTE COUNTS
;       2575
;       2576          if EX_SUP_PRG ()                           ! ISSUE AN "EXECUTE SUPPLIED PRG" CMD
;       2577          then                                       ! STATUS BIT INDICATES ERROR
;       2578              begin                                  ! THEN
;       2579              ERRDF (24, EXE_SUP_ERR, 0);            !
;       2580
;       2581              if .RET_STATUS then DECODE ();
```

```
;        2582
;        2583                CKLOOP;
;        2584                RETRIES = TRUE;
;        2585                end;
;        2586
;        2587            H_SADD = .FREE_MEM_ADDR;                ! LO BYTE FREE HOST MEMORY ADDRESS
;        2588            H_EADD = 0;                             ! HIGH BYTE FREE MEMORY ADDRESS
;        2589            BUF_LENGTH = 1024;                      ! TOTAL FREE HOST MEMORY SIZE
;        2590            CMD_REF = 4;                            ! COMMAND REFERENCE 04
;        2591            BUF_DESCRPTR = H_SADD;                  ! DESCRIPTOR ADDRESS
;        2592            BYTE_COUNT = 06;                        ! TOTAL BYTES TO BE TRANSFER
;        2593
;        2594            if SEND_DATA ()                         ! ISSUE SEND DATA COMMAND
;        2595            then                                    ! STATUS BIT INDICATES ERROR
;        2596                begin                               ! THEN
;        2597                ERRDF (25, SND_DATA_ERR, 0);        !
;        2598
;        2599                if .RET_STATUS then DECODE ();
;        2600
;        2601                CKLOOP;
;        2602                RETRIES = TRUE;
;        2603                end;
;        2604
;        2605            CMD_REF = 5;
;        2606            BUF_DESCRPTR = TIP;                     ! CLEAN THE BUFFER
;        2607            BYTE_COUNT = 02;                        ! SET BYTE COUNTS = 2
;        2608
;        2609                if REC_DATA ()                      ! SENT A RECEIVE DATA COMMAND
;        2610                then                                ! STATUS BIT INDICATES ERROR
;        2611                    begin                           ! THEN
;        2612                    ERRDF (26, RE_DATA_ERR, 0);     !
;        2613
;        2614                    if .RET_STATUS then DECODE ();
;        2615
;        2616                    CKLOOP;
;        2617                    RETRIES = TRUE;
;        2618                    end;
;        2619
;        2620                if .TIP nequ %o'104'                ! IS REMOTE PROGRAM SENT DONE FLAG -
;        2621                then                                ! TO THE HOST
;        2622                    begin                           ! NO.  THEN
;        2623                    ERRDF (27, DMC_ERR, 0);         ! REPORT ERROR
;        2624                    CKLOOP;
;        2625                    RETRIES = TRUE;
;        2626                    end;
;        2627
;        2628                end;
;        2629
;        2630        if (.RETRIES) then DO_RETRIES ();
;        2631
;        2632        if (.NUM_RETRIES eqlu ZERO) then exitloop;
;        2633
;        2634        end;
;        2635
;        2636    return;
;        2637    ENDTST;
```

```
                                                .SBTTL  $T9 TEST SECTION
000000  032767  000001  000000G       $T9:    BIT     #1,SWP.TRACE                    ;                               2545
000006  001407                                  BEQ     1$
000010  012746  000000G                         MOV     #DBM15,-(SP)
000014  012746  000001                          MOV     #1,-(SP)
000020  010600                                  MOV     SP,R0                           ; SP,*
000022  104417                                  TRAP    17
000024  022626                                  CMP     (SP)+,(SP)+
000026  005067  000000G               1$:     CLR     NUM.RETRIES                     ;                               2547
000032  026767  000000G 000000G       2$:     CMP     NUM.RETRIES,SWP.RETRIES         ;                               2549
000040  101401                                  BLOS    3$
000042  000207                                  RTS     PC
000044  004767  000000G               3$:     JSR     PC,AZTEC.READY                  ;                               2552
000050  006000                                  ROR     R0
000052  103022                                  BCC     6$
000054  104455                                  TRAP    55                              ;                               2555
000056  000027                                  .WORD   27
000060  000000G                                 .WORD   AZT.READY.ERR
000062  000000                                  .WORD   0
000064  032767  000001  000000G                BIT     #1,RET.STATUS                   ;                               2557
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$:     MOV     #1,RETRIES                      ;                               2560
000116  000573                                  BR      14$                             ;                               2552
000120  016767  000000G 000000G       6$:     MOV     FREE.MEM.ADDR,TEMP              ;                               2564
000126  005000                                  CLR     R0                              ; COUNT                         2566
000130  012777  125252  000000G       7$:     MOV     #-52526,@TEMP                   ;                               2568
000136  062767  000002  000000G               ADD     #2,TEMP                         ;                               2569
000144  005200                                  INC     R0                              ; COUNT                         2566
000146  020027  002000                          CMP     R0,#2000                        ; COUNT,*
000152  101766                                  BLOS    7$
000154  012767  000003  000000G               MOV     #3,CMD.REF                      ;                               2572
000162  012767  000000G 000000G               MOV     #DM.09,BUF.DESCRPTR             ;                               2573
000170  012767  000272  000000G               MOV     #272,BYTE.COUNT                 ;                               2574
000176  004767  000000G                       JSR     PC,EX.SUP.PRG                   ;                               2576
000202  006000                                  ROR     R0
000204  103020                                  BCC     9$
000206  104455                                  TRAP    55                              ;                               2579
000210  000030                                  .WORD   30
000212  000000G                                 .WORD   EXE.SUP.ERR
000214  000000                                  .WORD   0
000216  032767  000001  000000G                BIT     #1,RET.STATUS                   ;                               2581
000224  001402                                  BEQ     8$
000226  004767  000000G                         JSR     PC,DECODE
000232  104465                       8$:        TRAP    65
000234  006000                                  ROR     R0
000236  103536                                  BLO     16$
000240  012767  000001  000000G               MOV     #1,RETRIES                      ;                               2584
000246  016767  000000G 000000G       9$:     MOV     FREE.MEM.ADDR,H.SADD            ;                               2587
000254  005067  000000G                       CLR     H.EADD                          ;                               2588
000260  012767  002000  000000G               MOV     #2000,BUF.LENGTH                ;                               2589
000266  012767  000004  000000G               MOV     #4,CMD.REF                      ;                               2590
```

```
000274  012767  000000G 000000G          MOV     #H.SADD,BUF.DESCRPTR            ;                       2591
000302  012767  000006  000000G          MOV     #6,BYTE.COUNT                   ;                       2592
000310  004767  000000G                  JSR     PC,SEND.DATA                    ;                       2594
000314  006000                          ROR     R0
000316  103020                          BCC     11$
000320  104455                          TRAP    55                              ;                       2597
000322  000031                          .WORD   31
000324  000000G                         .WORD   SND.DATA.ERR
000326  000000                          .WORD   0
000330  032767  000001  000000G         BIT     #1,RET.STATUS                   ;                       2599
000336  001402                          BEQ     10$
000340  004767  000000G                  JSR     PC,DECODE
000344  104465                  10$:    TRAP    65
000346  006000                          ROR     R0
000350  103471                          BLO     16$
000352  012767  000001  000000G         MOV     #1,RETRIES                      ;                       2602
000360  012767  000005  000000G  11$:    MOV     #5,CMD.REF                      ;                       2605
000366  012767  000000G 000000G         MOV     #TIP,BUF.DESCRPTR               ;                       2606
000374  012767  000002  000000G         MOV     #2,BYTE.COUNT                   ;                       2607
000402  004767  000000G                  JSR     PC,REC.DATA                     ;                       2609
000406  006000                          ROR     R0
000410  103020                          BCC     13$
000412  104455                          TRAP    55                              ;                       2612
000414  000032                          .WORD   32
000416  000000G                         .WORD   RE.DATA.ERR
000420  000000                          .WORD   0
000422  032767  000001  000000G         BIT     #1,RET.STATUS                   ;                       2614
000430  001402                          BEQ     12$
000432  004767  000000G                  JSR     PC,DECODE
000436  104465                  12$:    TRAP    65
000440  006000                          ROR     R0
000442  103434                          BLO     16$
000444  012767  000001  000000G         MOV     #1,RETRIES                      ;                       2617
000452  026727  000000G 000104  13$:    CMP     TIP,#104                        ;                       2620
000460  001412                          BEQ     14$
000462  104455                          TRAP    55                              ;                       2623
000464  000033                          .WORD   33
000466  000000G                         .WORD   DMC.ERR
000470  000000                          .WORD   0
000472  104465                          TRAP    65
000474  006000                          ROR     R0
000476  103416                          BLO     16$
000500  012767  000001  000000G         MOV     #1,RETRIES                      ;                       2625
000506  032767  000001  000000G  14$:    BIT     #1,RETRIES                      ;                       2630
000514  001402                          BEQ     15$
000516  004767  000000G                  JSR     PC,DO.RETRIES
000522  005767  000000G          15$:    TST     NUM.RETRIES                     ;                       2632
000526  001402                          BEQ     16$
000530  000167  177276                  JMP     2$
000534  000207                  16$:    RTS     PC                              ;                       2523

; Routine Size:  175 words,     Routine Base:  AC$CODE + 5552
; Maximum stack depth per invocation:  4 words
```

```
                                        .SBTTL  T9 TEST SECTION
000000  004767  177236                T9::
000000                                  1$:     JSR     PC,$T9                  ;                       2636
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

; Routine Size:  6 words,      Routine Base:  AC$CODE + 6310
; Maximum stack depth per invocation:  2 words


:          2638  !
:          2639  BGNTST;
:          2640
:          2641  !++
:          2642  ! TEST #10 - NONEXISTENT MEMORY TEST
:          2643  !
:          2644  ! DESCRIPTION:
:          2645  !
:          2646  !       THIS "DIAGNOSTIC MACHINE" PROGRAM WILL ATTEMPT TO READ THE FIRST
:          2647  !       ADDRESS OF THE I/O PAGE OF THE HOST CPU.  THIS LOCATION IS RESERVED
:          2648  !       FOR DIAGNOSTICS AND A NXM SHOULD OCCUR.
:          2649  !
:          2650  !       IF THE CONTROLLER DOES NOT SEE THE NXM, THERE WILL BE A FRU CALLOUT
:          2651  !       OF THE ADAPTER CARD.
:          2652  !
:          2653  !       IF THE OPERATOR HAS SPECIFIED LOOP ON ERROR, LOOPING WILL BE FROM
:          2654  !       THE START OF THIS TEST.
:          2655  !--
:          2656
:          2657  if .SWP_TRACE then PRINTF (DBM16);                   ! TEST 10
:          2658
:          2659  NUM_RETRIES = ZERO;
:          2660
:          2661  while (.NUM_RETRIES lequ .SWP_RETRIES) do
:          2662      begin
:          2663      TIP = 0;                                          ! INIT TIP
:          2664
:          2665      if AZTEC_READY ()                                 ! GET AZTEC READY FOR OPERATION
:          2666      then
:          2667          begin
:          2668          ERRDF (28, AZT_READY_ERR, 0);                 !
:          2669
:          2670          if .RET_STATUS then DECODE ();
:          2671
:          2672          CKLOOP;
:          2673          RETRIES = TRUE;
:          2674          end
:          2675      else
:          2676          begin
:          2677          VEC_AD = 04;                                  ! SET INT. VECTOR ADDR. TO 4
:          2678          SETVEC (.VEC_AD, NXMI, PRI04);                !
:          2679          SET_INT_VECTOR ();                            ! SET THE VECTOR ADDR., SERVICE
:          2680          WRT_RC25 (RCSA, ONE);                         ! ROUTINE ADDR. AND INT. PRIORITY
:          2681          CMD_REF = 3;                                  ! COMMAND REFERENCE #
:          2682          BUF_DESCRPTR = DM_10;                         ! DMCODE STARTING ADDRESS
```

```
;        2683          BYTE_COUNT = 58*2;                          ! BYTE COUNTS
;        2684
;        2685          if EX_SUP_PRG ()                            ! ISSUE AN EXECUTE SUPPLIED CMD
;        2686          then                                        ! IF ERROR
;        2687              begin                                   ! THEN
;        2688              ERRDF (29, EXE_SUP_ERR, 0);             !
;        2689
;        2690              if .RET_STATUS then DECODE ();
;        2691
;        2692              CKLOOP;
;        2693              RETRIES = TRUE;
;        2694              end;
;        2695
;        2696      !
;        2697      ! WAIT FOR "DONE" SIGNAL FROM DM
;        2698      !
;        2699          CMD_REF = 4;                                ! COMMAND REFERENCE #
;        2700          BUF_DESCRPTR = TIP;                         ! CLEAN THE BUFFER
;        2701          BYTE_COUNT = 02;                            ! SET BYTE COUNTS = 2
;        2702
;        2703          if REC_DATA ()                              ! SENT A RECEIVE DATA COMMAND
;        2704          then                                        ! STATUS BIT INDICATES ERROR
;        2705              begin                                   ! THEN
;        2706              ERRDF (30, RE_DATA_ERR, 0);             !
;        2707
;        2708              if .RET_STATUS then DECODE ();
;        2709
;        2710              CKLOOP;
;        2711              RETRIES = TRUE;
;        2712              end;
;        2713
;        2714          if .TIP eqlu ZERO                           ! DID YOU GET SUCCESS FROM DM CODE?
;        2715          then                                        ! NO
;        2716              begin                                   ! THEN
;        2717              ERRDF (31, DMC_ERR, 0);                 ! REPORT ERROR
;        2718              CKLOOP;
;        2719              RETRIES = TRUE;
;        2720              end;
;        2721
;        2722          end;
;        2723
;        2724      if (.RETRIES) then DO_RETRIES ();
;        2725
;        2726      if (.NUM_RETRIES eqlu ZERO) then exitloop;
;        2727
;        2728      end;
;        2729
;        2730  return;
;        2731  ENDTST;


                                        .SBTTL  $T10 TEST SECTION
000000  010146                  $T10:   MOV     R1,-(SP)                ;                       2637
000002  032767  000001  000000G        BIT     #1,SWP.TRACE            ;                       2657
000010  001407                         BEQ     1$
000012  012746  000000G                MOV     #DBM16,-(SP)
000016  012746  000001                 MOV     #1,-(SP)
```

```
000022  010600                          MOV     SP,R0                           ; SP,*
000024  104417                          TRAP    17
000026  022626                          CMP     (SP)+,(SP)+
000030  005067  000000G          1$:    CLR     NUM.RETRIES                     ;                       2659
000034  026767  000000G 000000G  2$:    CMP     NUM.RETRIES,SWP.RETRIES         ;                       2661
000042  101402                          BLOS    3$
000044  000167  000422                  JMP     16$
000050  005067  000000G          3$:    CLR     TIP                             ;                       2663
000054  004767  000000G                  JSR     PC,AZTEC.READY                  ;                       2665
000060  006000                          ROR     R0
000062  103021                          BCC     5$
000064  104455                          TRAP    55                              ;                       2668
000066  000034                          .WORD   34
000070  000000G                         .WORD   AZT.READY.ERR
000072  000000                          .WORD   0
000074  032767  000001  000000G        BIT     #1,RET.STATUS                   ;                       2670
000102  001402                          BEQ     4$
000104  004767  000000G                  JSR     PC,DECODE
000110  104465                  4$:     TRAP    65
000112  006000                          ROR     R0
000114  103566                          BLO     16$
000116  012767  000001  000000G        MOV     #1,RETRIES                      ;                       2673
000124  000547                          BR      14$                             ;                       2665
000126  112767  000004  000000G  5$:    MOVB    #4,VEC.AD                       ;                       2677
000134  012746  000200                  MOV     #200,-(SP)                      ;                       2678
000140  012746  000000G                  MOV     #NXMI,-(SP)
000144  005046                          CLR     -(SP)
000146  116716  000000G                  MOVB    VEC.AD,(SP)
000152  012746  000003                  MOV     #3,-(SP)
000156  104437                          TRAP    37
000160  004767  000000G                  JSR     PC,SET.INT.VECTOR               ;                       2679
000164  012701  000001                  MOV     #1,R1                           ; *,RCM.REG             2680
000170  016700  000000G                  MOV     RC25.ADDR,R0
000174  010160  000002                  MOV     R1,2(R0)                        ; RCM.REG,*
000200  012767  000003  000000G        MOV     #3,CMD.REF                      ;                       2681
000206  012767  000000G 000000G        MOV     #DM.10,BUF.DESCRPTR             ;                       2682
000214  012767  000164  000000G        MOV     #164,BYTE.COUNT                 ;                       2683
000222  004767  000000G                  JSR     PC,EX.SUP.PRG                   ;                       2685
000226  006000                          ROR     R0
000230  103023                          BCC     8$
000232  104455                          TRAP    55                              ;                       2688
000234  000035                          .WORD   35
000236  000000G                         .WORD   EXE.SUP.ERR
000240  000000                          .WORD   0
000242  032767  000001  000000G        BIT     #1,RET.STATUS                   ;                       2690
000250  001402                          BEQ     6$
000252  004767  000000G                  JSR     PC,DECODE
000256  104465                  6$:     TRAP    65
000260  006000                          ROR     R0
000262  103003                          BCC     7$
000264  062706  000010                  ADD     #10,SP
000270  000500                          BR      16$
000272  012767  000001  000000G  7$:    MOV     #1,RETRIES                      ;                       2693
000300  012767  000004  000000G  8$:    MOV     #4,CMD.REF                      ;                       2699
000306  012767  000000G 000000G        MOV     #TIP,BUF.DESCRPTR               ;                       2700
000314  012767  000002  000000G        MOV     #2,BYTE.COUNT                   ;                       2701
000322  004767  000000G                  JSR     PC,REC.DATA                     ;                       2703
```

```
000326  006000                                  ROR     R0
000330  103023                                  BCC     11$
000332  104455                                  TRAP    55                              ;                       2706
000334  000036                                  .WORD   36
000336  000000G                                 .WORD   RE.DATA.ERR
000340  000000                                  .WORD   0
000342  032767  000001  000000G                BIT     #1,RET.STATUS                   ;                       2708
000350  001402                                  BEQ     9$
000352  004767  000000G                         JSR     PC,DECODE
000356  104465                          9$:     TRAP    65
000360  006000                                  ROR     R0
000362  103003                                  BCC     10$
000364  062706  000010                          ADD     #10,SP
000370  000440                                  BR      16$
000372  012767  000001  000000G        10$:    MOV     #1,RETRIES                      ;                       2711
000400  005767  000000G                11$:    TST     TIP                             ;                       2714
000404  001015                                  BNE     13$
000406  104455                                  TRAP    55                              ;                       2717
000410  000037                                  .WORD   37
000412  000000G                                 .WORD   DMC.ERR
000414  000000                                  .WORD   0
000416  104465                                  TRAP    65
000420  006000                                  ROR     R0
000422  103003                                  BCC     12$
000424  062706  000010                          ADD     #10,SP
000430  000420                                  BR      16$
000432  012767  000001  000000G        12$:    MOV     #1,RETRIES                      ;                       2719
000440  062706  000010                13$:    ADD     #10,SP                          ;                       2676
000444  032767  000001  000000G        14$:    BIT     #1,RETRIES                      ;                       2724
000452  001402                                  BEQ     15$
000454  004767  000000G                         JSR     PC,DO.RETRIES
000460  005767  000000G                15$:    TST     NUM.RETRIES                     ;                       2726
000464  001402                                  BEQ     16$
000466  000167  177342                         JMP     2$
000472  012601                          16$:    MOV     (SP)+,R1                        ;                       2637
000474  000207                                  RTS     PC

; Routine Size:  159 words,     Routine Base:  AC$CODE + 6324
; Maximum stack depth per invocation:  7 words



                                                .SBTTL  T10 TEST SECTION
000000  004767  177276                  T10::
000000                                  1$:     JSR     PC,$T10                         ;                       2730
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

; Routine Size:  6 words,       Routine Base:  AC$CODE + 7022
; Maximum stack depth per invocation:  2 words


;       2732  !
;       2733  BGNTST;
```

```

!++
! TEST #11 - BUS ADDRESSING/DATA TEST A
!
! DESCRIPTION:
!
!       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 AND DATA
!       EXPECTED ARE REPORTED.
!
!
!--

if .SWP_TRACE then PRINTF (DBI'17);                 ! TEST 11

NUM_RETRIES = ZERO;

while (.NUM_RETRIES lequ .SWP_RETRIES) do
    begin
    TIP = 11;

    if AZTEC_READY ()                               ! GET AZTEC READY FOR OPERATION
    then
        begin
        ERRDF (32, AZT_READY_ERR, 0);               !

        if .RET_STATUS then DECODE ();

        CKLOOP;
        RETRIES = TRUE;
        end
    else
        begin
!    SET_INT_VECTOR ();                             ! SET THE VECTOR ADDR., SERVICE
                                                    ! ROUTINE ADDR. AND INT. PRIORITY
        CMD_REF = 3;                                ! COMMAND REFERENCE #
        BUF_DESCRPTR = DM_11;                       ! DMCODE STARTING ADDRESS
        BYTE_COUNT = 100*2;                         ! BYTE COUNTS

        if EX_SUP_PRG ()                            ! ISSUE AN EXECUTE SUPPLIED -
        then                                        ! IF STATUS BIT INDICATES ERROR
            begin                                   ! THEN
            ERRDF (33, EXE_SUP_ERR, 0);             !

            if .RET_STATUS then DECODE ();

            CKLOOP;
            RETRIES = TRUE;
            end;

        H_SADD = .FREE_MEM_ADDR;                    ! LO-BYTE FREE HOST MEMORY ADDRESS
```

```
;        2791          TEMP = .H_SADD;                          ! LOAD START ADDRESS FOR INIT
;        2792          BUF_LENGTH = .MEM_SIZ;                   ! TOTAL FREE HOST MEMORY SIZE
;        2793          H_EADD = .H_SADD - 2 + (.BUF_LENGTH*2);  ! END OF FREE MEM ADDRESS
;        2794          CMD_REF = 4;                             ! COMMAND REFERENCE 04
;        2795          BUF_DESCRPTR = H_SADD;                   ! DESCRIPTOR ADDRESS
;        2796          BYTE_COUNT = 06;                         ! TOTAL BYTES TO BE TRANSFER
;        2797    ! INITIALIZE MEMORY BUFFER WITH A PATTERN BEFORE
;        2798    ! ASKING DM CODE TO WRITE TO THE BUFFER
;        2799
;        2800          incru COUNT from .H_SADD to .H_EADD by 2 do
;        2801              begin
;        2802              .TEMP = %o'177777';
;        2803              TEMP = .TEMP + 2;
;        2804              end;
;        2805
;        2806          H_EADD = 0;                              ! HIGH BYTE FREE MEMORY ADDRESS
;        2807
;        2808          if SEND_DATA ()                          ! ISSUE SEND DATA COMMAND
;        2809          then                                     ! STATUS BIT INDICATES ERROR
;        2810              begin                                ! THEN
;        2811              ERRDF (34, SND_DATA_ERR, 0);         !
;        2812
;        2813              if .RET_STATUS then DECODE ();
;        2814
;        2815              CKLOOP;
;        2816              RETRIES = TRUE;
;        2817              end;
;        2818
;        2819          CMD_REF = 5;
;        2820          BUF_DESCRPTR = TIP;                      ! CLEAN THE BUFFER
;        2821          BYTE_COUNT = 02;                         ! SET BYTE COUNTS = 2
;        2822
;        2823          if REC_DATA ()                           ! SENT A RECEIVE DATA COMMAND
;        2824          then                                     ! STATUS BIT INDICATES ERROR
;        2825              begin                                ! THEN
;        2826              ERRDF (35, RE_DATA_ERR, 0);          !
;        2827
;        2828              if .RET_STATUS then DECODE ();
;        2829
;        2830              CKLOOP;
;        2831              RETRIES = TRUE;
;        2832              end;
;        2833
;        2834    ! EXAMINE THE FREE HOST MEMORY
;        2835    !
;        2836          TIP = 2;                                 ! ADDRESS CONTAIN OWN ADDRESS
;        2837
;        2838          if EXAM_DATA ()
;        2839          then
;        2840              begin
;        2841              ERRDF (36, BUFF_ERR, RC25$ERR_RPT);
;        2842              CKLOOP;
;        2843              RETRIES = TRUE;
;        2844              end;
;        2845
;        2846          end;
;        2847
```

```
;         2848      if (.RETRIES) then DO_RETRIES ();
;         2849
;         2850      if (.NUM_RETRIES eqlu ZERO) then exitloop;
;         2851
;         2852      end;
;         2853
;         2854  return;
;         2855  ENDTST;


                                        .SBTTL  $T11 TEST SECTION
000000  010146                  $i11:   MOV     R1,-(SP)                        ;                       2731
000002  032767  000001  000000G        BIT     #1,SWP.TRACE                    ;                       2753
000010  001407                          BEQ     1$
000012  012746  000000G                 MOV     #DBM17,-(SP)
000016  012746  000001                  MOV     #1,-(SP)
000022  010600                          MOV     SP,R0                           ; SP,*
000024  104417                          TRAP    17
000026  022626                          CMP     (SP)+,(SP)+
000030  005067  000000G          1$:     CLR     NUM.RETRIES                     ;                       2755
000034  026767  000000G 000000G  2$:     CMP     NUM.RETRIES,SWP.RETRIES         ;                       2757
000042  101402                          BLOS    3$
000044  000167  000544                  JMP     17$
000050  012767  000013  000000G  3$:     MOV     #13,TIP                         ;                       2759
000056  004767  000000G                 JSR     PC,AZTEC.READY                  ;                       2761
000062  006000                          ROR     R0
000064  103024                          BCC     6$
000066  104455                          TRAP    55                              ;                       2764
000070  000040                          .WORD   40
000072  000000G                         .WORD   AZT.READY.ERR
000074  000000                          .WORD   0
000076  032767  000001  000000G        BIT     #1,RET.STATUS                   ;                       2766
000104  001402                          BEQ     4$
000106  004767  000000G                 JSR     PC,DECODE
000112  104465                  4$:     TRAP    65
000114  006000                          ROR     R0
000116  103002                          BHIS    5$
000120  000167  000470                  JMP     17$
000124  012767  000001  000000G  5$:     MOV     #1,RETRIES                      ;                       2769
000132  000167  000430                  JMP     15$                             ;                       2761
000136  012767  000003  000000G  6$:     MOV     #3,CMD.REF                      ;                       2775
000144  012767  000000G 000000G        MOV     #DM.11,BUF.DESCRPTR             ;                       2776
000152  012767  000310  000000G        MOV     #310,BYTE.COUNT                 ;                       2777
000160  004767  000000G                 JSR     PC,EX.SUP.PRG                   ;                       2779
000164  006000                          ROR     R0
000166  103020                          BCC     8$
000170  104455                          TRAP    55                              ;                       2782
000172  000041                          .WORD   41
000174  000000G                         .WORD   EXE.SUP.ERR
000176  000000                          .WORD   0
000200  032767  000001  000000G        BIT     #1,RET.STATUS                   ;                       2784
000206  001402                          BEQ     7$
000210  004767  000000G                 JSR     PC,DECODE
000214  104465                  7$:     TRAP    65
000216  006000                          ROR     R0
000220  103575                          BLO     17$
000222  012767  000001  000000G        MOV     #1,RETRIES                      ;                       2787
```

```
000230  016767  000000G 000000G        8$:     MOV     FREE.MEM.ADDR,H.SADD            ;                               2790
000236  016767  000000G 000000G                MOV     H.SADD,TEMP                     ;                               2791
000244  016767  000000G 000000G                MOV     MEM.SIZ,BUF.LENGTH              ;                               2792
000252  016700  000000G                        MOV     BUF.LENGTH,R0                   ;                               2793
000256  006300                                ASL     R0
000260  066700  000000G                        ADD     H.SADD,R0
000264  010067  000000G                        MOV     R0,H.EADD
000270  162767  000002  000000G               SUB     #2,H.EADD
000276  012767  000004  000000G               MOV     #4,CMD.REF                      ;                               2794
000304  012767  000000G 000000G               MOV     #H.SADD,BUF.DESCRPTR            ;                               2795
000312  012767  000006  000000G               MOV     #6,BYTE.COUNT                   ;                               2796
000320  016701  000000G                        MOV     H.EADD,R1                       ;                               2800
000324  016700  000000G                        MOV     H.SADD,R0                       ; *,COUNT
000330  000410                                BR      10$
000332  012777  177777  000000G       9$:     MOV     #-1,@TEMP                       ;                               2802
000340  062767  000002  000000G               ADD     #2,TEMP                         ;                               2803
000346  062700  000002                        ADD     #2,R0                           ; *,COUNT                       2800
000352  020001                        10$:    CMP     R0,R1                           ; COUNT,*
000354  101766                                BLOS    9$
000356  005067  000000G                        CLR     H.EADD                          ;                               2806
000362  004767  000000G                        JSR     PC,SEND.DATA                    ;                               2808
000366  006000                                ROR     R0
000370  103020                                BCC     12$
000372  104455                                TRAP    55                              ;                               2811
000374  000042                                .WORD   42
000376  000000G                               .WORD   SND.DATA.ERR
000400  000000                                .WORD   0
000402  032767  000001  000000G               BIT     #1,RET.STATUS                   ;                               2813
000410  001402                                BEQ     11$
000412  004767  000000G                        JSR     PC,DECODE
000416  104465                        11$:    TRAP    65
000420  006000                                ROR     R0
000422  103474                                BLO     17$
000424  012767  000001  000000G               MOV     #1,RETRIES                      ;                               2816
000432  012767  000005  000000G       12$:    MOV     #5,CMD.REF                      ;                               2819
000440  012767  000000G 000000G               MOV     #TIP,BUF.DESCRPTR               ;                               2820
000446  012767  000002  000000G               MOV     #2,BYTE.COUNT                   ;                               2821
000454  004767  000000G                        JSR     PC,REC.DATA                     ;                               2823
000460  006000                                ROR     R0
000462  103020                                BCC     14$
000464  104455                                TRAP    55                              ;                               2826
000466  000043                                .WORD   43
000470  000000G                               .WORD   RE.DATA.ERR
000472  000000                                .WORD   0
000474  032767  000001  000000G               BIT     #1,RET.STATUS                   ;                               2828
000502  001402                                BEQ     13$
000504  004767  000000G                        JSR     PC,DECODE
000510  104465                        13$:    TRAP    65
000512  006000                                ROR     R0
000514  103437                                BLO     17$
000516  012767  000001  000000G               MOV     #1,RETRIES                      ;                               2831
000524  012767  000002  000000G       14$:    MOV     #2,TIP                          ;                               2836
000532  004767  000000G                        JSR     PC,EXAM.DATA                    ;                               2838
000536  006000                                ROR     R0
000540  103012                                BCC     15$
000542  104455                                TRAP    55                              ;                               2841
000544  000044                                .WORD   44
```

```
000546  000000G                                 .WORD   BUFF.ERR
000550  000000G                                 .WORD   RC25$ERR.RPT
000552  104465                                  TRAP    65
000554  006000                                  ROR     R0
000556  103416                                  BLO     17$
000560  012767  000001  000000G                MOV     #1,RETRIES                      ;                       2843
000566  032767  000001  000000G        15$:    BIT     #1,RETRIES                      ;                       2848
000574  001402                                  BEQ     16$
000576  004767  000000G                         JSR     PC,DO.RETRIES
000602  005767  000000G                16$:    TST     NUM.RETRIES                     ;                       2850
000606  001402                                  BEQ     17$
000610  000167  177220                         JMP     2$
000614  012601                          17$:    MOV     (SP)+,R1                        ;                       2731
000616  000207                                  RTS     PC

; Routine Size:  200 words,     Routine Base:  AC$CODE + 7036
; Maximum stack depth per invocation:  5 words



                                                .SBTTL  T11 TEST SECTION
000000  004767  177154                  T11::
000000                                  1$:     JSR     PC,$T11                         ;                       2854
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

; Routine Size:  6 words,       Routine Base:  AC$CODE + 7656
; Maximum stack depth per invocation:  2 words

;          2856  !<BLF/PAGE>
```

```
2857  !
2858  BGNTST;
2859
2860  !++
2861  ! TEST #12 - BUS ADDRESSING/DATA TEST B
2862  !
2863  ! DESCRIPTION:
2864  !
2865  !       THIS TEST FIRST BRINGS AZTEC DRIVE READY AND ONLINE AND THEN
2866  !       LOADS DM_12 PROGRAM VECTOR TO PORT CONTROLLER MEMORY. THEN
2867  !       DOES THE FOLLOWING:
2868  !
2869  !               A.  GIVE FREE MEMORY ADDRESS AND BUFFER SIZE TO DM CODE
2870  !                   AND ASK DM CODE WRITE A PATTERN OF ONE'S COMPLEMENT
2871  !                   OF ADDRESS AT THE ADDRESS AND EXPECTS TO RECEIVE
2872  !                   SUCCESS OR FAILURE CODE FROM DM PROGRAM. THEN CHECKS
2873  !                   MEMORY BUFFER FOR THE EXPECTED PATTERN AND REPORTS
2874  !                   ERROR IF ENCOUNTERED.
2875  !
2876  !               B.  IF SUCCESS, ASKS DM CODE TO WRITE TO MEMORY A PATTERN
2877  !                   OF ALL ONES AND CHECKS FOR THE PATTERN IN MEMORY.
2878  !
2879  !               C.  IF SUCCESS, ASKS DM CODE TO WRITE TO MEMORY A PATTERN
2880  !                   OF ALL ZEROES AND CHECKS FOR THE PATTERN IN MEMORY.
2881  !
2882  !               IF OPERATOR ASKS FOR RETRIES THE WHOLE TEST WILL BE RETRIED
2883  !               ONLY IF FAILURE ENCOUNTERED.
2884  !--
2885
2886  if .SWP_TRACE then PRINTF (DBM18);                    ! TEST 12
2887
2888  NUM_RETRIES = ZERO;
2889
2890  while (.NUM_RETRIES lequ .SWP_RETRIES) do
2891      begin
2892      TIP = 12;
2893
2894      if AZTEC_READY ()                                 ! GET AZTEC READY FOR OPERATION
2895      then
2896          begin
2897          ERRDF (37, AZT_READY_ERR, 0);                 !
2898
2899          if .RET_STATUS then DECODE ();
2900
2901          CKLOOP;
2902          RETRIES = TRUE;
2903          end
2904      else
2905          begin
2906  !       SET_INT_VECTOR ();                            ! SET THE VECTOR ADDR., SERVICE
2907                                                        ! ROUTINE ADDR. AND INT. PRIORITY
2908  !
2909  ! SEND DOWN LINE LOAD THE DM CODE AND EXECUTE THE DM PROGRAM WHICH IT WILL
2910  ! WRITE THE FREE HOST MEMORY WITH COMPLEMENT THE TESTING ADDRESS
2911  !
2912          CMD_REF = 3;                                  ! COMMAND REFERENCE #
2913          BUF_DESCRPTR = DM_12;                         ! DMCODE STARTING ADDRESS
```

```
2914            BYTE_COUNT = 202*2;                             ! BYTE COUNTS
2915
2916            if EX_SUP_PRG ()                                ! ISSUE AN EXECUTE SUPPLIED -
2917            then                                            ! IF STATUS BIT INDICATES ERROR
2918                begin                                       ! THEN
2919                ERRDF (38, EXE_SUP_ERR, 0);                 !
2920
2921                if .RET_STATUS then DECODE ();
2922
2923                CKLOOP;
2924                RETRIES = TRUE;
2925                end;
2926
2927            incru COUNT from 0 to 2 do
2928                begin
2929                H_SADD = .FREE_MEM_ADDR;                    ! LO-BYTE FREE HOST MEMORY ADDRESS
2930                TEMP = .H_SADD;
2931                BUF_LENGTH = .MEM_SIZ;                      ! TOTAL FREE HOST MEMORY SIZE
2932                H_EADD = .FREE_MEM_ADDR - 2 + .BUF_LENGTH*2;        ! END ADDRESS OF BUFFER
2933   !
2934   ! SENT FREE HOST MEMORY ADDRESS AND IT LENGTH TO DM PROGRAM
2935   !
2936                CMD_REF = 4;                                ! COMMAND REFERENCE 04
2937                BUF_DESCRPTR = H_SADD;                      ! DESCRIPTOR ADDRESS
2938                BYTE_COUNT = 06;                            ! TOTAL BYTES TO BE TRANSFER
2939   ! INITIALIZE MEMORY BUFFER WITH A PATTERN BEFORE
2940   ! ASKING DM CODE TO WRITE TO THE BUFFER
2941
2942                incru LOOP from .H_SADD to .H_EADD by 2 do
2943                    begin
2944                    .TEMP = %o'125252';
2945                    TEMP = .TEMP + 2;
2946                    end;
2947
2948                H_EADD = 0;                                 ! HIGH BYTE FREE MEM ADDRESS
2949
2950                if SEND_DATA ()                             ! ISSUE SEND DATA COMMAND
2951                then                                        ! STATUS BIT INDICATES ERROR
2952                    begin                                   ! THEN
2953                    ERRDF (39, SND_DATA_ERR, 0);            !
2954
2955                    if .RET_STATUS then DECODE ();
2956
2957                    CKLOOP;
2958                    RETRIES = TRUE;
2959                    end;
2960
2961   !
2962   ! WAIT FOR "DONE" SIGNAL FROM DM
2963   !
2964                CMD_REF = 5;                                ! COMMAND REFERENCE #
2965                BUF_DESCRPTR = TIP;                         ! CLEAN THE BUFFER
2966                BYTE_COUNT = 02;                            ! SET BYTE COUNTS = 2
2967
2968                if REC_DATA ()                              ! SENT A RECEIVE DATA COMMAND
2969                then                                        ! STATUS BIT INDICATES ERROR
2970                    begin                                   ! THEN
```

```
;        2971                    ERRDF (40, RE_DATA_ERR, 0);     ! REPORT ERROR
;        2972
;        2973                    if .RET_STATUS then DECODE ();
;        2974
;        2975                    CKLOOP;
;        2976                    RETRIES = TRUE;
;        2977                    end;
;        2978
;        2979                if .TIP nequ %o'104'                ! IF DM RETURNS FAILURE CODE
;        2980                then                                ! THEN ABORT DM PROGRAM
;        2981                    begin
;        2982                    ERRDF (41, DMC_ERR, 0);
;        2983                    RETRIES = TRUE;
;        2984                    CKLOOP;
;        2985                    exitloop;
;        2986                    end;
;        2987
;        2988    !
;        2989    ! EXAMINE THE FREE HOST MEMORY
;        2990    !
;        2991
;        2992                if .COUNT eqlu 0 then TIP = 1;      ! ADDRESS CONTAINS COMPLEMENT
;        2993                                                    ! OF ADDRESS
;        2994
;        2995                if .COUNT eqlu 1 then TIP = ALL_ONES;       ! MEMORY PATTERN SECOND TIME
;        2996
;        2997                if .COUNT eqlu 2 then TIP = ZERO;   ! MEMORY PATTERN THIRD TIME
;        2998
;        2999                if EXAM_DATA ()
;        3000                then
;        3001                    begin
;        3002                    ERRDF (42, BUFF_ERR, RC25$ERR_RPT);
;        3003                    CKLOOP;
;        3004                    RETRIES = TRUE;
;        3005                    end;
;        3006
;        3007    !
;        3008    ! SIGNAL DM TO CONTINUE TO EXECUTE THE PROGRAM
;        3009    !
;        3010                    end;                            ! ASK DM CODE TO CONT.
;        3011
;        3012            end;
;        3013
;        3014        if (.RETRIES) then DO_RETRIES ();
;        3015
;        3016        if (.NUM_RETRIES eqlu ZERO) then exitloop;
;        3017
;        3018        end;
;        3019
;        3020    return;
;        3021    ENDTST;


                                    .SBTTL  $T12 TEST SECTION
000000  004167  000000G     $T12:   JSR     R1,$SAVE2                   ;                   2855
000004  032767  000001  000000G     BIT     #1,SWP.TRACE                ;                   2886
```

```
000012  001407                                  BEQ     1$
000014  012746  000000G                         MOV     #DBM18,-(SP)
000020  012746  000001                          MOV     #1,-(SP)
000024  010600                                  MOV     SP,R0                           ; SP,*
000026  104417                                  TRAP    17
000030  022626                                  CMP     (SP)+,(SP)+
000032  005067  000000G                 1$:     CLR     NUM.RETRIES                     ;                       2888
000036  026767  000000G 000000G         2$:     CMP     NUM.RETRIES,SWP.RETRIES         ;                       2890
000044  101401                                  BLOS    3$
000046  000207                                  RTS     PC
000050  012767  000014  000000G         3$:     MOV     #14,TIP                         ;                       2892
000056  004767  000000G                         JSR     PC,AZTEC.READY                  ;                       2894
000062  006000                                  ROR     R0
000064  103023                                  BCC     6$
000066  104455                                  TRAP    55                              ;                       2897
000070  000045                                  .WORD   45
000072  000000G                                 .WORD   AZT.READY.ERR
000074  000000                                  .WORD   0
000076  032767  000001  000000G                 BIT     #1,RET.STATUS                   ;                       2899
000104  001402                                  BEQ     4$
000106  004767  000000G                         JSR     PC,DECODE
000112  104465                          4$:     TRAP    65
000114  006000                                  ROR     R0
000116  103001                                  BHIS    5$
000120  000207                                  RTS     PC
000122  012767  000001  000000G         5$:     MOV     #1,RETRIES                      ;                       2902
000130  000167  000540                          JMP     22$                             ;                       2894
000134  012767  000003  000000G         6$:     MOV     #3,CMD.REF                      ;                       2912
000142  012767  000000G 000000G                 MOV     #DM.12,BUF.DESCRPTR             ;                       2913
000150  012767  000624  000000G                 MOV     #624,BYTE.COUNT                 ;                       2914
000156  004767  000000G                         JSR     PC,EX.SUP.PRG                   ;                       2916
000162  006000                                  ROR     R0
000164  103021                                  BCC     9$
000166  104455                                  TRAP    55                              ;                       2919
000170  000046                                  .WORD   46
000172  000000G                                 .WORD   EXE.SUP.ERR
000174  000000                                  .WORD   0
000176  032767  000001  000000G                 BIT     #1,RET.STATUS                   ;                       2921
000204  001402                                  BEQ     7$
000206  004767  000000G                         JSR     PC,DECODE
000212  104465                          7$:     TRAP    65
000214  006000                                  ROR     R0
000216  103001                                  BHIS    8$
000220  000207                                  RTS     PC
000222  012767  000001  000000G         8$:     MOV     #1,RETRIES                      ;                       2924
000230  005002                          9$:     CLR     R2                              ; COUNT                 2927
000232  016767  000000G 000000G         10$:    MOV     FREE.MEM.ADDR,H.SADD            ;                       2929
000240  016767  000000G 000000G                 MOV     H.SADD,TEMP                     ;                       2930
000246  016767  000000G 000000G                 MOV     MEM.SIZ,BUF.LENGTH              ;                       2931
000254  016700  000000G                         MOV     BUF.LENGTH,R0                   ;                       2932
000260  006300                                  ASL     R0
000262  066700  000000G                         ADD     FREE.MEM.ADDR,R0
000266  010067  000000G                         MOV     R0,H.EADD
000272  162767  000002  000000G                 SUB     #2,H.EADD
000300  012767  000004  000000G                 MOV     #4,CMD.REF                      ;                       2936
000306  012767  000000G 000000G                 MOV     #H.SADD,BUF.DESCRPTR            ;                       2937
000314  012767  000006  000000G                 MOV     #6,BYTE.COUNT                   ;                       2938
```

```
000322  016701  000000G                        MOV     H.EADD,R1                       ;                                       2942
000326  016700  000000G                        MOV     H.SADD,R0                       ; *,LOOP
000332  000410                                 BR      12$
000334  012777  125252  000000G       11$:    MOV     #-52526,@TEMP                   ;                                       2944
000342  062767  000002  000000G               ADD     #2,TEMP                         ;                                       2945
000350  062700  000002                         ADD     #2,R0                           ; *,LOOP                                2942
000354  020001                       12$:    CMP     R0,R1                           ; LOOP,*
000356  101766                               BLOS    11$
000360  005067  000000G                        CLR     H.EADD                          ;                                       2948
000364  004767  000000G                        JSR     PC,SEND.DATA                    ;                                       2950
000370  006000                               ROR     R0
000372  103020                               BCC     14$
000374  104455                               TRAP    55                              ;                                       2953
000376  000047                               .WORD   47
000400  000000G                              .WORD   SND.DATA.ERR
000402  000000                               .WORD   0
000404  032767  000001  000000G               BIT     #1,RET.STATUS                   ;                                       2955
000412  001402                               BEQ     13$
000414  004767  000000G                        JSR     PC,DECODE
000420  104465                       13$:    TRAP    65
000422  006000                               ROR     R0
000424  103536                               BLO     24$
000426  012767  000001  000000G               MOV     #1,RETRIES                      ;                                       2958
000434  012767  000005  000000G       14$:    MOV     #5,CMD.REF                      ;                                       2964
000442  012767  000000G 000000G               MOV     #TIP,BUF.DESCRPTR               ;                                       2965
000450  012767  000002  000000G               MOV     #2,BYTE.COUNT                   ;                                       2966
000456  004767  000000G                        JSR     PC,REC.DATA                     ;                                       2968
000462  006000                               ROR     R0
000464  103020                               BCC     16$
000466  104455                               TRAP    55                              ;                                       2971
000470  000050                               .WORD   50
000472  000000G                              .WORD   RE.DATA.ERR
000474  000000                               .WORD   0
000476  032767  000001  000000G               BIT     #1,RET.STATUS                   ;                                       2973
000504  001402                               BEQ     15$
000506  004767  000000G                        JSR     PC,DECODE
000512  104465                       15$:    TRAP    65
000514  006000                               ROR     R0
000516  103501                               BLO     24$
000520  012767  000001  000000G               MOV     #1,RETRIES                      ;                                       2976
000526  026727  000000G 000104       16$:    CMP     TIP,#104                        ;                                       2979
000534  001413                               BEQ     17$
000536  104455                               TRAP    55                              ;                                       2982
000540  000051                               .WORD   51
000542  000000G                              .WORD   DMC.ERR
000544  000000                               .WORD   0
000546  012767  000001  000000G               MOV     #1,RETRIES                      ;                                       2983
000554  104465                               TRAP    65
000556  006000                               ROR     R0
000560  103045                               BCC     22$
000562  000207                               RTS     PC
000564  005702                       17$:    TST     R2                              : COUNT                                 2992
000566  001003                               BNE     18$
000570  012767  000001  000000G               MOV     #1,TIP
000576  020227  000001                 18$:    CMP     R2,#1                           ; COUNT,*                               2996
000602  001003                               BNE     19$
000604  012767  177777  000000G               MOV     #-1,TIP
```

```
000612  020227  000002                  19$:    CMP     R2,#2                   ; COUNT,*                       2998
000616  001002                                  BNE     20$
000620  005067  000000G                         CLR     TIP
000624  004767  000000G                 20$:    JSR     PC,EXAM.DATA            ;                               3000
000630  006000                                  ROR     R0
000632  103012                                  BCC     21$
000634  104455                                  TRAP    55                      ;                               3003
000636  000052                                  .WORD   52
000640  000000G                                 .WORD   BUFF.ERR
000642  000000G                                 .WORD   RC25$ERR.RPT
000644  104465                                  TRAP    65
000646  006000                                  ROR     R0
000650  103424                                  BLO     24$
000652  012767  000001  000000G                MOV     #1,RETRIES              ;                               3005
000660  005202                          21$:    INC     R2                      ; COUNT                         2927
000662  020227  000002                          CMP     R2,#2                   ; COUNT,*
000666  101002                                  BHI     22$
000670  000167  177336                         JMP     10$
000674  032767  000001  000000G         22$:    BIT     #1,RETRIES              ;                               3015
000702  001402                                  BEQ     23$
000704  004767  000000G                         JSR     PC,DO.RETRIES
000710  005767  000000G                 23$:    TST     NUM.RETRIES             ;                               3017
000714  001402                                  BEQ     24$
000716  000167  177114                         JMP     2$
000722  000207                          24$:    RTS     PC                      ;                               2855

; Routine Size:  234 words,     Routine Base:  AC$CODE + 7672
; Maximum stack depth per invocation:  7 words



                                        .SBTTL  T12 TEST SECTION
000000  004767  177050                  T12::
000000                                  1$:     JSR     PC,$T12                 ;                               3021
000004  104466                                  TRAP    66
000006  006000                                  ROR     R0
000010  103773                                  BLO     1$
000012  000207                                  RTS     PC

; Routine Size:  6 words,       Routine Base:  AC$CODE + 10616
; Maximum stack depth per invocation:  2 words


;         3023  end
;         3024
;         3025  eludom


;                                       OTS external references
                                                .GLOBL  $SAVE4, $SAVE2, BL$SHF, BL$DIV


;                                       PSECT SUMMARY
;
;
;       Psect Name                      Words     Attributes
```

```
;       $OWN$                          336       RW ,  D  ,  LCL,  REL,  CON
;       AC$CODE                       2253       RO ,  I  ,  LCL,  REL,  CON


;                         LIBRARY STATISTICS
;
;                                      -------- Symbols --------      Blocks
;      File                            Total    Loaded   Percent        Read
;
;  SPIDER$USERS:[LAKSHMANA.11REL.REAL]AZTEC0.L16;1
;                                        523       231        44          59


;                              COMMAND QUALIFIERS

;      BLISS /PDP11/LIST ZRCFA3.B16/EN:NOEIS

; Size:          2253 code + 336 data words
; Run Time:         00:53.6
; Elapsed Time:     02:45.0
; Memory Used:   296 pages
; Compilation Complete
```

```
;       0001  MODULE ZRCFA4 (%TITLE 'CZRCFA0 RC25 FR END TEST'
;       0002                 IDENT = 'V01.0',
;       0003                 ADDRESSING_MODE (RELATIVE)
;       0004                 ) =
;       0005  BEGIN
;       0006  !<BLF/LOWERCASE_KEY>
;       0007
;       0008  library 'AZTEC0';                              ! AZTEC LIBRARY
;       0009
;       0010  require 'BLSMAC.REQ';                          ! DIAGNOSTIC SUPERVISR LIBRARY
;       1499  !
;       1500  !
;       1501  %sbttl 'DM PROGRAM'
;       1502
;       1503  !++
;       1504  !
;       1505  !       THIS MODULE CONTAINS DM CODE FOR SOME OF THE TESTS
;       1506  !       AS GLOBAL DATA. THE HOST PROGRAM WILL DOWN LINE LOAD
;       1507  !       THESE TESTS IN AZTEC CONTROLLER'S MEMORY FOR EXECUTION.
;       1508  !       THE DM CODE WAS FIRST ASSEMBLED AND LINKED UNDER RT
;       1509  !       AND THEN MADE AS VECTOR ARRAYS BY USING DMCONV.EXE
;       1510  !       THIS MODULE IS A COLLECTION OF ARRAYS FOR SPECIFIC
;       1511  !       TESTS.
;       1512  !--
;       1513
;       1514  !<BLF/PAGE>
```

```
;        1515  psect
;        1516      global = DM$CODE(nowrite, noexecute, global, concatenate);
;        1517
```

```
;          1518  %sbttl 'DM CODE DOWN LINE LOAD TEST'
;          1519
;          1520  global
;          1521      DM_09 : vector [93, word] preset (
;          1522  [0]      =  %o'000270',  ! THIS IS THE DM PROGRAM BYTE COUNT.
;          1523  [1]      =  %o'000000',
;          1524  [2]      =  %o'000000',  ! THIS IS THE DM OVERLAY BYTE COUNT.
;          1525  [3]      =  %o'000000',
;          1526  [4]      =  %o'042524',  ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
;          1527  [5]      =  %o'052123',  ! PROGRAM NAME IS 'TEST09'
;          1528  [6]      =  %o'034460',
;          1529  [7]      =  %o'000000',  ! THIS IS THE PROGRAM VERSION
;          1530  [8]      =  %o'126411',  ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
;          1531  [9]      =  %o'000000',
;          1532  [10]     =  %o'000000',
;          1533  [11]     =  %o'000000',
;          1534  [12]     =  %o'000000',
;          1535  [13]     =  %o'000000',
;          1536  [14]     =  %o'000000',
;          1537  [15]     =  %o'000000',
;          1538  [16]     =  %o'104206',  ! DM CODE STARTS HERE
;          1539  [17]     =  %o'003051',
;          1540  [18]     =  %o'114000',
;          1541  [19]     =  %o'003037',
;          1542  [20]     =  %o'104207',
;          1543  [21]     =  %o'003032',
;          1544  [22]     =  %o'104201',
;          1545  [23]     =  %o'000003',
;          1546  [24]     =  %o'060023',
;          1547  [25]     =  %o'103207',
;          1548  [26]     =  %o'177740',
;          1549  [27]     =  %o'115007',
;          1550  [28]     =  %o'012756',
;          1551  [29]     =  %o'003003',
;          1552  [30]     =  %o'114000',
;          1553  [31]     =  %o'003052',
;          1554  [32]     =  %o'104307',
;          1555  [33]     =  %o'003032',
;          1556  [34]     =  %o'104301',
;          1557  [35]     =  %o'003033',
;          1558  [36]     =  %o'104302',
;          1559  [37]     =  %o'003034',
;          1560  [38]     =  %o'104203',
;          1561  [39]     =  %o'003052',
;          1562  [40]     =  %o'060020',
;          1563  [41]     =  %o'103207',
;          1564  [42]     =  %o'177740',
;          1565  [43]     =  %o'115007',
;          1566  [44]     =  %o'013007',
;          1567  [45]     =  %o'115400',
;          1568  [46]     =  %o'003037',
;          1569  [47]     =  %o'106300',
;          1570  [48]     =  %o'003035',
;          1571  [49]     =  %o'003037',
;          1572  [50]     =  %o'032756',
;          1573  [51]     =  %o'104200',
;          1574  [52]     =  %o'000106',
```

```
;          1575  [53]    =  %o'003040',
;          1576  [54]    =  %o'003024',
;          1577  [55]    =  %o'104207',
;          1578  [56]    =  %o'003052',
;          1579  [57]    =  %o'104201',
;          1580  [58]    =  %o'125252',
;          1581  [59]    =  %o'104302',
;          1582  [60]    =  %o'003034',
;          1583  [61]    =  %o'106271',
;          1584  [62]    =  %o'053003',
;          1585  [63]    =  %o'117402',
;          1586  [64]    =  %o'053015',
;          1587  [65]    =  %o'104200',
;          1588  [66]    =  %o'000104',
;          1589  [67]    =  %o'003040',
;          1590  [68]    =  %o'104207',
;          1591  [69]    =  %o'003040',
;          1592  [70]    =  %o'104201',
;          1593  [71]    =  %o'000001',
;          1594  [72]    =  %o'060022',
;          1595  [73]    =  %o'060010',
;          1596  [74]    =  %o'000000',
;          1597  [75]    =  %o'000000',
;          1598  [76]    =  %o'000000',
;          1599  [77]    =  %o'000012',
;          1600  [78]    =  %o'000000',
;          1601  [79]    =  %o'000000',
;          1602  [80]    =  %o'000000',
;          1603  [81]    =  %o'000000',
;          1604  [82]    =  %o'000000',
;          1605  [83]    =  %o'000000',
;          1606  [84]    =  %o'000000',
;          1607  [85]    =  %o'000000',
;          1608  [86]    =  %o'000000',
;          1609  [87]    =  %o'000000',
;          1610  [88]    =  %o'000000',
;          1611  [89]    =  %o'000000',
;          1612  [90]    =  %o'000000',
;          1613  [91]    =  %o'144423',
;          1614  [92]    =  %o'000000');
;          1615
```

```
;       1616  %sbttl 'NONEXISTENT MEMORY TEST'
;       1617
;       1618  global
;       1619      DM_10 : vector [58, word] preset (
;       1620  [0]     =  %o'000162',  ! THIS IS THE DM PROGRAM BYTE COUNT.
;       1621  [1]     =  %o'000000',
;       1622  [2]     =  %o'000000',  ! THIS IS THE DM OVERLAY BYTE COUNT.
;       1623  [3]     =  %o'000000',
;       1624  [4]     =  %o'042524',  ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
;       1625  [5]     =  %o'052123',  ! PROGRAM NAME IS 'TEST10'
;       1626  [6]     =  %o'030061',
;       1627  [7]     =  %o'000000',  ! THIS IS THE PROGRAM VERSION
;       1628  [8]     =  %o'126411',  ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
;       1629  [9]     =  %o'000000',
;       1630  [10]    =  %o'000000',
;       1631  [11]    =  %o'000000',
;       1632  [12]    =  %o'000000',
;       1633  [13]    =  %o'000000',
;       1634  [14]    =  %o'000000',
;       1635  [15]    =  %o'000000',
;       1636  [16]    =  %o'104206',  ! DM CODE STARTS HERE
;       1637  [17]    =  %o'003007',
;       1638  [18]    =  %o'104207',
;       1639  [19]    =  %o'160000',
;       1640  [20]    =  %o'104201',
;       1641  [21]    =  %o'177777',
;       1642  [22]    =  %o'104202',
;       1643  [23]    =  %o'000001',
;       1644  [24]    =  %o'104203',
;       1645  [25]    =  %o'003500',
;       1646  [26]    =  %o'060021',
;       1647  [27]    =  %o'103207',
;       1648  [28]    =  %o'177740',
;       1649  [29]    =  %o'104070',
;       1650  [30]    =  %o'002765',
;       1651  [31]    =  %o'104207',
;       1652  [32]    =  %o'002765',
;       1653  [33]    =  %o'104201',
;       1654  [34]    =  %o'000001',
;       1655  [35]    =  %o'060022',
;       1656  [36]    =  %o'060010',
;       1657  [37]    =  %o'000000',
;       1658  [38]    =  %o'000000',
;       1659  [39]    =  %o'000000',
;       1660  [40]    =  %o'000000',
;       1661  [41]    =  %o'000000',
;       1662  [42]    =  %o'000000',
;       1663  [43]    =  %o'000000',
;       1664  [44]    =  %o'000000',
;       1665  [45]    =  %o'000000',
;       1666  [46]    =  %o'000000',
;       1667  [47]    =  %o'000000',
;       1668  [48]    =  %o'000000',
;       1669  [49]    =  %o'000000',
;       1670  [50]    =  %o'000000',
;       1671  [51]    =  %o'000000',
;       1672  [52]    =  %o'000000',
```

```
;         1673  [53]    =  %o'000000',
;         1674  [54]    =  %o'000000',
;         1675  [55]    =  %o'000000',
;         1676  [56]    =  %o'030037',
;         1677  [57]    =  %o'000000');
;         1678
```

```
;          1679  %sbttl 'BUS ADDRESSING/DATA TEST A'
;          1680
;          1681  global
;          1682      DM_11 : vector [100, word] preset (
;          1683  [0]     =  %o'000306',  ! THIS IS THE DM PROGRAM BYTE COUNT.
;          1684  [1]     =  %o'000000',
;          1685  [2]     =  %o'000000',  ! THIS IS THE DM OVERLAY BYTE COUNT.
;          1686  [3]     =  %o'000000',
;          1687  [4]     =  %o'042524',  ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
;          1688  [5]     =  %o'052123',  ! PROGRAM NAME IS 'TEST11'
;          1689  [6]     =  %o'030461',
;          1690  [7]     =  %o'000000',  ! THIS IS THE PROGRAM VERSION
;          1691  [8]     =  %o'126411',  ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
;          1692  [9]     =  %o'000000',
;          1693  [10]    =  %o'000000',
;          1694  [11]    =  %o'000000',
;          1695  [12]    =  %o'000000',
;          1696  [13]    =  %o'000000',
;          1697  [14]    =  %o'000000',
;          1698  [15]    =  %o'000000',
;          1699  [16]    =  %o'104206',  ! DM CODE STARTS HERE
;          1700  [17]    =  %o'003061',
;          1701  [18]    =  %o'104207',
;          1702  [19]    =  %o'003040',
;          1703  [20]    =  %o'104201',
;          1704  [21]    =  %o'000003',
;          1705  [22]    =  %o'060023',
;          1706  [23]    =  %o'103207',
;          1707  [24]    =  %o'177740',
;          1708  [25]    =  %o'115007',
;          1709  [26]    =  %o'012754',
;          1710  [27]    =  %o'003023',
;          1711  [28]    =  %o'104200',
;          1712  [29]    =  %o'000001',
;          1713  [30]    =  %o'003043',
;          1714  [31]    =  %o'104300',
;          1715  [32]    =  %o'003040',
;          1716  [33]    =  %o'003044',
;          1717  [34]    =  %o'104304',
;          1718  [35]    =  %o'003042',
;          1719  [36]    =  %o'114000',
;          1720  [37]    =  %o'003046',
;          1721  [38]    =  %o'104307',
;          1722  [39]    =  %o'003040',
;          1723  [40]    =  %o'104301',
;          1724  [41]    =  %o'003041',
;          1725  [42]    =  %o'104302',
;          1726  [43]    =  %o'003043',
;          1727  [44]    =  %o'104203',
;          1728  [45]    =  %o'003044',
;          1729  [46]    =  %o'060021',
;          1730  [47]    =  %o'103207',
;          1731  [48]    =  %o'177740',
;          1732  [49]    =  %o'115007',
;          1733  [50]    =  %o'013012',
;          1734  [51]    =  %o'115400',
;          1735  [52]    =  %o'003046',
```

```
;       1736  [53]    =  %o'106200',
;       1737  [54]    =  %o'000012',
;       1738  [55]    =  %o'003046',
;       1739  [56]    =  %o'032766',
;       1740  [57]    =  %o'003023',
;       1741  [58]    =  %o'117404',
;       1742  [59]    =  %o'013027',
;       1743  [60]    =  %o'105200',
;       1744  [61]    =  %o'000002',
;       1745  [62]    =  %o'003040',
;       1746  [63]    =  %o'104300',
;       1747  [64]    =  %o'003040',
;       1748  [65]    =  %o'003044',
;       1749  [66]    =  %o'002764',
;       1750  [67]    =  %o'104200',
;       1751  [68]    =  %o'000106',
;       1752  [69]    =  %o'003045',
;       1753  [70]    =  %o'003032',
;       1754  [71]    =  %o'104200',
;       1755  [72]    =  %o'000104',
;       1756  [73]    =  %o'003045',
;       1757  [74]    =  %o'104207',
;       1758  [75]    =  %o'003045',
;       1759  [76]    =  %o'104201',
;       1760  [77]    =  %o'000001',
;       1761  [78]    =  %o'060022',
;       1762  [79]    =  %o'060010',
;       1763  [80]    =  %o'000000',
;       1764  [81]    =  %o'000000',
;       1765  [82]    =  %o'000000',
;       1766  [83]    =  %o'000000',
;       1767  [84]    =  %o'000000',
;       1768  [85]    =  %o'000000',
;       1769  [86]    =  %o'000000',
;       1770  [87]    =  %o'000000',
;       1771  [88]    =  %o'000000',
;       1772  [89]    =  %o'000000',
;       1773  [90]    =  %o'000000',
;       1774  [91]    =  %o'000000',
;       1775  [92]    =  %o'000000',
;       1776  [93]    =  %o'000000',
;       1777  [94]    =  %o'000000',
;       1778  [95]    =  %o'000000',
;       1779  [96]    =  %o'000000',
;       1780  [97]    =  %o'000000',
;       1781  [98]    =  %o'056247',
;       1782  [99]    =  %o'000000');
;       1783
```

```
;          1784  %sbttl 'BUS ADDRESSING/DATA TEST B'
;          1785
;          1786  global
;          1787      DM_12 : vector [202, word] preset (
;          1788  [0]     =  %o'000622',  ! THIS IS THE DM PROGRAM BYTE COUNT.
;          1789  [1]     =  %o'000000',
;          1790  [2]     =  %o'000000',  ! THIS IS THE DM OVERLAY BYTE COUNT.
;          1791  [3]     =  %o'000000',
;          1792  [4]     =  %o'042524',  ! NEXT 3 WORDS = PROGRAM NAME (ASCII)
;          1793  [5]     =  %o'052123',  ! PROGRAM NAME IS 'TEST12'
;          1794  [6]     =  %o'031061',
;          1795  [7]     =  %o'000000',  ! THIS IS THE PROGRAM VERSION
;          1796  [8]     =  %o'177411',  ! UPPER BYTE=TIME OUT VAL. LOWER = FLAGS
;          1797  [9]     =  %o'000000',
;          1798  [10]    =  %o'000000',
;          1799  [11]    =  %o'000000',
;          1800  [12]    =  %o'000000',
;          1801  [13]    =  %o'000000',
;          1802  [14]    =  %o'000000',
;          1803  [15]    =  %o'000000',
;          1804  [16]    =  %o'104206',  ! DM CODE STARTS HERE
;          1805  [17]    =  %o'002767',
;          1806  [18]    =  %o'003004',
;          1807  [19]    =  %o'000000',
;          1808  [20]    =  %o'000000',
;          1809  [21]    =  %o'000000',
;          1810  [22]    =  %o'000000',
;          1811  [23]    =  %o'000000',
;          1812  [24]    =  %o'000000',
;          1813  [25]    =  %o'000000',
;          1814  [26]    =  %o'000000',
;          1815  [27]    =  %o'000000',
;          1816  [28]    =  %o'000000',
;          1817  [29]    =  %o'000000',
;          1818  [30]    =  %o'000000',
;          1819  [31]    =  %o'000000',
;          1820  [32]    =  %o'000000',
;          1821  [33]    =  %o'000000',
;          1822  [34]    =  %o'000000',
;          1823  [35]    =  %o'000000',
;          1824  [36]    =  %o'000000',
;          1825  [37]    =  %o'000000',
;          1826  [38]    =  %o'000000',
;          1827  [39]    =  %o'000000',
;          1828  [40]    =  %o'000000',
;          1829  [41]    =  %o'000000',
;          1830  [42]    =  %o'000000',
;          1831  [43]    =  %o'000104',
;          1832  [44]    =  %o'000106',
;          1833  [45]    =  %o'000000',
;          1834  [46]    =  %o'000000',
;          1835  [47]    =  %o'000000',
;          1836  [48]    =  %o'000000',
;          1837  [49]    =  %o'000000',
;          1838  [50]    =  %o'000000',
;          1839  [51]    =  %o'000000',
;          1840  [52]    =  %o'023016',
```

```
;          1841  [53]    =  %o'023031',
;          1842  [54]    =  %o'023210',
;          1843  [55]    =  %o'023120',
;          1844  [56]    =  %o'023126',
;          1845  [57]    =  %o'023210',
;          1846  [58]    =  %o'023120',
;          1847  [59]    =  %o'023155',
;          1848  [60]    =  %o'023210',
;          1849  [61]    =  %o'060010',
;          1850  [62]    =  %o'104207',
;          1851  [63]    =  %o'002770',
;          1852  [64]    =  %o'104201',
;          1853  [65]    =  %o'000003',
;          1854  [66]    =  %o'060023',
;          1855  [67]    =  %o'103207',
;          1856  [68]    =  %o'177740',
;          1857  [69]    =  %o'115007',
;          1858  [70]    =  %o'013030',
;          1859  [71]    =  %o'003203',
;          1860  [72]    =  %o'000000',
;          1861  [73]    =  %o'104300',
;          1862  [74]    =  %o'002770',
;          1863  [75]    =  %o'002777',
;          1864  [76]    =  %o'104300',
;          1865  [77]    =  %o'002771',
;          1866  [78]    =  %o'003000',
;          1867  [79]    =  %o'104301',
;          1868  [80]    =  %o'002772',
;          1869  [81]    =  %o'104207',
;          1870  [82]    =  %o'177777',
;          1871  [83]    =  %o'107307',
;          1872  [84]    =  %o'002777',
;          1873  [85]    =  %o'104070',
;          1874  [86]    =  %o'003002',
;          1875  [87]    =  %o'023063',
;          1876  [88]    =  %o'105200',
;          1877  [89]    =  %o'000002',
;          1878  [90]    =  %o'002777',
;          1879  [91]    =  %o'115000',
;          1880  [92]    =  %o'002777',
;          1881  [93]    =  %o'053060',
;          1882  [94]    =  %o'115400',
;          1883  [95]    =  %o'003000',
;          1884  [96]    =  %o'117401',
;          1885  [97]    =  %o'053041',
;          1886  [98]    =  %o'000000',
;          1887  [99]    =  %o'100467',
;          1888  [100]   =  %o'100461',
;          1889  [101]   =  %o'100462',
;          1890  [102]   =  %o'100463',
;          1891  [103]   =  %o'104307',
;          1892  [104]   =  %o'002777',
;          1893  [105]   =  %o'104301',
;          1894  [106]   =  %o'003000',
;          1895  [107]   =  %o'104202',
;          1896  [108]   =  %o'000001',
;          1897  [109]   =  %o'104203',
```

```
:          1898  [110]   =  %o'003002',
:          1899  [111]   =  %o'060021',
:          1900  [112]   =  %o'103207',
:          1901  [113]   =  %o'177740',
:          1902  [114]   =  %o'115007',
:          1903  [115]   =  %o'013113',
:          1904  [116]   =  %o'115400',
:          1905  [117]   =  %o'003003',
:          1906  [118]   =  %o'106200',
:          1907  [119]   =  %o'000012',
:          1908  [120]   =  %o'003003',
:          1909  [121]   =  %o'033067',
:          1910  [122]   =  %o'003203',
:          1911  [123]   =  %o'104263',
:          1912  [124]   =  %o'104262',
:          1913  [125]   =  %o'104261',
:          1914  [126]   =  %o'104267',
:          1915  [127]   =  %o'000000',
:          1916  [128]   =  %o'104207',
:          1917  [129]   =  %o'002775',
:          1918  [130]   =  %o'104201',
:          1919  [131]   =  %o'000001',
:          1920  [132]   =  %o'060023',
:          1921  [133]   =  %o'000000',
:          1922  [134]   =  %o'104300',
:          1923  [135]   =  %o'002770',
:          1924  [136]   =  %o'002777',
:          1925  [137]   =  %o'104300',
:          1926  [138]   =  %o'002771',
:          1927  [139]   =  %o'003000',
:          1928  [140]   =  %o'104301',
:          1929  [141]   =  %o'002772',
:          1930  [142]   =  %o'104200',
:          1931  [143]   =  %o'177777',
:          1932  [144]   =  %o'003002',
:          1933  [145]   =  %o'023063',
:          1934  [146]   =  %o'105200',
:          1935  [147]   =  %o'000002',
:          1936  [148]   =  %o'002777',
:          1937  [149]   =  %o'115000',
:          1938  [150]   =  %o'002777',
:          1939  [151]   =  %o'053152',
:          1940  [152]   =  %o'115400',
:          1941  [153]   =  %o'003000',
:          1942  [154]   =  %o'117401',
:          1943  [155]   =  %o'053141',
:          1944  [156]   =  %o'000000',
:          1945  [157]   =  %o'104300',
:          1946  [158]   =  %o'002770',
:          1947  [159]   =  %o'002777',
:          1948  [160]   =  %o'104300',
:          1949  [161]   =  %o'002771',
:          1950  [162]   =  %o'003000',
:          1951  [163]   =  %o'104301',
:          1952  [164]   =  %o'002772',
:          1953  [165]   =  %o'114000',
:          1954  [166]   =  %o'003002',
```

```
;          1955  [167]   =  %o'023063',
;          1956  [168]   =  %o'105200',
;          1957  [169]   =  %o'000002',
;          1958  [170]   =  %o'002777',
;          1959  [171]   =  %o'115000',
;          1960  [172]   =  %o'002777',
;          1961  [173]   =  %o'053200',
;          1962  [174]   =  %o'115400',
;          1963  [175]   =  %o'003000',
;          1964  [176]   =  %o'117401',
;          1965  [177]   =  %o'053167',
;          1966  [178]   =  %o'000000',
;          1967  [179]   =  %o'104300',
;          1968  [180]   =  %o'002774',
;          1969  [181]   =  %o'002776',
;          1970  [182]   =  %o'023215',
;          1971  [183]   =  %o'060010',
;          1972  [184]   =  %o'104300',
;          1973  [185]   =  %o'002773',
;          1974  [186]   =  %o'002776',
;          1975  [187]   =  %o'023215',
;          1976  [188]   =  %o'000000',
;          1977  [189]   =  %o'104207',
;          1978  [190]   =  %o'002776',
;          1979  [191]   =  %o'104201',
;          1980  [192]   =  %o'000001',
;          1981  [193]   =  %o'060022',
;          1982  [194]   =  %o'103207',
;          1983  [195]   =  %o'177740',
;          1984  [196]   =  %o'115007',
;          1985  [197]   =  %o'013227',
;          1986  [198]   =  %o'060010',
;          1987  [199]   =  %o'000000',
;          1988  [200]   =  %o'165572',
;          1989  [201]   =  %o'000000');
;          1990
;          1991  end
;          1992
;          1993  eludom
```

```
                                .TITLE  ZRCFA4 CZRCFA0 RC25 FR END TEST
                                .IDENT  /V01.0/


000000                          .PSECT  DM$CODE,  RO ,  D  ,  GBL
000000  000270          DM.09:: .WORD   270
000002  000000                  .WORD   0
000004  000000                  .WORD   0
000006  000000                  .WORD   0
000010  042524                  .WORD   42524
000012  052123                  .WORD   52123
000014  034460                  .WORD   34460
000016  000000                  .WORD   0
000020  126411                  .WORD   -51367
000022  000000                  .WORD   0
000024  000000                  .WORD   0
```

```
000026  000000                  .WORD   0
000030  000000                  .WORD   0
000032  000000                  .WORD   0
000034  000000                  .WORD   0
000036  000000                  .WORD   0
000040  104206                  .WORD   -73572
000042  003051                  .WORD   3051
000044  114000                  .WORD   -64000
000046  003037                  .WORD   3037
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
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  177777                          .WORD   -1
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
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
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
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
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
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
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


;                               PSECT SUMMARY
;
;       Psect Name              Words     Attributes
;        DM$CODE                  453       RO ,  D  ,  GBL,  REL,  CON


;                           LIBRARY STATISTICS
;
;                                    -------- Symbols --------    Blocks
```

```
;       File                                    Total    Loaded   Percent      Read
;
;  SPIDER$USERS:[LAKSHMANA.11REL.REAL]AZTEC0.L16;1
;                                                523         4         0        19
;
;
;
;
;
;
;                                       COMMAND QUALIFIERS
;
;       BLISS /PDP11/LIST ZRCFA4.B16/EN:NOEIS
;
; Size:          0 code + 453 data words
; Run Time:        00:23.9
; Elapsed Time:    00:56.1
; Memory Used:  196 pages
; Compilation Complete
```

```
;          0001  MODULE ZRCFA5 =
;          0002  BEGIN
;          0003  %TITLE 'LASTAD AND SETUP'
;          0004  !
;          0005  !
;          0006  REQUIRE 'BLSMAC.REQ';
;          1495
;          1496  LIBRARY 'AZTEC0';
;          1497
;          1498  %SBTTL 'LAST ADDRESS AND SETUP SECTION'
;          1499  LASTAD;
;          1500  BGNSETUP (0);
;          1501  ENDSETUP;

                                        .TITLE  ZRCFA5 LASTAD AND SETUP

                                        .PSECT  $XYZ$,  RO
000000                                  .WORD   T$FREE
000000  000004'                BL$LAS::.WORD   <<T$FREE-<BL$LAS+4>>/2>
000002  000000C                         .WORD   0
000004  000000                 T$FREE::.WORD   0

        000004'                 L$LAST==        BL$LAS+4
        000000                  T$PTHV==        0

                                        .SBTTL  $END.LINK LAST ADDRESS AND SETUP SECTION
000000  000207                 $END.LINK::
                                        RTS     PC                              ;                       1496

; Routine Size:  1 word,        Routine Base:  $XYZ$ + 0006
; Maximum stack depth per invocation:  0 words

;          1502  END
;          1503
;          1504  ELUDOM


;                                       PSECT SUMMARY
;
;       Psect Name                      Words     Attributes
;        $XYZ$                            4       RO ,  I  ,  LCL,  REL,  CON


;                               LIBRARY STATISTICS
;
;                                       -------- Symbols --------     Blocks
;       File                            Total    Loaded   Percent       Read
;
;  SPIDER$USERS:[LAKSHMANA.11REL.REAL]AZTEC0.L16;1
;                                        523         0         0         16
;
```

```
;                                       COMMAND QUALIFIERS

;       BLISS /PDP11/LIST ZRCFA5.B16/EN:NOEIS

; Size:          1 code + 3 data words
; Run Time:        00:06.1
; Elapsed Time:    00:15.1
; Memory Used:  103 pages
; Compilation Complete
```

```
CZRCFA.EXE   Memory allocation map  TKB M40.02      Page 1
                    8-JUL-83   15:46



Partition name : DUMMY
Identification : V01.0
Task  UIC      : [300,10]
Task attributes: -HD
Total address windows: 1.
Task  image  size  : 12128. words
Task address limits: 002000 061247
R-W disk blk limits: 000002 000061 000060 00048.

*** Root segment: ZRCFA1


R/W mem  limits: 002000 061247 057250 24232.
Disk blk limits: 000002 000061 000060 00048.


Memory allocation synopsis:

Section                                          Title  Ident  File
-------                                          -----  -----  ----
. BLK.:(RW,I,LCL,REL,CON)      002000 000000 00000.
AA$COD:(RO,I,LCL,REL,CON)      002000 001260 00688.
                               002000 000232 00154.  ZRCFA1 V01.0  ZRCFA1.OBJ;1
                               002232 001026 00534.  ZRCFA2 V01.0  ZRCFA2.OBJ;1
AB$COD:(RO,I,LCL,REL,CON)      003260 015132 06746.
                               003260 015132 06746.  ZRCFA2 V01.0  ZRCFA2.OBJ;1
AC$COD:(RO,I,LCL,REL,CON)      020412 010632 04506.
                               020412 010632 04506.  ZRCFA3 V01.0  ZRCFA3.OBJ;1
BL$COD:(RO,I,LCL,REL,CON)      031244 000424 00276.
                               031244 000316 00206.  B16MUL V3.0   NEISLB.OLB;4
                               031562 000106 00070.  B16SAV V3.0   NEISLB.OLB;4
DM$COD:(RO,D,GBL,REL,CON)      031670 001612 00906.
                               031670 001612 00906.  ZRCFA4 V01.0  ZRCFA4.OBJ;1
$GLOB$:(RO,D,GBL,REL,CON)      033502 005714 03020.
                               033502 005714 03020.  ZRCFA1 V01.0  ZRCFA1.OBJ;1
$OWN$ :(RW,D,LCL,REL,CON)      041416 001240 00672.
                               041416 001240 00672.  ZRCFA3 V01.0  ZRCFA3.OBJ;1
$PLIT$:(RO,D,GBL,REL,CON)      042656 016360 07408.
                               042656 016360 07408.  ZRCFA1 V01.0  ZRCFA1.OBJ;1
$XYZ$ :(RO,I,LCL,REL,CON)      061236 000010 00008.
                               061236 000010 00008.  ZRCFA5 NONE   ZRCFA5.OBJ;1
```

Global symbols:

ADAPTO	042714-R	BL$DIV	031470-R	BYTE.C	041270-R	CLK.ST	035530-R	DATA1	041302-R	DBM14	044614-R	DBM22	045256-R
AHEAD.	050674-R	BL$LAS	061236-R	BYT.CN	041406-R	CLK.TY	035522-R	DATA2	041304-R	DBM15	044650-R	DBM23	045324-R
AVAILA	011762-R	BL$MOD	031502-R	B.MASK	041321-R	CLOCK.	003530-R	DATA3	041306-R	DBM16	044712-R	DBM24	045406-R
AVERAG	016344-R	BL$MUL	031244-R	CANCEL	041346-R	CMD.RE	041266-R	DATA4	041310-R	DBM17	044756-R	DBM25	045452-R
AZP.IN	005110-R	BL$SHF	031514-R	CHEAD.	050764-R	CMD.SL	041350-R	DBM1	044010-R	DBM18	045026-R	DBM26	045510-R
AZTEC.	017500-R	BRERR	050132-R	CLK.AD	035520-R	COM.AR	035546-R	DBM10	044410-R	DBM19	045076-R	DBM27	045554-R
AZT.IN	004236-R	BUFF.E	047662-R	CLK.CS	035524-R	CONTRO	042746-R	DBM11	044452-R	DBM2	044110-R	DBM28	045614-R
AZT.RE	052216-R	BUF.DE	041264-R	CLK.HE	035526-R	CTO.ER	050632-R	DBM12	044520-R	DBM20	045136-R	DBM29	045664-R
BHEAD.	050730-R	BUF.LE	041374-R	CLK.IN	003276-R	DATA.X	016764-R	DBM13	044560-R	DBM21	045206-R	DBM3	044124-R

DBM30	045724-R	ERR.02	046450-R	L$AU	003250-R	L$NDHW	002176-R	MSG.TK	051054-R	QST12	043612-R	SMSCP.	061204-R
DBM31	045762-R	EXAM.D	016522-R	L$AUT	002070-R	L$NDSF	002416-R	MSG.WR	052062-R	QST13	043640-R	SND.DA	052316-R
DBM32	046024-R	EXE.SU	052256-R	L$AUTO	003142-R	L$NDSW	002222-R	MSG.01	046372-R	QST14	043664-R	SND.EN	037764-R
DBM33	046060-R	EX.SUP	006464-R	L$CCP	002106-R	L$PRIO	002042-R	MSG.1	047254-R	QST15	043736-R	SWITCH	041324-R
DBM34	046100-R	FAL.CO	041402-R	L$CLEA	003224-R	L$PROT	002224-R	MSG.10	047506-R	QST2	043100-R	SWP.CO	002214-R
DBM35	046122-R	FIND.C	003344-R	L$CO	002032-R	L$PRT	002112-R	MSG.11	047554-R	QST3	043110-R	SWP.EN	002210-R
DBM36	046146-R	FMT$A	047162-R	L$DEPO	002011-R	L$REPP	002062-R	MSG.13	047612-R	QST4	043122-R	SWP.LI	002204-R
DBM37	046210-R	FMT$C	046516-R	L$DESC	002256-R	L$REV	002010-R	MSG.14	047634-R	QST6	043146-R	SWP.MA	002216-R
DBM38	046254-R	FMT1	046524-R	L$DESP	002076-R	L$RPT	002440-R	MSG.17	050210-R	QST7	043220-R	SWP.RE	002212-R
DBM39	046314-R	FMT2	046610-R	L$DEVP	002060-R	L$SFTL	002366-R	MSG.18	050256-R	QST8	043310-R	SWP.ST	002206-R
DBM4	044136-R	FMT3	046670-R	L$DISP	002124-R	L$SOFT	002370-R	MSG.19	050334-R	QST9	043330-R	SWP.TO	002202-R
DBM5	044152-R	FMT4	046764-R	L$DLY	002116-R	L$SPC	002056-R	MSG.2	047304-R	QS10.1	043426-R	SWP.TR	002220-R
DBM6	044166-R	FMT5	047022-R	L$DTP	002040-R	L$SPCP	002020-R	MSG.20	050372-R	QS10.2	043456-R	TEMP	041414-R
DBM7	044200-R	FMT6	047102-R	L$DTYP	002034-R	L$SPTP	002024-R	MSG.21	050440-R	RANDOM	016264-R	TICKS	041272-R
DBM8	044244-R	FREE.M	041364-R	L$DU	003236-R	L$STA	002030-R	MSG.28	050504-R	RCV.DA	034520-R	TIME	042660-R
DBM9	044326-R	FRU	042664-R	L$DUT	002072-R	L$SW	002202-R	MSG.29	050532-R	RC.STR	060210-R	TIP	041300-R
DECODE	017764-R	GET.CM	015464-R	L$DVTY	002232-R	L$SWLE	002200-R	MSG.30	050572-R	RC25$E	003554-R	T$FREE	061242-R
DFPTBL	002166-R	GET.RE	015532-R	L$EF	002052-R	L$TEST	002114-R	MSG.7	047334-R	RC25.A	035540-R	T$PTHV	000000
DHEAD.	051020-R	GET.UN	014770-R	L$ENVI	002044-R	L$TIML	002014-R	MSG.8	047406-R	RC25.D	035542-R	T1	021122-R
DMC.ER	047740-R	GP$1	002320-R	L$ERRT	002154-R	L$UNIT	002012-R	MSG.9	047450-R	READ.C	013310-R	T10	027434-R
DMC.TE	041404-R	GP$2	002330-R	L$ETP	002102-R	MANU.S	041322-R	NUM.RE	041376-R	READ.F	014166-R	T11	030270-R
DM.ADD	017244-R	GP$3	002340-R	L$EXP1	002046-R	MECHAN	043032-R	NXMI	003260-R	RECEIV	035760-R	T12	031230-R
DM.REC	041410-R	GP$4	002352-R	L$EXP4	002064-R	MEM.SI	041366-R	ON.LIN	012376-R	REC.DA	010270-R	T2	021346-R
DM.XMT	041412-R	GP$5	002370-R	L$EXP5	002066-R	MINUTE	041276-R	PFE.ER	050654-R	REC.EN	035764-R	T3	022144-R
DM.09	031670-R	GP$6	002402-R	L$HARD	002320-R	MSGADR	041314-R	PFE.ST	054376-R	REC.ST	015744-R	T4	022760-R
DM.10	032162-R	GP$7	002410-R	L$HIME	002120-R	MSG.AD	051364-R	PRT$FR	004040-R	RES.SL	041352-R	T5	024314-R
DM.11	032346-R	HEAD.A	035756-R	L$HPCP	002016-R	MSG.AV	052146-R	P.BR.L	002172-R	RETRIE	041400-R	T6	025310-R

```
DM.12  032656-R  HWP.TA 033516-R  L$HPTP 002022-R  MSG.BU 051320-R  P.IP.A 002166-R  RET.ST 041344-R  T7     025600-R
DO.RET 017674-R  H.EADD 041372-R  L$HRDL 002316-R  MSG.CO 051746-R  P.MASK 041320-R  RET.UN 041326-R  T8     026150-R
DRIVE. 043002-R  H.SADD 041370-R  L$HW   002166-R  MSG.DA 051432-R  P.UNIT 002174-R  RE.DAT 052354-R  T9     026722-R
DUP.MS 015600-R  INIT.C 005706-R  L$HWLE 002164-R  MSG.ER 051514-R  P.VECT 002170-R  RINGBA 035556-R  UNIT   035532-R
D$PCNT 002122-R  INI.MS 047776-R  L$ICP  002104-R  MSG.HS 051546-R  P1     041330-R  RT     033502-R  VEC.AD 035536-R
EMSG.S 054722-R  I.AM.N 041312-R  L$INIT 003130-R  MSG.LB 051134-R  P2     041332-R  RT.TAB 033514-R  WRT.PR 017266-R
END.LB 041316-R  LBN    041354-R  L$LADP 002026-R  MSG.PT 052000-R  P3     041334-R  SDUP.S 060542-R  XMT.DA 033520-R
END.MS 050060-R  LBN.ED 041360-R  L$LAST 061242-R  MSG.PW 047220-R  P4     041336-R  SECOND 041274-R  $END.L 061244-R
ERRBLK 002162-R  LBN.ST 041356-R  L$LOAD 002100-R  MSG.RE 051646-R  P5     041340-R  SEND.D 007446-R  $SAVE2 031562-R
ERRMSG 002160-R  LBN.SZ 041362-R  L$LUN  002074-R  MSG.SA 051672-R  P6     041342-R  SEND.R 035762-R  $SAVE3 031576-R
ERRNBR 002156-R  LOG.UN 035534-R  L$MREV 002050-R  MSG.SE 051472-R  QST1   043064-R  SET.CN 011112-R  $SAVE4 031614-R
ERRTYP 002154-R  L$ACP  002110-R  L$NAME 002000-R  MSG.ST 051240-R  QST10  043346-R  SET.IN 015672-R  $SAVE5 031634-R
ERR.01 046424-R  L$APT  002036-R  L$NDHR 002364-R  MSG.SU 051576-R  QST11  043536-R  SFPTBL 002202-R

*** Task builder statistics:

    Total work file references: 62863.
    Work  file  reads: 0.
    Work  file writes: 0.
    Size of core pool: 5486. words (21. pages)
    Size of work file: 3584. words (14. pages)

    Elapsed time:00:00:20
```

CZRCFA CREATED BY TKB ON 8-JUL-83 AT 15:46 PAGE 1

GLOBAL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES...			
ADAPTO	042714-R	# ZRCFA1	ZRCFA2	ZRCFA3	
AHEAD.	050674-R	# ZRCFA1	ZRCFA3		
AVAILA	011762-R	# ZRCFA2	ZRCFA3		
AVERAG	016344-R	# ZRCFA2	ZRCFA3		
AZP.IN	005110-R	# ZRCFA2	ZRCFA3		
AZTEC.	017500-R	# ZRCFA2	ZRCFA3		
AZT.IN	004236-R	# ZRCFA2	ZRCFA3		
AZT.RE	052216-R	# ZRCFA1	ZRCFA3		
BHEAD.	050730-R	# ZRCFA1	ZRCFA3		
BL$DIV	031470-R	# B16MUL	ZRCFA2	ZRCFA3	
BL$LAS	061236-R	# ZRCFA5			
BL$MOD	031502-R	# B16MUL	ZRCFA2		
BL$MUL	031244-R	# B16MUL	ZRCFA2		
BL$SHF	031514-R	# B16MUL	ZRCFA2	ZRCFA3	
BRERR	050132-R	# ZRCFA1	ZRCFA3		
BUFF.E	047662-R	# ZRCFA1	ZRCFA3		
BUF.DE	041264-R	# ZRCFA1	ZRCFA2	ZRCFA3	
BUF.LE	041374-R	# ZRCFA1	ZRCFA2	ZRCFA3	
BYTE.C	041270-R	# ZRCFA1	ZRCFA2	ZRCFA3	
BYT.CN	041406-R	# ZRCFA1	ZRCFA2		
B.MASK	041321-R	# ZRCFA1	ZRCFA2	ZRCFA3	
CANCEL	041346-R	# ZRCFA1	ZRCFA2	ZRCFA3	
CHEAD.	050764-R	# ZRCFA1	ZRCFA3		
CLK.AD	035520-R	# ZRCFA1	ZRCFA2	ZRCFA3	
CLK.CS	035524-R	# ZRCFA1	ZRCFA2	ZRCFA3	
CLK.HE	035526-R	# ZRCFA1	ZRCFA2		
CLK.IN	003276-R	# ZRCFA2			
CLK.ST	035530-R	# ZRCFA1	ZRCFA2	ZRCFA3	
CLK.TY	035522-R	# ZRCFA1	ZRCFA2		
CLOCK.	003530-R	# ZRCFA2	ZRCFA3		
CMD.RE	041266-R	# ZRCFA1	ZRCFA2	ZRCFA3	
CMD.SL	041350-R	# ZRCFA1	ZRCFA2	ZRCFA3	
COM.AR	035546-R	# ZRCFA1	ZRCFA2	ZRCFA3	
CONTRO	042746-R	# ZRCFA1	ZRCFA2		
CTO.ER	050632-R	# ZRCFA1	ZRCFA2	ZRCFA3	
DATA.X	016764-R	# ZRCFA2	ZRCFA3		
DATA1	041302-R	# ZRCFA1	ZRCFA2	ZRCFA3	
DATA2	041304-R	# ZRCFA1	ZRCFA2	ZRCFA3	
DATA3	041306-R	# ZRCFA1	ZRCFA2	ZRCFA3	
DATA4	041310-R	# ZRCFA1	ZRCFA2	ZRCFA3	
DBM1	044010-R	# ZRCFA1	ZRCFA2		
DBM10	044410-R	# ZRCFA1	ZRCFA3		
DBM11	044452-R	# ZRCFA1	ZRCFA3		
DBM12	044520-R	# ZRCFA1	ZRCFA3		
DBM13	044560-R	# ZRCFA1	ZRCFA3		
DBM14	044614-R	# ZRCFA1	ZRCFA3		
DBM15	044650-R	# ZRCFA1	ZRCFA3		
DBM16	044712-R	# ZRCFA1	ZRCFA3		
DBM17	044756-R	# ZRCFA1	ZRCFA3		
DBM18	045026-R	# ZRCFA1	ZRCFA3		
DBM19	045076-R	# ZRCFA1	ZRCFA3		
DBM2	044110-R	# ZRCFA1	ZRCFA2		

CZRCFA CREATED BY TKB ON 8-JUL-83 AT 15:46 PAGE 2

GLOBAL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES...				
DBM20	045136-R	# ZRCFA1	ZRCFA3			
DBM21	045206-R	# ZRCFA1	ZRCFA3			
DBM22	045256-R	# ZRCFA1	ZRCFA3			
DBM23	045324-R	# ZRCFA1	ZRCFA3			
DBM24	045406-R	# ZRCFA1	ZRCFA3			
DBM25	045452-R	# ZRCFA1	ZRCFA3			
DBM26	045510-R	# ZRCFA1	ZRCFA3			
DBM27	045554-R	# ZRCFA1	ZRCFA3			
DBM28	045614-R	# ZRCFA1	ZRCFA3			
DBM29	045664-R	# ZRCFA1	ZRCFA3			
DBM3	044124-R	# ZRCFA1	ZRCFA2			
DBM30	045724-R	# ZRCFA1	ZRCFA3			
DBM31	045762-R	# ZRCFA1	ZRCFA3			
DBM32	046024-R	# ZRCFA1	ZRCFA3			
DBM33	046060-R	# ZRCFA1	ZRCFA2			
DBM34	046100-R	# ZRCFA1	ZRCFA2			
DBM35	046122-R	# ZRCFA1	ZRCFA2			
DBM36	046146-R	# ZRCFA1	ZRCFA3			
DBM37	046210-R	# ZRCFA1	ZRCFA3			
DBM38	046254-R	# ZRCFA1	ZRCFA3			
DBM39	046314-R	# ZRCFA1	ZRCFA3			
DBM4	044136-R	# ZRCFA1	ZRCFA2			
DBM5	044152-R	# ZRCFA1	ZRCFA2			
DBM6	044166-R	# ZRCFA1	ZRCFA2			
DBM7	044200-R	# ZRCFA1	ZRCFA3			
DBM8	044244-R	# ZRCFA1	ZRCFA3			
DBM9	044326-R	# ZRCFA1	ZRCFA3			
DECODE	017764-R	# ZRCFA2	ZRCFA3			
DFPTBL	002166-R	# ZRCFA1				
DHEAD.	051020-R	# ZRCFA1	ZRCFA3			
DMC.ER	047740-R	# ZRCFA1	ZRCFA3			
DMC.TE	041404-R	# ZRCFA1	ZRCFA2			
DM.ADD	017244-R	# ZRCFA2	ZRCFA3			
DM.REC	041410-R	# ZRCFA1	ZRCFA2			
DM.XMT	041412-R	# ZRCFA1	ZRCFA2			
DM.09	031670-R	ZRCFA3	# ZRCFA4			
DM.10	032162-R	ZRCFA3	# ZRCFA4			
DM.11	032346-R	ZRCFA3	# ZRCFA4			
DM.12	032656-R	ZRCFA3	# ZRCFA4			
DO.RET	017674-R	# ZRCFA2	ZRCFA3			
DRIVE.	043002-R	# ZRCFA1	ZRCFA2			
DUP.MS	015600-R	# ZRCFA2				
D$PCNT	002122-R	# ZRCFA1				
EMSG.S	054722-R	# ZRCFA1	ZRCFA2			
END.LB	041316-R	# ZRCFA1	ZRCFA2	ZRCFA3		
END.MS	050060-R	# ZRCFA1	ZRCFA3			
ERRBLK	002162-R	# ZRCFA1				
ERRMSG	002160-R	# ZRCFA1				
ERRNBR	002156-R	# ZRCFA1				
ERRTYP	002154-R	# ZRCFA1				
ERR.01	046424-R	# ZRCFA1	ZRCFA2			
ERR.02	046450-R	# ZRCFA1	ZRCFA2			

```
SYMBOL  VALUE      REFERENCES...

EXAM.D  016522-R  # ZRCFA2    ZRCFA3
EXE.SU  052256-R  # ZRCFA1    ZRCFA3
EX.SUP  006464-R  # ZRCFA2    ZRCFA3
FAL.CO  041402-R  # ZRCFA1    ZRCFA2
FIND.C  003344-R  # ZRCFA2    ZRCFA3
FMT$A   047162-R  # ZRCFA1    ZRCFA2    ZRCFA3
FMT$C   046516-R  # ZRCFA1    ZRCFA2
FMT1    046524-R  # ZRCFA1    ZRCFA3
FMT2    046610-R  # ZRCFA1    ZRCFA2    ZRCFA3
FMT3    046670-R  # ZRCFA1    ZRCFA2    ZRCFA3
FMT4    046764-R  # ZRCFA1    ZRCFA3
FMT5    047022-R  # ZRCFA1    ZRCFA3
FMT6    047102-R  # ZRCFA1    ZRCFA3
FREE.M  041364-R  # ZRCFA1    ZRCFA2    ZRCFA3
FRU     042664-R  # ZRCFA1    ZRCFA2    ZRCFA3
GET.CM  015464-R  # ZRCFA2    ZRCFA3
GET.RE  015532-R  # ZRCFA2
GET.UN  014770-R  # ZRCFA2    ZRCFA3
GP$1    002320-R  # ZRCFA2
GP$2    002330-R  # ZRCFA2
GP$3    002340-R  # ZRCFA2
GP$4    002352-R  # ZRCFA2
GP$5    002370-R  # ZRCFA2
GP$6    002402-R  # ZRCFA2
GP$7    002410-R  # ZRCFA2
HEAD.A  035756-R  # ZRCFA1    ZRCFA2    ZRCFA3
HWP.TA  033516-R  # ZRCFA1    ZRCFA2
H.EADD  041372-R  # ZRCFA1    ZRCFA2    ZRCFA3
H.SADD  041370-R  # ZRCFA1    ZRCFA2    ZRCFA3
INIT.C  005706-R  # ZRCFA2    ZRCFA3
INI.MS  047776-R  # ZRCFA1    ZRCFA3
I.AM.N  041312-R  # ZRCFA1    ZRCFA2    ZRCFA3
LBN     041354-R  # ZRCFA1    ZRCFA2    ZRCFA3
LBN.ED  041360-R  # ZRCFA1    ZRCFA2    ZRCFA3
LBN.ST  041356-R  # ZRCFA1    ZRCFA2    ZRCFA3
LBN.SZ  041362-R  # ZRCFA1    ZRCFA3
LOG.UN  035534-R  # ZRCFA1    ZRCFA2    ZRCFA3
L$ACP   002110-R  # ZRCFA1
L$APT   002036-R  # ZRCFA1
L$AU    003250-R    ZRCFA1  # ZRCFA2
L$AUT   002070-R  # ZRCFA1
L$AUTO  003142-R    ZRCFA1  # ZRCFA2
L$CCP   002106-R  # ZRCFA1
L$CLEA  003224-R    ZRCFA1  # ZRCFA2
L$CO    002032-R  # ZRCFA1
L$DEPO  002011-R  # ZRCFA1
L$DESC  002256-R    ZRCFA1  # ZRCFA2
L$DESP  002076-R  # ZRCFA1
L$DEVP  002060-R  # ZRCFA1
L$DISP  002124-R  # ZRCFA1
L$DLY   002116-R  # ZRCFA1    ZRCFA2    ZRCFA3
L$DTP   002040-R  # ZRCFA1
```

CZRCFA CREATED BY TKB ON 8-JUL-83 AT 15:46 PAGE 4

GLOBAL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES...				
L$DTYP	002034-R	# ZRCFA1				
L$DU	003236-R	ZRCFA1	# ZRCFA2			
L$DUT	002072-R	# ZRCFA1				
L$DVTY	002232-R	ZRCFA1	# ZRCFA2			
L$EF	002052-R	# ZRCFA1				
L$ENVI	002044-R	# ZRCFA1				
L$ERRT	002154-R	# ZRCFA1				
L$ETP	002102-R	# ZRCFA1				
L$EXP1	002046-R	# ZRCFA1				
L$EXP4	002064-R	# ZRCFA1				
L$EXP5	002066-R	# ZRCFA1				
L$HARD	002320-R	ZRCFA1	# ZRCFA2			
L$HIME	002120-R	# ZRCFA1				
L$HPCP	002016-R	# ZRCFA1				
L$HPTP	002022-R	# ZRCFA1				
L$HRDL	002316-R	# ZRCFA2				
L$HW	002166-R	# ZRCFA1				
L$HWLE	002164-R	# ZRCFA1				
L$ICP	002104-R	# ZRCFA1				
L$INIT	003130-R	ZRCFA1	# ZRCFA2			
L$LADP	002026-R	# ZRCFA1				
L$LAST	061242-R	ZRCFA1	# ZRCFA5			
L$LOAD	002100-R	# ZRCFA1				
L$LUN	002074-R	# ZRCFA1				
L$MREV	002050-R	# ZRCFA1				
L$NAME	002000-R	# ZRCFA1				
L$NDHR	002364-R	# ZRCFA2				
L$NDHW	002176-R	# ZRCFA1				
L$NDSF	002416-R	# ZRCFA2				
L$NDSW	002222-R	# ZRCFA1				
L$PRIO	002042-R	# ZRCFA1				
L$PROT	002224-R	# ZRCFA1				
L$PRT	002112-R	# ZRCFA1				
L$REPP	002062-R	# ZRCFA1				
L$REV	002010-R	# ZRCFA1				
L$RPT	002440-R	ZRCFA1	# ZRCFA2			
L$SFTL	002366-R	# ZRCFA2				
L$SOFT	002370-R	ZRCFA1	# ZRCFA2			
L$SPC	002056-R	# ZRCFA1				
L$SPCP	002020-R	# ZRCFA1				
L$SPTP	002024-R	# ZRCFA1				
L$STA	002030-R	# ZRCFA1				
L$SW	002202-R	# ZRCFA1				
L$SWLE	002200-R	# ZRCFA1				
L$TEST	002114-R	# ZRCFA1				
L$TIML	002014-R	# ZRCFA1				
L$UNIT	002012-R	# ZRCFA1	ZRCFA2			
MANU.S	041322-R	# ZRCFA1	ZRCFA2	ZRCFA3		
MECHAN	043032-R	# ZRCFA1	ZRCFA2			
MEM.SI	041366-R	# ZRCFA1	ZRCFA2	ZRCFA3	ZRCFA3	
MINUTE	041276-R	# ZRCFA1	ZRCFA2	ZRCFA3		
MSGADR	041314-R	# ZRCFA1	ZRCFA2	ZRCFA3		

SYMBOL	VALUE	REFERENCES...			
MSG.AD	051364-R	# ZRCFA1	ZRCFA3		
MSG.AV	052146-R	# ZRCFA1	ZRCFA3		
MSG.BU	051320-R	# ZRCFA1	ZRCFA3		
MSG.CO	051746-R	# ZRCFA1	ZRCFA3		
MSG.DA	051432-R	# ZRCFA1	ZRCFA3		
MSG.ER	051514-R	# ZRCFA1	ZRCFA3		
MSG.HS	051546-R	# ZRCFA1	ZRCFA3		
MSG.LB	051134-R	# ZRCFA1	ZRCFA3		
MSG.PT	052000-R	# ZRCFA1	ZRCFA3		
MSG.PW	047220-R	# ZRCFA1	ZRCFA2		
MSG.RE	051646-R	# ZRCFA1	ZRCFA3		
MSG.SA	051672-R	# ZRCFA1	ZRCFA3		
MSG.SE	051472-R	# ZRCFA1	ZRCFA3		
MSG.ST	051240-R	# ZRCFA1	ZRCFA2	ZRCFA3	
MSG.SU	051576-R	# ZRCFA1	ZRCFA3		
MSG.TK	051054-R	# ZRCFA1	ZRCFA3		
MSG.WR	052062-R	# ZRCFA1	ZRCFA3		
MSG.01	046372-R	# ZRCFA1			
MSG.1	047254-R	# ZRCFA1	ZRCFA3		
MSG.10	047506-R	# ZRCFA1	ZRCFA3		
MSG.11	047554-R	# ZRCFA1	ZRCFA3		
MSG.13	047612-R	# ZRCFA1	ZRCFA3		
MSG.14	047634-R	# ZRCFA1	ZRCFA2	ZRCFA3	
MSG.17	050210-R	# ZRCFA1	ZRCFA3		
MSG.18	050256-R	# ZRCFA1	ZRCFA3		
MSG.19	050334-R	# ZRCFA1	ZRCFA3		
MSG.2	047304-R	# ZRCFA1	ZRCFA3		
MSG.20	050372-R	# ZRCFA1	ZRCFA3		
MSG.21	050440-R	# ZRCFA1	ZRCFA3		
MSG.28	050504-R	# ZRCFA1	ZRCFA3		
MSG.29	050532-R	# ZRCFA1	ZRCFA3		
MSG.30	050572-R	# ZRCFA1	ZRCFA3		
MSG.7	047334-R	# ZRCFA1	ZRCFA3		
MSG.8	047406-R	# ZRCFA1	ZRCFA3		
MSG.9	047450-R	# ZRCFA1	ZRCFA3		
NUM.RE	041376-R	# ZRCFA1	ZRCFA2	ZRCFA3	
NXMI	003260-R	# ZRCFA2	ZRCFA3		
ON.LIN	012376-R	# ZRCFA2	ZRCFA3		
PFE.ER	050654-R	# ZRCFA1	ZRCFA2		
PFE.ST	054376-R	# ZRCFA1	ZRCFA2		
PRT$FR	004040-R	# ZRCFA2			
P.BR.L	002172-R	# ZRCFA1			
P.IP.A	002166-R	# ZRCFA1	ZRCFA3		
P.MASK	041320-R	# ZRCFA1	ZRCFA2	ZRCFA3	
P.UNIT	002174-R	# ZRCFA1			
P.VECT	002170-R	# ZRCFA1	ZRCFA3		
P1	041330-R	# ZRCFA1	ZRCFA2	ZRCFA3	
P2	041332-R	# ZRCFA1	ZRCFA2	ZRCFA3	
P3	041334-R	# ZRCFA1	ZRCFA2	ZRCFA3	
P4	041336-R	# ZRCFA1	ZRCFA2	ZRCFA3	
P5	041340-R	# ZRCFA1	ZRCFA2	ZRCFA3	
P6	041342-R	# ZRCFA1	ZRCFA2	ZRCFA3	

CZRCFA CREATED BY TKB ON 8-JUL-83 AT 15:46 PAGE 6

GLOBAL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES...			
QST1	043064-R	# ZRCFA1	ZRCFA2		
QST10	043346-R	# ZRCFA1	ZRCFA2		
QST11	043536-R	# ZRCFA1	ZRCFA2		
QST12	043612-R	# ZRCFA1	ZRCFA3		
QST13	043640-R	# ZRCFA1	ZRCFA3		
QST14	043664-R	# ZRCFA1	ZRCFA2	ZRCFA3	
QST15	043736-R	# ZRCFA1	ZRCFA2	ZRCFA3	
QST2	043100-R	# ZRCFA1	ZRCFA2		
QST3	043110-R	# ZRCFA1	ZRCFA2		
QST4	043122-R	# ZRCFA1	ZRCFA2		
QST6	043146-R	# ZRCFA1	ZRCFA2		
QST7	043220-R	# ZRCFA1	ZRCFA2		
QST8	043310-R	# ZRCFA1	ZRCFA2		
QST9	043330-R	# ZRCFA1	ZRCFA2		
QS10.1	043426-R	# ZRCFA1	ZRCFA2		
QS10.2	043456-R	# ZRCFA1	ZRCFA2		
RANDOM	016264-R	# ZRCFA2	ZRCFA3		
RCV.DA	034520-R	# ZRCFA1	ZRCFA2	ZRCFA3	
RC.STR	060210-R	# ZRCFA1	ZRCFA2		
RC25$E	003554-R	# ZRCFA2	ZRCFA3		
RC25.A	035540-R	# ZRCFA1	ZRCFA2	ZRCFA3	
RC25.D	035542-R	# ZRCFA1	ZRCFA2	ZRCFA3	
READ.C	013310-R	# ZRCFA2	ZRCFA3		
READ.F	014166-R	# ZRCFA2	ZRCFA3		
RECEIV	035760-R	# ZRCFA1	ZRCFA2	ZRCFA3	
REC.DA	010270-R	# ZRCFA2	ZRCFA3		
REC.EN	035764-R	# ZRCFA1	ZRCFA2	ZRCFA3	
REC.ST	015744-R	# ZRCFA2	ZRCFA3		
RES.SL	041352-R	# ZRCFA1	ZRCFA2	ZRCFA3	
RETRIE	041400-R	# ZRCFA1	ZRCFA2	ZRCFA3	
RET.ST	041344-R	# ZRCFA1	ZRCFA2	ZRCFA3	
RET.UN	041326-R	# ZRCFA1	ZRCFA2	ZRCFA3	
RE.DAT	052354-R	# ZRCFA1	ZRCFA3		
RINGBA	035556-R	# ZRCFA1	ZRCFA2	ZRCFA3	
RT	033502-R	# ZRCFA1	ZRCFA2		
RT.TAB	033514-R	# ZRCFA1	ZRCFA2	ZRCFA3	
SDUP.S	060542-R	# ZRCFA1	ZRCFA2		
SECOND	041274-R	# ZRCFA1	ZRCFA2	ZRCFA3	
SEND.D	007446-R	# ZRCFA2	ZRCFA3		
SEND.R	035762-R	# ZRCFA1	ZRCFA2	ZRCFA3	
SET.CN	011112-R	# ZRCFA2	ZRCFA3		
SET.IN	015672-R	# ZRCFA2	ZRCFA3		
SFPTBL	002202-R	# ZRCFA1			
SMSCP.	061204-R	# ZRCFA1	ZRCFA2		
SND.DA	052316-R	# ZRCFA1	ZRCFA3		
SND.EN	037764-R	# ZRCFA1	ZRCFA2	ZRCFA3	
SWITCH	041324-R	# ZRCFA1	ZRCFA2	ZRCFA3	
SWP.CO	002214-R	# ZRCFA1	ZRCFA2	ZRCFA3	
SWP.EN	002210-R	# ZRCFA1	ZRCFA3		
SWP.LI	002204-R	# ZRCFA1			
SWP.MA	002216-R	# ZRCFA1	ZRCFA3		
SWP.RE	002212-R	# ZRCFA1	ZRCFA2	ZRCFA3	

GLOBAL CROSS REFERENCE CREF V01

SYMBOL	VALUE	REFERENCES...				
SWP.ST	002206-R	# ZRCFA1	ZRCFA3			
SWP.TO	002202-R	# ZRCFA1				
SWP.TR	002220-R	# ZRCFA1	ZRCFA2	ZRCFA3		
TEMP	041414-R	# ZRCFA1	ZRCFA2	ZRCFA3		
TICKS	041272-R	# ZRCFA1	ZRCFA2	ZRCFA3		
TIME	042660-R	# ZRCFA1	ZRCFA3			
TIP	041300-R	# ZRCFA1	ZRCFA2	ZRCFA3		
T$FREE	061242-R	# ZRCFA5				
T$PTHV	000000	ZRCFA1	# ZRCFA5			
T1	021122-R	ZRCFA1	# ZRCFA3			
T10	027434-R	ZRCFA1	# ZRCFA3			
T11	030270-R	ZRCFA1	# ZRCFA3			
T12	031230-R	ZRCFA1	# ZRCFA3			
T2	021346-R	ZRCFA1	# ZRCFA3			
T3	022144-R	ZRCFA1	# ZRCFA3			
T4	022760-R	ZRCFA1	# ZRCFA3			
T5	024314-R	ZRCFA1	# ZRCFA3			
T6	025310-R	ZRCFA1	# ZRCFA3			
T7	025600-R	ZRCFA1	# ZRCFA3			
T8	026150-R	ZRCFA1	# ZRCFA3			
T9	026722-R	ZRCFA1	# ZRCFA3			
UNIT	035532-R	# ZRCFA1	ZRCFA2	ZRCFA3		
VEC.AD	035536-R	# ZRCFA1	ZRCFA2	ZRCFA3		
WRT.PR	017266-R	# ZRCFA2	ZRCFA3			
XMT.DA	033520-R	# ZRCFA1	ZRCFA2	ZRCFA3		
$END.L	061244-R	# ZRCFA5				
$SAVE2	031562-R	B16MUL	# B16SAV	ZRCFA2	ZRCFA3	
$SAVE3	031576-R	# B16SAV	ZRCFA2			
$SAVE4	031614-R	# B16SAV	ZRCFA3			
$SAVE5	031634-R	B16MUL	# B16SAV	ZRCFA2		