

RQDX1 RX50
RD51 RD52

RQDX1 EXERCISER
CZRQACO

COPYRIGHT (c) 1983
AH-T399C-MC
FICHE 1 OF 3

APR 1984
digital
Made In USA

This microfiche card contains a grid of frames. The frames are arranged in approximately 12 rows and 12 columns. Each frame contains a small, high-contrast image or diagram, likely representing a circuit board or a specific component of the system. The images are very small and difficult to discern, but they appear to be technical drawings or photographs of hardware. The overall layout is a dense grid of these small images, typical of a microfiche card used for storing large amounts of technical data.

RQDX1 RX50
RD51 RD52

RQDX1 EXERCISER
CZRQACO

COPYRIGHT (c) 1983
AH-T399C-MC
FICHE 2 OF 3

APR 1984
digital
Made In USA

This microfiche card contains a grid of frames. Each frame contains a small, high-contrast image of a document page. The frames are arranged in approximately 12 rows and 12 columns. The content of the frames is mostly illegible due to the low resolution and high contrast of the microfiche reproduction. Some frames appear to contain text, while others contain what might be diagrams or tables. The overall appearance is that of a dense array of tiny document pages.

RQDX1 RX50 RQDX1 EXERCISER
RD51 RD52 CZRQACO

COPYRIGHT (c) 1983
AH-T399C-MC
FICHE 3 OF 3

APR 1984
digital
Made In USA

```

:      0001 module ZRQAM1 (
:      0002
:      0003 *title 'RD/RX EXERCISER'
:      0004             ident = 'V01.2',
:      0005             addressing_mode (absolute),
:      0006             environment (noeis)
:      0007             ) =
:      0008
:      0009 begin
:      0010
:      0011
:      C 0012 *(
:      C 0013             IDENTIFICATION
:      C 0014             -----
:      C 0015
:      C 0016             PRODUCT CODE:          AC-T398C-MC
:      C 0017
:      C 0018             PRODUCT NAME:         CZRQACO RQDX1 EXERCISER
:      C 0019
:      C 0020             PRODUCT DATE:         5 DEC 1983
:      C 0021
:      C 0022             MAINTAINER:          DIAGNOSTIC ENGINEERING
:      C 0023
:      C 0024             AUTHOR:              RICHARD F. DIETZ
:      C 0025
:      C 0026
:      C 0027             COPYRIGHT (C) 1983
:      C 0028
:      C 0029             DIGITAL EQUIPMENT CORPOTRATION, MAYNARD, MASSACHUSETTS 01754
:      C 0030
:      C 0031             THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
:      C 0032             COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE
:      C 0033             ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF,
:      C 0034             MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON
:      C 0035             EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE
:      C 0036             TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES
:      C 0037             REMAIN IN DEC.
:      C 0038
:      C 0039             THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE
:      C 0040             AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
:      C 0041             CORPORATION.
:      C 0042
:      C 0043             DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
:      C 0044             SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
:      C 0045
:      C 0046
:      C 0047             THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
:      C 0048
:      C 0049             DIGITAL          PDP          UNIBUS          MASSBUS
:      C 0050             DEC              DECUS         DECTAPE

```

:	C 0051	
:	C 0052	
:	C 0053	
:	C 0054	
:	C 0055	
:	C 0056	
:	C 0057	
:	C 0058	TABLE OF CONTENTS
:	C 0059	
:	C 0060	
:	C 0061	
:	C 0062	1.0 GENERAL INFORMATION
:	C 0063	1.1 PROGRAM ABSTRACT
:	C 0064	1.2 SYSTEM REQUIREMENTS
:	C 0065	1.2.1 HARDWARE REQUIREMENTS
:	C 0066	1.2.2 SOFTWARE REQUIREMENTS
:	C 0067	1.3 RELATED DOCUMENTS AND STANDARDS
:	C 0068	1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
:	C 0069	1.5 ASSUMPTIONS
:	C 0070	1.6 MEMORY MAP
:	C 0071	
:	C 0072	2.0 OPERATING INSTRUCTIONS
:	C 0073	2.1 HARDWARE QUESTIONS
:	C 0074	2.2 SOFTWARE QUESTIONS
:	C 0075	
:	C 0076	3.0 ERROR TYPES
:	C 0077	3.1 ERROR INFORMATION
:	C 0078	3.2 INITIALIZATION ERRORS
:	C 0079	3.3 EXERCISER ERRORS
:	C 0080	3.4 ERROR LOG MESSAGES
:	C 0081	3.5 MSCP ERRORS
:	C 0082	3.6 SAMPLE MSCP ERROR STATEMENT
:	C 0083	3.7 DUP ERRORS
:	C 0084	3.8 SAMPLE DUP ERROR STATEMENT
:	C 0085	4.0 PERFORMANCE AND PROGRESS REPORTS
:	C 0086	
:	C 0087	5.0 TEST SUMMARY
:	C 0088	5.1 INITIALIZATION SUBTEST
:	C 0089	5.2 EXERCISER
:	C 0090	5.3 DROP UNIT SUMMARY
:	C 0091	
:	C 0092	6.0 ERROR LIST
:	C 0093	
:	C 0094	7.0 DATA PATTERNS

```

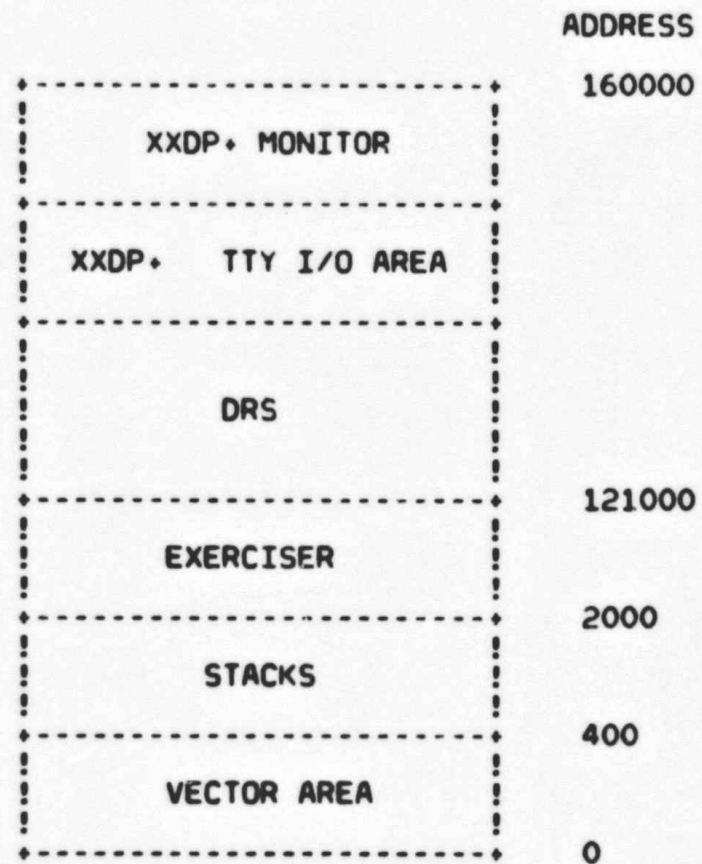
: C 0095
: C 0096
: C 0097      1.0  GENERAL INFORMATION
: C 0098      -----
: C 0099
: C 0100
: C 0101      1.1  PROGRAM ABSTRACT
: C 0102      -----
: C 0103
: C 0104      This program will functionally verify and exercise RQDX1
: C 0105      Controller/Disk Drive subsystems. It is designed to verify
: C 0106      that the subsystem is functioning correctly and operating
: C 0107      within design specifications.
: C 0108
: C 0109
: C 0110
: C 0111      1.2  SYSTEM REQUIREMENTS
: C 0112      -----
: C 0113
: C 0114      1.2.1 HARDWARE REQUIREMENTS
: C 0115      -----
: C 0116
: C 0117      LSI - 11/23 processor with 28K or more of memory, console
: C 0118      device (eg. VT100) and RQDX1 CONTROLLER board and attached
: C 0119      RD-51 WINCHESTER drive(s) and RX-50 FLOPPY drive(s)
: C 0120
: C 0121      1.2.2 SOFTWARE REQUIREMENTS
: C 0122      -----
: C 0123
: C 0124      This diagnostic is designed to run with the Diagnostic
: C 0125      Supervisor as described in paragraph 2.0.
: C 0126
: C 0127
: C 0128      1.3  RELATED DOCUMENTS AND STANDARDS
: C 0129      -----
: C 0130
: C 0131      XXDP+ SUPERVISOR/USERS MANUAL   CHQUS
: C 0132      UQSSP UNIBUS/Q-BUS STORAGE SYSTEMS PORT
: C 0133      MSCP  MASS STORAGE SYSTEM PROTOCOL
: C 0134      DUP   DIAGNOSTIC/UTILITIES PROTOCOL
: C 0135
: C 0136      1.4  DIAGNOSTIC HIERARCY PREREQUISITES
: C 0137      -----
: C 0138
: C 0139      NONE
: C 0140
: C 0141
: C 0142      1.5  ASSUMPTIONS
: C 0143      -----
: C 0144
: C 0145      The hardware, other than the subsystem being tested is
: C 0146      assumed to work properly. False errors may be reported if
: C 0147      the processor, memory, etc., do not function properly.

```

: C 0148
: C 0149
: C 0150
: C 0151
: C 0152
: C 0153
: C 0154
: C 0155
: C 0156
: C 0157
: C 0158
: C 0159
: C 0160
: C 0161
: C 0162
: C 0163
: C 0164
: C 0165
: C 0166
: C 0167
: C 0168
: C 0169
: C 0170
: C 0171
: C 0172
: C 0173
: C 0174
: C 0175
: C 0176
: C 0177
: C 0178
: C 0179
: C 0180
: C 0181
: C 0182
: C 0183
: C 0184
: C 0185
: C 0186
: C 0187
: C 0188
: C 0189

1.6 MEMORY MAP

Memory layout on 28k machine - XXDP environment



In a machine with more memory, free space will occur between the exerciser and the DRS.

```

: C 0190
: C 0191
: C 0192      2.0  OPERATING INSTRUCTIONS
: C 0193      -----
: C 0194
: C 0195
: C 0196      This is a Rev C Supervisor Diagnostic: for operating
: C 0197      instructions, please see chapter 5 of XXDP+ operator's
: C 0198      manual. They are no longer included in the diagnostic
: C 0199      because it is desired that a change in those instruc-
: C 0200      tions not require a re-assembly of all Supervisor Diag-
: C 0201      nostics.
: C 0202
: C 0203
: C 0204      2.1  HARDWARE QUESTIONS
: C 0205      -----
: C 0206
: C 0207      The following series of questions comprise the para-
: C 0208      meters necessary to identify each disk subsystem.
: C 0209
: C 0210
: C 0211      Hardware Configuration Questions
: C 0212      -----
: C 0213
: C 0214      The program will ask the following questions in
: C 0215      response to a START command (non-script).
: C 0216
: C 0217      1.  CHANGE HW (L) Y ?
: C 0218
: C 0219      Answer NO to use the pre-built answers for all hardware
: C 0220      questions. This program will be released pre-built to
: C 0221      test three unit with default answers shown below. The
: C 0222      pre-built answers may be changed at any time with the
: C 0223      setup utility. Answer YES if you want all the hardware
: C 0224      questions to be asked.
: C 0225
: C 0226      2.  NUMBER OF UNITS (D) ?
: C 0227
: C 0228      No default. Answer with the number of disk drive units
: C 0229      to be exercised or tested. This answer will determine
: C 0230      how many times the following questions are asked. A
: C 0231      range of 1 to 4 units may be specified. A unit number
: C 0232      will be assigned sequentially from 0 by the Diagnostic
: C 0233      supervisor for each unit.
: C 0234
: C 0235      3.  IP ADDRESS (O) 172150 ?
: C 0236
: C 0237      Enter the address of the IP register of one RQDX1 as ad-
: C 0238      dressed by the processor with memory management turned
: C 0239      off. The program expects an even 16-bit address in the
: C 0240      range of 160000 to 177774. 172150 is the default.

```



```

: C 0241
: C 0242
: C 0243
: C 0244
: C 0245
: C 0246
: C 0247
: C 0248
: C 0249
: C 0250
: C 0251
: C 0252
: C 0253
: C 0254
: C 0255
: C 0256
: C 0257
: C 0258
: C 0259
: C 0260
: C 0261
: C 0262
: C 0263
: C 0264
: C 0265
: C 0266
: C 0267
: C 0268
: C 0269
: C 0270
: C 0271
: C 0272
: C 0273
: C 0274
: C 0275
: C 0276
: C 0277
: C 0278
: C 0279
: C 0280
: C 0281
: C 0282
: C 0283
: C 0284
: C 0285
: C 0286
: C 0287
: C 0288
: C 0289
: C 0290
: C 0291
: C 0292
: C 0293

```

4. VECTOR ADDRESS (0) 154 ?

Answer with the interrupt vector of same RQDX1 in the above question. A vector address in the range of 4 to 774 may be specified. 154 is the default.

5. BR LEVEL (D) 4 ?

Answer with the bus request interrupt level used by the above RQDX1. Levels 4 through 7 are acceptable. 4 is the default.

6. RQDX1 DRIVE NUMBER (D) 0 ?

Enter the logical unit number for one drive associated with the IP address above. Drive numbers are in the range of 0 through 3. The number entered here must match the unit plug on the front panel of the device. 0 is the default answer.

7. ALSO EXERCISE DIAGNOSTIC AREA (-NON-CUSTOMER AREA RDs ONLY) ?

A "Yes" answer to this question will turn on the DUP (Diagnostic/Utilities Protocol) Exerciser. The DUP Exerciser will read the DBNs (Diagnostic Blocks) in a sequential order from 0 to octal (217). DBNs can only be accessed through DUP protocol making the blocks unreachable through normal customer use. The DUP Exerciser is contained in the middle of the MSCP (Mass Storage Control Protocol) Exerciser which reads LBNs (Logical blocks). So physically a large amount of LBNs will be READ and/or WRITTEN then the DUP Exerciser will turn on and READ and/or WRITE a few DBNs. The DUP Exerciser will then reinitialize the controller so the MSCP Exerciser can continue where it left off. The processes of reinitializing takes a few seconds.

NOTE This question only pertains to Winchester type devices. If a yes answer is given for an RX-50 nothing will happen.

8. WRITE ON DIAGNOSTIC AREA?

A "yes" answer to this statement will allow the DUP Exerciser to WRITE/READ and compare the DBNs (Diagnostic Blocks). Where a "NO" answer will allow the program to READ only the DBNs. If this is an RX-50 nothing will be affected.

NOTE ** IF yes ANSWER, DATA ON DBNs WILL BE DESTROYED.

: C 0294
: C 0295
: C 0296
: C 0297
: C 0298
: C 0299
: C 0300
: C 0301
: C 0302
: C 0303
: C 0304
: C 0305
: C 0306
: C 0307
: C 0308
: C 0309
: C 0310
: C 0311
: C 0312
: C 0313
: C 0314
: C 0315
: C 0316
: C 0317
: C 0318
: C 0319
: C 0320
: C 0321
: C 0322
: C 0323
: C 0324
: C 0325
: C 0326
: C 0327
: C 0328
: C 0329
: C 0330
: C 0331
: C 0332
: C 0333
: C 0334
: C 0335

9. STARTING LBN (D) 0 ?

Enter the starting logical block number of the customer data area that you are going to test. LBNs range from 0 to 21599 (RD51), or 0 to 47999 (RD51), or 0 to 799 (RX50), with 0 as the default. If too large a number is given for that particular device the program will default to 0.

10. ENDING LBN (MAX - RX50: 799, RD51: 21599) (D) 21599 ?

Answer this question for the last customer LBN you wish to test. LBNs range from 0 to 21599 (RD51), or 0 to 47999 (RD52), or 0 to 799 (RX50), with the default being larger than all units. If the number given is larger than the amount of LBNs for that particular unit the program will default the end track to the highest track for that unit.

NOTE: The two previous questions are generally Software Parameter questions, but since two different disk devices exist on the RQDX1 subsystem, this becomes a unit by unit question. It is possible to specify an LBN which is too large since we are dealing with different devices. The program will check for block number bounds and if they are exceeded, will assign the maximum bound for that device.

11. EXERCISE ON CUSTOMER DATA AREA ON THIS DISK UNIT (L) ?

Answering YES will destroy any customer data that is on the disk; therefore, the following warning message will appear, followed by a confirmation prompt:

** WARNING - CUSTOMER DATA AREA WILL BE OVERWRITTEN! ...
CONFIRM (L) ?

This question will default to NO if the operator has decided to bypass the hardware questions. Otherwise, there is no default.

```

: C 0336
: C 0337
: C 0338
: C 0339
: C 0340
: C 0341
: C 0342
: C 0343
: C 0344
: C 0345
: C 0346
: C 0347
: C 0348
: C 0349
: C 0350
: C 0351
: C 0352
: C 0353
: C 0354
: C 0355
: C 0356
: C 0357
: C 0358
: C 0359
: C 0360
: C 0361
: C 0362
: C 0363
: C 0364
: C 0365
: C 0366
: C 0367
: C 0368
: C 0369
: C 0370
: C 0371
: C 0372
: C 0373
: C 0374
: C 0375
: C 0376
: C 0377
: C 0378
: C 0379
: C 0380
: C 0381
: C 0382
: C 0383
: C 0384
: C 0385
: C 0386
: C 0387
: C 0388

```

2.2 SOFTWARE QUESTIONS

Software Parameter Questions

The program will ask the following questions in response to the START, RESTART, and CONTINUE commands.

1. CHANGE SW (L) Y ?

Answer NO to bypass the following questions in this section. This question should normally be answered NO when the Exerciser is first run. A NO answer will cause the Exerciser to select the default parameters shown with each question below. Then, depending on the errors detected, it may be desirable to change this answer to YES to alter the test parameters and further isolate the problem.

2. HARD ERROR LIMIT (D) 32 ?

Enter the number of hard errors allowed before a unit is dropped from testing. A number in the range of 1 to 65535 will be accepted.

3. TRANSFER LIMIT IN MEGABYTES (0 FOR QUICK PASS) (D) 0 ?

When the specified number of bytes have been transferred to/from a unit, the unit will be dropped from testing. When all units are dropped, an end-of-pass will be indicated. This is the method used to determine how long the Exerciser is to run.

The only other way the Exerciser will declare end-of-pass is if all units are dropped because the error limit on each is exceeded. However, the operator can always abort the program at any time by typing CONTROL-C.

4. PERCENTAGE OF RD51 OPERATIONS OUT OF TOTAL OPERATIONS (D) 99 ?

In order to maintain typical usage for the devices of this exercise, a certain percentage of operations must be directed to the RD51s (the rest go to the RX50s). It turns out that this percentage is very high (as indicated by the 99% figure given as the default). It may be desirable in some cases to direct more activity to the RX50s. This is easily done by directing a smaller percentage of the operations to the RD51s. The numbers associated with usage are adjusted internally by the program according to device type and percentage.

```

: C 0389
: C 0390
: C 0391
: C 0392
: C 0393
: C 0394
: C 0395
: C 0396
: C 0397
: C 0398
: C 0399
: C 0400
: C 0401
: C 0402
: C 0403
: C 0404
: C 0405
: C 0406
: C 0407
: C 0408
: C 0409
: C 0410
: C 0411
: C 0412
: C 0413
: C 0414
: C 0415
: C 0416
: C 0417
: C 0418
: C 0419
: C 0420
: C 0421
: C 0422
: C 0423
: C 0424
: C 0425
: C 0426
: C 0427
: C 0428
: C 0429
: C 0430

```

5. NUMBER OF DBNs TRANSFERRED AT ONE "DUP" PASS (11) ?

This variable adjusts the amount of DBNs read in one pass of the DUP Exerciser. The DUP Exerciser has to reinit the controller every time it goes back into the MSCP Exerciser. This Hard reinitialization takes a few seconds. The ratio of LBN tranfers to DBN tranfers DOESN'T change only the amount of DBNs read on one pass. So the higher this number the less reinitializing the program does and the less time to run. The lower this number the more reinitializations the longer amount of time to tranfer the same amount LBNs and DBNs.

6. CLEAR STATISTICAL TABLES AFTER PRINTING (L) N ?

Answering YES causes the statistical fields to be cleared to zero after the report is printed (either at end of pass, or at operator request). Otherwise, cumulative totals are maintained.

7. RANDOM UNIT MODE (L) Y ?

Answer YES causes the units to be pick in a random fashion starting with either floppys at random or winchesters at random. A NO answer causes the units to be in a sequential order.

8. PERCENTAGE OF WINCHESTER OPERATIONS OUT OF TOTAL OPERATIONS (98)?

This question is only asked when a YES answer is given to the random unit mode. Any number from 0 to 100 may be given to define the percentage of winchester transfers to total transfers.

9. RANDOM BLOCK MODE (Y)?

A YES answer to this question will pick LBN blocks at random. A NO answer will pick LBN blocks in a sequential order starting at the beginning LBN and going up to the ending LBN and then back.

```

: C 0431
: C 0432
: C 0433
: C 0434
: C 0435
: C 0436
: C 0437
: C 0438
: C 0439
: C 0440
: C 0441
: C 0442
: C 0443
: C 0444
: C 0445
: C 0446
: C 0447
: C 0448
: C 0449
: C 0450
: C 0451
: C 0452
: C 0453
: C 0454
: C 0455
: C 0456
: C 0457
: C 0458
: C 0459
: C 0460
: C 0461
: C 0462
: C 0463
: C 0464
: C 0465
: C 0466
: C 0467
: C 0468
: C 0469
: C 0470
: C 0471
: C 0472
: C 0473
: C 0474
: C 0475
: C 0476
: C 0477
: C 0478

```

10. READ-COMPARES PERFORMED AT THE CONTROLLER (L) N ?

Answering YES causes all read commands to include the "compare" modifier. This essentially forces the controller to perform two read operations on the same disk address, and to compare the results.

The following message will appear after the operator has answered this question:

THE REMAINING QUESTIONS ONLY APPLY TO UNPROTECTED DISK UNITS.

11. WRITE-COMPARES PERFORMED AT THE CONTROLLER (L) N ?

Answering YES causes all write I/O requests to be changed to write-compare. After each write, the controller will read the data and compare it to data re-obtained from the host.

12. CHECK ALL WRITES AT HOST BY READING (L) N ?

This question will only be asked if the previous question was answered NO. Answering YES causes all writes to be checked by the host by reading the data immediately after the write operation. This option consumes extra CPU time, and doubles the amount of storage required for writes. Therefore, it is only recommended when device write-compare operations are suspect.

13. USER-DEFINED DATA PATTERN (L) N ?

An answer of YES allows the operator to define his/her own data pattern to be used in all write operations. A NO answer will allow the operator to select a pre-defined data pattern in the next question.

14. SELECT PRE-DEFINED DATA PATTERN (0 FOR SEQUENTIAL SELECTION) (D) 0 ?

There are 21 predefined data patterns available, selected as 1 to 21 (see section 4.9). A zero answer will cause patterns 1 to 21 to be sequentially selected for each write. (Note that pattern 1 consists entirely of random numbers).

L1

ZRQAM1
V01.2

RD/RX EXERCISER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (11)

SEQ 0011
Page 11

: C 0479
: C 0480
: C 0481
: C 0482
: C 0483
: C 0484
: C 0485
: C 0486

15. NUMBER OF WORDS IN DATA PATTERN (16 MAXIMUM) (D) 16 ?
PATTERN VALUES (O) ?

These questions will only be asked if the operator has
decided to define his/her own data pattern. The actual
bit patterns will be entered as octal (PDP-11).

```

: C 0487
: C 0488
: C 0489
: C 0490
: C 0491
: C 0492
: C 0493
: C 0494
: C 0495
: C 0496
: C 0497
: C 0498
: C 0499
: C 0500
: C 0501
: C 0502
: C 0503
: C 0504
: C 0505
: C 0506
: C 0507
: C 0508
: C 0509
: C 0510
: C 0511
: C 0512
: C 0513
: C 0514
: C 0515
: C 0516
: C 0517
: C 0518
: C 0519
: C 0520
: C 0521
: C 0522
: C 0523
: C 0524
: C 0525
: C 0526
: C 0527
: C 0528
: C 0529
: C 0530
: C 0531
: C 0532
: C 0533
: C 0534
: C 0535

```

3.0 ERROR TYPES

This program has four types of error classifications:
system fatal, device fatal, hard and soft.

SYSTEM FATAL ERRORS

System fatal errors are used to indicate that an error
was detected by the Diagnostic Supervisor in relation
to loading/controlling the diagnostic process.

The content of each error is such that it should be
self explanatory. However, the messages utilize some
terms that are specific to the disk subsystem, and may
require some getting use to.

DEVICE FATAL ERRORS

Device fatal errors are a result of:

an error that is considered fatal to the device, but
testing will continue.

HARD ERRORS

Hard errors are a result of:

1. retries of a soft error or *
2. a non-recoverable error
3. a soft error if retrys are not set.

* Note: Retries are executed in the controller

SOFT ERRORS

Soft errors are media related errors. All soft errors
will be retried by the controller.

Note: Soft errors are retrieved from the controller via
the error log capabilities of MSCP.

: C 0536
 : C 0537
 : C 0538
 : C 0539
 : C 0540
 : C 0541
 : C 0542
 : C 0543
 : C 0544
 : C 0545
 : C 0546
 : C 0547
 : C 0548
 : C 0549
 : C 0550
 : C 0551
 : C 0552
 : C 0553
 : C 0554
 : C 0555
 : C 0556
 : C 0557
 : C 0558
 : C 0559
 : C 0560
 : C 0561
 : C 0562
 : C 0563
 : C 0564
 : C 0565
 : C 0566
 : C 0567
 : C 0568
 : C 0569
 : C 0570
 : C 0571
 : C 0572
 : C 0573
 : C 0574
 : C 0575
 : C 0576
 : C 0577
 : C 0578
 : C 0579
 : C 0580
 : C 0581
 : C 0582
 : C 0583
 : C 0584
 : C 0585
 : C 0586

3.1 ERROR INFORMATION

All general error messages will include the type of error (system-fatal, device-fatal, hard, soft) and a unit number. If the error applies to a controller, then only the first unit number of the controller will be given. (The user will know the other unit numbers when subsequent "drop unit" messages are printed).

Basic error messages provide more details about the error. The Exerciser will print all basic error messages, along with the disk address, if applicable. In some cases where a device-fatal error applies to a controller, the controller's IP address will be printed.

Extended error messages will be used to print the relevant fields of command and end message packets, status codes, SA register contents, and error log messages. All values will be in octal (PDP-11).

The error messages in this section do not include errors detected and printed by the Diagnostic Supervisor.

3.2 INITIALIZATION ERRORS

Two kinds of errors will be reported to the operator during the Initialization Test. The System-fatal error is - too many units specified. A system-fatal error will cause the Exerciser to abort.

Device-fatal errors only affect the unit(s) involved. Testing will continue on all other units. This class of errors includes, but is not limited to, the following:

1. Register Existence Test failure (no device present)
2. Vector Test failure
3. BR Level Test failure
4. Initialization sequence failure
5. Online failed
6. Access failed


```

: C 0587
: C 0588
: C 0589
: C 0590
: C 0591
: C 0592
: C 0593
: C 0594
: C 0595
: C 0596
: C 0597
: C 0598
: C 0599
: C 0600
: C 0601
: C 0602
: C 0603
: C 0604
: C 0605
: C 0606
: C 0607
: C 0608
: C 0609
: C 0610
: C 0611
: C 0612
: C 0613
: C 0614
: C 0615
: C 0616
: C 0617
: C 0618
: C 0619
: C 0620
: C 0621
: C 0622
: C 0623
: C 0624
: C 0625
: C 0626
: C 0627
: C 0628
: C 0629
: C 0630
: C 0631
: C 0632
: C 0633
: C 0634
: C 0635

```

3.3 EXERCISER ERRORS

Most errors reported during this test will originate from MSCP end message packets. The status code field will be converted to text and printed as part of a basic error message. Any subcode value will follow if extended error messages are enabled.

The following list represents some of the error conditions reported via MSCP:

1. Disk unit went offline (a sub-code may follow detailing the reason)
2. Compare error
3. Data error (a sub-code may follow)
4. Drive error (a sub-code may follow)
5. Host buffer access error
6. Media format error (a sub-code may follow)

3.4 ERROR LOG MESSAGES

The contents of the error-log messages received from the controller are printed as received, and should be deciphered using the MSCP specs.

3.5 MSCP ERRORS

An MSCP error occurs when the host receives an Invalid Command End Message from the RQDX1. In such cases, the host will print out the erroneous command followed by the reason for the error. If extended printouts are enabled, then the entire contents of the end message will be displayed in octal without interpretation of the data.

3.6 SAMPLE MSCP ERROR STATEMENT

The errors listed by the exerciser are usually very descriptive and are self explanatory. The following is an example error statement. This error statement is the extended error message.

(example)	(comments)
* DISK: XXX	!DISK UNIT NUMBER
CRN: XXXXX	!CONTROLLER PACKET NUMBER
MESSAGE TYPE: - SEQUENTIAL	!THIS IS THE PORT MESSAGE TYPE
COMMAND: -MSCP-READ-COMPARE	!CONNECTION ID AND COMMAND AND MODIFIER GIVEN TO DRIVE
STATUS CODE: UNIT OFFLINE	! STATUS CODE OF COMMAND
STATUS SUB-CODE: NO VOLUME MOUNTED OR DRIVE DISABLED BY SWITCH	! SUB CODE
BYTE COUNT IN COMMAND XXXXX	!NUMBER OF BYTES WANTED TO READ
ACTUAL # OF BYTES TRANSFERED XXXXX	!NUMBER OF BYTES ACTUALLY READ
I/O BUFFER ADDRESS (32 BITS) XXXXXX XXXXXX	
LBN: XXX	! LBN WANTED TO READ
END MESSAGE FLAGS:	! THERE ARE NO END FLAGS FOR THIS ERROR

The status code in an end messages is broken into two pieces. The first 5 bits represent the major status which is given by the "invalid command" message. The 11 remaining bits represent the sub-code, which tells in greater detail the error in the controller. The LBN is the logical block on the disk the controller was trying to read. The byte count refers to the number of bytes the controller was going to read off the LBN. The actual number of bytes transferred refers to the number of bytes read before the error. The end message flags give any flags that might have been set by the controller. It is pretty apparent that this error was caused by something physically switching the disk offline.

3.7 DUP ERRORS

A DUP error occurs when the host receives an Invalid Command End Message from the RQDX1. In such cases, the host will print out the erroneous command followed by the reason for the error.

There are two major places where errors in DUP will occur. The first being in the status code much the same as MSCP, and the second in the DUP I/O Buffer. Using the DUP sub-protocol (using Receive/Send commands to communicate with controller local programs (p. 25 of DUP.V05)) the controller may send an error

D2

ZRQAM1
V01.2

RD/RX EXERCISER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (15)

SEQ 0016
Page 16

: C 0689 message to the host by writing in the DUP I/O Buffer.

: C 0690
: C 0691
: C 0692
: C 0693
: C 0694
: C 0695
: C 0696
: C 0697
: C 0698
: C 0699
: C 0700
: C 0701
: C 0702
: C 0703
: C 0704
: C 0705
: C 0706
: C 0707
: C 0708
: C 0709
: C 0710
: C 0711
: C 0712
: C 0713
: C 0714
: C 0715
: C 0716
: C 0717
: C 0718
: C 0719
: C 0720
: C 0721
: C 0722
: C 0723
: C 0724
: C 0725
: C 0726
: C 0727
: C 0728
: C 0729
: C 0730

3.8 SAMPLE DUP ERROR STATEMENT

The errors listed by the exerciser are usually very descriptive and are self explanatory. The following is an example error statement. This error statement is the extended error message.

(example)	(comments)
* DISK: XXX	!DISK UNIT NUMBER
CRN: XXXXX	!CONTROLLER PACKET NUMBER
MESSAGE TYPE: - SEQUENTIAL	!THIS IS THE PORT MESSAGE TYPE
COMMAND: -DUP-RECEIVE DATA	!CONNECTION ID AND COMMAND GIVEN TO DRIVE
STATUS CODE: -SUCCESS	! STATUS CODE OF COMMAND
ACTUAL # OF BYTES TRANSFERED XXXXX	!NUMBER OF BYTES ACTUALLY READ
I/O BUFFER ADDRESS (32 BITS) XXXXXX XXXXXX	
MESSAGE TYPE:	! TYPE OF COMMUNICATION
** FATAL ERROR	
MESSAGE NUMBER:	! MESSAGE TOBE GIVEN OR RESULT EXPECTED
-- SUCCESS/FAILURE CODE	
MESSAGE ERROR CODE:	! ERROR LISTING
- ILLEGAL UNIT NUMBER	
DBN: XXX	

The DUP Error messages are almost like the MSCP messages but they contain an extra three classifications. These are the MESSAGE TYPE, MESSAGE NUMBER, MESSAGE ERROR CODE. These are part of a DUP sub-protocol for communicating with a controller local program. The I/O buffer address contains the address of the DUP I/O Buffer which contains the MESSAGE information.

The status code in this end message is succesful which means the controller was OK but it did not understand the host message or something on the host side caused the controller to error while running the controller local program. This error was produced by an ILLEGAL UNIT NUMBER. The disk went off line.

4.0 PERFORMANCE AND PROGRESS REPORTS

A summary report is printed at the end of each pass of the Exerciser or upon demand by the operator. The fields may be cleared to zero after the report is printed depending on the operator's response to this option in the software questions. Any units added to the test cycle will also begin with cleared statistics.

There are two basic listings one for LBNs and one for DBNs. The DBN listing will only contain RD51s for the simple reason that they contain DBNs and RX50s do not. The DBNs are READ and WRITTEN by blocks instead of by bytes. All units contain LBNs. The errors for each unit will be listed next to the LBNs.

Errors are grouped into two basic categories: hard and soft. Each is sub divided into four more categories, depending on the most probable classification for that error.

The sub categories are:

1. disk related errors
2. seek (or format) related errors
3. controller or drive related errors
4. host (the CPU) related errors.

All numeric values are in decimal radix.

UNIT		# OF BYTS	# OF	BYTES	-- HRD ERS --	-- SFT ERS --							
#	TYPE	READS	READ	WRITES	WRITTEN	DAT	SEK	DRV	HST	DAT	SEK	DRV	HST
X	RD51	XXXX	XXXX	XXXXX	XXXXXX	X	X	X	X	X	X	X	X
.	RX50
.	RX50

UNIT	DISK		# OF	# BLKS	# OF	# BLKS
#	#	TYPE	READS	READ	WRITES	WRITTEN
X	X	DBNRD51	XXXXX	XXXXX	XXXXX	XXXXX

: C 0731
: C 0732
: C 0733
: C 0734
: C 0735
: C 0736
: C 0737
: C 0738
: C 0739
: C 0740
: C 0741
: C 0742
: C 0743
: C 0744
: C 0745
: C 0746
: C 0747
: C 0748
: C 0749
: C 0750
: C 0751
: C 0752
: C 0753
: C 0754
: C 0755
: C 0756
: C 0757
: C 0758
: C 0759
: C 0760
: C 0761
: C 0762
: C 0763
: C 0764
: C 0765
: C 0766
: C 0767
: C 0768
: C 0769
: C 0770
: C 0771
: C 0772
: C 0773
: C 0774
: C 0775
: C 0776
: C 0777

```

: C 0778
: C 0779
: C 0780
: C 0781
: C 0782
: C 0783
: C 0784
: C 0785
: C 0786
: C 0787
: C 0788
: C 0789
: C 0790
: C 0791
: C 0792
: C 0793
: C 0794
: C 0795
: C 0796
: C 0797
: C 0798
: C 0799
: C 0800
: C 0801
: C 0802
: C 0803
: C 0804
: C 0805
: C 0806
: C 0807
: C 0808
: C 0809
: C 0810
: C 0811
: C 0812
: C 0813
: C 0814
: C 0815
: C 0816
: C 0817
: C 0818
: C 0819
: C 0820
: C 0821
: C 0822
: C 0823
: C 0824
: C 0825
: C 0826
: C 0827
: C 0828
: C 0829

```

5.0 TEST SUMMARY

The RQDX1 functional tester and exerciser consists of two parts, the initialization subtest and the performance exerciser. The operator is not able to select which of these two parts he/she wishes to run; they both must be executed.

5.1 INITIALIZATION SUBTEST

The purpose of this subtest is to verify the hardware configuration as specified by the operator, and to bring each unit online. The Initialization Subtest will always precede the execution of any other test.

First, the presence of each device register will be verified, along with a check on the BR level specified by the operator. Then, an initialization will be issued to each controller configured for testing. When the initialization sequence has been completed, an attempt will be made to bring each unit online. If this succeeds, one or two MSCP reads will be issued to the inner-most LBN of each selected disk to ensure that each disk drive can seek and be read.

Any device-fatal or hard errors encountered during this test will cause the appropriate unit(s) to be dropped. If basic error messages are enabled, then the program will print out the specific reason for dropping the unit(s). Henceforth, the failed unit(s) will not be tested unless the operator intervenes (adds unit(s) or restarts Exerciser).

Upon successful completion of the Initialization Subtest, the program will begin executing the Exerciser.

5.2 EXERCISER

The purpose of this subtest is to exercise the disk drives in a manner similar to the typical usage under standard operating systems. Execution of this test should give an indication of the operating performance of the disk drive subunits. This test will utilize random disk addresses, random word counts, and data patterns, all subject to the limits and specifications made by the operator. All protected disks will be subject to read-only operations, while unprotected disks may be read or written, depending on the answers given to the software parameter questions. End-of-pass will be declared when the specified number of bytes have been transferred for all the disks taken as a whole.

: C 0830
 : C 0831
 : C 0832
 : C 0833
 : C 0834
 : C 0835
 : C 0836
 : C 0837
 : C 0838
 : C 0839
 : C 0840
 : C 0841
 : C 0842
 : C 0843
 : C 0844
 : C 0845
 : C 0846
 : C 0847
 : C 0848
 : C 0849
 : C 0850
 : C 0851
 : C 0852
 : C 0853
 : C 0854
 : C 0855
 : C 0856
 : C 0857
 : C 0858
 : C 0859
 : C 0860
 : C 0861
 : C 0862
 : C 0863
 : C 0864
 : C 0865
 : C 0866
 : C 0867
 : C 0868
 : C 0869
 : C 0870
 : C 0871
 : C 0872
 : C 0873
 : C 0874
 : C 0875
 : C 0876

If a read/write error occurs during this test, then the RQDX1 CONTROLLER will initiate an appropriate number of retries. If all retries fail, then a hard error will be reported to the host, an error message will be displayed on the console terminal and the error will be tallied for the summary report. The unit will be dropped if the hard error count has exceeded the specified limit.

The Exerciser is actually two exercisers combined together. The main MSCP exerciser writes and reads LBNs while the less used DUP exerciser writes and reads DBNs. The DUP exerciser is used once per 25 LBN transfers or slot less than the MSCP exerciser. The two Exercisers use a somewhat different protocol to transfer I/O. It is possible to go from MSCP protocol to DUP protocol without reinitializing the controller but impossible to go from DUP to MSCP protocol. Therefore after the DUP Exercise pass the controller is reinitialize and control given back to the MSCP Exerciser. The reinitialization process takes a few seconds. For this reason a variable is placed in the DUP Exerciser to allow it to transfer more than 1 DBN per pass. The variable, "X", is multiplied by 25 to give the amount of MSCP transfers before the DUP Exerciser can transfer "X" amount of DBNs. The higher the variable the less reinit the controller must do.

5.3 DROP UNIT SUMMARY

During the Initialization Subtest, individual units will be dropped from the test sequence if they are unable to be brought online or the operator specified device does not match the hardware.

During the Exercise, the program will drop a unit for one of three reasons. The normal path is for each unit to complete the transfer of N megabytes, where N is specified by the operator during SW questioning and be soft-dropped. Otherwise, a unit will be hard-dropped if the number of hard errors encountered exceeds the operator-specified limit, or if a fatal error is detected. Units hard-dropped may later be added to the test cycle. However, statistics for the hard-added unit will be cleared to zero; if a transfer limit was specified, in which case the unit was soft-dropped, the statistics may or may not be cleared depending on the operators answer to Software question 12.

6.0 ERROR CODES GENERATED BY ZRQA EXERCISER

SYSTEM FATAL ERRORS

1 More than 4 units specified

DEVICE FATAL ERRORS

10	Controller couldn't be addressed at the address given.	Wrong IP address selected
11	Controller didn't interrupt at the interrupt vector given.	Wrong vector address selected.
12	Controller didn't interrupt at the BR level given.	Wrong BR level selected.
13	Init sequence failed.	Either one of the four initialization steps did not receive the correct response from the Controller, or one of the steps timed-out.
14	Fatal Controller error.	The error bit (bit 15) in the SA register was set.
15	Failed to bring unit on-line.	On-line response had an error code. (see also #s 22 and 23.)
16	Write protect conflict.	The unit was hardware write protected and write operations were requested on the unit.
17	Access to inner track failed.	Innermost track's header may be corrupted.
18	Unit went off-line.	---
20	Failed to send 'Set Controller Characteristics' command.	Either the unit is off-line or the Diagnostic is corrupted because of any problems with its RAM.

```

: C 0877
: C 0878
: C 0879
: C 0880
: C 0881
: C 0882
: C 0883
: C 0884
: C 0885
: C 0886
: C 0887
: C 0888
: C 0889
: C 0890
: C 0891
: C 0892
: C 0893
: C 0894
: C 0895
: C 0896
: C 0897
: C 0898
: C 0899
: C 0900
: C 0901
: C 0902
: C 0903
: C 0904
: C 0905
: C 0906
: C 0907
: C 0908
: C 0909
: C 0910
: C 0911
: C 0912
: C 0913
: C 0914
: C 0915
: C 0916
: C 0917
: C 0918
: C 0919
: C 0920
: C 0921
: C 0922
: C 0923
: C 0924
: C 0925
: C 0926
: C 0927

```



```

: C 0928
: C 0929
: C 0930      21 Controller returned wrong 'end
: C 0931      code' for the 'Set Controller
: C 0932      Characteristics' command.      Problem with the Control-
: C 0933      DMA interface.
: C 0934      22 Failed to send 'On-line' command      Either the unit is off-
: C 0935      line or the diagnostic is
: C 0936      corrupted because of any
: C 0937      problems with its RAM.
: C 0938
: C 0939      23 Controller returned wrong 'end
: C 0940      code' for the 'On-line' command.      Problem with the Control-
: C 0941      ler's microcode or the
: C 0942      port/DMA interface.
: C 0943      24 Device went to available state      Fault switch or door mechanism
: C 0944
: C 0945
: C 0946      25 Unknown device type      Drive not identifying itself
: C 0947      properly
: C 0948
: C 0949
: C 0950      HARD ERRORS
: C 0951      -----
: C 0952
: C 0953      MSCP ERRORS
: C 0954
: C 0955      31 Controller received an invalid
: C 0956      command.      The diagnostic is corrup-
: C 0957      ted because of any prob-
: C 0958      lems with its RAM, or
: C 0959      there is a problem with
: C 0960      the Controller microde
: C 0961      (RAM or ROM) or there is
: C 0962      problem with the port/DMA
: C 0963      interface.
: C 0964      32 Command aborted by the Control-
: C 0965      ler.      Command timed-out in the
: C 0966      Controller.
: C 0967      35 Media format error.      ---
: C 0968
: C 0969      36 Device write protected.      ---
: C 0970
: C 0971      37 Controller read or write com-
: C 0972      pare error.      ---
: C 0973
: C 0974      38 Data error.      CRC error in the data
: C 0975      field of a disk block.
: C 0976
: C 0977      39 Host buffer access error      ---
: C 0978
: C 0979      40 Controller error.      Difficult to catagorize
: C 0980      without looking at the

```

K2

ZRQAM1
V01.2

RD/RX EXERCISER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (21)

SEQ 0023
Page 23

: C 0981
: C 0982
: C 0983

error sub-code or any
associated error-log mes-
sage.

:	C 0984		
:	C 0985		
:	C 0986	41 Drive error.	See #40.
:	C 0987		
:	C 0988	42 Host write compare error.	Error detected when Host
:	C 0989		CPU compared the data
:	C 0990		written and read back. May
:	C 0991		be a problem with the Host
:	C 0992		or Controller RAM.
:	C 0993		
:	C 0994	43 Message from internal diagnostics	See #40.
:	C 0995		
:	C 0996	44 Duplicate unit number detected	---
:	C 0997	by the Controller.	
:	C 0998		
:	C 0999	45 Unknown end code received.	Problem with the Control-
:	C 1000		ler microcode or the port/
:	C 1001		DMA interface.
:	C 1002		
:	C 1003	DUP ERRORS	
:	C 1004	Host found error	
:	C 1005	46 DBN compare error.	see # 42
:	C 1006		
:	C 1007	Message errors	
:	C 1008	47 No local media	Controller local program on
:	C 1009		RAM may be corrupt
:	C 1010		
:	C 1011	48 Illegal Unit #	Unit went offline
:	C 1012		
:	C 1013	49 Illegal relative or physical BLK #	see # 31
:	C 1014		
:	C 1015	50 Device Error	Possible write protected
:	C 1016		
:	C 1017	51 Zero length message	see # 31
:	C 1018		
:	C 1019	Status errors	
:	C 1020	52 Invalid Command	see # 31
:	C 1021		
:	C 1022	53 No region available	see # 31
:	C 1023		
:	C 1024	54 No region suitable	see # 31
:	C 1025		
:	C 1026	55 Program not known	see # 47
:	C 1027		
:	C 1028	56 Load failure	---
:	C 1029		
:	C 1030	57 Stand alone type program	see # 31
:	C 1031		
:	C 1032	58 DUP unkown status code	see # 31

: C 1033
: C 1034
: C 1035
: C 1036
: C 1037
: C 1038
: C 1039
: C 1040
: C 1041
: C 1042
: C 1043
: C 1044
: C 1045
: C 1046
: C 1047
: C 1048
: C 1049

SOFT ERRORS

- 60 Controller error. See error-log packet for details as the exact cause may not be evident.
- 61 Host memory access error. See #50.
- 62 Disk transfer error. See #50.
- 63 'Standard Disk Inteconnect' error. See #50.
- 64 'Small Disk' error. See #50.

7.0 DATA PATTERNS

	HEX	OCTAL	BINARY
	---	-----	-----
C 1050			
C 1051			
C 1052			
C 1053			
C 1054			
C 1055			
C 1056			
C 1057			
C 1058			
C 1059	Pattern 1	R A N D O M	N U M B E R S
C 1060			
C 1061	Pattern 2	0000	000000 0 000 000 000 000 000
C 1062			
C 1063	Pattern 3	FFFF	1 111 111 111 111 111
C 1064			
C 1065	Pattern 4	8B8B	1 000 101 110 001 011
C 1066			
C 1067	Pattern 5	3333	0 011 001 100 110 011
C 1068			
C 1069	Pattern 6	3091	0 011 000 010 010 001
C 1070			
C 1071	Pattern 7	0001	0 000 000 000 000 001
C 1072		0003	0 000 000 000 000 011
C 1073		0007	0 000 000 000 000 111
C 1074		000F	0 000 000 000 001 111
C 1075		001F	0 000 000 000 011 111
C 1076		003F	0 000 000 000 111 111
C 1077		007F	0 000 000 001 111 111
C 1078		00FF	0 000 000 011 111 111
C 1079		01FF	0 000 000 111 111 111
C 1080		03FF	0 000 001 111 111 111
C 1081		07FF	0 000 011 111 111 111
C 1082		0FFF	0 000 111 111 111 111
C 1083		1FFF	0 001 111 111 111 111
C 1084		3FFF	0 011 111 111 111 111
C 1085		7FFF	0 111 111 111 111 111
C 1086		FFFF	1 111 111 111 111 111
C 1087			
C 1088			
C 1089	Pattern 8	FFFE	1 111 111 111 111 110
C 1090		FFFC	1 111 111 111 111 100
C 1091		FFF8	1 111 111 111 111 000
C 1092		FFF0	1 111 111 111 110 000
C 1093		FFE0	1 111 111 111 100 000
C 1094		FFC0	1 111 111 111 000 000
C 1095		FF80	1 111 111 110 000 000
C 1096		FF00	1 111 111 100 000 000
C 1097		FE00	1 111 111 000 000 000
C 1098		FC00	1 111 110 000 000 000
C 1099		F800	1 111 100 000 000 000
C 1100		F000	1 111 000 000 000 000
C 1101		E000	1 110 000 000 000 000
C 1102		C000	1 100 000 000 000 000

B3

ZRQAM1
V01.2

RD/RX EXERCISER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (24)

SEQ 0027
Page 27

:	C 1103	8000	100000
:	C 1104	0000	000000

1	000	000	000	000	000
0	000	000	000	000	000

:	C	1105									
:	C	1106									
:	C	1107	Pattern 9	0000	000000	0	000	000	000	000	000
:	C	1108		0000	000000	0	000	000	000	000	000
:	C	1109		0000	000000	0	000	000	000	000	000
:	C	1110		FFFF	177777	1	111	111	111	111	111
:	C	1111		FFFF	177777	1	111	111	111	111	111
:	C	1112		FFFF	177777	1	111	111	111	111	111
:	C	1113		0000	000000	0	000	000	000	000	000
:	C	1114		0000	000000	0	000	000	000	000	000
:	C	1115		FFFF	177777	1	111	111	111	111	111
:	C	1116		FFFF	177777	1	111	111	111	111	111
:	C	1117		0000	000000	0	000	000	000	000	000
:	C	1118		FFFF	177777	1	111	111	111	111	111
:	C	1119		0000	000000	0	000	000	000	000	000
:	C	1120		FFFF	177777	1	111	111	111	111	111
:	C	1121		0000	000000	0	000	000	000	000	000
:	C	1122		FFFF	177777	1	111	111	111	111	111
:	C	1123									
:	C	1124	Pattern 10	B6D9	133331	1	011	011	011	011	001
:	C	1125									
:	C	1126	Pattern 11	5555	052525	0	101	010	101	010	101
:	C	1127		5555	052525	0	101	010	101	010	101
:	C	1128		5555	052525	0	101	010	101	010	101
:	C	1129		AAAA	125252	1	010	101	010	101	010
:	C	1130		AAAA	125252	1	010	101	010	101	010
:	C	1131		AAAA	125252	1	010	101	010	101	010
:	C	1132		5555	052525	0	101	010	101	010	101
:	C	1133		5555	052525	0	101	010	101	010	101
:	C	1134		AAAA	125252	1	010	101	010	101	010
:	C	1135		AAAA	125252	1	010	101	010	101	010
:	C	1136		5555	052525	0	101	010	101	010	101
:	C	1137		AAAA	125252	1	010	101	010	101	010
:	C	1138		5555	052525	0	101	010	101	010	101
:	C	1139		AAAA	125252	1	010	101	010	101	010
:	C	1140		5555	052525	0	101	010	101	010	101
:	C	1141		AAAA	125252	1	010	101	010	101	010
:	C	1142									
:	C	1143	Pattern 12	2D2D	026455	0	010	110	100	101	101
:	C	1144		2D2D	026455	0	010	110	100	101	101
:	C	1145		2D2D	026455	0	010	110	100	101	101
:	C	1146		D2D2	151322	1	101	001	011	010	010
:	C	1147		D2D2	151322	1	101	001	011	010	010
:	C	1148		D2D2	151322	1	101	001	011	010	010
:	C	1149		2D2D	026455	0	010	110	100	101	101
:	C	1150		2D2D	026455	0	010	110	100	101	101
:	C	1151		D2D2	151322	1	101	001	011	010	010
:	C	1152		D2D2	151322	1	101	001	011	010	010
:	C	1153		2D2D	026455	0	010	110	100	101	101
:	C	1154		2D2D	026455	0	010	110	100	101	101
:	C	1155		D2D2	151322	1	101	001	011	010	010
:	C	1156		2D2D	026455	0	010	110	100	101	101
:	C	1157		D2D2	151322	1	101	001	011	010	010

D3

ZRQAM1
V01.2

RD/RX EXERCISER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (25)

SEQ 0029
Page 29

:	C 1158	2020	026455	0 010 110 100 101 101
:	C 1159	D2D2	151322	1 101 001 011 010 010
:	C 1160	2020	026455	0 010 110 100 101 101
:	C 1161	D2D2	151322	1 101 001 011 010 010
:	C 1162	2020	026455	0 010 110 100 101 101

E3

ZRQAM1
V01.2

RD/RX EXERCISER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (26)

SEQ 0030
Page 30

:	C 1163								
:	C 1164								
:	C 1165	Pattern 13	6DB6	066666	0	110	110	110	110
:	C 1166								
:	C 1167	Pattern 14	0001	000001	0	000	000	000	001
:	C 1168		0002	000002	0	000	000	000	010
:	C 1169		0004	000004	0	000	000	000	100
:	C 1170		0008	000010	0	000	000	000	001
:	C 1171		0010	000020	0	000	000	000	010
:	C 1172		0020	000040	0	000	000	000	100
:	C 1173		0040	000100	0	000	000	001	000
:	C 1174		0080	000200	0	000	000	010	000
:	C 1175		0100	000400	0	000	000	100	000
:	C 1176		0200	001000	0	000	001	000	000
:	C 1177		0400	002000	0	000	010	000	000
:	C 1178		0800	004000	0	000	100	000	000
:	C 1179		1000	010000	0	001	000	000	000
:	C 1180		2000	020000	0	010	000	000	000
:	C 1181		4000	040000	0	100	000	000	000
:	C 1182		8000	100000	1	000	000	000	000
:	C 1183								
:	C 1184	Pattern 15	FFFE	177776	1	111	111	111	110
:	C 1185		FFFD	177775	1	111	111	111	101
:	C 1186		FFFB	177773	1	111	111	111	011
:	C 1187		FFF7	177767	1	111	111	111	110
:	C 1188		FFEF	177757	1	111	111	111	101
:	C 1189		FFDF	177737	1	111	111	111	011
:	C 1190		FFBF	177677	1	111	111	110	111
:	C 1191		FF7F	177577	1	111	111	101	111
:	C 1192		FEFF	177377	1	111	111	011	111
:	C 1193		FDFE	176777	1	111	110	111	111
:	C 1194		F8FF	175777	1	111	101	111	111
:	C 1195		F7FF	173777	1	111	011	111	111
:	C 1196		EFFE	167777	1	110	111	111	111
:	C 1197		DFFF	157777	1	101	111	111	111
:	C 1198		BFFF	137777	1	011	111	111	111
:	C 1199		7FFF	077777	0	111	111	111	111
:	C 1200								
:	C 1201	Pattern 16	B6D9	133331	1	011	011	011	001
:	C 1202		B6D9	133331	1	011	011	011	001
:	C 1203		B6D9	133331	1	011	011	011	001
:	C 1204		D86C	155554	1	101	101	101	100
:	C 1205		D86C	155554	1	101	101	101	100
:	C 1206		D86C	155554	1	101	101	101	100
:	C 1207		B6D9	133331	1	011	011	011	001
:	C 1208		B6D9	133331	1	011	011	011	001
:	C 1209		D86C	155554	1	101	101	101	100
:	C 1210		D86C	155554	1	101	101	101	100
:	C 1211		B6D9	133331	1	011	011	011	001
:	C 1212		D86C	155554	1	101	101	101	100
:	C 1213		B6D9	133331	1	011	011	011	001
:	C 1214		D86C	155554	1	101	101	101	100
:	C 1215		B6D9	133331	1	011	011	011	001

F3

ZRQAM1
V01.2

RD/RX EXERCISER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 B1es-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (26)

SEQ 0031
Page 31

: C 1216

DB6C 155554

1 101 101 101 101 100

		LBN	LBN	LBN					
:	C 1217								
:	C 1218								
:	C 1219	Pattern 17	LBN	LBN	LBN				
:	C 1220		8D36	106466	1 000	110	100	110	110
:	C 1221		8D36	106466	1 000	110	100	110	110
:	C 1222		72C9	071311	0 111	001	011	001	001
:	C 1223		72C9	071311	0 111	001	011	001	001
:	C 1224		72C9	071311	0 111	001	011	001	001
:	C 1225		8D36	106466	1 000	110	100	110	110
:	C 1226		8D36	106466	1 000	110	100	110	110
:	C 1227		8D36	106466	1 000	110	100	110	110
:	C 1228		8D36	106466	1 000	110	100	110	110
:	C 1229		72C9	071311	0 111	001	011	001	001
:	C 1230		72C9	071311	0 111	001	011	001	001
:	C 1231		72C9	071311	0 111	001	011	001	001
:	C 1232		72C9	071311	0 111	001	011	001	001
:	C 1233		72C9	071311	0 111	001	011	001	001
:	C 1234		8D36	106466	1 000	110	100	110	110
:	C 1235		8D36	106466	1 000	110	100	110	110
:	C 1236		8D36	106466	1 000	110	100	110	110
:	C 1237		8D36	106466	1 000	110	100	110	110
:	C 1238		8D36	106466	1 000	110	100	110	110
:	C 1239		8D36	106466	1 000	110	100	110	110
:	C 1240								
:	C 1241	Pattern 18	8D36	106466	1 000	110	100	110	110
:	C 1242		LBN	LBN	LBN				
:	C 1243		72C9	071311	0 111	001	011	001	001
:	C 1244		8D36	106466	1 000	110	100	110	110
:	C 1245		8D36	106466	1 000	110	100	110	110
:	C 1246		8D36	106466	1 000	110	100	110	110
:	C 1247		72C9	071311	0 111	001	011	001	001
:	C 1248		72C9	071311	0 111	001	011	001	001
:	C 1249		72C9	071311	0 111	001	011	001	001
:	C 1250		72C9	071311	0 111	001	011	001	001
:	C 1251		8D36	106466	1 000	110	100	110	110
:	C 1252		8D36	106466	1 000	110	100	110	110
:	C 1253		8D36	106466	1 000	110	100	110	110
:	C 1254		8D36	106466	1 000	110	100	110	110
:	C 1255		8D36	106466	1 000	110	100	110	110
:	C 1256		72C9	071311	0 111	001	011	001	001
:	C 1257		72C9	071311	0 111	001	011	001	001
:	C 1258		72C9	071311	0 111	001	011	001	001
:	C 1259		72C9	071311	0 111	001	011	001	001
:	C 1260		72C9	071311	0 111	001	011	001	001
:	C 1261		72C9	071311	0 111	001	011	001	001

		LBN	LBN	LBN		
:	C 1262					
:	C 1263					
:	C 1264	Pattern 19				
:	C 1265	B999	134631	1 011 100 110 011 001		
:	C 1266	B999	134631	1 011 100 110 011 001		
:	C 1267	4666	043146	0 100 011 001 100 110		
:	C 1268	4666	043146	0 100 011 001 100 110		
:	C 1269	4666	043146	0 100 011 001 100 110		
:	C 1270	B999	134631	1 011 100 110 011 001		
:	C 1271	B999	134631	1 011 100 110 011 001		
:	C 1272	B999	134631	1 011 100 110 011 001		
:	C 1273	B999	134631	1 011 100 110 011 001		
:	C 1274	4666	043146	0 100 011 001 100 110		
:	C 1275	4666	043146	0 100 011 001 100 110		
:	C 1276	4666	043146	0 100 011 001 100 110		
:	C 1277	4666	043146	0 100 011 001 100 110		
:	C 1278	4666	043146	0 100 011 001 100 110		
:	C 1279	B999	134631	1 011 100 110 011 001		
:	C 1280	B999	134631	1 011 100 110 011 001		
:	C 1281	B999	134631	1 011 100 110 011 001		
:	C 1282	B999	134631	1 011 100 110 011 001		
:	C 1283	B999	134631	1 011 100 110 011 001		
:	C 1284	B999	134631	1 011 100 110 011 001		
:	C 1285					
:	C 1286	Pattern 20	B999 134631	1 011 100 110 011 001		
:	C 1287		LBN			
:	C 1288		4666 043146	0 100 011 001 100 110		
:	C 1289		B999 134631	1 011 100 110 011 001		
:	C 1290		B999 134631	1 011 100 110 011 001		
:	C 1291		B999 134631	1 011 100 110 011 001		
:	C 1292		4666 043146	0 100 011 001 100 110		
:	C 1293		4666 043146	0 100 011 001 100 110		
:	C 1294		4666 043146	0 100 011 001 100 110		
:	C 1295		4666 043146	0 100 011 001 100 110		
:	C 1296		B999 134631	1 011 100 110 011 001		
:	C 1297		B999 134631	1 011 100 110 011 001		
:	C 1298		B999 134631	1 011 100 110 011 001		
:	C 1299		B999 134631	1 011 100 110 011 001		
:	C 1300		B999 134631	1 011 100 110 011 001		
:	C 1301		4666 043146	0 100 011 001 100 110		
:	C 1302		4666 043146	0 100 011 001 100 110		
:	C 1303		4666 043146	0 100 011 001 100 110		
:	C 1304		4666 043146	0 100 011 001 100 110		
:	C 1305		4666 043146	0 100 011 001 100 110		
:	C 1306		4666 043146	0 100 011 001 100 110		
:	C 1307					
:	C 1308	Pattern 21	LBN LBN	LBN		
:	C 1309					
:	C 1310)				

ZRQAM1
VOL.2RD/RX EXERCISER
PROGRAM HEADER5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (29)

Page 34

```
: 1311 #sbttl 'PROGRAM HEADER'
: 1312
: 1313 library 'ZRQACO.L16';           ! RDRX EXERCISER GLOBAL LIBRARY
: 1314
: 1315 require 'BLSMAC.REQ';         ! DIAGNOSTIC SUPERVISOR LIBRARY
: 2806
: 2807 literal
: 2808     DS#NBR_OF_TESTS = 1;       ! NUMBER OF TESTS IN THIS DIAGNOSTIC
: 2809
: 2810 EQUALS;
: 2811
: 2812 POINTER (ALL);
: 2813
: 2814 !+
: 2815 ! THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: 2816 ! THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
: 2817 !-
: 2818
: 2819 HEADER (#ascii'ZRQA', #ascii'C', #ascii'O', 32767, 1, PRI00);
```

J3

ZRQAM1
V01.2

RD/RX EXERCISER
DISPATCH TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (30)

SEQ 0035
Page 35

```
: 2820 #sbttl 'DISPATCH TABLE'  
:  
: 2821  
: 2822 !+  
: 2823 ! THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.  
: 2824 ! IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.  
: 2825 !-  
:  
: 2826  
: 2827 DISPATCH (DS#NBR_OF_TESTS);
```

```

: 2828 #sbttl 'GLOBAL DATA SECTION'
: 2829
: 2830 !+
: 2831 ! THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
: 2832 ! IN MORE THAN ONE TEST.
: 2833 !-
: 2834
: 2835 psect
: 2836     global = $FFF$ (read, nowrite, execute, local, concatenate);
: 2837
: 2838 global
: 2839     CST : blockvector [MAX_CTLR, CST_LEN, word] field (CST_FIELDS),
: 2840           ! RUN-TIME CONTROLLER STATUS TABLES
: 2841     CST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 2842           ! CONTROLLER STATUS TABLE ADDRESS OF "CURRENT" CONTROLLER
: 2843     DCT : blockvector [MAX_CTLR, DCT_LEN, word] field (DCT_FIELDS),
: 2844           ! DRIVER CONTROLLER TABLES
: 2845     DCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 2846           ! ADDRESS OF "CURRENT" DRIVER CONTROLLER TABLE
: 2847     RDRX_ADDR : ref rdrx field (RC_REG),
: 2848           ! DEVICE ADDRESS OF "CURRENT" CONTROLLER
: 2849     IRDRX_ADDR : ref rdrx field (RC_REG),
: 2850     ICOM_ADDR : ref block [COM_LEN, word] field (COM_FIELDS),
: 2851           ! ADDRESS OF INTERRUPTING CONTROLLER'S COMMUNICATION AREA
: 2852     ICST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 2853           ! ADDRESS OF INTERRUPTING CONTROLLER'S CST
: 2854     IDCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 2855           ! ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 2856     RDM_CNT : word initial (RDM_LEN),           ! NUMBER OF RANDOM NUMBERS \ KEEP
: 2857     RANDOM : vector [RDM_LEN, word],           ! RANDOM NUMBER TABLE / TOGETHER
: 2858           ! DEVICE ADDRESS OF INTERRUPTING CONTROLLER
: 2859     TRK_SGN : vector [MAX_UNITS, byte, signed] initial (byte (rep MAX_UNITS of (1))),           ! CURRENT TRACK DIRE
:
: 2860     DUPPKT : BLOCK [257, WORD] field (DP_FIELDS),
: 2861           ! BUFFER CONTAINING DUP INFORMATION FROM RECEIVE AND SEND COMMANDS
: 2862     BST : blockvector [MAX_UNITS, 2, WORD],
: 2863           ! CONTAIN LBNS(LO & HI FIELDS) FOR SEQUENTIAL I/O TRANSFER FOR EACH UNIT
: 2864     TALLY : vector [MAX_UNITS * TALLY_LEN, word] field (T_FIELDS),
: 2865           ! STATISTICS TABLES
: 2866     T_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 2867           ! ADDRESS OF STATISTICS TABLE (TALLY) FOR CURRENT UNIT
: 2868     C_ERR_TBL : blockvector [MAX_CTLR, C_ERR_LEN, word] field (C_ERR_FIELDS),
: 2869           ! STATISTICS TABLE FOR CONTROLLER ERRORS
: 2870     MSCP_PKT : blockvector [PKT_CNT, PKT_LEN, word] field (PKT_FIELDS),
: 2871           ! MSCP PACKET POOL
: 2872     IPKT_ADDR : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 2873           ! ADDRESS OF AN MSCP PACKET (INTERUPT PROCESSING)
: 2874     PKT_USE : vector [PKT_CNT, byte, signed],
: 2875           ! MSCP PACKET POOL ALLOCATION TABLE
: 2876     RETPKT : blockvector [RP_CNT, RP_LEN, word] field (RP_FIELDS),
: 2877           ! RETURN PACKET POOL
: 2878     RP_USE : vector [RP_CNT, byte, signed],
: 2879           ! RETURN PACKET POOL ALLOCATION TABLE
: 2880     RP_INDX : word,           ! CURRENT RETURN PACKET INDEX

```

```

: 2881 RP_ADDR : ref block [RP_LEN, word] field (RP_FIELDS),
: 2882 ! CURRENT RETURN PACKET ADDRESS
: 2883 ELOG_PKT : blockvector [EP_CNT, EP_LEN, word] field (EP_FIELDS),
: 2884 ! ERROR-LOG PACKET SAVE AREA
: 2885 BUFF_ADDR : vector [MAX_BUF_CNT], ! TABLE OF I/O BUFFER DESCRIPTORS
: 2886 BUFF_OWN : vector [MAX_BUF_CNT, byte, signed], ! I/O BUFFER OWNERSHIP (CTLR NUMBER)
: 2887 IODQ : vector [IODQ_LEN, byte],
: 2888 ! I/O DONE QUEUE - CIRCULAR QUEUE OF RETPKT INDECES
: 2889 IODQ_IN : word, ! I/O DONE QUEUE IN POINTER
: 2890 IODQ_OUT : word, ! I/O DONE QUEUE OUT POINTER
: 2891 ENTRY_REASON : byte, ! CURRENT OPERATOR COMMAND
: 2892 EOP_FLAG : byte, ! END-OF-PASS FLAG
: 2893 DUP_FLAGS : WORD, ! DUP FLAGS
: 2894 CCTLR : word, ! NUMBER OF "CURRENT" CTLR
: 2895 CDISK : word, ! CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 2896 CUOFF : word, ! CURRENT UNIT CST OFFSET
: 2897 CTLR_CNT : word, ! TOTAL NUMBER OF CONFIGURED CTLRS
: 2898 DUR : vector [MAX_UNITS, byte], ! DROP UNIT REASON
: 2899 GIO : vector [MAX_CTLR, byte], ! NUMBER OF OUTSTANDING GIOS PER CTLR
: 2900 FREE_MEM_ADDR, ! START OF FREE MEMORY
: 2901 BYTS_PER_GIO : word, ! SIZE (BYTES) OF AN I/O BUFFER
: 2902 ST_CODE : word, ! CURRENT STATUS CODE
: 2903 SB_CODE : word, ! CURRENT SUB-CODE
: 2904 STEP : word, ! CURRENT STEP IN HARD_INIT
: 2905 OF_RC : signed word, ! OFFSET (0 OR 2) TO READ IP OR SA
: 2906 SA_REG : word, ! STORAGE FOR SA REGISTER READS AND WRITES
: 2907 CMD_TIME : word, ! COMMAND TIMEOUT VALUE (IN SECONDS)
: 2908 NEX : word, ! NON-EXISTENT MEMORY TRAP INDICATOR
: 2909 CRN_LOW : word, ! COMMAND REF NUMBER OF LAST COMMAND SENT
: 2910 CRN_HIGH : word, ! COMMAND REF NUMBER (HI ORDER)
: 2911 P_INDEX : signed word, ! CURRENT message PACKET INDEX
: 2912 S_DUPPKT : WORD, ! DBN BYTE COUNTER
: 2913 S_PATTERN : WORD, ! THE PATTERN WRITTEN TO DBN'S
: 2914 CREDIT_BAL : word, ! CREDIT BALANCE
: 2915 INIT_OCCURED : BYTE, ! INDICATES IF EXERCISER FINISHED INIT SEQUENCE
: 2916 NXT_PKT_2USE : byte, ! POINTER TO NEXT ENTRY IN PKT_USE TABLE
: 2917
: 2918 ERR_TBL;

```



```

: 2919 #sbttl 'GLOBAL TEXT SECTION'
: 2920
: 2921 !+
: 2922 ! THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: 2923 ! MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: 2924 ! MORE THAN ONE TEST.
: 2925 !-
: 2926
: 2927 global bind
: 2928 !
: 2929 ! HARDWARE DIALOG
: 2930 !
: 2931 HWQ1 = uplit (#asciz'IP ADDRESS'),
: 2932 HWQ2 = uplit (#asciz'VECTOR'),
: 2933 HWQ3 = uplit (#asciz'BR LEVEL'),
: 2934 HWQ4 = uplit (#asciz'RD/RX DRIVE NUMBER'),
: 2935 HWQ6 = uplit (#asciz'STARTING LBN'),
: 2936 HWQ7 = uplit (#asciz'ENDING LBN (MAXIMUM - RX50: 799, RD51: 21599, RD52: 47999)'),
: 2937 HWQ8 = uplit (#asciz'WRITE ON CUSTOMER DATA AREA ON THIS DISK'),
: 2938 HWQ9 = uplit (#asciz'** WARNING - CUSTOMER DATA AREA MAY BE OVERWRITTEN! ... CONFIRM'),
: 2939 HWQ10 = uplit (#asciz'EXERCISE ON DIAGNOSTIC AREA (-NON-CUSTOMER AREA RDs ONLY)'),
: 2940 HWQ11 = uplit (#asciz'WRITE ON DIAGNOSTIC AREA'),
: 2941
: 2942 !
: 2943 ! SOFTWARE DIALOG
: 2944 !
: 2945 SWQ1 = uplit (#asciz'HARD ERROR LIMIT'),
: 2946 SWQ2 = uplit (#asciz'TRANSFER LIMIT IN MEGABYTES (0 for "QUICK PASS")'),
: 2947 SWQ4 = uplit (#asciz'RANDOM UNIT MODE'),
: 2948 SWQ7 = uplit (#asciz'READ-COMPARES PERFORMED AT THE CTRL'),
: 2949 SWQ9 = uplit (#asciz'WRITE-COMPARES PERFORMED AT THE CTRL'),
: 2950 SWQ10 = uplit (#asciz'CHECK ALL WRITES AT MOST BY READING'),
: 2951 SWQ11 = uplit (#asciz'USER-DEFINED DATA PATTERN'),
: 2952 SWQ12 = uplit (#asciz'SELECT PRE-DEFINED DATA PATTERN (0 for sequential selection)'),
: 2953 SWQ13 = uplit (#asciz'NUMBER OF WORDS IN DATA PATTERN (16 maximum)'),
: 2954 SWQ14 = uplit (#asciz'PATTERN VALUE'),
: 2955 SWQ15 = uplit (#asciz'CLEAR STATISTICAL TABLES AFTER PRINTING'),
: 2956 SWQ17 = uplit (#asciz'PERCENTAGE OF WINCHESTER OPERATIONS OUT OF TOTAL OPERATIONS'),
: 2957 SWQ19 = uplit (#asciz'RANDOM BLOCK MODE'),
: 2958 ! SWQ20 = uplit (#asciz'WANT TO REWRITE BLOCKS WHEN "FORCED ERROR" DETECTED ON READS'),
: 2959 ! SWQ21 = uplit (#asciz'IF "HALT ON ERROR" FLAG SET, WANT TO HALT ON HARD/SOFT ERRORS ALSO'),
: 2960 SWQ22 = uplit (#asciz'NUMBER OF DBNs TRANSFERRED AT ONE "DUP" PASS'),
: 2961 SWM1 = uplit (#asciz'THE REMAINING QUESTIONS ONLY APPLY TO UNPROTECTED DISKS'),
: 2962 NULL = uplit (#asciz''),
: 2963
: 2964 !+
: 2965 ! THE FOLLOWING DBMs ARE DEBUG MESSAGES, AND SHOULD BE REMOVED BEFORE
: 2966 ! RELEASING THE PROGRAM. THEY INCLUDE THE NAMES OF EACH ROUTINE, PLUS
: 2967 ! FORMAT STATEMENTS FOR PRINTING OUT OTHER INFORMATION.
: 2968 !--
: 2969
: 2970 DBM5 = uplit (#asciz'#N#A** DROP UNIT #D2'),
: 2971 DBM12 = uplit (#asciz'#N#A** PROC_RETPKT: CONN ID = #D5#A RECEIVED'),

```

```

: 2972 ! DBM15 = uplit (%asciz'%N%A** MULTI-DRIVE TEST'),
: 2973 DBM18 = uplit (%asciz'%N%A** FTL ERR: RETPKT NOT AVAILABLE'),
: 2974 DBM19 = uplit (%asciz'%N%A** FSET_UPAR: CAN'T FIND DISK #D3#A IN CST #D1'),
: 2975 DBM20 = uplit (%asciz'%N%A** BAD CONN ID = #D5#A RECEIVED FROM #06'),
: 2976 DBM21 = uplit (%asciz'%N%A** MESSAGE TYPE #D2#A RECEIVED IN MSCP PACKET'),
: 2977 ! DBM22 = uplit (%asciz'%N%A** SEQUEN: RETPKT NOT AVAILABLE'),
: 2978 DBM23 = uplit (%asciz'%N%A** ERR IN SET_CTLR_CHAR'),
: 2979 DBM25 = uplit (%asciz'%N%A** CTLR TIMEOUT = #D3#A SECONDS'),
: 2980 DBM26 = uplit (%asciz'%N%A** ERR IN UNIT_INIT'),
: 2981 ! DBM27 = uplit (%asciz'%N%A** UNIT_INIT: RETPKT HAS BAD ENDCODE'),
: 2982 ! DBM28A = uplit (%asciz'%N%A** UNIT SIZE (LO) = #D5'),
: 2983 ! DBM28B = uplit (%asciz'%N%A** UNIT SIZE (HI) = #D5'),
: 2984 DBM29 = uplit (%asciz'%N%A** ACCESS: RETPKT HAS BAD ENDCODE'),
: 2985 ! DBM32 = uplit (%asciz'%N%A** GIO_UNIT: CST #D1#A NO UNIT SELECTED'),
: 2986 ! DBM101 = uplit (%asciz'%N%A** UNIT # IS: #06'),
: 2987 ! DBM104 = uplit (%asciz'%N%A** RX50 IS SELECTED'),
: 2988 ! DBM105 = uplit (%asciz'%N%A** RD51 IS SELECTED'),
: 2989 DBM107 = uplit (%asciz'%N%A** ILLEGAL FUNCTION: #06'),
: 2990 DBM108 = uplit (%asciz'%N%A** COMMAND REF # #D5#A, NOT SENT BY HOST'),
: 2991 DBM109 = uplit (%asciz'%N%A** UNKNOWN ERROR LOG FORMAT #D3#A, RECEIVED'),
: 2992 DBM110 = uplit (%asciz'%N%A** ERROR-LOG SAVE AREA FULL'),
: 2993 ! DBM111 = uplit (%asciz'%N%A** REINIT MSCP EXERCISER'),
: 2994 DBM112 = uplit (%asciz'%N%A** DUP: PKT NOT AVAILABLE'),
: 2995 !DER10 = uplit (%asciz'%N%A** DUP'),
: 2996 !DER13 = uplit (%asciz'%N%A** DUP COMMAND'),
: 2997 !
: 2998 ! DROP UNIT MESSAGES
: 2999 !
: 3000 DU_MSG = uplit (%asciz'%N%AUNIT#D2#A DROPPED - '),
: 3001 DU_RSN = uplit (
: 3002 uplit (%asciz'%AUSER CMD#N'),
: 3003 uplit (%asciz'%ACONFIGURATION ERR#N'),
: 3004 uplit (%asciz'%AINIT ERR#N'),
: 3005 uplit (%asciz'%ATRANSFER LIMIT REACHED#N'),
: 3006 uplit (%asciz'%AERR LIMIT REACHED#N'),
: 3007 uplit (%asciz'%AUNRECOVERABLE DEV ERR#N'),
: 3008 uplit (%asciz'%AUNRECOVERABLE CTLR ERR#N'),
: 3009 uplit (%asciz'%AFAILED TO COME ONLINE#N'),
: 3010 uplit (%asciz'%AFAILED TO ACCESS LAST TRACK DURING INIT#N'),
: 3011 uplit (%asciz'%ADISK WRITE PROTECTED#N'),
: 3012 uplit (%asciz'%ACMD TIME OUT#N'),
: 3013 uplit (%asciz'%AUNIT WENT TO AVAILABLE STATE#N')) : vector [11],
: 3014 !
: 3015 ! SYSTEM MESSAGES (PRINTF)
: 3016 !
: 3017 MSG_01 = uplit (%asciz'%N%APOWER DELAY - WAITING'),
: 3018 MSG_02 = uplit (%asciz'%N%AFUNCTIONAL TEST STARTED'),
: 3019 MSG_03 = uplit (%asciz'%N%N%AEXERCISER STARTED#N'),
: 3020 !
: 3021 ! REPORT MESSAGES (PRINTS)
: 3022 !
: 3023 RPT1 = uplit (%asciz'%N%N%AUNT DSK#S8#A# OF # BYTES # OF # BYTES'),
: 3024 RPT2 = uplit (%asciz'%A --HARD ERRGRS-- --SOFT ERRORS--'),

```

```

: 3025 RPT3 = uplit (#asciz'#N#A # # TYPE READS READ WRITES WRITTEN'),
: 3026 RPT4 = uplit (#asciz'#A SEK DAT DRV HST SEK DAT DRV HST'),
: 3027 RPT5 = uplit (#asciz'#N#A-----' ),
: 3028 RPT6 = uplit (#asciz'#A-----' ),
: 3029 RPT7 = uplit (#asciz'#N#D2#D4#A RX50'),
: 3030 RPT8 = uplit (#asciz'#N#D2#D4#A RD51'),
: 3031 RPT9 = uplit (#asciz'#D4#Z3#D3#A,#Z3#A,#Z3'),
: 3032 RPT10 = uplit (#asciz'#D4#D4#D4#D4#D4#D4#D4#D4'),
: 3033 RPT11 = uplit (#asciz'#N#A . CNTR .....'),
: 3034 RPT12 = uplit (#asciz'#A .#D4#A . .#D4#A . .'),
: 3035 RPT13 = UPLIT(#ASCIZ'#N#N#AUNIT DISK # OF # BLKS # OF # BLKS '),
: 3036 RPT14 = UPLIT(#ASCIZ'#N#A # # TYPE READS READ WRITES WRITTEN '),
: 3037 RPT15 = UPLIT(#ASCIZ'#N#A-----' ),
: 3038 RPT16 = UPLIT(#ASCIZ'#N#S1#D2#S4#D2#A DBNRD51 #D6#S3#D6#S5#D6#S3#D6'),
: 3039 RPT17 = uplit (#asciz'#N#D2#D4#A RD52'),
: 3040 RPT18 = UPLIT(#ASCIZ'#N#S1#D2#S4#D2#A DBNRD52 #D6#S3#D6#S5#D6#S3#D6'),
: 3041 RPT19 = uplit (#asciz'#N#D2#D4#A ????' ),
: 3042 :
: 3043 : GENERAL ERROR MESSAGES
: 3044 :
: 3045 : SYSTEM FATAL (ERRSF)
: 3046 :
: 3047 EGS_01 = uplit (#asciz'TOO MANY UNITS'),
: 3048 EGS_02 = uplit (#asciz'NOT ENOUGH FREE MEMORY FOR ALLOCATING READ/WRITE BUFFERS'),
: 3049 :
: 3050 : DEVICE FATAL (ERRDF)
: 3051 :
: 3052 EGD_10 = uplit (#asciz'REG EXISTENCE TEST FAILED'),
: 3053 EGD_11 = uplit (#asciz'VECTOR TEST FAILED'),
: 3054 EGD_12 = uplit (#asciz'BR LEVEL TEST FAILED'),
: 3055 EGD_13 = uplit (#asciz'INIT SEQUENCE FAILED'),
: 3056 EGD_14 = uplit (#asciz'FATAL CTLR ERR'),
: 3057 EGD_15 = uplit (#asciz'ONLINE FAILED'),
: 3058 EGD_16 = uplit (#asciz'WRITE-PROTECT CONFLICT'),
: 3059 EGD_17 = uplit (#asciz'ACCESS FAILED'),
: 3060 EGD_18 = uplit (#asciz'FATAL I/O ERR'),
: 3061 EGD_19 = uplit (#asciz'CTLR TIMEOUT'),
: 3062 EGD_20 = uplit (#asciz'FAILED TO SEND SET-CTLR-CHARACTERISTICS COMMAND'),
: 3063 EGD_21 = uplit (#asciz'SET-CTLR-CHARACTERISTICS RESPONSE HAS BAD ENCODE OR FLAGS IN ERR'),
: 3064 EGD_22 = uplit (#asciz'FAILED TO SEND ON-LINE COMMAND'),
: 3065 EGD_23 = uplit (#asciz'ON-LINE RESPONSE HAS BAD ENCODE'),
: 3066 EGD_24 = uplit (#asciz'ON-LINE RESPONSE HAS UNKNOWN DEVICE'),
: 3067 :
: 3068 : HARD or SOFT (ERRHRD or ERRSOFT)
: 3069 :
: 3070 EGH_30 = uplit (#asciz'I/O REQUEST FAILED'),
: 3071 :
: 3072 : BASIC ERROR MESSAGES (PRINTB)
: 3073 :
: 3074 : SYSTEM FATAL (ERRSF)
: 3075 :
: 3076 EBS_01 = uplit (#asciz'#AMORE THAN #D2#A UNITS SPECIFIED'),
: 3077 :

```

```

: 3078 : DEVICE FATAL (ERRDF)
: 3079 :
: 3080 EBD_10 = uplit (#asciz'#A* NO RESPONSE AT ADDRESS #06'),
: 3081 EBD_12 = uplit (#asciz'#A* INCORRECT BR LEVEL FOR DEV #06'),
: 3082 EBD_13 = uplit (#asciz'#A* STEP #D1#A READ ERR'),
: 3083 EBD_14 = uplit (#asciz'#A* BAD SA CODE FROM DEV #06'),
: 3084 EBD_18 = uplit (#asciz'#A* DISK#D2#A WENT OFFLINE'),
: 3085 EBD_19 = uplit (#asciz'#A* DEVICE #06#A NOT PROCESSING COMMAND PACKETS'),
: 3086 :
: 3087 : HARD or SOFT (ERRHRD or ERRSOFT)
: 3088 :
: 3089 EH_0 = UPLIT (#ASCIZ' - unrecognized MESSAGE TYPE'),
: 3090 EH_1 = UPLIT (#ASCIZ' - unrecognized connection id'),
: 3091 EH_2 = UPLIT (#ASCIZ' - unrecognized RETURN message'),
: 3092 EH_3 = UPLIT (#ASCIZ' - unrecognized RETURN PACKET'),
: 3093 EH_4 = UPLIT (#ASCIZ' - unrecognized CRN'),
: 3094 EH_5 = UPLIT (#ASCIZ' - UNRECOGNIZED OPCODE'),
: 3095 EH_6 = UPLIT (#ASCIZ' - MSCP STATUS CODE ERR'),
: 3096 EH_7 = UPLIT (#ASCIZ' - DUP STATUS CODE ERR'),
: 3097 EH_8 = UPLIT (#ASCIZ' - unrecognized STATUS CODE'),
: 3098 EH_9 = UPLIT (#ASCIZ' - LBN HOST COMPARE ERR'),!
: 3099 EH_10 = UPLIT (#ASCIZ' - DBN HOST COMPARE ERR'),
: 3100 EH_12 = UPLIT (#ASCIZ' - UNABLE TO LOAD DUP MEDIA '),
: 3101 EH_13 = UPLIT (#ASCIZ' - ERR IN DUP-PKT WHEN USING CTLR LC PRG'),
: 3102 ERR_COD = uplit (
: 3103     uplit (#asciz'#AINVALID COMMAND'),
: 3104     uplit (#asciz'#ACOMMAND ABORTED'),
: 3105     uplit (#asciz'#AUNIT OFFLINE'),
: 3106     uplit (#asciz'#ATRANSITION TO AVAILABLE STATE'),
: 3107     uplit (#asciz'#AMEDIA FORMAT ERR'),
: 3108     uplit (#asciz'#AWRITE-PROTECTED'),
: 3109     uplit (#asciz'#ADEVICE COMPARE ERR'),
: 3110     uplit (#asciz'#ADATA ERR'),
: 3111     uplit (#asciz'#AHOST BUFFER ACCESS ERR'),
: 3112     uplit (#asciz'#ACTLR ERR'),
: 3113     uplit (#asciz'#ADRIVE ERR'),
: 3114     uplit (#asciz'#AMESSAGE FROM INTERNAL DIAGNOSTICS'),
: 3115     uplit (#asciz'#AHOST COMPARE ERR'),
: 3116     uplit (#asciz'#ACOMMAND TIMEOUT')) : vector [14],
: 3117 :
: 3118 : ERROR LOG MESSAGE (ERRSOFT)
: 3119 :
: 3120 ELG_00 = uplit (#asciz'#N#AERR LOG MESSAGE RECEIVED:#N'),
: 3121 ELG_FMT = uplit (
: 3122     uplit (#asciz'#A* CTLR ERR#N'),
: 3123     uplit (#asciz'#A* HOST MEMORY ACCESS ERR#N'),
: 3124     uplit (#asciz'#A* DISK#D2#A - DISK TRANSFER ERR#N'),
: 3125     uplit (#asciz'#A* DISK#D2#A - "STANDARD DISK INTERCONNECT" ERR#N'),
: 3126     uplit (#asciz'#A* DISK#D2#A - "SMALL DISK" ERR#N')) : vector [5],
: 3127 :
: 3128 : EXTENDED ERROR MESSAGES (PRINTX)
: 3129 :
: 3130

```

```

: 3131 EX_BDR = uplit (#asciz'#N#AI/O BUFFER ADDRESS FOR READ (32 BITS): #06#A #06#N'),
: 3132 EX_BDW = uplit (#asciz'#N#AI/O BUFFER ADDRESS FOR WRITE (32 BITS): #06#A #06#N'),
: 3133 EX_LBR = uplit (#asciz'#N#ALBN: (READ) #D5#A. (OCT #06#A #06#A)'),
: 3134 EX_LBW = uplit (#asciz'#N#ALBN: (WRITE) #D5#A. (OCT #06#A #06#A)'),
: 3135 EX_RBN = uplit (#asciz'#N#AREPLACEMENT BLOCK NO. #D5#A. (OCT #06#A #06#A)'),
: 3136 EX_CBR = uplit (#asciz'#N#ABYTE COUNT IN READ COMMAND: #D5#A.'),
: 3137 EX_CBW = uplit (#asciz'#N#ABYTE COUNT IN WRITE COMMAND: #D5#A.'),
: 3138
: 3139 XX13 = UPLIT (#ASCIZ'#N#A * DISK : #D2'),
: 3140 XX14 = uplit (#asciz'#N#ASA: #06#A'),
: 3141 XX15 = uplit (#asciz'#N#ASTATUS CODE: '),
: 3142 XX16 = uplit (#asciz'#N#ASTATUS SUB-CODE: '),
: 3143 XX17 = uplit (#asciz'#N#ACOMMAND: '),
: 3144 XX18 = uplit (#asciz'#A-DUP-'),
: 3145 XX19 = uplit (#asciz'#A-MSCP-'),
: 3146 XX20 = uplit (#asciz'#A-COMPARE'),
: 3147 XX21 = uplit (#asciz'#N#ABAD BLOCK REPORTED: #06#A #06#A.'),
: 3148 XX22 = uplit (#asciz'#N#ALBN: #D5#A. (OCT #06#A #06#A)'),
: 3149 XX23 = UPLIT (#ASCIZ'#N#ADBN: #D5#A. (OCT #06#A #06#A)'),
: 3150 XX24 = uplit (#asciz'#N#ABYTE COUNT IN COMMAND: #D5#A.'),
: 3151 XX25 = uplit (#asciz'#N#AACTUAL # OF BYTES TRANSFERRED: #D5#A.'),
: 3152 XX26 = uplit (#asciz'#N#AI/O BUFFER ADDRESS (32 BITS): #06#A #06#A'),
: 3153 XX27 = uplit (#asciz'#N#ACONTENTS OF RETURN PACKET: #N#A'),
: 3154 XX29 = UPLIT (#ASCIZ'#N#AMESSAGE TYPE: '),
: 3155 XX30 = UPLIT (#ASCIZ'#N#AMESSAGE NUMBERS: '),
: 3156 XX31 = UPLIT (#ASCIZ'#N#AMESSAGE ERR CODES: '),
: 3157 XX32 = UPLIT (#ASCIZ'#N#ABYTE NUMBER: #D3#A'),
: 3158 XX33 = UPLIT (#ASCIZ'#N#ARANDOM WRITTEN WORD :#B16#A'),
: 3159 XX34 = UPLIT (#ASCIZ'#N#ARANDOM READ WORD bin:#B16#A oct:#06#A'),
: 3160 XX35 = UPLIT (#ASCIZ'#N#ACRN : #06#A #06#A'),
: 3161 !XX36 = UPLIT (#ASCIZ'#N#ATHE EXPECTED CRN : #06#A #06#A'),
: 3162 XX37 = UPLIT (#ASCIZ'#A - UNKNOWN : #D2#A'),
: 3163 XX38 = UPLIT (#ASCIZ'#A - UNKNOWN CONNECTION ID: #D3#A -'),
: 3164 XX39 = UPLIT (#ASCIZ'#N#ACTLR FLAGS: '),
: 3165 XX40 = UPLIT (#ASCIZ'#N#AUNIT FLAGS: '),
: 3166 XX41 = UPLIT (#ASCIZ'#N#AEND MESSAGE FLAGS: '),
: 3167 XX42 = UPLIT (#asciz'#N#ACONTENTS OF DUP I/O PACKET: #N#A'),
: 3168
: 3169 !
: 3170 ! UNKNOWN RETURN MESSAGES
: 3171 !
: 3172 EB_DCT = UPLIT (#ASCIZ'#N#A DRIVER CTLR TABLE = ADDR: #D6#A'),
: 3173 EB_COMM = UPLIT (#ASCIZ'#N#A CMD INT, RSP INT, COMMAND RING = ADDR: #D6#A'),
: 3174 EB_PKT = UPLIT (#ASCIZ'#N#A ALL PACKETS IN MESSAGE AREA'),
: 3175 EB_RAL = UPLIT (#ASCIZ'#N#A ALL RETURN PACKETS IN AREA'),
: 3176 EB_ADDR = UPLIT (#ASCIZ'#N#A ADDR: #D6#A PACKET = #N#A'),
: 3177 EB_NEX1 = UPLIT (#ASCIZ'#N#A ADDR OF RESPONSE RING TO BE POLLED #D6#A'),
: 3178 EB_NEX2 = UPLIT (#ASCIZ'#N#A ADDR OF MESSAGE PACKET RESPONSE RING SLOT POINTS TO #D6#A'),
: 3179 EB_NEX3 = UPLIT (#ASCIZ'#N#A ADDR OF MESSAGE PACKET COMMAND RING SLOT POINTS TO #D6#A'),
: 3180 EB_SA = UPLIT (#ASCIZ'#N#A Contents of SA Register'),
: 3181 !
: 3182 ! CONFIGURATION ERROR MESSAGES (PRINTF)
: 3183 !

```

```

: 3184 CER_01 = uplit (#asciz'#N#ADUPLICATE UNIT:#D2#A AT IP: #06'),
: 3185 CER_02 = uplit (#asciz'#N#AMORE THAN #D1#A DIFFERENT IP ADDRESSES'),
: 3186
: 3187 :
: 3188 : MESSAGE TYPES
: 3189 :
: 3190 EX_SEQ = UPLIT (#ASCIZ'#A- SEQUENTIAL'),
: 3191 EX_CRD = UPLIT (#ASCIZ'#A- CREDIT NOTIFICATION'),
: 3192 EX_MTN = UPLIT (#ASCIZ'#A- MAINTENANCE'),
: 3193 EX_DGM = UPLIT (#ASCIZ'#A- DATAGRAM'),
: 3194 :
: 3195 : commands
: 3196 : mscp
: 3197 EX_RD = uplit (#asciz'#AREAD'),
: 3198 EX_WRT = uplit (#asciz'#AWRITE'),
: 3199 EX_ACC = uplit (#asciz'#AACCESS'),
: 3200 EX_ONL = uplit (#asciz'#AON LINE'),
: 3201 EX_SCC = uplit (#asciz'#ASET CONTROL CHAR. '),
: 3202 : dup
: 3203 EX_GDS = uplit (#asciz'#AGET DUST STATUS'),
: 3204 EX_ESP = uplit (#asciz'#AEXECUTE SUPPLIED PRG'),
: 3205 EX_ELP = uplit (#asciz'#AEXECUTE LOCAL PRG'),
: 3206 EX_SDD = uplit (#asciz'#ASEND DATA'),
: 3207 EX_RCD = uplit (#asciz'#ARECEIVE DATA'),
: 3208 EX_ABP = uplit (#asciz'#AABORT'),
: 3209 :
: 3210 : ERROR/EVENT SUB CODES (PRINTX)
: 3211 :
: 3212 SC_SDI = uplit (#asciz'#ASPIN-DOWN IGNORED'),
: 3213 SC_CON = uplit (#asciz'#ASTILL CONNECTED'),
: 3214 SC_DUP = uplit (#asciz'#ADUPLICATE UNIT NUMBER'),
: 3215 SC_ONL = uplit (#asciz'#AALREADY ONLINE'),
: 3216 SC_SON = uplit (#asciz'#ASTILL ONLINE'),
: 3217 SC_UNK = uplit (#asciz'#AUNIT UNKNOWN OR ONLINE TO ANOTHER CTRLR'),
: 3218 SC_VOL = uplit (#asciz'#AND VOLUME MOUNTED OR DRIVE DISABLED BY SWITCH'),
: 3219 SC_IOP = uplit (#asciz'#AUNIT INOPERATIVE (RDS1 WRITE FAULT)'),
: 3220 SC_DIS = uplit (#asciz'#AUNIT DISABLED BY FIELD SERVICE OR INTERNAL DIAGNOSTICS'),
: 3221 SC_FER = uplit (#asciz'#A"FORCED ERR" DETECTED WHILE ACCESSING FCT OR RCT'),
: 3222 SC_FE2 = uplit (#asciz'#ASECTOR WRITTEN WITH "FORCED ERR" MODIFIER'),
: 3223 SC_ISH = uplit (#asciz'#AFCT OR RCT UNREADABLE - INVALID SECTOR HEADER'),
: 3224 SC_IS2 = uplit (#asciz'#AHEADER COMPARE ERR (VALID HEADER NOT FOUND)'),
: 3225 SC_DST = uplit (#asciz'#AFCT OR RCT UNREADABLE - DATA SYNC TIMEOUT'),
: 3226 SC_DS2 = uplit (#asciz'#ADATA SYNC NOT FOUND (DATA SYNC TIMEOUT)'),
: 3227 SC_ECC = uplit (#asciz'#AFCT OR RCT UNREADABLE - UNCORRECTABLE ECC ERR'),
: 3228 SC_ECD = uplit (#asciz'#AUNCORRECTABLE ECC ERR'),
: 3229 SC_RCT = uplit (#asciz'#ARCT CORRUPTED'),
: 3230 SC_FUL = uplit (#asciz'#AND REPLACEMENT BLOCK AVAILABLE (RCT FULL)'),
: 3231 SC_576 = uplit (#asciz'#ADISK NOT FORMATTED WITH 512 BYTE SECTORS'),
: 3232 SC_FCT = uplit (#asciz'#ADISK NOT FORMATTED OR FCT CORRUPTED'),
: 3233 SC_EC1 = uplit (#asciz'#AONE SYMBOL ECC ERR'),
: 3234 SC_EC2 = uplit (#asciz'#ATWO SYMBOL ECC ERR'),
: 3235 SC_EC3 = uplit (#asciz'#ATHREE SYMBOL ECC ERR'),
: 3236 SC_EC4 = uplit (#asciz'#AFOUR SYMBOL ECC ERR'),

```

```

: 3237 SC_EC5 = uplit ('asciz'AFIVE SYMBOL ECC ERR'),
: 3238 SC_EC6 = uplit ('asciz'ASIX SYMBOL ECC ERR'),
: 3239 SC_EC7 = uplit ('asciz'ASEVEN SYMBOL ECC ERR'),
: 3240 SC_EC8 = uplit ('asciz'AEIGHT SYMBOL ECC ERR'),
: 3241 SC_EC9 = uplit ('asciz'ACORRECTABLE ERR IN ECC FIELD'),
: 3242 SC_SWP = uplit ('asciz'ASUNIT SOFTWARE WRITE PROTECTED'),
: 3243 SC_HWP = uplit ('asciz'ASUNIT HARDWARE WRITE PROTECTED'),
: 3244 SC_ODA = uplit ('asciz'ASODD TRANSFER ADDRESS'),
: 3245 SC_ODB = uplit ('asciz'ASODD BYTE COUNT'),
: 3246 SC_NXM = uplit ('asciz'ANON-EXISTENT HOST MEMORY'),
: 3247 SC_PAR = uplit ('asciz'AHOST MEMORY PARITY ERR'),
: 3248 SC_CTO = uplit ('asciz'ACOMMAND TIMEOUT OR RETRY LIMIT EXCEEDED'),
: 3249 SC_SDS = uplit ('asciz'ASERIALIZER/DESERIALIZER OVERRUN OR UNDERRUN'),
: 3250 SC_EDC = uplit ('asciz'AEDC ERR'),
: 3251 SC_IDS = uplit ('asciz'AINCONSISTENT INTERNAL DATA STRUCTURE'),
: 3252 SC_SRT = uplit ('asciz'ADRIVE COMMAND TIMEOUT (NO RESPONSE OR SEEK INCOMPLETE)'),
: 3253 SC_SRI = uplit ('asciz'ACTLR DETECTED TRANSMISSION OR PROTOCOL ERR'),
: 3254 SC_POE = uplit ('asciz'ASPOSITION ERR (MIS-SEEK)'),
: 3255 SC_RDY = uplit ('asciz'ALOST READ/WRITE READY DURING/BETWEEN TRANSFERS'),
: 3256 SC_CLK = uplit ('asciz'ADRIVE CLOCK DROPOUT'),
: 3257 SC_RSP = uplit ('asciz'ALOST RECEIVER READY BETWEEN SECTORS'),
: 3258 SC_SUR = uplit ('asciz'ADRIVE DETECTED ERR'),
: 3259 SC_PSP = uplit ('asciz'ACTLR DETECTED PULSE OR STATE PARITY ERR'),
: 3260 :
: 3261 : MSCP END MESSAGE FLAGS
: 3262 :
: 3263 !F_0 = uplit ('asciz'NSA) Bad Block Reported'),
: 3264 F_1 = uplit ('asciz'NSA) Bad Block Unreported'),
: 3265 F_2 = uplit ('asciz'NSA) Err Log Generated'),
: 3266 F_3 = uplit ('asciz'NSA) Serious Exception'),
: 3267 ! MSCP Controller Flags
: 3268 F_4 = uplit ('asciz'NSA) Enable Attention Messages'),
: 3269 F_5 = uplit ('asciz'NSA) Enable Miscellaneous Err Log Messages'),
: 3270 F_6 = uplit ('asciz'NSA) Enable Other Hosts Err Log Messages'),
: 3271 F_7 = uplit ('asciz'NSA) Enable This Hosts Err Log Messages'),
: 3272 F_8 = uplit ('asciz'NSA) Ctlr Initiated Bad Block Rplcmt'),
: 3273 F_9 = uplit ('asciz'NSA) Shadowing'),
: 3274 F_10 = uplit ('asciz'NSA) 576 Byte Sectors'),
: 3275 :
: 3276 ! MSCP UNIT FLAGS
: 3277 :
: 3278 F_11 = uplit ('asciz'NSA) Compare Reads'),
: 3279 F_12 = uplit ('asciz'NSA) Compare Writes'),
: 3280 F_13 = uplit ('asciz'NSA) Ctlr Initiated Bad Block Rplcmt'),
: 3281 F_14 = uplit ('asciz'NSA) Inactive Shadow Set Unit'),
: 3282 F_15 = uplit ('asciz'NSA) Removable Media'),
: 3283 F_16 = uplit ('asciz'NSA) Suppress Caching (high speed)'),
: 3284 F_17 = uplit ('asciz'NSA) Suppress Caching (low speed)'),
: 3285 F_18 = uplit ('asciz'NSA) Write-back (non-volatile)'),
: 3286 F_19 = uplit ('asciz'NSA) Write Protect (hardware)'),
: 3287 F_20 = uplit ('asciz'NSA) Write Protect (software or volume)'),
: 3288 F_21 = uplit ('asciz'NSA) 576 Byte Sectors'),
: 3289 !

```

```

: 3290 ! DUP RETURN PACKET MESSAGES
: 3291 ! STATUS CODE
: 3292   EBH_30 = uplit (#asciz'#A - SUCCESS'),
: 3293   EBH_44 = UPLIT (#ASCIZ'#A - INVALID COMMAND(SERVER nonIDLE or no media if EX-LC-PRG cmd)'),
: 3294   EBH_45 = UPLIT (#ASCIZ'#A - NO REGION AVAILABLE'),
: 3295   EBH_46 = UPLIT (#ASCIZ'#A - NO REGION SUITABLE'),
: 3296   EBH_47 = UPLIT (#ASCIZ'#A - PROGRAM NOT KNOWN (NO SUCH PROGRAM ON MEDIA)'),
: 3297   EBH_48 = UPLIT (#ASCIZ'#A - LOAD FAILURE (INPUT ERR WHILE LOADING PROGRAM)'),
: 3298   EBH_49 = UPLIT (#ASCIZ'#A - STANDALONE (STANDALONE MODIFIER NOT SPECIFIED FOR STAND ALONE PRG.)'),
: 3299
: 3300 !
: 3301 ! DUP GET DUST STATUS FLAGS
: 3302   df_0 = uplit (#asciz'#N#A) One Server at a Time'),
: 3303   df_1 = uplit (#asciz'#N#A) Contains Local Media'),
: 3304   df_2 = uplit (#asciz'#N#A) Execute Local Prg cmd is UNSUPPORTED'),
: 3305   df_3 = uplit (#asciz'#N#A) Currently in Active State'),
: 3306 !
: 3307 ! DUP EXECUTE LOCAL PRG END FLAGS
: 3308   df_4 = uplit (#asciz'#N#A) Standalone Prg'),
: 3309   df_5 = uplit (#asciz'#N#A) Needs overlay'),
: 3310   df_6 = uplit (#asciz'#N#A) Needs Writeable/Readable Overlay'),
: 3311   df_7 = uplit (#asciz'#N#A) Uses Std Dup Dialog; REC/SEND/REC'),
: 3312 !
: 3313 ! DUP LOCAL PROGRAM PACKET MESSAGES
: 3314 !
: 3315   T_QUE = uplit (#asciz'#N#A ** QUESTION'),
: 3316   T_DEF = uplit (#asciz'#N#A ** DEFAULT QUESTION'),
: 3317   T_INF = uplit (#asciz'#N#A ** INFORMATION'),
: 3318   T_TER = uplit (#asciz'#N#A ** TERMINATION'),
: 3319   T_FAT = uplit (#asciz'#N#A ** FATAL ERR'),
: 3320   T_SPL = uplit (#asciz'#N#A ** SPECIAL'),
: 3321 !   E_SUC = UPLIT (#ASCIZ'#N#A - SUCCESS'),
: 3322   E_UNT = UPLIT (#ASCIZ'#N#A - ILLEGAL UNIT NUMBER'),
: 3323   E_BLK = UPLIT (#ASCIZ'#N#A - ILLEGAL PHYSICAL OR RELATIVE BLOCK NUMBER'),
: 3324   E_DEV = UPLIT (#ASCIZ'#N#A - DEVICE ERR'),
: 3325   E_ZER = UPLIT (#ASCIZ'#N#A - ZERO LENGHT MESSAGE'),
: 3326   M_ASC = UPLIT (#ASCIZ'#N#A -- ASCII INFORMATION'),
: 3327   M_BIN = UPLIT (#ASCIZ'#N#A -- NON-ASCII INFORMATION'),
: 3328   M_TER = UPLIT (#ASCIZ'#N#A -- TERMINATION MESSAGE'),
: 3329   M_COD = UPLIT (#ASCIZ'#N#A -- SUCCESS/FAILURE CODE'),
: 3330   M_DAT = UPLIT (#ASCIZ'#N#A -- SEND BINARY DATA'),
: 3331   M_UR = UPLIT (#ASCIZ'#N#A -- SEND UNIT NUMBER, RELATIVE DBN'),
: 3332   M_URP = UPLIT (#ASCIZ'#N#A -- SEND UNIT NUMBER, RELATIVE DBN, WRITE PATTERN'),
: 3333   M_UP = UPLIT (#ASCIZ'#N#A -- SEND UNIT NUMBER, PHYSICAL BLOCK NUMBER'),
: 3334   M_UL = UPLIT (#ASCIZ'#N#A -- SEND UNIT NUMBER, LOGICAL BLOCK NUMBER'),
: 3335 !
: 3336 ! CONTROLLER GENERIC ERROR CODES
: 3337 !
: 3338   CNTR_ERR = uplit (
: 3339     uplit (#asciz'#ACTLR TIMEOUT'),
: 3340     uplit (#asciz'#AENVELOPE/PACKET READ ERR (PARITY OR TIMEOUT)'),
: 3341     uplit (#asciz'#AENVELOPE/PACKET WRITE ERR (PARITY OR TIMEOUT)'),
: 3342     uplit (#asciz'#ACTLR ROM AND RAM PARITY ERR'),

```



```

: 3343      uplit (#asciz'#ACTLR RAM PARITY ERR'),
: 3344      uplit (#asciz'#ACTLR ROM PARITY ERR'),
: 3345      uplit (#asciz'#ARING READ ERR (PARITY OR TIMEOUT)'),
: 3346      uplit (#asciz'#ARING WRITE ERR (PARITY OR TIMEOUT)'),
: 3347      uplit (#asciz'#INTERRUPT MASTER FAILURE'),
: 3348      uplit (#asciz'#AHOST ACCESS TIMEOUT (HIGHER LEVEL PROTOCOL DEPENDENT)'),
: 3349      uplit (#asciz'#ACREDIT LIMIT EXCEEDED'),
: 3350      uplit (#asciz'#AQ-BUS MASTER ERR'),
: 3351      uplit (#asciz'#ACTLR FATAL ERR'),
: 3352      uplit (#asciz'#AINSTRUCTION LOOP TIMEOUT'),
: 3353      uplit (#asciz'#AILLEGAL VIRTUAL CIRCUIT ID'),
: 3354      uplit (#asciz'#AINTERRUPT VECTOR ILLEGAL'),
: 3355      uplit (#asciz'#AMAINTENANCE READ/WRITE INVALID REGION IDENTIFIER'),
: 3356      uplit (#asciz'#AMAINTENANCE WRITE LOAD TO NON-LOADABLE CTRLR'),
: 3357      uplit (#asciz'#ACTLR RAM ERR (NON-PARITY)'),
: 3358      uplit (#asciz'#AINIT SEQUENCE ERR'),
: 3359      uplit (#asciz'#AHIGHER LEVEL PROTOCOL INCOMPATIBILITY ERR'),
: 3360      uplit (#asciz'#APURGE/POLL HARDWARE FAILURE'),
: 3361      uplit (#asciz'#AMAPPING REGISTER READ FAILURE (PARITY OR TIMEOUT)') : vector [23].
: 3362      :
: 3363      : RD/RX CONTROLLER DEPENDENT ERRORS CODES
: 3364      :
: 3365      RDRX_ERR = uplit (
: 3366          uplit (#asciz'#AT11 CPU FAILURE'),
: 3367          uplit (#asciz'#ANON-PARITY RAM ERR'),
: 3368          uplit (#asciz'#ASTATE MACHINE FAILURE - T11 ADDRESS REGISTER'),
: 3369          uplit (#asciz'#ASTATE MACHINE FAILURE - Q-BUS ADDRESS REGISTER'),
: 3370          uplit (#asciz'#ASTATE MACHINE FAILURE - CRC REGISTER'),
: 3371          uplit (#asciz'#ASTATE MACHINE FAILURE - SERIALIZER/DESERIALIZER REGISTER'),
: 3372          uplit (#asciz'#ASTATE MACHINE FAILURE - WRONG HARDWARE VERSION') : vector [7].
: 3373      :
: 3374      : MISCELLANEOUS
: 3375      :
: 3376      EX_WRD = uplit (#asciz'#A #06'),
: 3377      EX_OP = uplit (#asciz'#Aoct #04'),
: 3378      SPACE4 = uplit (#asciz'#S4'),
: 3379      CRLF = uplit (#asciz'#N'),
: 3380      DASH = uplit (#asciz'#A - '),
: 3381      ASTERISK = uplit (#asciz'#A* ');

```

ZRGAM1
V01.2RD/RX EXERCISER
DEFAULT HARDWARE P-TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 B1116-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRGACO.BL1;82 (33)SEQ 0047
Page 47

```

: 3382 #sbttl 'DEFAULT HARDWARE P-TABLE'
: 3383
: 3384 !+
: 3385 ! THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: 3386 ! THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: 3387 ! IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
: 3388 ! AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
: 3389 !-
: 3390
: 3391 BGNHW (DFPTBL);
: 3392
: 3393 global
: 3394     HWPT_IP_ADDR : word initial (INIT_IP_ADDR), ! IP ADDRESS
: 3395     HWPT_VECTOR : word initial (INIT_INTR_VECT), ! VECTOR ADDRESS
: 3396     HWPT_BR_LEVEL : word initial (INIT_BR_LEVEL), ! BR LEVEL
: 3397     HWPT_DISK : word initial (#0'100034'), ! DISK NUMBER, TYPE, PROTECTON BIT
: 3398     HWPT_S_TRK : word initial (0), ! STARTING TRACK
: 3399     HWPT_E_TRK : word initial (#0'177777'); ! ENDING TRACK ACTUALLY THE PROGRAM WILL
: 3400 ! FIGURE OUT MAXIMUM IN UNIT_INIT ROUTINE
: 3401
: 3402 ENDHW;

```

ZRGAM1
V01.2RD/RX EXERCISER
SOFTWARE P-TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRGACO.BL1;82 (34)SEQ 0048
Page 48

```

: 3403 #sbttl 'SOFTWARE P-TABLE'
: 3404
: 3405 !+
: 3406 ! THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
: 3407 ! PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE
: 3408 ! SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
: 3409 ! AT RUN TIME.
: 3410 !-
: 3411
: 3412 BGNSW (SFPTBL);
: 3413
: 3414 global
: 3415     SWP_ERROR : word initial (32),           ! HARD ERROR LIMIT FOR DROPPING UNIT
: 3416     SWP_XFER : word initial (0),           ! TRANSFER LIMIT FOR DROPPING UNIT
: 3417     SWP_FLAGS : word initial ('1002'),    ! FLAGS (SEE DOCUMENTATION)
: 3418     SWP_DPAT : word initial (0),           ! DATA PATTERN NUMBER
: 3419     SWP_RAT : word initial (98),          ! RDS1 OPERATION RATIO
: 3420     dupround : word initial (11),         ! NUMBER OF DBN'S WRITTEN AT ONE TIME
:
: 3421     SWP_UCNT : word initial (MAX_UDP_CNT), ! USER DATA PATTERN COUNT
: 3422     SWP_UDPAT : vector [MAX_UDP_CNT, word], ! USER DATA PATTERN
: 3423
: 3424 ENDSW;

```

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

```

: 3425 #sbttl 'PROTECTION TABLE'
: 3426
: 3427 !+
: 3428 ! THIS TABLE IS USED BY THE RUNTIME SERVICES
: 3429 ! TO PROTECT THE LOAD MEDIA.
: 3430 !-
: 3431
: 3432 BGNPROT (0, -1, 6);
: 3433
: 3434 !1ST ARG =      OFFSET INTO P-TABLE FOR CSR ADDRESS
: 3435 !2ND ARG =      OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
: 3436 !3RD ARG =      OFFSET INTO P-TABLE FOR DRIVE NUMBER
: 3437
: 3438 ENDPROT;
: 3439 end
: 3440
: 3441 eludom
    
```

```

                                .TITLE ZRQAM1 RD/RX EXERCISER
                                .IDENT  /V01.2/
                                .ENABL  AMA

000000                                .PSECT  $CODE$,  RO
000000      132      122      121      L$NAME:: .ASCII  /ZRQ/
000003      101                                .ASCII  /A/
000004      000                                .BYTE   0
000005      000                                .BYTE   0
000006      000                                .BYTE   0
000007      000                                .BYTE   0
000010                                L$REV::
000010      103                                .ASCII  /C/
000011      060                                .ASCII  /O/
000012      000000G      L$UNIT:: .WORD   T$PTHV
000014      077777      L$TIML:: .WORD   77777
000016      000000G      L$MPCP:: .WORD   L$HARD
000020      000000G      L$SPCP:: .WORD   L$SOFT
000022      024532'      L$MPTP:: .WORD   L$HW
000024      024552'      L$SPTP:: .WORD   L$SW
000026      000000G      L$LADP:: .WORD   L$LAST
000030      000000      L$STA::  .WORD   0
000032      000000      L$CO::   .WORD   0
000034      000001      L$DTYP:: .WORD   1
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
    
```

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

000054 000000
000056 000000
000060 000000G
000062 000000G
000064 000000
000066 000000
000070 000000G
000072 000000G
000074 000000
000076 000000G
000100 104035
000102 000126'
000104 000000G
000106 000000G
000110 000000G
000112 024632'
000114 000000
000116 000000
000120 000000
000122 000001
000124 000000G

000126
000130
000132
000134
000136 111 120 040
000141 101 104 104
000144 122 105 123
000147 123 000 000
000152 126 105 103
000155 124 117 122
000160 000 000
000162 102 122 040
000165 114 105 126
000170 105 114 000
000173 000
000174 122 104 057
000177 122 130 040
000202 104 122 111
000205 126 105 040
000210 116 125 115
000213 102 105 122
000216 000 000
000220 123 124 101
000223 122 124 111
000226 116 107 040
000231 114 102 116
000234 000 000
000236 105 116 104
000241 111 116 107
000244 040 114 102
000247 116 040 050

.WORD 0
L\$SPC:: .WORD 0
L\$DEVP:: .WORD L\$DVTYP
L\$REPP:: .WORD L\$RPT
L\$EXP4:: .WORD 0
L\$EXP5:: .WORD 0
L\$AUT:: .WORD L\$AU
L\$DUT:: .WORD L\$DU
L\$LUN:: .WORD 0
L\$DESP:: .WORD L\$DESC
L\$LOAD:: .WORD -73743
L\$ETP:: .WORD L\$ERRTBL
L\$ICP:: .WORD L\$INIT
L\$CCP:: .WORD L\$CLEAN
L\$ACP:: .WORD L\$AUTO
L\$PRT:: .WORD L\$PROT
L\$TEST:: .WORD 0
L\$DLY:: .WORD 0
L\$HIME:: .WORD 0
D\$PCNT:: .WORD 1
L\$DISPATCH::
.WORD T1
ERRTYP:: .BLKW 1
ERRNBR:: .BLKW 1
ERRMSG:: .BLKW 1
ERRBLK:: .BLKW 1
P.AAA: .ASCII /IP /
.ASCII /ADD/
.ASCII /RES/
.ASCII /S/<00><00>
P.AAB: .ASCII /VEC/
.ASCII /TOR/
.ASCII <00><00>
P.AAC: .ASCII /BR /
.ASCII /LEV/
.ASCII /EL/<00>
.ASCII <00>
P.AAD: .ASCII /RD/<57>
.ASCII /RX /
.ASCII /DRI/
.ASCII /VE /
.ASCII /NUM/
.ASCII /BER/
.ASCII <00><00>
P.AAE: .ASCII /STA/
.ASCII /RTI/
.ASCII /NG /
.ASCII /LBN/
.ASCII <00><00>
P.AAF: .ASCII /END/
.ASCII /ING/
.ASCII /LB /
.ASCII /N (/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

000252	115	101	130	.ASCII	/MAX/
000255	111	115	125	.ASCII	/IMU/
000260	115	040	055	.ASCII	/M -/
000263	040	040	122	.ASCII	/ R/
000266	130	065	060	.ASCII	/X50/
000271	072	040	067	.ASCII	/: 7/
000274	071	071	054	.ASCII	/99./
000277	040	122	104	.ASCII	/ RD/
000302	065	061	072	.ASCII	/51:/
000305	040	062	061	.ASCII	/ 21/
000310	065	071	071	.ASCII	/599/
000313	054	040	122	.ASCII	/, R/
000316	104	065	062	.ASCII	/D52/
000321	072	040	064	.ASCII	/: 4/
000324	067	071	071	.ASCII	/799/
000327	071	051	000	.ASCII	/9)/<00>
000332	127	122	111	P.AAG: .ASCII	/WRI/
000335	124	105	040	.ASCII	/TE /
000340	117	116	040	.ASCII	/ON /
000343	103	125	123	.ASCII	/CUS/
000346	124	117	115	.ASCII	/TOM/
000351	105	122	040	.ASCII	/ER /
000354	104	101	124	.ASCII	/DAT/
000357	101	040	101	.ASCII	/A A/
000362	122	105	101	.ASCII	/REA/
000365	040	117	116	.ASCII	/ ON/
000370	040	124	110	.ASCII	/ TH/
000373	111	123	040	.ASCII	/IS /
000376	104	111	123	.ASCII	/DIS/
000401	113	000	000	.ASCII	/K/<00><00>
000404	052	052	040	P.AAH: .ASCII	/** /
000407	127	101	122	.ASCII	/WAR/
000412	116	111	116	.ASCII	/NIN/
000415	107	040	055	.ASCII	/G -/
000420	040	103	125	.ASCII	/ CU/
000423	123	124	117	.ASCII	/STO/
000426	115	105	122	.ASCII	/MER/
000431	040	104	101	.ASCII	/ DA/
000434	124	101	040	.ASCII	/TA /
000437	101	122	105	.ASCII	/ARE/
000442	101	040	115	.ASCII	/A M/
000445	101	131	040	.ASCII	/AY /
000450	102	105	040	.ASCII	/BE /
000453	117	126	105	.ASCII	/OVE/
000456	122	127	122	.ASCII	/RWR/
000461	111	124	124	.ASCII	/ITT/
000464	105	116	041	.ASCII	/EN!//
000467	040	056	056	.ASCII	/ .. /
000472	056	040	103	.ASCII	/ . C/
000475	117	116	106	.ASCII	/ONF/
000500	111	122	115	.ASCII	/IRM/
000503	000			.ASCII	<00>
000504	105	130	105	P.AAI: .ASCII	/EXE/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

000507	122	103	111	.ASCII	/RCI/
000512	123	105	040	.ASCII	/SE /
000515	117	116	040	.ASCII	/ON /
000520	104	111	101	.ASCII	/DIA/
000523	107	116	117	.ASCII	/GNO/
000526	123	124	111	.ASCII	/STI/
000531	103	040	101	.ASCII	/C A/
000534	122	105	101	.ASCII	/REA/
000537	040	050	055	.ASCII	/ (- /
000542	116	117	116	.ASCII	/NON/
000545	055	103	125	.ASCII	/-CU/
000550	123	124	117	.ASCII	/STO/
000553	115	105	122	.ASCII	/MER/
000556	040	101	122	.ASCII	/ AR/
000561	105	101	040	.ASCII	/EA /
000564	122	104	163	.ASCII	/RDs/
000567	040	117	116	.ASCII	/ ON/
000572	114	131	051	.ASCII	/LY)/
000575	000			.ASCII	<00>
000576	127	122	111	P.AAJ:	.ASCII /WRI/
000601	124	105	040	.ASCII	/TE /
000604	117	116	040	.ASCII	/ON /
000607	104	111	101	.ASCII	/DIA/
000612	107	116	117	.ASCII	/GNO/
000615	123	124	111	.ASCII	/STI/
000620	103	040	101	.ASCII	/C A/
000623	122	105	101	.ASCII	/REA/
000626	000	000		.ASCII	<00><00>
000630	110	101	122	P.AAK:	.ASCII /HAR/
000633	104	040	105	.ASCII	/D E/
000636	122	122	117	.ASCII	/RRO/
000641	122	040	114	.ASCII	/R L/
000644	111	115	111	.ASCII	/IMI/
000647	124	000	000	.ASCII	/T/<00><00>
000652	124	122	101	P.AAL:	.ASCII /TRA/
000655	116	123	106	.ASCII	/NSF/
000660	105	122	040	.ASCII	/ER /
000663	114	111	115	.ASCII	/LIM/
000666	111	124	040	.ASCII	/IT /
000671	111	116	040	.ASCII	/IN /
000674	115	105	107	.ASCII	/MEG/
000677	101	102	131	.ASCII	/ABY/
000702	124	105	123	.ASCII	/TES/
000705	040	050	060	.ASCII	/ (O/
000710	040	146	157	.ASCII	/ fo/
000713	162	040	042	.ASCII	/r "/
000716	121	125	111	.ASCII	/QUI/
000721	103	113	040	.ASCII	/CK /
000724	120	101	123	.ASCII	/PAS/
000727	123	042	051	.ASCII	/S"/
000732	000	000		.ASCII	<00><00>
000734	122	101	116	P.AAM:	.ASCII /RAN/
000737	104	117	115	.ASCII	/DOM/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

000742	040	125	116	.ASCII	/ UN/
000745	111	124	040	.ASCII	/IT /
000750	115	117	104	.ASCII	/MOD/
000753	105	000	000	.ASCII	/E/<00><00>
000756	122	105	101	P.AAN:	.ASCII /REA/
000761	104	055	103	.ASCII	/D-C/
000764	117	115	120	.ASCII	/OMP/
000767	101	122	105	.ASCII	/ARE/
000772	123	040	120	.ASCII	/S P/
000775	105	122	106	.ASCII	/ERF/
001000	117	122	115	.ASCII	/ORM/
001003	105	104	040	.ASCII	/ED /
001006	101	124	040	.ASCII	/AT /
001011	124	110	105	.ASCII	/THE/
001014	040	103	124	.ASCII	/ CT/
001017	114	122	000	P.AAO:	.ASCII /LR/<00>
001022	127	122	111	.ASCII	/MRI/
001025	124	105	055	.ASCII	/TE -/
001030	103	117	115	.ASCII	/COM/
001033	120	101	122	.ASCII	/PAR/
001036	105	123	040	.ASCII	/ES /
001041	120	105	122	.ASCII	/PER/
001044	106	117	122	.ASCII	/FOR/
001047	115	105	104	.ASCII	/MED/
001052	040	101	124	.ASCII	/ AT/
001055	040	124	110	.ASCII	/ TH/
001060	105	040	103	.ASCII	/E C/
001063	124	114	122	.ASCII	/TLR/
001066	000	000		.ASCII	<00><00>
001070	103	110	105	P.AAP:	.ASCII /CHE/
001073	103	113	040	.ASCII	/CK /
001076	101	114	114	.ASCII	/ALL/
001101	040	127	122	.ASCII	/ WR/
001104	111	124	105	.ASCII	/ITE/
001107	123	040	101	.ASCII	/S A/
001112	124	040	110	.ASCII	/T H/
001115	117	123	124	.ASCII	/OST/
001120	040	102	131	.ASCII	/ BY/
001123	040	122	105	.ASCII	/ RE/
001126	101	104	111	.ASCII	/ADI/
001131	116	107	000	.ASCII	/NG/<00>
001134	125	123	105	P.AAQ:	.ASCII /USE/
001137	122	055	104	.ASCII	/R-D/
001142	105	106	111	.ASCII	/EFI/
001145	116	105	104	.ASCII	/NED/
001150	040	104	101	.ASCII	/ DA/
001153	124	101	040	.ASCII	/TA /
001156	120	101	124	.ASCII	/PAT/
001161	124	105	122	.ASCII	/TER/
001164	116	000		.ASCII	/N/<00>
001166	123	105	114	P.AAR:	.ASCII /SEL/
001171	105	103	124	.ASCII	/ECT/
001174	040	120	122	.ASCII	/ PR/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

001177	105	055	104	.ASCII	/E-D/
001202	105	106	111	.ASCII	/EFI/
001205	116	105	104	.ASCII	/NED/
001210	040	104	101	.ASCII	/DA/
001213	124	101	040	.ASCII	/TA /
001216	120	101	124	.ASCII	/PAT/
001221	124	105	122	.ASCII	/TER/
001224	116	040	050	.ASCII	/N (/
001227	060	040	146	.ASCII	/O f/
001232	157	162	040	.ASCII	/or /
001235	163	145	161	.ASCII	/seq/
001240	165	145	156	.ASCII	/uen/
001243	164	151	141	.ASCII	/tia/
001246	154	040	163	.ASCII	/l e/
001251	145	154	145	.ASCII	/ele/
001254	143	164	151	.ASCII	/cti/
001257	157	156	051	.ASCII	/on)/
001262	000	000		.ASCII	<00><00>
001264	116	125	115	P.AAS:	.ASCII /NUM/
001267	102	105	122		.ASCII /BER/
001272	040	117	106		.ASCII / OF/
001275	040	127	117		.ASCII / WO/
001300	122	104	123		.ASCII /RDS/
001303	040	111	116		.ASCII / IN/
001306	040	104	101		.ASCII / DA/
001311	124	101	040		.ASCII /TA /
001314	120	101	124		.ASCII /PAT/
001317	124	105	122		.ASCII /TER/
001322	116	040	050		.ASCII /N (/
001325	061	066	040		.ASCII /16 /
001330	155	141	170		.ASCII /max/
001333	151	155	165		.ASCII /imu/
001336	155	051	000		.ASCII /m)/<00>
001341	000				.ASCII <00>
001342	120	101	124	P.AAT:	.ASCII /PAT/
001345	124	105	122		.ASCII /TER/
001350	116	040	126		.ASCII /N V/
001353	101	114	125		.ASCII /ALU/
001356	105	000			.ASCII /E/<00>
001360	103	114	105	P.AAU:	.ASCII /CLE/
001363	101	122	040		.ASCII /AR /
001366	123	124	101		.ASCII /STA/
001371	124	111	123		.ASCII /TIS/
001374	124	111	103		.ASCII /TIC/
001377	101	114	040		.ASCII /AL /
001402	124	101	102		.ASCII /TAB/
001405	114	105	123		.ASCII /LES/
001410	040	101	106		.ASCII / AF/
001413	124	105	122		.ASCII /TER/
001416	040	120	122		.ASCII / PR/
001421	111	116	124		.ASCII /INT/
001424	111	116	107		.ASCII /ING/
001427	000				.ASCII <00>

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

001430	120	105	122	P.AAV:	.ASCII	/PER/
001433	103	105	116		.ASCII	/CEN/
001436	124	101	107		.ASCII	/TAG/
001441	105	040	117		.ASCII	/E O/
001444	106	040	127		.ASCII	/F W/
001447	111	116	103		.ASCII	/INC/
001452	110	105	123		.ASCII	/HES/
001455	124	105	122		.ASCII	/TER/
001460	040	117	120		.ASCII	/ OP/
001463	105	122	101		.ASCII	/ERA/
001466	124	111	117		.ASCII	/TIO/
001471	116	123	040		.ASCII	/NS /
001474	117	125	124		.ASCII	/OUT/
001477	040	117	106		.ASCII	/ OF/
001502	040	124	117		.ASCII	/ TO/
001505	124	101	114		.ASCII	/TAL/
001510	040	117	120		.ASCII	/ OP/
001513	105	122	101		.ASCII	/ERA/
001516	124	111	117		.ASCII	/TIO/
001521	116	123	000		.ASCII	/NS/<00>
001524	122	101	116	P.AAW:	.ASCII	/RAN/
001527	104	117	115		.ASCII	/DOM/
001532	040	102	114		.ASCII	/ BL/
001535	117	103	113		.ASCII	/OCK/
001540	040	115	117		.ASCII	/ MO/
001543	104	105	000		.ASCII	/DE/<00>
001546	116	125	115	P.AAX:	.ASCII	/NUM/
001551	102	105	122		.ASCII	/BER/
001554	040	117	106		.ASCII	/ OF/
001557	040	104	102		.ASCII	/ DB/
001562	116	163	040		.ASCII	/No /
001565	124	122	101		.ASCII	/TRA/
001570	116	123	106		.ASCII	/NSF/
001573	105	122	122		.ASCII	/ERR/
001576	105	104	040		.ASCII	/ED /
001601	101	124	040		.ASCII	/AT /
001604	117	116	105		.ASCII	/ONE/
001607	040	042	104		.ASCII	/ "D/
001612	125	120	042		.ASCII	/UP"/
001615	040	120	101		.ASCII	/ PA/
001620	123	123	000		.ASCII	/SS/<00>
001623	000				.ASCII	<00>
001624	124	110	105	P.AAY:	.ASCII	/THE/
001627	040	122	105		.ASCII	/ RE/
001632	115	101	111		.ASCII	/MAI/
001635	116	111	116		.ASCII	/NIN/
001640	107	040	121		.ASCII	/G Q/
001643	125	105	123		.ASCII	/UES/
001646	124	111	117		.ASCII	/TIO/
001651	116	123	040		.ASCII	/NS /
001654	117	116	114		.ASCII	/ONL/
001657	131	040	101		.ASCII	/Y A/
001662	120	120	114		.ASCII	/PPL/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

001665	131	040	124	.ASCII	/Y T/
001670	117	040	125	.ASCII	/O U/
001673	116	120	122	.ASCII	/NPR/
001676	117	124	105	.ASCII	/OTE/
001701	103	124	105	.ASCII	/CTE/
001704	104	040	104	.ASCII	/D D/
001707	111	123	113	.ASCII	/ISK/
001712	123	000		.ASCII	/S/<00>
001714	000	000		P.AAZ:	.ASCII <00><00>
001716	045	116	045	P.ABA:	.ASCII /#Ns/
001721	101	052	052		.ASCII /A**/
001724	040	104	122		.ASCII / DR/
001727	117	120	040		.ASCII /OP /
001732	125	116	111		.ASCII /UNI/
001735	124	040	045		.ASCII /T #/
001740	104	062	000		.ASCII /D2/<00>
001743	000				.ASCII <00>
001744	045	116	045	P.ABB:	.ASCII /#Ns/
001747	101	052	052		.ASCII /A**/
001752	040	120	122		.ASCII / PR/
001755	117	103	137		.ASCII /OC /
001760	122	105	124		.ASCII /RET/
001763	120	113	124		.ASCII /PKT/
001766	072	040	103		.ASCII /: C/
001771	117	116	116		.ASCII /ONN/
001774	040	111	104		.ASCII / ID/
001777	040	075	040		.ASCII / = /
002002	045	104	065		.ASCII /#D5/
002005	045	101	040		.ASCII /#A /
002010	122	105	103		.ASCII /REC/
002013	105	111	126		.ASCII /EIV/
002016	105	104	000		.ASCII /ED/<00>
002021	000				.ASCII <00>
002022	045	116	045	P.ABC:	.ASCII /#Ns/
002025	101	052	052		.ASCII /A**/
002030	040	106	124		.ASCII / FT/
002033	114	040	105		.ASCII /L E/
002036	122	122	072		.ASCII /RR:/
002041	040	122	105		.ASCII / RE/
002044	124	120	113		.ASCII /TPK/
002047	124	040	116		.ASCII /T N/
002052	117	124	040		.ASCII /OT /
002055	101	126	101		.ASCII /AVA/
002060	111	114	101		.ASCII /ILA/
002063	102	114	105		.ASCII /BLE/
002066	000	000			.ASCII <00><00>
002070	045	116	045	P.ABD:	.ASCII /#Ns/
002073	101	052	052		.ASCII /A**/
002076	040	106	123		.ASCII / FS/
002101	105	124	137		.ASCII /ET /
002104	125	120	101		.ASCII /UPA/
002107	122	072	040		.ASCII /R: /
002112	103	101	116		.ASCII /CAN/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

002115	047	124	040	.ASCII	/'T /
002120	106	111	116	.ASCII	/FIN/
002123	104	040	104	.ASCII	/D D/
002126	111	123	113	.ASCII	/ISK/
002131	040	045	104	.ASCII	/ #D/
002134	063	045	101	.ASCII	/3#A/
002137	040	111	116	.ASCII	/ IN/
002142	040	103	123	.ASCII	/ CS/
002145	124	040	045	.ASCII	/T #/
002150	104	061	000	.ASCII	/D1/<00>
002153	000			.ASCII	<00>
002154	045	116	045	P.ABE: .ASCII	/#N#/
002157	101	052	052	.ASCII	/A**/
002162	040	102	101	.ASCII	/ BA/
002165	104	040	103	.ASCII	/D C/
002170	117	116	116	.ASCII	/ONN/
002173	040	111	104	.ASCII	/ ID/
002176	040	075	040	.ASCII	/ = /
002201	045	104	065	.ASCII	/#D5/
002204	045	101	040	.ASCII	/#A /
002207	122	105	103	.ASCII	/REC/
002212	105	111	126	.ASCII	/EIV/
002215	105	104	040	.ASCII	/ED /
002220	106	122	117	.ASCII	/FRO/
002223	115	040	045	.ASCII	/M #/
002226	117	066	000	.ASCII	/06/<00>
002231	000			.ASCII	<00>
002232	045	116	045	P.ABF: .ASCII	/#N#/
002235	101	052	052	.ASCII	/A**/
002240	040	115	105	.ASCII	/ ME/
002243	123	123	101	.ASCII	/SSA/
002246	107	105	040	.ASCII	/GE /
002251	124	131	120	.ASCII	/TYP/
002254	105	040	045	.ASCII	/E #/
002257	104	062	045	.ASCII	/D2#/
002262	101	040	122	.ASCII	/A R/
002265	105	103	105	.ASCII	/ECE/
002270	111	126	105	.ASCII	/IVE/
002273	104	040	111	.ASCII	/D I/
002276	116	040	115	.ASCII	/N M/
002301	123	103	120	.ASCII	/SCP/
002304	040	120	101	.ASCII	/ PA/
002307	103	113	105	.ASCII	/CKE/
002312	124	000		.ASCII	/T/<00>
002314	045	116	045	P.ABG: .ASCII	/#N#/
002317	101	052	052	.ASCII	/A**/
002322	040	105	122	.ASCII	/ ER/
002325	122	040	111	.ASCII	/R I/
002330	116	040	123	.ASCII	/N S/
002333	105	124	137	.ASCII	/ET_/
002336	103	124	114	.ASCII	/CTL/
002341	122	137	103	.ASCII	/R_C/
002344	110	101	122	.ASCII	/HAR/

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)SEQ 0058
Page 58

002347	000				.ASCII <00>
002350	045	116	045	P.ABH:	.ASCII /#N#/
002353	101	052	052		.ASCII /A**/
002356	040	103	124		.ASCII /CT/
002361	114	122	040		.ASCII /LR /
002364	124	111	115		.ASCII /TIM/
002367	105	117	125		.ASCII /EOU/
002372	124	040	075		.ASCII /T =/
002375	040	045	104		.ASCII / #D/
002400	063	045	101		.ASCII /3#A/
002403	040	123	105		.ASCII /SE/
002406	103	117	116		.ASCII /CON/
002411	104	123	000		.ASCII /DS/<00>
002414	045	116	045	P.ABI:	.ASCII /#N#/
002417	101	052	052		.ASCII /A**/
002422	040	105	122		.ASCII /ER/
002425	122	040	111		.ASCII /R I/
002430	116	040	125		.ASCII /N U/
002433	116	111	124		.ASCII /NIT/
002436	137	111	116		.ASCII /IN/
002441	111	124	000		.ASCII /IT/<00>
002444	045	116	045	P.ABJ:	.ASCII /#N#/
002447	101	052	052		.ASCII /A**/
002452	040	101	103		.ASCII /AC/
002455	103	105	123		.ASCII /CES/
002460	123	072	040		.ASCII /S: /
002463	122	105	124		.ASCII /RET/
002466	120	113	124		.ASCII /PKT/
002471	040	110	101		.ASCII /HA/
002474	123	040	102		.ASCII /S B/
002477	101	104	040		.ASCII /AD /
002502	105	116	104		.ASCII /END/
002505	103	117	104		.ASCII /COD/
002510	105	000			.ASCII /E/<00>
002512	045	116	045	P.ABK:	.ASCII /#N#/
002515	101	052	052		.ASCII /A**/
002520	040	111	114		.ASCII /IL/
002523	114	105	107		.ASCII /LEG/
002526	101	114	040		.ASCII /AL /
002531	106	125	116		.ASCII /FUN/
002534	103	124	111		.ASCII /CTI/
002537	117	116	072		.ASCII /ON:/
002542	040	045	117		.ASCII / #0/
002545	066	000	000		.ASCII /6/<00><00>
002550	045	116	045	P.ABL:	.ASCII /#N#/
002553	101	052	052		.ASCII /A**/
002556	040	103	117		.ASCII /CO/
002561	115	115	101		.ASCII /MMA/
002564	116	104	040		.ASCII /ND /
002567	122	105	106		.ASCII /REF/
002572	040	043	040		.ASCII / # /
002575	045	104	065		.ASCII /#D5/
002600	045	101	056		.ASCII /#A./

002603	040	116	117	.ASCII	/ NO/	
002606	124	040	123	.ASCII	/T S/	
002611	105	116	124	.ASCII	/ENT/	
002614	040	102	131	.ASCII	/ BY/	
002617	040	110	117	.ASCII	/ HO/	
002622	123	124	000	.ASCII	/ST/<00>	
002625	000			.ASCII	<00>	
002626	045	116	045	P.ABM:	.ASCII	/#Ns/
002631	101	052	052	.ASCII	/A**/	
002634	040	125	116	.ASCII	/ UN/	
002637	113	116	117	.ASCII	/KNO/	
002642	127	116	040	.ASCII	/WN /	
002645	105	122	122	.ASCII	/ERR/	
002650	117	122	040	.ASCII	/OR /	
002653	114	117	107	.ASCII	/LOG/	
002656	040	106	117	.ASCII	/ FO/	
002661	122	115	101	.ASCII	/RMA/	
002664	124	040	045	.ASCII	/T #/	
002667	104	063	045	.ASCII	/D3#/	
002672	101	056	040	.ASCII	/A. /	
002675	122	105	103	.ASCII	/REC/	
002700	105	111	126	.ASCII	/EIV/	
002703	105	104	000	.ASCII	/ED/<00>	
002706	045	116	045	P.ABN:	.ASCII	/#Ns/
002711	101	052	052	.ASCII	/A**/	
002714	040	105	122	.ASCII	/ ER/	
002717	122	117	122	.ASCII	/ROR/	
002722	055	114	117	.ASCII	/-LO/	
002725	107	040	123	.ASCII	/G S/	
002730	101	126	105	.ASCII	/AVE/	
002733	040	101	122	.ASCII	/ AR/	
002736	105	101	040	.ASCII	/EA /	
002741	106	125	114	.ASCII	/FUL/	
002744	114	000		.ASCII	/L/<00>	
002746	045	116	045	P.ABO:	.ASCII	/#Ns/
002751	101	052	052	.ASCII	/A**/	
002754	040	104	125	.ASCII	/ DU/	
002757	120	072	040	.ASCII	/P: /	
002762	120	113	124	.ASCII	/PKT/	
002765	040	116	117	.ASCII	/ NO/	
002770	124	040	101	.ASCII	/T A/	
002773	126	101	111	.ASCII	/VAI/	
002776	114	101	102	.ASCII	/LAB/	
003001	114	105	000	.ASCII	/LE/<00>	
003004	045	116	045	P.ABP:	.ASCII	/#Ns/
003007	101	125	116	.ASCII	/AUN/	
003012	111	124	045	.ASCII	/IT#/	
003015	104	062	045	.ASCII	/D2#/	
003020	101	040	104	.ASCII	/A D/	
003023	122	117	120	.ASCII	/ROP/	
003026	120	105	104	.ASCII	/PED/	
003031	040	055	040	.ASCII	/ - /	
003034	000	000		.ASCII	<00><00>	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0060
Page 60
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

003036	045	101	125	P.ABR:	.ASCII	/AU/
003041	123	105	122		.ASCII	/SER/
003044	040	103	115		.ASCII	/CM/
003047	104	045	116		.ASCII	/DN/
003052	000	000			.ASCII	<00><00>
003054	045	101	103	P.ABS:	.ASCII	/AC/
003057	117	116	106		.ASCII	/ONF/
003062	111	107	125		.ASCII	/IGU/
003065	122	101	124		.ASCII	/RAT/
003070	111	117	116		.ASCII	/ION/
003073	040	105	122		.ASCII	/ER/
003076	122	045	116		.ASCII	/RN/
003101	000				.ASCII	<00>
003102	045	101	111	P.ABT:	.ASCII	/AI/
003105	116	111	124		.ASCII	/NIT/
003110	040	105	122		.ASCII	/ER/
003113	122	045	116		.ASCII	/RN/
003116	000	000			.ASCII	<00><00>
003120	045	101	124	P.ABU:	.ASCII	/AT/
003123	122	101	116		.ASCII	/RAN/
003126	123	106	105		.ASCII	/SFE/
003131	122	040	114		.ASCII	/R L/
003134	111	115	111		.ASCII	/IMI/
003137	124	040	122		.ASCII	/T R/
003142	105	101	103		.ASCII	/EAC/
003145	110	105	104		.ASCII	/MED/
003150	045	116	000		.ASCII	/N/<00>
003153	000				.ASCII	<00>
003154	045	101	105	P.ABV:	.ASCII	/AE/
003157	122	122	040		.ASCII	/RR /
003162	114	111	115		.ASCII	/LIM/
003165	111	124	040		.ASCII	/IT /
003170	122	105	101		.ASCII	/REA/
003173	103	110	105		.ASCII	/CHE/
003176	104	045	116		.ASCII	/DN/
003201	000				.ASCII	<00>
003202	045	101	125	P.ABW:	.ASCII	/AU/
003205	116	122	105		.ASCII	/NRE/
003210	103	117	126		.ASCII	/COV/
003213	105	122	101		.ASCII	/ERA/
003216	102	114	105		.ASCII	/BLE/
003221	040	104	105		.ASCII	/DE/
003224	126	040	105		.ASCII	/V E/
003227	122	122	045		.ASCII	/RR/
003232	116	000			.ASCII	/N/<00>
003234	045	101	125	P.ABX:	.ASCII	/AU/
003237	116	122	105		.ASCII	/NRE/
003242	103	117	126		.ASCII	/COV/
003245	105	122	101		.ASCII	/ERA/
003250	102	114	105		.ASCII	/BLE/
003253	040	103	124		.ASCII	/CT/
003256	114	122	040		.ASCII	/LR /
003261	105	122	122		.ASCII	/ERR/

003264	045	116	000		.ASCII	/N/<00>
003267	000				.ASCII	<00>
003270	045	101	106	P.ABY:	.ASCII	/AF/
003273	101	111	114		.ASCII	/AIL/
003276	105	104	040		.ASCII	/ED /
003301	124	117	040		.ASCII	/TO /
003304	103	117	115		.ASCII	/COM/
003307	105	040	117		.ASCII	/E O/
003312	116	114	111		.ASCII	/NLI/
003315	116	105	045		.ASCII	/NE/
003320	116	000			.ASCII	/N/<00>
003322	045	101	106	P.ABZ:	.ASCII	/AF/
003325	101	111	114		.ASCII	/AIL/
003330	105	104	040		.ASCII	/ED /
003333	124	117	040		.ASCII	/TO /
003336	101	103	103		.ASCII	/ACC/
003341	105	123	123		.ASCII	/ESS/
003344	040	114	101		.ASCII	/ LA/
003347	123	124	040		.ASCII	/ST /
003352	124	122	101		.ASCII	/TRA/
003355	103	113	040		.ASCII	/CK /
003360	104	125	122		.ASCII	/DUR/
003363	111	116	107		.ASCII	/ING/
003366	040	111	116		.ASCII	/ IN/
003371	111	124	045		.ASCII	/IT/
003374	116	000			.ASCII	/N/<00>
003376	045	101	104	P.ACA:	.ASCII	/AD/
003401	111	123	113		.ASCII	/ISK/
003404	040	127	122		.ASCII	/ WR/
003407	111	124	105		.ASCII	/ITE/
003412	040	120	122		.ASCII	/ PR/
003415	117	124	105		.ASCII	/OTE/
003420	103	124	105		.ASCII	/CTE/
003423	104	045	116		.ASCII	/DIN/
003426	000	000			.ASCII	<00><00>
003430	045	101	103	P.ACB:	.ASCII	/AC/
003433	115	104	040		.ASCII	/MD /
003436	124	111	115		.ASCII	/TIM/
003441	105	040	117		.ASCII	/E O/
003444	125	124	045		.ASCII	/UT/
003447	116	000	000		.ASCII	/N/<00><00>
003452	045	101	125	P.ACC:	.ASCII	/AU/
003455	116	111	124		.ASCII	/NIT/
003460	040	127	105		.ASCII	/ WE/
003463	116	124	040		.ASCII	/NT /
003466	124	117	040		.ASCII	/TO /
003471	101	126	101		.ASCII	/AVA/
003474	111	114	101		.ASCII	/ILA/
003477	102	114	105		.ASCII	/BLE/
003502	040	123	124		.ASCII	/ ST/
003505	101	124	105		.ASCII	/ATE/
003510	045	116	000		.ASCII	/N/<00>
003513	000				.ASCII	<00>

003514	003036'			P.ABQ:	.WORD	P.ABR
003516	003054'				.WORD	P.ABS
003520	003102'				.WORD	P.ABT
003522	003120'				.WORD	P.ABU
003524	003154'				.WORD	P.ABV
003526	003202'				.WORD	P.ABW
003530	003234'				.WORD	P.ABX
003532	003270'				.WORD	P.ABY
003534	003322'				.WORD	P.ABZ
003536	003376'				.WORD	P.ACA
003540	003430'				.WORD	P.ACB
003542	003452'				.WORD	P.ACC
003544	045	116	045	P.ACD:	.ASCII	/N#
003547	101	120	117		.ASCII	/APO/
003552	127	105	122		.ASCII	/WER/
003555	040	104	105		.ASCII	/ DE/
003560	114	101	131		.ASCII	/LAY/
003563	040	055	040		.ASCII	/ - /
003566	127	101	111		.ASCII	/WAI/
003571	124	111	116		.ASCII	/TIN/
003574	107	000			.ASCII	/G/<00>
003576	045	116	045	P.ACE:	.ASCII	/N#
003601	101	106	125		.ASCII	/AFU/
003604	116	103	124		.ASCII	/NCT/
003607	111	117	116		.ASCII	/ION/
003612	101	114	040		.ASCII	/AL /
003615	124	105	123		.ASCII	/TES/
003620	124	040	123		.ASCII	/T S/
003623	124	101	122		.ASCII	/TAR/
003626	124	105	104		.ASCII	/TED/
003631	000				.ASCII	<00>
003632	045	116	045	P.ACF:	.ASCII	/N#
003635	116	045	101		.ASCII	/N#A/
003640	105	130	105		.ASCII	/EXE/
003643	122	103	111		.ASCII	/RCI/
003646	123	105	122		.ASCII	/SER/
003651	040	123	124		.ASCII	/ ST/
003654	101	122	124		.ASCII	/ART/
003657	105	104	045		.ASCII	/ED#
003662	116	000			.ASCII	/N/<00>
003664	045	116	045	P.ACG:	.ASCII	/N#
003667	116	045	101		.ASCII	/N#A/
003672	125	116	124		.ASCII	/UNT/
003675	040	104	123		.ASCII	/ DS/
003700	113	045	123		.ASCII	/K#S/
003703	070	045	101		.ASCII	/B#A/
003706	043	040	117		.ASCII	/# 0/
003711	106	040	040		.ASCII	/F /
003714	040	043	040		.ASCII	/ # /
003717	102	131	124		.ASCII	/BYT/
003722	105	123	040		.ASCII	/ES /
003725	040	040	043		.ASCII	/ # /
003730	040	117	106		.ASCII	/ OF /

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

003733	040	040	040	.ASCII	/ /
003736	040	043	040	.ASCII	/ # /
003741	102	131	124	.ASCII	/BYT/
003744	105	123	000	.ASCII	/ES/<00>
003747	000			.ASCII	<00>
003750	045	101	040	P.ACH: .ASCII	/#A /
003753	040	055	055	.ASCII	/ -- /
003756	110	101	122	.ASCII	/HAR/
003761	104	040	105	.ASCII	/D E/
003764	122	122	117	.ASCII	/RRO/
003767	122	123	055	.ASCII	/RS-/
003772	055	040	055	.ASCII	/- - /
003775	055	123	117	.ASCII	/-SO/
004000	106	124	040	.ASCII	/FT /
004003	105	122	122	.ASCII	/ERR/
004006	117	122	123	.ASCII	/ORS/
004011	055	055	000	.ASCII	/--/<00>
004014	045	116	045	P.ACI: .ASCII	/#N# /
004017	101	040	043	.ASCII	/A # /
004022	040	040	040	.ASCII	/ /
004025	043	040	040	.ASCII	/# /
004030	124	131	120	.ASCII	/TYP/
004033	105	040	040	.ASCII	/E /
004036	122	105	101	.ASCII	/REA/
004041	104	123	040	.ASCII	/DS /
004044	040	040	040	.ASCII	/ /
004047	040	122	105	.ASCII	/ RE/
004052	101	104	040	.ASCII	/AD /
004055	040	040	127	.ASCII	/ W/
004060	122	111	124	.ASCII	/RIT/
004063	105	123	040	.ASCII	/ES /
004066	040	040	127	.ASCII	/ W/
004071	122	111	124	.ASCII	/RIT/
004074	124	105	116	.ASCII	/TEN/
004077	000			.ASCII	<00>
004100	045	101	040	P.ACJ: .ASCII	/#A /
004103	040	123	105	.ASCII	/ SE/
004106	113	040	104	.ASCII	/K D/
004111	101	124	040	.ASCII	/AT /
004114	104	122	126	.ASCII	/DRV/
004117	040	110	123	.ASCII	/ HS/
004122	124	040	123	.ASCII	/T S/
004125	105	113	040	.ASCII	/EK /
004130	104	101	124	.ASCII	/DAT/
004133	040	104	122	.ASCII	/ DR/
004136	126	040	110	.ASCII	/V H/
004141	123	124	000	.ASCII	/ST/<00>
004144	045	116	045	P.ACK: .ASCII	/#N# /
004147	101	055	055	.ASCII	/A-- /
004152	055	040	055	.ASCII	/- - /
004155	055	055	040	.ASCII	/-- /
004160	055	055	055	.ASCII	/--- /
004163	055	040	040	.ASCII	/- /

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

004166	055	055	055	.ASCII	/---/
004171	055	055	040	.ASCII	/-- /
004174	040	055	055	.ASCII	/ --/
004177	055	055	055	.ASCII	/---/
004202	055	055	055	.ASCII	/---/
004205	055	040	055	.ASCII	/- -/
004210	055	055	055	.ASCII	/---/
004213	055	055	040	.ASCII	/-- /
004216	040	055	055	.ASCII	/ --/
004221	055	055	055	.ASCII	/---/
004224	055	055	055	.ASCII	/---/
004227	055	000	000	.ASCII	/-/<00><00>
004232	045	101	040	P.ACL: .ASCII	/#A /
004235	055	055	055	.ASCII	/---/
004240	040	055	055	.ASCII	/ --/
004243	055	040	055	.ASCII	/- -/
004246	055	055	040	.ASCII	/-- /
004251	055	055	055	.ASCII	/---/
004254	040	055	055	.ASCII	/ --/
004257	055	040	055	.ASCII	/- -/
004262	055	055	040	.ASCII	/-- /
004265	055	055	055	.ASCII	/---/
004270	040	055	055	.ASCII	/ --/
004273	055	000	000	.ASCII	/-/<00><00>
004276	045	116	045	P.ACM: .ASCII	/#N#/
004301	104	062	045	.ASCII	/D2#/
004304	104	064	045	.ASCII	/D4#/
004307	101	040	040	.ASCII	/A /
004312	122	130	065	.ASCII	/RX5/
004315	060	000	000	.ASCII	/0/<00><00>
004320	045	116	045	P.ACN: .ASCII	/#N#/
004323	104	062	045	.ASCII	/D2#/
004326	104	064	045	.ASCII	/D4#/
004331	101	040	040	.ASCII	/A /
004334	122	104	065	.ASCII	/RD5/
004337	061	000	000	.ASCII	/1/<00><00>
004342	045	104	064	P.ACO: .ASCII	/#D4/
004345	045	132	063	.ASCII	/#Z3/
004350	045	104	063	.ASCII	/#D3/
004353	045	101	054	.ASCII	/#A, /
004356	045	132	063	.ASCII	/#Z3/
004361	045	101	054	.ASCII	/#A, /
004364	045	132	063	.ASCII	/#Z3/
004367	000			.ASCII	<00>
004370	045	104	064	P.ACP: .ASCII	/#D4/
004373	045	104	064	.ASCII	/#D4/
004376	045	104	064	.ASCII	/#D4/
004401	045	104	064	.ASCII	/#D4/
004404	045	104	064	.ASCII	/#D4/
004407	045	104	064	.ASCII	/#D4/
004412	045	104	064	.ASCII	/#D4/
004415	045	104	064	.ASCII	/#D4/
004420	000	000		.ASCII	<00><00>

Address	Column 1	Column 2	Column 3	Column 4	Column 5
004422	045	116	045	P.ACQ:	.ASCII /%N%/
004425	101	040	056		.ASCII /A ./
004430	040	040	040		.ASCII / ./
004433	056	040	040		.ASCII / ./
004436	103	116	124		.ASCII /CNT/
004441	122	040	040		.ASCII /R ./
004444	040	040	040		.ASCII / ./
004447	040	056	040		.ASCII / ./
004452	040	056	056		.ASCII / ./
004455	056	056	056		.ASCII /.../
004460	056	056	056		.ASCII /.../
004463	056	040	040		.ASCII / ./
004466	040	040	040		.ASCII / ./
004471	040	056	040		.ASCII / ./
004474	040	056	056		.ASCII / ./
004477	056	056	056		.ASCII /.../
004502	056	056	056		.ASCII /.../
004505	056	000	000		.ASCII /./<00><00>
004510	045	101	040	P.ACR:	.ASCII /%A /
004513	040	040	056		.ASCII / ./
004516	040	040	040		.ASCII / ./
004521	056	045	104		.ASCII /.%D/
004524	064	045	101		.ASCII /4%A/
004527	040	040	040		.ASCII / ./
004532	056	040	040		.ASCII / ./
004535	040	056	040		.ASCII / ./
004540	040	040	056		.ASCII / ./
004543	045	104	064		.ASCII /%D4/
004546	045	101	040		.ASCII /%A /
004551	040	040	056		.ASCII / ./
004554	000	000			.ASCII <00><00>
004556	045	116	045	P.ACS:	.ASCII /%N%/
004561	116	045	101		.ASCII /N%A/
004564	125	116	111		.ASCII /UNI/
004567	124	040	040		.ASCII /T /
004572	104	111	123		.ASCII /DIS/
004575	113	040	040		.ASCII /K /
004600	040	040	040		.ASCII / ./
004603	040	040	040		.ASCII / ./
004606	040	040	040		.ASCII / ./
004611	040	043	040		.ASCII / # /
004614	117	106	040		.ASCII /OF /
004617	040	040	043		.ASCII / # /
004622	040	102	114		.ASCII / BL/
004625	113	123	040		.ASCII /KS /
004630	040	040	040		.ASCII / ./
004633	040	040	040		.ASCII / ./
004636	043	040	117		.ASCII /# O/
004641	106	040	040		.ASCII /F /
004644	040	040	043		.ASCII / # /
004647	040	102	114		.ASCII / BL/
004652	113	123	040		.ASCII /KS /
004655	000				.ASCII <00>

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

004656	045	116	045	P.ACT:	.ASCII	/#Ns/
004661	101	040	040		.ASCII	/A /
004664	043	040	040		.ASCII	/0 /
004667	040	040	040		.ASCII	/ /
004672	043	040	040		.ASCII	/0 /
004675	040	040	040		.ASCII	/ /
004700	124	131	120		.ASCII	/TYP/
004703	105	040	040		.ASCII	/E /
004706	040	122	105		.ASCII	/RE/
004711	101	104	123		.ASCII	/ADS/
004714	040	040	040		.ASCII	/ /
004717	040	040	122		.ASCII	/R/
004722	105	101	104		.ASCII	/EAD/
004725	040	040	040		.ASCII	/ /
004730	040	040	040		.ASCII	/ /
004733	127	122	111		.ASCII	/WRI/
004736	124	105	123		.ASCII	/TES/
004741	040	040	127		.ASCII	/W/
004744	122	111	124		.ASCII	/RIT/
004747	124	105	116		.ASCII	/TEN/
004752	040	000			.ASCII	/ /<00>
004754	045	116	045	P.ACU:	.ASCII	/#Ns/
004757	101	055	055		.ASCII	/A--/
004762	055	055	040		.ASCII	/-- /
004765	040	055	055		.ASCII	/-- /
004770	055	055	040		.ASCII	/-- /
004773	040	055	055		.ASCII	/-- /
004776	055	055	055		.ASCII	/---/
005001	055	055	040		.ASCII	/-- /
005004	040	055	055		.ASCII	/-- /
005007	055	055	055		.ASCII	/---/
005012	055	040	040		.ASCII	/- /
005015	040	055	055		.ASCII	/-- /
005020	055	055	055		.ASCII	/---/
005023	055	040	040		.ASCII	/- /
005026	040	040	040		.ASCII	/ /
005031	055	055	055		.ASCII	/---/
005034	055	055	055		.ASCII	/---/
005037	040	040	040		.ASCII	/ /
005042	055	055	055		.ASCII	/---/
005045	055	055	055		.ASCII	/---/
005050	040	040	000		.ASCII	/ /<00>
005053	000				.ASCII	<00>
005054	045	116	045	P.ACV:	.ASCII	/#Ns/
005057	123	061	045		.ASCII	/S1#/
005062	104	062	045		.ASCII	/D2#/
005065	123	064	045		.ASCII	/S4#/
005070	104	062	045		.ASCII	/D2#/
005073	101	040	040		.ASCII	/A /
005076	040	104	102		.ASCII	/DB/
005101	116	122	104		.ASCII	/NRD/
005104	065	061	040		.ASCII	/51 /
005107	040	045	104		.ASCII	/ #D/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

005112	066	045	123	.ASCII	/6#S/	
005115	063	045	104	.ASCII	/3#D/	
005120	066	045	123	.ASCII	/6#S/	
005123	065	045	104	.ASCII	/5#D/	
005126	066	045	123	.ASCII	/6#S/	
005131	063	045	104	.ASCII	/3#D/	
005134	066	000		.ASCII	/6/<00>	
005136	045	116	045	P.ACW:	.ASCII	/#N#/
005141	104	062	045	.ASCII	/D2#/	
005144	104	064	045	.ASCII	/D4#/	
005147	101	040	040	.ASCII	/A /	
005152	122	104	065	.ASCII	/RD5/	
005155	062	000	000	P.ACX:	.ASCII	/2/<00><00>
005160	045	116	045	.ASCII	/#N#/	
005163	123	061	045	.ASCII	/S1#/	
005166	104	062	045	.ASCII	/D2#/	
005171	123	064	045	.ASCII	/S4#/	
005174	104	062	045	.ASCII	/D2#/	
005177	101	040	040	.ASCII	/A /	
005202	040	104	102	.ASCII	/ DB/	
005205	116	122	104	.ASCII	/NRD/	
005210	065	062	040	.ASCII	/52 /	
005213	040	045	104	.ASCII	/ #D/	
005216	066	045	123	.ASCII	/6#S/	
005221	063	045	104	.ASCII	/3#D/	
005224	066	045	123	.ASCII	/6#S/	
005227	065	045	104	.ASCII	/5#D/	
005232	066	045	123	.ASCII	/6#S/	
005235	063	045	104	.ASCII	/3#D/	
005240	066	000		.ASCII	/6/<00>	
005242	045	116	045	P.ACY:	.ASCII	/#N#/
005245	104	062	045	.ASCII	/D2#/	
005250	104	064	045	.ASCII	/D4#/	
005253	101	040	040	.ASCII	/A /	
005256	077	077	077	.ASCII	/???/	
005261	077	000	000	P.ACZ:	.ASCII	/?/<00><00>
005264	124	117	117	.ASCII	/T00/	
005267	040	115	101	.ASCII	/ MA/	
005272	116	131	040	.ASCII	/NY /	
005275	125	116	111	.ASCII	/UNI/	
005300	124	123	000	.ASCII	/TS/<00>	
005303	000			.ASCII	<00>	
005304	116	117	124	P.ADA:	.ASCII	/NOT/
005307	040	105	116	.ASCII	/ EN/	
005312	117	125	107	.ASCII	/OUG/	
005315	110	040	106	.ASCII	/H F/	
005320	122	105	105	.ASCII	/REE/	
005323	040	115	105	.ASCII	/ ME/	
005326	115	117	122	.ASCII	/MOR/	
005331	131	040	106	.ASCII	/Y F/	
005334	117	122	040	.ASCII	/OR /	
005337	101	114	114	.ASCII	/ALL/	
005342	117	103	101	.ASCII	/OCA/	

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

005345	124	111	116	.ASCII	/TIN/
005350	107	040	122	.ASCII	/G R/
005353	105	101	104	.ASCII	/EAD/
005356	057	127	122	.ASCII	<57>/WR/
005361	111	124	105	.ASCII	/ITE/
005364	040	102	125	.ASCII	/ BU/
005367	106	106	105	.ASCII	/FFE/
005372	122	123	000	.ASCII	/RS/<00>
005375	000			.ASCII	<00>
005376	122	105	107	P.ADB:	.ASCII /REG/
005401	040	105	130		.ASCII / EX/
005404	111	123	124		.ASCII /IST/
005407	105	116	103		.ASCII /ENC/
005412	105	040	124		.ASCII /E T/
005415	105	123	124		.ASCII /EST/
005420	040	106	101		.ASCII / FA/
005423	111	114	105		.ASCII /ILE/
005426	104	000			.ASCII /D/<00>
005430	126	105	103	P.ADC:	.ASCII /VEC/
005433	124	117	122		.ASCII /TOR/
005436	040	124	105		.ASCII / TE/
005441	123	124	040		.ASCII /ST /
005444	106	101	111		.ASCII /FAI/
005447	114	105	104		.ASCII /LED/
005452	000	000			.ASCII <00><00>
005454	102	122	040	P.ADD:	.ASCII /BR /
005457	114	105	126		.ASCII /LEV/
005462	105	114	040		.ASCII /EL /
005465	124	105	123		.ASCII /TES/
005470	124	040	106		.ASCII /T F/
005473	101	111	114		.ASCII /AIL/
005476	105	104	000		.ASCII /ED/<00>
005501	000				.ASCII <00>
005502	111	116	111	P.ADE:	.ASCII /INI/
005505	124	040	123		.ASCII /T S/
005510	105	121	125		.ASCII /EQU/
005513	105	116	103		.ASCII /ENC/
005516	105	040	106		.ASCII /E F/
005521	101	111	114		.ASCII /AIL/
005524	105	104	000		.ASCII /ED/<00>
005527	000				.ASCII <00>
005530	106	101	124	P.ADF:	.ASCII /FAT/
005533	101	114	040		.ASCII /AL /
005536	103	124	114		.ASCII /CTL/
005541	122	040	105		.ASCII /R E/
005544	122	122	000		.ASCII /RR/<00>
005547	000				.ASCII <00>
005550	117	116	114	P.ADG:	.ASCII /ONL/
005553	111	116	105		.ASCII /INE/
005556	040	106	101		.ASCII / FA/
005561	111	114	105		.ASCII /ILE/
005564	104	000			.ASCII /D/<00>
005566	127	122	111	P.ADH:	.ASCII /WRI/

ZRGAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

005571	124	105	055	.ASCII	/TE-/
005574	120	122	117	.ASCII	/PRO/
005577	124	105	103	.ASCII	/TEC/
005602	124	040	103	.ASCII	/T C/
005605	117	116	106	.ASCII	/ONF/
005610	114	111	103	.ASCII	/LIC/
005613	124	000	000	.ASCII	/T/<00><00>
005616	101	103	103	P.ADI:	.ASCII /ACC/
005621	105	123	123	.ASCII	/ESS/
005624	040	106	101	.ASCII	/FA/
005627	111	114	105	.ASCII	/ILE/
005632	104	000		.ASCII	/D/<00>
005634	106	101	124	P.ADJ:	.ASCII /FAT/
005637	101	114	040	.ASCII	/AL /
005642	111	057	117	.ASCII	/I/<57>/0/
005645	040	105	122	.ASCII	/ER/
005650	122	000		.ASCII	/R/<00>
005652	103	124	114	P.ADK:	.ASCII /CTL/
005655	122	040	124	.ASCII	/R T/
005660	111	115	105	.ASCII	/IME/
005663	117	125	124	.ASCII	/OUT/
005666	000	000		.ASCII	<00><00>
005670	106	101	111	P.ADL:	.ASCII /FAI/
005673	114	105	104	.ASCII	/LED/
005676	040	124	117	.ASCII	/TO/
005701	040	123	105	.ASCII	/SE/
005704	116	104	040	.ASCII	/ND /
005707	123	105	124	.ASCII	/SET/
005712	055	103	124	.ASCII	/-CT/
005715	114	122	055	.ASCII	/LR-/
005720	103	110	101	.ASCII	/CHA/
005723	122	101	103	.ASCII	/RAC/
005726	124	105	122	.ASCII	/TER/
005731	111	123	124	.ASCII	/IST/
005734	111	103	123	.ASCII	/ICS/
005737	040	103	117	.ASCII	/CO/
005742	115	115	101	.ASCII	/MMA/
005745	116	104	000	.ASCII	/ND/<00>
005750	123	105	124	P.ADM:	.ASCII /SET/
005753	055	103	124	.ASCII	/-CT/
005756	114	122	055	.ASCII	/LR-/
005761	103	110	101	.ASCII	/CHA/
005764	122	101	103	.ASCII	/RAC/
005767	124	105	122	.ASCII	/TER/
005772	111	123	124	.ASCII	/IST/
005775	111	103	123	.ASCII	/ICS/
006000	040	122	105	.ASCII	/RE/
006003	123	120	117	.ASCII	/SPO/
006006	116	123	105	.ASCII	/NSE/
006011	040	110	101	.ASCII	/HA/
006014	123	040	102	.ASCII	/S B/
006017	101	104	040	.ASCII	/AD /
006022	105	116	104	.ASCII	/END/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

006025	103	117	104	.ASCII	/COD/	
006030	105	040	117	.ASCII	/E O/	
006033	122	040	106	.ASCII	/R F/	
006036	114	101	107	.ASCII	/LAG/	
006041	123	040	111	.ASCII	/S I/	
006044	116	040	105	.ASCII	/N E/	
006047	122	122	000	.ASCII	/RR/<00>	
006052	106	101	111	P.ADN:	.ASCII	/FAI/
006055	114	105	104	.ASCII	/LED/	
006060	040	124	117	.ASCII	/ TO/	
006063	040	123	105	.ASCII	/ SE/	
006066	116	104	040	.ASCII	/ND /	
006071	117	116	055	.ASCII	/ON-/	
006074	114	111	116	.ASCII	/LIN/	
006077	105	040	103	.ASCII	/E C/	
006102	117	115	115	.ASCII	/OMM/	
006105	101	116	104	.ASCII	/AND/	
006110	000	000		.ASCII	<00><00>	
006112	117	116	055	P.ADQ:	.ASCII	/ON-/
006115	114	111	116	.ASCII	/LIN/	
006120	105	040	122	.ASCII	/E R/	
006123	105	123	120	.ASCII	/ESP/	
006126	117	116	123	.ASCII	/ONS/	
006131	105	040	110	.ASCII	/E H/	
006134	101	123	040	.ASCII	/AS /	
006137	102	101	104	.ASCII	/BAD/	
006142	040	105	116	.ASCII	/ EN/	
006145	104	103	117	.ASCII	/DCO/	
006150	104	105	000	.ASCII	/DE/<00>	
006153	000			.ASCII	<00>	
006154	117	116	055	P.ADP:	.ASCII	/ON-/
006157	114	111	116	.ASCII	/LIN/	
006162	105	040	122	.ASCII	/E R/	
006165	105	123	120	.ASCII	/ESP/	
006170	117	116	123	.ASCII	/ONS/	
006173	105	040	110	.ASCII	/E H/	
006176	101	123	040	.ASCII	/AS /	
006201	125	116	113	.ASCII	/UNK/	
006204	116	117	127	.ASCII	/NOW/	
006207	116	040	104	.ASCII	/N D/	
006212	105	126	111	.ASCII	/EVI/	
006215	103	105	000	.ASCII	/CE/<00>	
006220	111	057	117	P.ADQ:	.ASCII	/I/<57>/0/
006223	040	122	105	.ASCII	/ RE/	
006226	121	125	105	.ASCII	/QUE/	
006231	123	124	040	.ASCII	/ST /	
006234	106	101	111	.ASCII	/FAI/	
006237	114	105	104	.ASCII	/LED/	
006242	000	000		.ASCII	<00><00>	
006244	045	101	115	P.ADR:	.ASCII	/AM/
006247	117	122	105	.ASCII	/ORE/	
006252	040	124	110	.ASCII	/ TH/	
006255	101	116	040	.ASCII	/AN /	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

006260	045	104	062	.ASCII	/#D2/
006263	045	101	040	.ASCII	/#A /
006266	125	116	111	.ASCII	/UNI/
006271	124	123	040	.ASCII	/TS /
006274	123	120	105	.ASCII	/SPE/
006277	103	111	106	.ASCII	/CIF/
006302	111	105	104	.ASCII	/IED/
006305	000			.ASCII	<00>
006306	045	101	052	P.ADS:	.ASCII /#A*/
006311	040	116	117	.ASCII	/ NO/
006314	040	122	105	.ASCII	/ RE/
006317	123	120	117	.ASCII	/SPO/
006322	116	123	105	.ASCII	/NSE/
006325	040	101	124	.ASCII	/ AT/
006330	040	101	104	.ASCII	/ AD/
006333	104	122	105	.ASCII	/DRE/
006336	123	123	040	.ASCII	/SS /
006341	045	117	066	.ASCII	/#06/
006344	000	000		.ASCII	<00><00>
006346	045	101	052	P.ADT:	.ASCII /#A*/
006351	040	111	116	.ASCII	/ IN/
006354	103	117	122	.ASCII	/COR/
006357	122	105	103	.ASCII	/REC/
006362	124	040	102	.ASCII	/T B/
006365	122	040	114	.ASCII	/R L/
006370	105	126	105	.ASCII	/EVE/
006373	114	040	106	.ASCII	/L F/
006376	117	122	040	.ASCII	/OR /
006401	104	105	126	.ASCII	/DEV/
006404	040	045	117	.ASCII	/ #0/
006407	066	000	000	.ASCII	/6/<00><00>
006412	045	101	052	P.ADU:	.ASCII /#A*/
006415	040	123	124	.ASCII	/ ST/
006420	105	120	040	.ASCII	/EP /
006423	045	104	061	.ASCII	/#D1/
006426	045	101	040	.ASCII	/#A /
006431	122	105	101	.ASCII	/REA/
006434	104	040	105	.ASCII	/D E/
006437	122	122	000	.ASCII	/RR/<00>
006442	045	101	052	P.ADV:	.ASCII /#A*/
006445	040	102	101	.ASCII	/ BA/
006450	104	040	123	.ASCII	/D S/
006453	101	040	103	.ASCII	/A C/
006456	117	104	105	.ASCII	/ODE/
006461	040	106	122	.ASCII	/ FR/
006464	117	115	040	.ASCII	/OM /
006467	104	105	126	.ASCII	/DEV/
006472	040	045	117	.ASCII	/ #0/
006475	066	000	000	.ASCII	/6/<00><00>
006500	045	101	052	P.ADW:	.ASCII /#A*/
006503	040	104	111	.ASCII	/ DI/
006506	123	113	045	.ASCII	/SK#/
006511	104	062	045	.ASCII	/D2#/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

006514	101	040	127	.ASCII	/A W/	
006517	105	116	124	.ASCII	/ENT/	
006522	040	117	106	.ASCII	/ OF/	
006525	106	114	111	.ASCII	/FLI/	
006530	116	105	000	.ASCII	/NE/<00>	
006533	000			.ASCII	<00>	
006534	045	101	052	P.ADX:	.ASCII	/#A*/
006537	040	104	105	.ASCII	/ DE/	
006542	126	111	103	.ASCII	/VIC/	
006545	105	040	045	.ASCII	/E #/	
006550	117	066	045	.ASCII	/O6#/	
006553	101	040	116	.ASCII	/A N/	
006556	117	124	040	.ASCII	/OT /	
006561	120	122	117	.ASCII	/PRO/	
006564	103	105	123	.ASCII	/CES/	
006567	123	111	116	.ASCII	/SIN/	
006572	107	040	103	.ASCII	/G C/	
006575	117	115	115	.ASCII	/OMM/	
006600	101	116	104	.ASCII	/AND/	
006603	040	120	101	.ASCII	/ PA/	
006606	103	113	105	.ASCII	/CKE/	
006611	124	123	000	P.ADY:	.ASCII	/TS/<00>
006614	040	055	040	.ASCII	/ - /	
006617	165	156	162	.ASCII	/unr/	
006622	145	143	157	.ASCII	/eco/	
006625	147	156	151	.ASCII	/gni/	
006630	172	145	144	.ASCII	/zed/	
006633	040	115	105	.ASCII	/ ME/	
006636	123	123	101	.ASCII	/SSA/	
006641	107	105	040	.ASCII	/GE /	
006644	124	131	120	.ASCII	/TYP/	
006647	105	000	000	P.ADZ:	.ASCII	/E/<00><00>
006652	040	055	040	.ASCII	/ - /	
006655	165	156	162	.ASCII	/unr/	
006660	145	143	157	.ASCII	/eco/	
006663	147	156	151	.ASCII	/gni/	
006666	172	145	144	.ASCII	/zed/	
006671	040	143	157	.ASCII	/ co/	
006674	156	156	145	.ASCII	/nne/	
006677	143	164	151	.ASCII	/cti/	
006702	157	156	040	.ASCII	/on /	
006705	151	144	000	P.AEA:	.ASCII	/id/<00>
006710	040	055	040	.ASCII	/ - /	
006713	165	156	162	.ASCII	/unr/	
006716	145	143	157	.ASCII	/eco/	
006721	147	156	151	.ASCII	/gni/	
006724	172	145	144	.ASCII	/zed/	
006727	040	122	105	.ASCII	/ RE/	
006732	124	125	122	.ASCII	/TUR/	
006735	116	040	155	.ASCII	/N m/	
006740	145	163	163	.ASCII	/ese/	
006743	141	147	145	.ASCII	/ege/	
006746	000	000		.ASCII	<00><00>	

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 B111-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

006750	040	055	040	P.AEB:	.ASCII / - /
006753	165	156	162		.ASCII /unr/
006756	145	143	157		.ASCII /eco/
006761	147	156	151		.ASCII /gni/
006764	172	145	144		.ASCII /zed/
006767	040	122	105		.ASCII / RE/
006772	124	125	122		.ASCII /TUR/
006775	116	040	120		.ASCII /N P/
007000	101	103	113		.ASCII /ACK/
007003	105	124	000		.ASCII /ET/<00>
007006	040	055	040	P.AEC:	.ASCII / - /
007011	165	156	162		.ASCII /unr/
007014	145	143	157		.ASCII /eco/
007017	147	156	151		.ASCII /gni/
007022	172	145	144		.ASCII /zed/
007025	040	103	122		.ASCII / CR/
007030	116	000			.ASCII /N/<00>
007032	040	055	040	P.AED:	.ASCII / - /
007035	125	116	122		.ASCII /UNR/
007040	105	103	117		.ASCII /ECO/
007043	107	116	111		.ASCII /GNI/
007046	132	105	104		.ASCII /ZED/
007051	040	117	120		.ASCII / OP/
007054	103	117	104		.ASCII /COD/
007057	105	000	000		.ASCII /E/<00><00>
007062	040	055	040	P.AEE:	.ASCII / - /
007065	115	123	103		.ASCII /MSC/
007070	120	040	123		.ASCII /P S/
007073	124	101	124		.ASCII /TAT/
007076	125	123	040		.ASCII /US /
007101	103	117	104		.ASCII /COD/
007104	105	040	105		.ASCII /E E/
007107	122	122	000		.ASCII /RR/<00>
007112	040	055	040	P.AEF:	.ASCII / - /
007115	104	125	120		.ASCII /DUP/
007120	040	123	124		.ASCII / ST/
007123	101	124	125		.ASCII /ATU/
007126	123	040	103		.ASCII /S C/
007131	117	104	105		.ASCII /ODE/
007134	040	105	122		.ASCII / ER/
007137	122	000	000		.ASCII /R/<00><00>
007142	040	055	040	P.AEG:	.ASCII / - /
007145	165	156	162		.ASCII /unr/
007150	145	143	157		.ASCII /eco/
007153	147	156	151		.ASCII /gni/
007156	172	145	144		.ASCII /zed/
007161	040	123	124		.ASCII / ST/
007164	101	124	125		.ASCII /ATU/
007167	123	040	103		.ASCII /S C/
007172	117	104	105		.ASCII /ODE/
007175	000				.ASCII <00>
007176	040	055	040	P.AEH:	.ASCII / - /
007201	114	102	116		.ASCII /LBN/

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

007204	040	110	117	.ASCII	/ HO/
007207	123	124	040	.ASCII	/ST /
007212	103	117	115	.ASCII	/COM/
007215	120	101	122	.ASCII	/PAR/
007220	105	040	105	.ASCII	/E E/
007223	122	122	000	.ASCII	/RR/<00>
007226	040	055	040	P.AEI:	.ASCII / - /
007231	104	102	116	.ASCII	/DBN/
007234	040	110	117	.ASCII	/ HO/
007237	123	124	040	.ASCII	/ST /
007242	103	117	115	.ASCII	/COM/
007245	120	101	122	.ASCII	/PAR/
007250	105	040	105	.ASCII	/E E/
007253	122	122	000	.ASCII	/RR/<00>
007256	040	055	040	P.AEJ:	.ASCII / - /
007261	125	116	101	.ASCII	/UNA/
007264	102	114	105	.ASCII	/BLE/
007267	040	124	117	.ASCII	/ TO/
007272	040	114	117	.ASCII	/ LO/
007275	101	104	040	.ASCII	/AD /
007300	104	125	120	.ASCII	/DUP/
007303	040	115	105	.ASCII	/ ME/
007306	104	111	101	.ASCII	/DIA/
007311	040	000	000	.ASCII	/ /<00><00>
007314	040	055	040	P.AEK:	.ASCII / - /
007317	105	122	122	.ASCII	/ERR/
007322	040	111	116	.ASCII	/ IN/
007325	040	104	125	.ASCII	/ DU/
007330	120	055	120	.ASCII	/P-P/
007333	113	124	040	.ASCII	/KT /
007336	127	110	105	.ASCII	/WHE/
007341	116	040	125	.ASCII	/N U/
007344	123	111	116	.ASCII	/SIN/
007347	107	040	103	.ASCII	/G C/
007352	124	114	122	.ASCII	/TLR/
007355	040	114	103	.ASCII	/ LC/
007360	040	120	122	.ASCII	/ PR/
007363	107	000	000	.ASCII	/G/<00><00>
007366	045	101	111	P.AEM:	.ASCII /AI/
007371	116	126	101	.ASCII	/NVA/
007374	114	111	104	.ASCII	/LID/
007377	040	103	117	.ASCII	/ CO/
007402	115	115	101	.ASCII	/MMA/
007405	116	104	000	.ASCII	/ND/<00>
007410	045	101	103	P.AEN:	.ASCII /AC/
007413	117	115	115	.ASCII	/OMM/
007416	101	116	104	.ASCII	/AND/
007421	040	101	102	.ASCII	/ AB/
007424	117	122	124	.ASCII	/ORT/
007427	105	104	000	.ASCII	/ED/<00>
007432	045	101	125	P.AEO:	.ASCII /AU/
007435	116	111	124	.ASCII	/NIT/
007440	040	117	106	.ASCII	/ OF/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

007443	106	114	111		.ASCII	/FLI/
007446	116	105	000		.ASCII	/NE/<00>
007451	000				.ASCII	<00>
007452	045	101	124	P.AEP:	.ASCII	/MAT/
007455	122	101	116		.ASCII	/RAN/
007460	123	111	124		.ASCII	/SIT/
007463	111	117	116		.ASCII	/ION/
007466	040	124	117		.ASCII	/ TO/
007471	040	101	126		.ASCII	/ AV/
007474	101	111	114		.ASCII	/AIL/
007477	101	102	114		.ASCII	/ABL/
007502	105	040	123		.ASCII	/E S/
007505	124	101	124		.ASCII	/TAT/
007510	105	000			.ASCII	/E/<00>
007512	045	101	115	P.AEQ:	.ASCII	/MAM/
007515	105	104	111		.ASCII	/EDI/
007520	101	040	106		.ASCII	/A F/
007523	117	122	115		.ASCII	/ORM/
007526	101	124	040		.ASCII	/AT /
007531	105	122	122		.ASCII	/ERR/
007534	000	000			.ASCII	<00><00>
007536	045	101	127	P.AER:	.ASCII	/MAW/
007541	122	111	124		.ASCII	/RIT/
007544	105	055	120		.ASCII	/E-P/
007547	122	117	124		.ASCII	/ROT/
007552	105	103	124		.ASCII	/ECT/
007555	105	104	000		.ASCII	/ED/<00>
007560	045	101	104	P.AES:	.ASCII	/MAD/
007563	105	126	111		.ASCII	/EVI/
007566	103	105	040		.ASCII	/CE /
007571	103	117	115		.ASCII	/COM/
007574	120	101	122		.ASCII	/PAR/
007577	105	040	105		.ASCII	/E E/
007602	122	122	000		.ASCII	/RR/<00>
007605	000				.ASCII	<00>
007606	045	101	104	P.AET:	.ASCII	/MAD/
007611	101	124	101		.ASCII	/ATA/
007614	040	105	122		.ASCII	/ ER/
007617	122	000	000		.ASCII	/R/<00><00>
007622	045	101	110	P.AEU:	.ASCII	/MAH/
007625	117	123	124		.ASCII	/OST/
007630	040	102	125		.ASCII	/ BU/
007633	106	106	105		.ASCII	/FFE/
007636	122	040	101		.ASCII	/R A/
007641	103	103	105		.ASCII	/CCE/
007644	123	123	040		.ASCII	/SS /
007647	105	122	122		.ASCII	/ERR/
007652	000	000			.ASCII	<00><00>
007654	045	101	103	P.AEV:	.ASCII	/MAC/
007657	124	114	122		.ASCII	/TLR/
007662	040	105	122		.ASCII	/ ER/
007665	122	000	000		.ASCII	/R/<00><00>
007670	045	101	104	P.AEW:	.ASCII	/MAD/

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK#USER2:(DIETZ.RDRX)ZRQACO.BL1;82 (35)

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

007673	122	111	126	.ASCII	/RIV/
007676	105	040	105	.ASCII	/E E/
007701	122	122	000	.ASCII	/RR/<00>
007704	045	101	115	P.AEX:	.ASCII /#AM/
007707	105	123	123	.ASCII	/ESS/
007712	101	107	105	.ASCII	/AGE/
007715	040	106	122	.ASCII	/ FR/
007720	117	115	040	.ASCII	/OM /
007723	111	116	124	.ASCII	/INT/
007726	105	122	116	.ASCII	/ERN/
007731	101	114	040	.ASCII	/AL /
007734	104	111	101	.ASCII	/DIA/
007737	107	116	117	.ASCII	/GNO/
007742	123	124	111	.ASCII	/STI/
007745	103	123	000	P.AEY:	.ASCII /CS/<00>
007750	045	101	110	.ASCII	/#AH/
007753	117	123	124	.ASCII	/OST/
007756	040	103	117	.ASCII	/ CO/
007761	115	120	101	.ASCII	/MPA/
007764	122	105	040	.ASCII	/RE /
007767	105	122	122	.ASCII	/ERR/
007772	000	000		P.AEZ:	.ASCII <00><00>
007774	045	101	103	.ASCII	/#AC/
007777	117	115	115	.ASCII	/OMM/
010002	101	116	104	.ASCII	/AND/
010005	040	124	111	.ASCII	/ TI/
010010	115	105	117	.ASCII	/MEO/
010013	125	124	000	P.AEL:	.ASCII /UT/<00>
010016	007366'			.WORD	P.AEM
010020	007410'			.WORD	P.AEN
010022	007432'			.WORD	P.AEO
010024	007452'			.WORD	P.AEP
010026	007512'			.WORD	P.AEQ
010030	007536'			.WORD	P.AER
010032	007560'			.WORD	P.AES
010034	007606'			.WORD	P.AET
010036	007622'			.WORD	P.AEU
010040	007654'			.WORD	P.AEV
010042	007670'			.WORD	P.AEW
010044	007704'			.WORD	P.AEX
010046	007750'			.WORD	P.AEY
010050	007774'			.WORD	P.AEZ
010052	045	116	045	P.AFA:	.ASCII /#N#/
010055	101	105	122	.ASCII	/AER/
010060	122	040	114	.ASCII	/R L/
010063	117	107	040	.ASCII	/OG /
010066	115	105	123	.ASCII	/MES/
010071	123	101	107	.ASCII	/SAG/
010074	105	040	122	.ASCII	/E R/
010077	105	103	105	.ASCII	/ECE/
010102	111	126	105	.ASCII	/IVE/
010105	104	072	045	.ASCII	/D: #/
010110	116	000		.ASCII	/N/<00>

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

010112	045	101	052	P.AFC:	.ASCII	/#A*/
010115	040	103	124		.ASCII	/ CT/
010120	114	122	040		.ASCII	/LR /
010123	105	122	122		.ASCII	/ERR/
010126	045	116	000		.ASCII	/#N/<00>
010131	000				.ASCII	<00>
010132	045	101	052	P.AFD:	.ASCII	/#A*/
010135	040	110	117		.ASCII	/ HO/
010140	123	124	040		.ASCII	/ST /
010143	115	105	115		.ASCII	/MEM/
010146	117	122	131		.ASCII	/ORY/
010151	040	101	103		.ASCII	/ AC/
010154	103	105	123		.ASCII	/CES/
010157	123	040	105		.ASCII	/S E/
010162	122	122	045		.ASCII	/RR#/
010165	116	000	000		.ASCII	/N/<00><00>
010170	045	101	052	P.AFE:	.ASCII	/#A*/
010173	040	104	111		.ASCII	/ DI/
010176	123	113	045		.ASCII	/SK#/
010201	104	062	045		.ASCII	/D2#/
010204	101	040	055		.ASCII	/A -/
010207	040	104	111		.ASCII	/ DI/
010212	123	113	040		.ASCII	/SK /
010215	124	122	101		.ASCII	/TRA/
010220	116	123	106		.ASCII	/NSF/
010223	105	122	040		.ASCII	/ER /
010226	105	122	122		.ASCII	/ERR/
010231	045	116	000		.ASCII	/#N/<00>
010234	045	101	052	P.AFF:	.ASCII	/#A*/
010237	040	104	111		.ASCII	/ DI/
010242	123	113	045		.ASCII	/SK#/
010245	104	062	045		.ASCII	/D2#/
010250	101	040	055		.ASCII	/A -/
010253	040	042	123		.ASCII	/ "S/
010256	124	101	116		.ASCII	/TAN/
010261	104	101	122		.ASCII	/DAR/
010264	104	040	104		.ASCII	/D D/
010267	111	123	113		.ASCII	/ISK/
010272	040	111	116		.ASCII	/ IN/
010275	124	105	122		.ASCII	/TER/
010300	103	117	116		.ASCII	/CON/
010303	116	105	103		.ASCII	/NEC/
010306	124	042	040		.ASCII	/T" /
010311	105	122	122		.ASCII	/ERR/
010314	045	116	000		.ASCII	/#N/<00>
010317	000				.ASCII	<00>
010320	045	101	052	P.AFG:	.ASCII	/#A*/
010323	040	104	111		.ASCII	/ DI/
010326	123	113	045		.ASCII	/SK#/
010331	104	062	045		.ASCII	/D2#/
010334	101	040	055		.ASCII	/A -/
010337	040	042	123		.ASCII	/ "S/
010342	115	101	114		.ASCII	/MAL/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0078
Page 78
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

010345	114	040	104	.ASCII	/L D/
010350	111	123	113	.ASCII	/ISK/
010353	042	040	105	.ASCII	/" E/
010356	122	122	045	.ASCII	/RR#/
010361	116	000	000	.ASCII	/N/<00><00>
010364	010112'			P.AFB:	.WORD P.AFC
010366	010132'				.WORD P.AFD
010370	010170'				.WORD P.AFE
010372	010234'				.WORD P.AFF
010374	010320'				.WORD P.AFG
010376	045	116	045	P.AFH:	.ASCII /#N#/
010401	101	111	057		.ASCII /AI/<57>
010404	117	040	102		.ASCII /O B/
010407	125	106	106		.ASCII /UFF/
010412	105	122	040		.ASCII /ER /
010415	101	104	104		.ASCII /ADD/
010420	122	105	123		.ASCII /RES/
010423	123	040	106		.ASCII /S F/
010426	117	122	040		.ASCII /OR /
010431	122	105	101		.ASCII /REA/
010434	104	040	050		.ASCII /D (/
010437	063	062	040		.ASCII /32 /
010442	102	111	124		.ASCII /BIT/
010445	123	051	072		.ASCII /S):/
010450	040	045	117		.ASCII / #0/
010453	066	045	101		.ASCII /6#A/
010456	040	045	117		.ASCII / #0/
010461	066	045	116		.ASCII /6#N/
010464	000	000			.ASCII <00><00>
010466	045	116	045	P.AFI:	.ASCII /#N#/
010471	101	111	057		.ASCII /AI/<57>
010474	117	040	102		.ASCII /O B/
010477	125	106	106		.ASCII /UFF/
010502	105	122	040		.ASCII /ER /
010505	101	104	104		.ASCII /ADD/
010510	122	105	123		.ASCII /RES/
010513	123	040	106		.ASCII /S F/
010516	117	122	040		.ASCII /OR /
010521	127	122	111		.ASCII /WRI/
010524	124	105	040		.ASCII /TE /
010527	050	063	062		.ASCII /(32/
010532	040	102	111		.ASCII / BI/
010535	124	123	051		.ASCII /TS)/
010540	072	040	045		.ASCII /: #/
010543	117	066	045		.ASCII /06#/
010546	101	040	045		.ASCII /A #/
010551	117	066	000		.ASCII /06/<00>
010554	045	116	045	P.AFJ:	.ASCII /#N#/
010557	101	114	102		.ASCII /ALB/
010562	116	072	040		.ASCII /N: /
010565	050	122	105		.ASCII /(RE/
010570	101	104	051		.ASCII /AD)/
010573	040	045	104		.ASCII / #D/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

010576	065	045	101	.ASCII	/5#A/
010601	056	040	050	.ASCII	/. (/
010604	117	103	124	.ASCII	/OCT/
010607	040	045	117	.ASCII	/ #0/
010612	066	045	101	.ASCII	/6#A/
010615	040	045	117	.ASCII	/ #0/
010620	066	045	101	.ASCII	/6#A/
010623	051	000	000	.ASCII	/)/<00><00>
010626	045	116	045	P.AFK: .ASCII	/#N#/
010631	101	114	102	.ASCII	/ALB/
010634	116	072	040	.ASCII	/N: /
010637	050	127	122	.ASCII	/(WR/
010642	111	124	105	.ASCII	/ITE/
010645	051	040	045	.ASCII	/) #/
010650	104	065	045	.ASCII	/D5#/
010653	101	056	040	.ASCII	/A. /
010656	050	117	103	.ASCII	/(OC/
010661	124	040	045	.ASCII	/T #/
010664	117	066	045	.ASCII	/06#/
010667	101	040	045	.ASCII	/A #/
010672	117	066	045	.ASCII	/06#/
010675	101	051	000	.ASCII	/A)/<00>
010700	045	116	045	P.AFL: .ASCII	/#N#/
010703	101	122	105	.ASCII	/ARE/
010706	120	114	101	.ASCII	/PLA/
010711	103	105	115	.ASCII	/CEM/
010714	105	116	124	.ASCII	/ENT/
010717	040	102	114	.ASCII	/ BL/
010722	117	103	113	.ASCII	/OCK/
010725	040	116	117	.ASCII	/ NO/
010730	056	040	045	.ASCII	/. #/
010733	104	065	045	.ASCII	/D5#/
010736	101	056	040	.ASCII	/A. /
010741	050	117	103	.ASCII	/(OC/
010744	124	040	045	.ASCII	/T #/
010747	117	066	045	.ASCII	/06#/
010752	101	040	045	.ASCII	/A #/
010755	117	066	045	.ASCII	/06#/
010760	101	051	000	.ASCII	/A)/<00>
010763	000			.ASCII	<00>
010764	045	116	045	P.AFM: .ASCII	/#N#/
010767	101	102	131	.ASCII	/ABY/
010772	124	105	040	.ASCII	/TE /
010775	103	117	125	.ASCII	/COU/
011000	116	124	040	.ASCII	/NT /
011003	111	116	040	.ASCII	/IN /
011006	122	105	101	.ASCII	/REA/
011011	104	040	103	.ASCII	/D C/
011014	117	115	115	.ASCII	/OPM/
011017	101	116	104	.ASCII	/AND/
011022	072	040	045	.ASCII	/: #/
011025	104	065	045	.ASCII	/D5#/
011030	101	056	000	.ASCII	/A./<00>

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2
RD/RX EXERCISER
PROTECTION TABLE

011033	000				
011034	045	116	045	P.AFN:	.ASCII <00>
011037	101	102	131		.ASCII /#N#/
011042	124	105	040		.ASCII /ABY/
011045	103	117	125		.ASCII /TE /
011050	116	124	040		.ASCII /COU/
011053	111	116	040		.ASCII /NT /
011056	127	122	111		.ASCII /IN /
011061	124	105	040		.ASCII /WRI/
011064	103	117	115		.ASCII /TE /
011067	115	101	116		.ASCII /COM/
011072	104	072	040		.ASCII /MAN/
011075	045	104	065		.ASCII /D: /
011100	045	101	056		.ASCII /#D5/
011103	000				.ASCII /#A. /
011104	045	116	045	P.AFO:	.ASCII <00>
011107	101	040	052		.ASCII /#N#/
011112	040	104	111		.ASCII /A #/
011115	123	113	040		.ASCII / DI/
011120	072	040	045		.ASCII /SK /
011123	104	062	000		.ASCII /: #/
011126	045	116	045	P.AFP:	.ASCII /D2/<00>
011131	101	123	101		.ASCII /#N#/
011134	072	040	045		.ASCII /ASA/
011137	117	066	000		.ASCII /: #/
011142	045	116	045	P.AFQ:	.ASCII /06/<00>
011145	101	123	124		.ASCII /#N#/
011150	101	124	125		.ASCII /AST/
011153	123	040	103		.ASCII /ATU/
011156	117	104	105		.ASCII /S C/
011161	072	040	000		.ASCII /ODE/
011164	045	116	045	P.AFR:	.ASCII /: /<00>
011167	101	123	124		.ASCII /#N#/
011172	101	124	125		.ASCII /AST/
011175	123	040	123		.ASCII /ATU/
011200	125	102	055		.ASCII /S S/
011203	103	117	104		.ASCII /UB -/
011206	105	072	040		.ASCII /COD/
011211	000				.ASCII /E: /
011212	045	116	045	P.AFS:	.ASCII <00>
011215	101	103	117		.ASCII /#N#/
011220	115	115	101		.ASCII /ACO/
011223	116	104	072		.ASCII /MMA/
011226	040	000			.ASCII /ND: /
011230	045	101	055	P.AFT:	.ASCII / /<00>
011233	104	125	120		.ASCII /#A -/
011236	055	000			.ASCII /DUP/
011240	045	101	055	P.AFU:	.ASCII /- /<00>
011243	115	123	103		.ASCII /#A -/
011246	120	055	000		.ASCII /MSC/
011251	000				.ASCII /P - /<00>
011252	045	101	055	P.AFV:	.ASCII <00>
011255	103	117	115		.ASCII /#A -/
					.ASCII /COM/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

011260	120	101	122	.ASCII	/PAR/
011263	105	000	000	.ASCII	/E/<00><00>
011266	045	116	045	P.AFW:	.ASCII /#N#/
011271	101	102	101	.ASCII	/ABA/
011274	104	040	102	.ASCII	/D B/
011277	114	117	103	.ASCII	/LOC/
011302	113	040	122	.ASCII	/K R/
011305	105	120	117	.ASCII	/EPO/
011310	122	124	105	.ASCII	/RTE/
011313	104	072	040	.ASCII	/D: /
011316	045	117	066	.ASCII	/#06/
011321	045	101	040	.ASCII	/#A /
011324	045	117	066	.ASCII	/#06/
011327	045	101	056	.ASCII	/#A /
011332	000	000		.ASCII	<00><00>
011334	045	116	045	P.AFX:	.ASCII /#N#/
011337	101	114	102	.ASCII	/ALB/
011342	116	072	040	.ASCII	/N: /
011345	045	104	065	.ASCII	/#D5/
011350	045	101	056	.ASCII	/#A /
011353	040	050	117	.ASCII	/ (O/
011356	103	124	040	.ASCII	/CT /
011361	045	117	066	.ASCII	/#06/
011364	045	101	040	.ASCII	/#A /
011367	045	117	066	.ASCII	/#06/
011372	045	101	051	.ASCII	/#A)/
011375	000			.ASCII	<00>
011376	045	116	045	P.AFY:	.ASCII /#N#/
011401	101	104	102	.ASCII	/ADB/
011404	116	072	040	.ASCII	/N: /
011407	045	104	065	.ASCII	/#D5/
011412	045	101	056	.ASCII	/#A /
011415	040	050	117	.ASCII	/ (O/
011420	103	124	040	.ASCII	/CT /
011423	045	117	066	.ASCII	/#06/
011426	045	101	051	.ASCII	/#A)/
011431	000			.ASCII	<00>
011432	045	116	045	P.AFZ:	.ASCII /#N#/
011435	101	102	131	.ASCII	/ABY/
011440	124	105	040	.ASCII	/TE /
011443	103	117	125	.ASCII	/COU/
011446	116	124	040	.ASCII	/NT /
011451	111	116	040	.ASCII	/IN /
011454	103	117	115	.ASCII	/COM/
011457	115	101	116	.ASCII	/MAN/
011462	104	072	040	.ASCII	/D: /
011465	045	104	065	.ASCII	/#D5/
011470	000	000		.ASCII	<00><00>
011472	045	116	045	P.AGA:	.ASCII /#N#/
011475	101	101	103	.ASCII	/AAC/
011500	124	125	101	.ASCII	/TUA/
011503	114	040	043	.ASCII	/L #/
011506	040	117	106	.ASCII	/ OF/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

011511	040	102	131	.ASCII	/BY/
011514	124	105	123	.ASCII	/TES/
011517	040	124	122	.ASCII	/TR/
011522	101	116	123	.ASCII	/ANS/
011525	106	105	122	.ASCII	/FER/
011530	122	105	104	.ASCII	/RED/
011533	072	040	045	.ASCII	/: #/
011536	104	065	000	.ASCII	/D5/<00>
011541	000			.ASCII	<00>
011542	045	116	045	P.AGB: .ASCII	/#N#/
011545	101	111	057	.ASCII	/AI/<57>
011550	117	040	102	.ASCII	/O B/
011553	125	106	106	.ASCII	/UFF/
011556	105	122	040	.ASCII	/ER /
011561	101	104	104	.ASCII	/ADD/
011564	122	105	123	.ASCII	/RES/
011567	123	040	050	.ASCII	/S (/
011572	063	062	040	.ASCII	/32 /
011575	102	111	124	.ASCII	/BIT/
011600	123	051	072	.ASCII	/S):/
011603	040	045	117	.ASCII	/ #0/
011606	066	045	101	.ASCII	/6#A/
011611	040	045	117	.ASCII	/ #0/
011614	066	000		.ASCII	/6/<00>
011616	045	116	045	P.AGC: .ASCII	/#N#/
011621	101	103	117	.ASCII	/ACO/
011624	116	124	105	.ASCII	/NTE/
011627	116	124	123	.ASCII	/NTS/
011632	040	117	106	.ASCII	/ OF/
011635	040	122	105	.ASCII	/ RE/
011640	124	125	122	.ASCII	/TUR/
011643	116	040	120	.ASCII	/N P/
011646	101	103	113	.ASCII	/ACK/
011651	105	124	072	.ASCII	/ET:/
011654	045	116	000	.ASCII	/#N/<00>
011657	000			.ASCII	<00>
011660	045	116	045	P.AGD: .ASCII	/#N#/
011663	101	115	105	.ASCII	/AME/
011666	123	123	101	.ASCII	/SSA/
011671	107	105	040	.ASCII	/GE /
011674	124	131	120	.ASCII	/TYP/
011677	105	072	040	.ASCII	/E: /
011702	000	000		.ASCII	<00><00>
011704	045	116	045	P.AGE: .ASCII	/#N#/
011707	101	115	105	.ASCII	/AME/
011712	123	123	101	.ASCII	/SSA/
011715	107	105	040	.ASCII	/GE /
011720	116	125	115	.ASCII	/NUM/
011723	102	105	122	.ASCII	/BER/
011726	123	072	040	.ASCII	/S: /
011731	000			.ASCII	<00>
011732	045	116	045	P.AGF: .ASCII	/#N#/
011735	101	115	105	.ASCII	/AME/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

011740	123	123	101	.ASCII	/SSA/
011743	107	105	040	.ASCII	/GE /
011746	105	122	122	.ASCII	/ERR/
011751	040	103	117	.ASCII	/ CO/
011754	104	105	123	.ASCII	/DES/
011757	072	040	000	.ASCII	/: /<00>
011762	045	116	045	P.AGG:	.ASCII /#N#/
011765	101	102	131	.ASCII	/ABY/
011770	124	105	040	.ASCII	/TE /
011773	116	125	115	.ASCII	/NUM/
011776	102	105	122	.ASCII	/BER/
012001	072	040	045	.ASCII	/: #/
012004	104	063	000	.ASCII	/D3/<00>
012007	000			.ASCII	<00>
012010	045	116	045	P.AGH:	.ASCII /#N#/
012013	101	122	101	.ASCII	/ARA/
012016	116	104	117	.ASCII	/NDO/
012021	115	040	127	.ASCII	/M W/
012024	122	111	124	.ASCII	/RIT/
012027	124	105	116	.ASCII	/TEN/
012032	040	127	117	.ASCII	/ WO/
012035	122	104	040	.ASCII	/RD /
012040	072	045	102	.ASCII	/:#B/
012043	061	066	000	.ASCII	/16/<00>
012046	045	116	045	P.AGI:	.ASCII /#N#/
012051	101	122	101	.ASCII	/ARA/
012054	116	104	117	.ASCII	/NDO/
012057	115	040	122	.ASCII	/M R/
012062	105	101	104	.ASCII	/EAD/
012065	040	127	117	.ASCII	/ WO/
012070	122	104	040	.ASCII	/RD /
012073	142	151	156	.ASCII	/bin/
012076	072	045	102	.ASCII	/:#B/
012101	061	066	045	.ASCII	/16#/
012104	101	040	157	.ASCII	/A o/
012107	143	164	072	.ASCII	/ct:/
012112	045	117	066	.ASCII	/#06/
012115	000			.ASCII	<00>
012116	045	116	045	P.AGJ:	.ASCII /#N#/
012121	101	103	122	.ASCII	/ACR/
012124	116	040	072	.ASCII	/N :/
012127	040	045	117	.ASCII	/ #0/
012132	066	045	101	.ASCII	/6#A/
012135	040	040	045	.ASCII	/ #/
012140	117	066	000	.ASCII	/06/<00>
012143	000			.ASCII	<00>
012144	045	101	040	P.AGK:	.ASCII /#A /
012147	055	040	125	.ASCII	/- U/
012152	116	113	116	.ASCII	/NKN/
012155	117	127	116	.ASCII	/OWN/
012160	040	072	040	.ASCII	/ : /
012163	045	104	062	.ASCII	/#D2/
012166	000	000		.ASCII	<00><00>

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

012170	045	101	040	P.AGL:	.ASCII	/#A /
012173	055	040	125		.ASCII	/- U/
012176	116	113	116		.ASCII	/NKN/
012201	117	127	116		.ASCII	/OWN/
012204	040	103	117		.ASCII	/ CO/
012207	116	116	105		.ASCII	/NNE/
012212	103	124	111		.ASCII	/CTI/
012215	117	116	040		.ASCII	/ON /
012220	111	104	072		.ASCII	/ID:/
012223	040	045	104		.ASCII	/ #D/
012226	063	045	101		.ASCII	/3#A/
012231	040	055	000		.ASCII	/ -/<00>
012234	045	116	045	P.AGM:	.ASCII	/#N#/
012237	101	103	124		.ASCII	/ACT/
012242	114	122	040		.ASCII	/LR /
012245	106	114	101		.ASCII	/FLA/
012250	107	123	072		.ASCII	/GS:/
012253	000				.ASCII	<00>
012254	045	116	045	P.AGN:	.ASCII	/#N#/
012257	101	125	116		.ASCII	/AUN/
012262	111	124	040		.ASCII	/IT /
012265	106	114	101		.ASCII	/FLA/
012270	107	123	072		.ASCII	/GS:/
012273	000				.ASCII	<00>
012274	045	116	045	P.AGO:	.ASCII	/#N#/
012277	101	105	116		.ASCII	/AEN/
012302	104	040	115		.ASCII	/D M/
012305	105	123	123		.ASCII	/ESS/
012310	101	107	105		.ASCII	/AGE/
012313	040	106	114		.ASCII	/ FL/
012316	101	107	123		.ASCII	/AGS/
012321	072	000	000		.ASCII	/:/<00><00>
012324	045	116	045	P.AGP:	.ASCII	/#N#/
012327	101	103	117		.ASCII	/ACO/
012332	116	124	105		.ASCII	/NTE/
012335	116	124	123		.ASCII	/NTS/
012340	040	117	106		.ASCII	/ OF/
012343	040	104	125		.ASCII	/ DU/
012346	120	040	111		.ASCII	/P I/
012351	057	117	040		.ASCII	<57>/O /
012354	120	101	103		.ASCII	/PAC/
012357	113	105	124		.ASCII	/KET/
012362	072	045	116		.ASCII	/:#N/
012365	000				.ASCII	<00>
012366	045	116	045	P.AGQ:	.ASCII	/#N#/
012371	101	040	104		.ASCII	/A D/
012374	122	111	126		.ASCII	/RIV/
012377	105	122	040		.ASCII	/ER /
012402	103	124	114		.ASCII	/CTL/
012405	122	040	124		.ASCII	/R T/
012410	101	102	114		.ASCII	/ABL/
012413	105	040	075		.ASCII	/E =/
012416	040	101	104		.ASCII	/ AD/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

012421	104	122	072	.ASCII	/DR:/
012424	040	045	104	.ASCII	/ #D/
012427	066	045	116	.ASCII	/6#N/
012432	000	000		.ASCII	<00><00>
012434	045	116	045	P.AGR:	.ASCII /#N#/
012437	101	040	103	.ASCII	/A C/
012442	115	104	040	.ASCII	/MD /
012445	111	116	124	.ASCII	/INT/
012450	054	040	122	.ASCII	/, R/
012453	123	120	040	.ASCII	/SP /
012456	111	116	124	.ASCII	/INT/
012461	054	040	103	.ASCII	/, C/
012464	117	115	115	.ASCII	/OMM/
012467	101	116	104	.ASCII	/AND/
012472	040	122	111	.ASCII	/ RI/
012475	116	107	040	.ASCII	/NG /
012500	075	040	101	.ASCII	/= A/
012503	104	104	122	.ASCII	/DDR/
012506	072	040	045	.ASCII	/: #/
012511	104	066	045	.ASCII	/D6#/
012514	116	000		P.AGS:	.ASCII /N/<00>
012516	045	116	045	.ASCII	/#N#/
012521	101	040	101	.ASCII	/A A/
012524	114	114	040	.ASCII	/LL /
012527	120	101	103	.ASCII	/PAC/
012532	113	105	124	.ASCII	/KET/
012535	123	040	111	.ASCII	/S I/
012540	116	040	115	.ASCII	/N M/
012543	105	123	123	.ASCII	/ESS/
012546	101	107	105	.ASCII	/AGE/
012551	040	101	122	.ASCII	/ AR/
012554	105	101	000	.ASCII	/EA/<00>
012557	000			.ASCII	<00>
012560	045	116	045	P.AGT:	.ASCII /#N#/
012563	101	040	101	.ASCII	/A A/
012566	114	114	040	.ASCII	/LL /
012571	122	105	124	.ASCII	/RET/
012574	125	122	116	.ASCII	/URN/
012577	040	120	101	.ASCII	/ PA/
012602	103	113	105	.ASCII	/CKE/
012605	124	123	040	.ASCII	/TS /
012610	111	116	040	.ASCII	/IN /
012613	101	122	105	.ASCII	/ARE/
012616	101	000		.ASCII	/A/<00>
012620	045	116	045	P.AGU:	.ASCII /#N#/
012623	101	040	101	.ASCII	/A A/
012626	104	104	122	.ASCII	/DDR/
012631	072	040	045	.ASCII	/: #/
012634	104	066	045	.ASCII	/D6#/
012637	101	040	040	.ASCII	/A /
012642	040	040	040	.ASCII	/ /
012645	120	101	103	.ASCII	/PAC/
012650	113	105	124	.ASCII	/KET/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

012653	040	075	040	.ASCII	/ = /
012656	045	116	000	.ASCII	/#N/<00>
012661	000			.ASCII	<00>
012662	045	116	045	P.AGV:	.ASCII /#N# /
012665	101	040	101	.ASCII	/A A /
012670	104	104	122	.ASCII	/DDR /
012673	040	117	106	.ASCII	/ OF /
012676	040	122	105	.ASCII	/ RE /
012701	123	120	117	.ASCII	/SPO /
012704	116	123	105	.ASCII	/NSE /
012707	040	122	111	.ASCII	/ RI /
012712	116	107	040	.ASCII	/NG /
012715	124	117	040	.ASCII	/TO /
012720	102	105	040	.ASCII	/BE /
012723	120	117	114	.ASCII	/POL /
012726	114	105	104	.ASCII	/LED /
012731	040	045	104	.ASCII	/ #D /
012734	066	000		.ASCII	/6/<00>
012736	045	116	045	P.AGW:	.ASCII /#N# /
012741	101	040	101	.ASCII	/A A /
012744	104	104	122	.ASCII	/DDR /
012747	040	117	106	.ASCII	/ OF /
012752	040	115	105	.ASCII	/ ME /
012755	123	123	101	.ASCII	/SSA /
012760	107	105	040	.ASCII	/GE /
012763	120	101	103	.ASCII	/PAC /
012766	113	105	124	.ASCII	/KET /
012771	040	122	105	.ASCII	/ RE /
012774	123	120	117	.ASCII	/SPO /
012777	116	123	105	.ASCII	/NSE /
013002	040	122	111	.ASCII	/ RI /
013005	116	107	040	.ASCII	/NG /
013010	123	114	117	.ASCII	/SLO /
013013	124	040	120	.ASCII	/T P /
013016	117	111	116	.ASCII	/OIN /
013021	124	123	040	.ASCII	/TS /
013024	124	117	040	.ASCII	/TO /
013027	045	104	066	.ASCII	/#D6 /
013032	000	000		.ASCII	<00><00>
013034	045	116	045	P.AGX:	.ASCII /#N# /
013037	101	040	101	.ASCII	/A A /
013042	104	104	122	.ASCII	/DDR /
013045	040	117	106	.ASCII	/ OF /
013050	040	115	105	.ASCII	/ ME /
013053	123	123	101	.ASCII	/SSA /
013056	107	105	040	.ASCII	/GE /
013061	120	101	103	.ASCII	/PAC /
013064	113	105	124	.ASCII	/KET /
013067	040	103	117	.ASCII	/ CO /
013072	115	115	101	.ASCII	/MMA /
013075	116	104	040	.ASCII	/ND /
013100	122	111	116	.ASCII	/RIN /
013103	107	040	123	.ASCII	/G S /

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

013106	114	117	124	.ASCII	/LOT/	
013111	040	120	117	.ASCII	/PO/	
013114	111	116	124	.ASCII	/INT/	
013117	123	040	124	.ASCII	/S T/	
013122	117	040	045	.ASCII	/O #/	
013125	104	066	000	.ASCII	/D6/<00>	
013130	045	116	045	P.AGY:	.ASCII	/#N#/
013133	101	040	103	.ASCII	/A C/	
013136	157	156	164	.ASCII	/ont/	
013141	145	156	164	.ASCII	/ent/	
013144	163	040	157	.ASCII	/s o/	
013147	146	040	123	.ASCII	/f S/	
013152	101	040	122	.ASCII	/A R/	
013155	145	147	151	.ASCII	/egi/	
013160	163	164	145	.ASCII	/ste/	
013163	162	000	000	.ASCII	/r/<00><00>	
013166	045	116	045	P.AGZ:	.ASCII	/#N#/
013171	101	104	125	.ASCII	/ADU/	
013174	120	114	111	.ASCII	/PLI/	
013177	103	101	124	.ASCII	/CAT/	
013202	105	040	125	.ASCII	/E U/	
013205	116	111	124	.ASCII	/NIT/	
013210	072	045	104	.ASCII	/: #D/	
013213	062	045	101	.ASCII	/2#A/	
013216	040	101	124	.ASCII	/ AT/	
013221	040	111	120	.ASCII	/ IP/	
013224	072	040	045	.ASCII	/: #/	
013227	117	066	000	.ASCII	/O6/<00>	
013232	045	116	045	P.AHA:	.ASCII	/#N#/
013235	101	115	117	.ASCII	/AMO/	
013240	122	105	040	.ASCII	/RE /	
013243	124	110	101	.ASCII	/THA/	
013246	116	040	045	.ASCII	/N #/	
013251	104	061	045	.ASCII	/D1#/	
013254	101	040	104	.ASCII	/A D/	
013257	111	106	106	.ASCII	/IFF/	
013262	105	122	105	.ASCII	/ERE/	
013265	116	124	040	.ASCII	/NT /	
013270	111	120	040	.ASCII	/IP /	
013273	101	104	104	.ASCII	/ADD/	
013276	122	105	123	.ASCII	/RES/	
013301	123	105	123	.ASCII	/SES/	
013304	000	000		.ASCII	<00><00>	
013306	045	101	055	P.AHB:	.ASCII	/#A-/
013311	040	123	105	.ASCII	/ SE/	
013314	121	125	105	.ASCII	/QUE/	
013317	116	124	111	.ASCII	/NTI/	
013322	101	114	000	.ASCII	/AL/<00>	
013325	000			.ASCII	<00>	
013326	045	101	055	P.AHC:	.ASCII	/#A-/
013331	040	103	122	.ASCII	/ CR/	
013334	105	104	111	.ASCII	/EDI/	
013337	124	040	116	.ASCII	/T N/	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

013342	117	124	111	.ASCII	/OTI/	
013345	106	111	103	.ASCII	/FIC/	
013350	101	124	111	.ASCII	/ATI/	
013353	117	116	000	.ASCII	/ON/<00>	
013356	045	101	055	P.AHD:	.ASCII	/WA-/
013361	040	115	101	.ASCII	/MA/	
013364	111	116	124	.ASCII	/INT/	
013367	105	116	101	.ASCII	/ENA/	
013372	116	103	105	.ASCII	/NCE/	
013375	000			.ASCII	<00>	
013376	045	101	055	P.AHE:	.ASCII	/WA-/
013401	040	104	101	.ASCII	/DA/	
013404	124	101	107	.ASCII	/TAG/	
013407	122	101	115	.ASCII	/RAM/	
013412	000	000		.ASCII	<00><00>	
013414	045	101	122	P.AHF:	.ASCII	/WAR/
013417	105	101	104	.ASCII	/EAD/	
013422	000	000		.ASCII	<00><00>	
013424	045	101	127	P.AHG:	.ASCII	/WAW/
013427	122	111	124	.ASCII	/RIT/	
013432	105	000		.ASCII	/E/<00>	
013434	045	101	101	P.AHH:	.ASCII	/WAA/
013437	103	103	105	.ASCII	/CCE/	
013442	123	123	000	.ASCII	/SS/<00>	
013445	000			.ASCII	<00>	
013446	045	101	117	P.AHI:	.ASCII	/WAO/
013451	116	040	114	.ASCII	/N L/	
013454	111	116	105	.ASCII	/INE/	
013457	000			.ASCII	<00>	
013460	045	101	123	P.AHJ:	.ASCII	/WAS/
013463	105	124	040	.ASCII	/ET /	
013466	103	117	116	.ASCII	/CON/	
013471	124	122	117	.ASCII	/TRO/	
013474	114	040	103	.ASCII	/L C/	
013477	110	101	122	.ASCII	/HAR/	
013502	056	000		.ASCII	./<00>	
013504	045	101	107	P.AHK:	.ASCII	/WAG/
013507	105	124	040	.ASCII	/ET /	
013512	104	125	123	.ASCII	/DUS/	
013515	124	040	123	.ASCII	/T S/	
013520	124	101	124	.ASCII	/TAT/	
013523	125	123	000	.ASCII	/US/<00>	
013526	045	101	105	P.AHL:	.ASCII	/WAE/
013531	130	105	103	.ASCII	/XEC/	
013534	125	124	105	.ASCII	/UTE/	
013537	040	123	125	.ASCII	/SU/	
013542	120	120	114	.ASCII	/PPL/	
013545	111	105	104	.ASCII	/IED/	
013550	040	120	122	.ASCII	/PR/	
013553	107	000	000	.ASCII	/G/<00><00>	
013556	045	101	105	P.AHM:	.ASCII	/WAE/
013561	130	105	103	.ASCII	/XEC/	
013564	125	124	105	.ASCII	/UTE/	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0089
Page 89
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

013567	040	114	117	.ASCII	/ LO/
013572	103	101	114	.ASCII	/CAL/
013575	040	120	122	.ASCII	/ PR/
013600	107	000		.ASCII	/G/<00>
013602	045	101	123	P.AHN:	.ASCII /#AS/
013605	105	116	104		.ASCII /END/
013610	040	104	101		.ASCII / DA/
013613	124	101	000		.ASCII /TA/<00>
013616	045	101	122	P.AHO:	.ASCII /#AR/
013621	105	103	105		.ASCII /ECE/
013624	111	126	105		.ASCII /IVE/
013627	040	104	101		.ASCII / DA/
013632	124	101	000		.ASCII /TA/<00>
013635	000				.ASCII <00>
013636	045	101	101	P.AHP:	.ASCII /#AA/
013641	102	117	122		.ASCII /BOR/
013644	124	000			.ASCII /T/<00>
013646	045	101	123	P.AHQ:	.ASCII /#AS/
013651	120	111	116		.ASCII /PIN/
013654	055	104	117		.ASCII /-DO/
013657	127	116	040		.ASCII /MN /
013662	111	107	116		.ASCII /IGN/
013665	117	122	105		.ASCII /ORE/
013670	104	000			.ASCII /D/<00>
013672	045	101	123	P.AHR:	.ASCII /#AS/
013675	124	111	114		.ASCII /TIL/
013700	114	040	103		.ASCII /L C/
013703	117	116	116		.ASCII /ONN/
013706	105	103	124		.ASCII /ECT/
013711	105	104	000		.ASCII /ED/<00>
013714	045	101	104	P.AHS:	.ASCII /#AD/
013717	125	120	114		.ASCII /UPL/
013722	111	103	101		.ASCII /ICA/
013725	124	105	040		.ASCII /TE /
013730	125	116	111		.ASCII /UNI/
013733	124	040	116		.ASCII /T N/
013736	125	115	102		.ASCII /UMB/
013741	105	122	000		.ASCII /ER/<00>
013744	045	101	101	P.AHT:	.ASCII /#AA/
013747	114	122	105		.ASCII /LRE/
013752	101	104	131		.ASCII /ADY/
013755	040	117	116		.ASCII / ON/
013760	114	111	116		.ASCII /LIN/
013763	105	000	000		.ASCII /E/<00><00>
013766	045	101	123	P.AHU:	.ASCII /#AS/
013771	124	111	114		.ASCII /TIL/
013774	114	040	117		.ASCII /L O/
013777	116	114	111		.ASCII /NLI/
014002	116	105	000		.ASCII /NE/<00>
014005	000				.ASCII <00>
014006	045	101	125	P.AHV:	.ASCII /#AU/
014011	116	111	124		.ASCII /NIT/
014014	040	125	116		.ASCII / UN/

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555

SEQ 0090
Page 90
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

014017	113	116	117	.ASCII	/KNO/
014022	127	116	040	.ASCII	/WN /
014025	117	122	040	.ASCII	/OR /
014030	117	116	114	.ASCII	/ONL/
014033	111	116	105	.ASCII	/INE/
014036	040	124	117	.ASCII	/ TO/
014041	040	101	116	.ASCII	/ AN/
014044	117	124	110	.ASCII	/OTH/
014047	105	122	040	.ASCII	/ER /
014052	103	124	114	.ASCII	/CTL/
014055	122	000	000	.ASCII	/R/<00><00>
014060	045	101	116	P.AHW: .ASCII	/AN/
014063	117	040	126	.ASCII	/O V/
014066	117	114	125	.ASCII	/OLU/
014071	115	105	040	.ASCII	/ME /
014074	115	117	125	.ASCII	/MOU/
014077	116	124	105	.ASCII	/NTE/
014102	104	040	117	.ASCII	/D O/
014105	122	040	104	.ASCII	/R D/
014110	122	111	126	.ASCII	/RIV/
014113	105	040	104	.ASCII	/E D/
014116	111	123	101	.ASCII	/ISA/
014121	102	114	105	.ASCII	/BLE/
014124	104	040	102	.ASCII	/D B/
014127	131	040	123	.ASCII	/Y S/
014132	127	111	124	.ASCII	/WIT/
014135	103	110	000	.ASCII	/CH/<00>
014140	045	101	125	P.AHX: .ASCII	/AU/
014143	116	111	124	.ASCII	/NIT/
014146	040	111	116	.ASCII	/ IN/
014151	117	120	105	.ASCII	/OPE/
014154	122	101	124	.ASCII	/RAT/
014157	111	126	105	.ASCII	/IVE/
014162	040	050	122	.ASCII	/ (R/
014165	104	065	061	.ASCII	/D51/
014170	040	127	122	.ASCII	/ WR/
014173	111	124	105	.ASCII	/ITE/
014176	040	106	101	.ASCII	/ FA/
014201	125	114	124	.ASCII	/ULT/
014204	000	000		.ASCII	<00><00>
014206	045	101	125	P.AHY: .ASCII	/AU/
014211	116	111	124	.ASCII	/NIT/
014214	040	104	111	.ASCII	/ DI/
014217	123	101	102	.ASCII	/SAB/
014222	114	105	104	.ASCII	/LED/
014225	040	102	131	.ASCII	/ BY/
014230	040	106	111	.ASCII	/ FI/
014233	105	114	104	.ASCII	/ELD/
014236	040	123	105	.ASCII	/ SE/
014241	122	126	111	.ASCII	/RVI/
014244	103	105	040	.ASCII	/CE /
014247	117	122	040	.ASCII	/OR /
014252	111	116	124	.ASCII	/INT/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

014255	105	122	116	.ASCII	/ERN/	
014260	101	114	040	.ASCII	/AL /	
014263	104	111	101	.ASCII	/DIA/	
014266	107	116	117	.ASCII	/GNO/	
014271	123	124	111	.ASCII	/STI/	
014274	103	123	000	.ASCII	/CS/<00>	
014277	000			.ASCII	<00>	
014300	045	101	042	P.AHZ:	.ASCII	/#A"/
014303	106	117	122	.ASCII	/FOR/	
014306	103	105	104	.ASCII	/CED/	
014311	040	105	122	.ASCII	/ ER/	
014314	122	042	040	.ASCII	/R" /	
014317	104	105	124	.ASCII	/DET/	
014322	105	103	124	.ASCII	/ECT/	
014325	105	104	040	.ASCII	/ED /	
014330	127	110	111	.ASCII	/WHI/	
014333	114	105	040	.ASCII	/LE /	
014336	101	103	103	.ASCII	/ACC/	
014341	105	123	123	.ASCII	/ESS/	
014344	111	116	107	.ASCII	/ING/	
014347	040	106	103	.ASCII	/ FC/	
014352	124	040	117	.ASCII	/T O/	
014355	122	040	122	.ASCII	/R R/	
014360	103	124	000	.ASCII	/CT/<00>	
014363	000			.ASCII	<00>	
014364	045	101	123	P.AIA:	.ASCII	/#AS/
014367	105	103	124	.ASCII	/ECT/	
014372	117	122	040	.ASCII	/OR /	
014375	127	122	111	.ASCII	/WRI/	
014400	124	124	105	.ASCII	/TTE/	
014403	116	040	127	.ASCII	/N W/	
014406	111	124	110	.ASCII	/ITH/	
014411	040	042	106	.ASCII	/ "F/	
014414	117	122	103	.ASCII	/ORC/	
014417	105	104	040	.ASCII	/ED /	
014422	105	122	122	.ASCII	/ERR/	
014425	042	040	115	.ASCII	/ " M/	
014430	117	104	111	.ASCII	/ODI/	
014433	106	111	105	.ASCII	/FIE/	
014436	122	000		P.AIB:	.ASCII	/R/<00>
014440	045	101	106	.ASCII	/#AF/	
014443	103	124	040	.ASCII	/CT /	
014446	117	122	040	.ASCII	/OR /	
014451	122	103	124	.ASCII	/RCT/	
014454	040	125	116	.ASCII	/ UN/	
014457	122	105	101	.ASCII	/REA/	
014462	104	101	102	.ASCII	/DAB/	
014465	114	105	040	.ASCII	/LE /	
014470	055	040	111	.ASCII	/- I/	
014473	116	126	101	.ASCII	/NVA/	
014476	114	111	104	.ASCII	/LID/	
014501	040	123	105	.ASCII	/ SE/	
014504	103	124	117	.ASCII	/CTO/	

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

014507	122	040	110	.ASCII	/R H/
014512	105	101	104	.ASCII	/EAD/
014515	105	122	000	.ASCII	/ER/<00>
014520	045	101	110	P.AIL:	.ASCII /#AH/
014523	105	101	104	.ASCII	/EAD/
014526	105	122	040	.ASCII	/ER /
014531	103	117	115	.ASCII	/COM/
014534	120	101	122	.ASCII	/PAR/
014537	105	040	105	.ASCII	/E E/
014542	122	122	040	.ASCII	/RR /
014545	050	126	101	.ASCII	/(VA/
014550	114	111	104	.ASCII	/LID/
014553	040	110	105	.ASCII	/ HE/
014556	101	104	105	.ASCII	/ADE/
014561	122	040	116	.ASCII	/R N/
014564	117	124	040	.ASCII	/OT /
014567	106	117	125	.ASCII	/FOU/
014572	116	104	051	.ASCII	/ND)/
014575	000			.ASCII	<00>
014576	045	101	106	P.AID:	.ASCII /#AF/
014601	103	124	040	.ASCII	/CT /
014604	117	122	040	.ASCII	/OR /
014607	122	103	124	.ASCII	/RCT/
014612	040	125	116	.ASCII	/ UN/
014615	122	105	101	.ASCII	/REA/
014620	104	101	102	.ASCII	/DAB/
014623	114	105	040	.ASCII	/LE /
014626	055	040	104	.ASCII	/- D/
014631	101	124	101	.ASCII	/ATA/
014634	040	123	131	.ASCII	/ SY/
014637	116	103	040	.ASCII	/NC /
014642	124	111	115	.ASCII	/TIM/
014645	105	117	125	.ASCII	/EQU/
014650	124	000		.ASCII	/T/<00>
014652	045	101	104	P.AIE:	.ASCII /#AD/
014655	101	124	101	.ASCII	/ATA/
014660	040	123	131	.ASCII	/ SY/
014663	116	103	040	.ASCII	/NC /
014666	116	117	124	.ASCII	/NOT/
014671	040	106	117	.ASCII	/ FO/
014674	125	116	104	.ASCII	/UND/
014677	040	050	104	.ASCII	/ (D/
014702	101	124	101	.ASCII	/ATA/
014705	040	123	131	.ASCII	/ SY/
014710	116	103	040	.ASCII	/NC /
014713	124	111	115	.ASCII	/TIM/
014716	105	117	125	.ASCII	/EQU/
014721	124	051	000	.ASCII	/T)/<00>
014724	045	101	106	P.AIF:	.ASCII /#AF/
014727	103	124	040	.ASCII	/CT /
014732	117	122	040	.ASCII	/OR /
014735	122	103	124	.ASCII	/RCT/
014740	040	125	116	.ASCII	/ UN/

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

014743	122	105	101	.ASCII	/REA/
014746	104	101	102	.ASCII	/DAB/
014751	114	105	040	.ASCII	/LE /
014754	055	040	125	.ASCII	/- U/
014757	116	103	117	.ASCII	/NCO/
014762	122	122	105	.ASCII	/RRE/
014765	103	124	101	.ASCII	/CTA/
014770	102	114	105	.ASCII	/BLE/
014773	040	105	103	.ASCII	/ EC/
014776	103	040	105	.ASCII	/C E/
015001	122	122	000	.ASCII	/RR/<00>
015004	045	101	125	P.AIG:	.ASCII /#AU/
015007	116	103	117	.ASCII	/NCO/
015012	122	122	105	.ASCII	/RRE/
015015	103	124	101	.ASCII	/CTA/
015020	102	114	105	.ASCII	/BLE/
015023	040	105	103	.ASCII	/ EC/
015026	103	040	105	.ASCII	/C E/
015031	122	122	000	.ASCII	/RR/<00>
015034	045	101	122	P.AIH:	.ASCII /#AR/
015037	103	124	040	.ASCII	/CT /
015042	103	117	122	.ASCII	/COR/
015045	122	125	120	.ASCII	/RUP/
015050	124	105	104	.ASCII	/TED/
015053	000			.ASCII	<00>
015054	045	101	116	P.AII:	.ASCII /#AN/
015057	117	040	122	.ASCII	/O R/
015062	105	120	114	.ASCII	/EPL/
015065	101	103	105	.ASCII	/ACE/
015070	115	105	116	.ASCII	/MEN/
015073	124	040	102	.ASCII	/T B/
015076	114	117	103	.ASCII	/LOC/
015101	113	040	101	.ASCII	/K A/
015104	126	101	111	.ASCII	/VAI/
015107	114	101	102	.ASCII	/LAB/
015112	114	105	040	.ASCII	/LE /
015115	050	122	103	.ASCII	/(RC/
015120	124	040	106	.ASCII	/T F/
015123	125	114	114	.ASCII	/ULL/
015126	051	000		.ASCII	/)/<00>
015130	045	101	104	P.AIJ:	.ASCII /#AD/
015133	111	123	113	.ASCII	/ISK/
015136	040	116	117	.ASCII	/ NO/
015141	124	040	106	.ASCII	/T F/
015144	117	122	115	.ASCII	/ORM/
015147	101	124	124	.ASCII	/ATT/
015152	105	104	040	.ASCII	/ED /
015155	127	111	124	.ASCII	/WIT/
015160	110	040	065	.ASCII	/H 5/
015163	061	062	040	.ASCII	/12 /
015166	102	131	124	.ASCII	/BYT/
015171	105	040	123	.ASCII	/E S/
015174	105	103	124	.ASCII	/ECT/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

015177	117	122	123		.ASCII	/ORS/
015202	000	000			.ASCII	<00><00>
015204	045	101	104	P.AIK:	.ASCII	/AD/
015207	111	123	113		.ASCII	/ISK/
015212	040	116	117		.ASCII	/NO/
015215	124	040	106		.ASCII	/T F/
015220	117	122	115		.ASCII	/ORM/
015223	101	124	124		.ASCII	/ATT/
015226	105	104	040		.ASCII	/ED /
015231	117	122	040		.ASCII	/OR /
015234	106	103	124		.ASCII	/FCT/
015237	040	103	117		.ASCII	/CO/
015242	122	122	125		.ASCII	/RRU/
015245	120	124	105		.ASCII	/PTE/
015250	104	000			.ASCII	/D/<00>
015252	045	101	117	P.AIL:	.ASCII	/AO/
015255	116	105	040		.ASCII	/NE /
015260	123	131	115		.ASCII	/SYM/
015263	102	117	114		.ASCII	/BOL/
015266	040	105	103		.ASCII	/EC/
015271	103	040	105		.ASCII	/C E/
015274	122	122	000		.ASCII	/RR/<00>
015277	000				.ASCII	<00>
015300	045	101	124	P.AIM:	.ASCII	/AT/
015303	127	117	040		.ASCII	/WO /
015306	123	131	115		.ASCII	/SYM/
015311	102	117	114		.ASCII	/BOL/
015314	040	105	103		.ASCII	/EC/
015317	103	040	105		.ASCII	/C E/
015322	122	122	000		.ASCII	/RR/<00>
015325	000				.ASCII	<00>
015326	045	101	124	P.AIN:	.ASCII	/AT/
015331	110	122	105		.ASCII	/HRE/
015334	105	040	123		.ASCII	/E S/
015337	131	115	102		.ASCII	/YMB/
015342	117	114	040		.ASCII	/OL /
015345	105	103	103		.ASCII	/ECC/
015350	040	105	122		.ASCII	/ER/
015353	122	000	000		.ASCII	/R/<00><00>
015356	045	101	106	P.AIO:	.ASCII	/AF/
015361	117	125	122		.ASCII	/OUR/
015364	040	123	131		.ASCII	/SY/
015367	115	102	117		.ASCII	/MBO/
015372	114	040	105		.ASCII	/L E/
015375	103	103	040		.ASCII	/CC /
015400	105	122	122		.ASCII	/ERR/
015403	000				.ASCII	<00>
015404	045	101	106	P.AIP:	.ASCII	/AF/
015407	111	126	105		.ASCII	/IVE/
015412	040	123	131		.ASCII	/SY/
015415	115	102	117		.ASCII	/MBO/
015420	114	040	105		.ASCII	/L E/
015423	103	103	040		.ASCII	/CC /

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0095
Page 95
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

015426	105	122	122		.ASCII	/ERR/
015431	000				.ASCII	<00>
015432	045	101	123	P.AIQ:	.ASCII	/AS/
015435	111	130	040		.ASCII	/IX /
015440	123	131	115		.ASCII	/SYM/
015443	102	117	114		.ASCII	/BOL/
015446	040	105	103		.ASCII	/ EC/
015451	103	040	105		.ASCII	/C E/
015454	122	122	000		.ASCII	/RR/<00>
015457	000				.ASCII	<00>
015460	045	101	123	P.AIR:	.ASCII	/AS/
015463	105	126	105		.ASCII	/EVE/
015466	116	040	123		.ASCII	/N S/
015471	131	115	102		.ASCII	/YMB/
015474	117	114	040		.ASCII	/OL /
015477	105	103	103		.ASCII	/ECC/
015502	040	105	122		.ASCII	/ ER/
015505	122	000	000		.ASCII	/R/<00><00>
015510	045	101	105	P.AIS:	.ASCII	/AE/
015513	111	107	110		.ASCII	/IGH/
015516	124	040	123		.ASCII	/T S/
015521	131	115	102		.ASCII	/YMB/
015524	117	114	040		.ASCII	/OL /
015527	105	103	103		.ASCII	/ECC/
015532	040	105	122		.ASCII	/ ER/
015535	122	000	000		.ASCII	/R/<00><00>
015540	045	101	103	P.AIT:	.ASCII	/AC/
015543	117	122	122		.ASCII	/ORR/
015546	105	103	124		.ASCII	/ECT/
015551	101	102	114		.ASCII	/ABL/
015554	105	040	105		.ASCII	/E E/
015557	122	122	040		.ASCII	/RR /
015562	111	116	040		.ASCII	/IN /
015565	105	103	103		.ASCII	/ECC/
015570	040	106	111		.ASCII	/ FI/
015573	105	114	104		.ASCII	/ELD/
015576	000	000			.ASCII	<00><00>
015600	045	101	125	P.AIU:	.ASCII	/AU/
015603	116	111	124		.ASCII	/NIT/
015606	040	123	117		.ASCII	/ SO/
015611	106	124	127		.ASCII	/FTW/
015614	101	122	105		.ASCII	/ARE/
015617	040	127	122		.ASCII	/ WR/
015622	111	124	105		.ASCII	/ITE/
015625	040	120	122		.ASCII	/ PR/
015630	117	124	105		.ASCII	/OTE/
015633	103	124	105		.ASCII	/CTE/
015636	104	000			.ASCII	/D/<00>
015640	045	101	125	P.AIV:	.ASCII	/AU/
015643	116	111	124		.ASCII	/NIT/
015646	040	110	101		.ASCII	/ HA/
015651	122	104	127		.ASCII	/RDW/
015654	101	122	105		.ASCII	/ARE/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

015657	040	127	122	.ASCII	/ WR/	
015662	111	124	105	.ASCII	/ITE/	
015665	040	120	122	.ASCII	/ PR/	
015670	117	124	105	.ASCII	/OTE/	
015673	103	124	105	.ASCII	/CTE/	
015676	104	000		.ASCII	/D/<00>	
015700	045	101	117	P.AIW:	.ASCII	/#AO/
015703	104	104	040		.ASCII	/DD /
015706	124	122	101		.ASCII	/TRA/
015711	116	123	106		.ASCII	/NSF/
015714	105	122	040		.ASCII	/ER /
015717	101	104	104		.ASCII	/ADD/
015722	122	105	123		.ASCII	/RES/
015725	123	000	000	P.AIX:	.ASCII	/S/<00><00>
015730	045	101	117		.ASCII	/#AO/
015733	104	104	040		.ASCII	/DD /
015736	102	131	124		.ASCII	/BYT/
015741	105	040	103		.ASCII	/E C/
015744	117	125	116		.ASCII	/OUN/
015747	124	000	000	P.AIY:	.ASCII	/T/<00><00>
015752	045	101	116		.ASCII	/#AN/
015755	117	116	055		.ASCII	/ON-/
015760	105	130	111		.ASCII	/EXI/
015763	123	124	105		.ASCII	/STE/
015766	116	124	040		.ASCII	/NT /
015771	110	117	123		.ASCII	/HOS/
015774	124	040	115		.ASCII	/T M/
015777	105	115	117		.ASCII	/EMO/
016002	122	131	000		.ASCII	/RY/<00>
016005	000			P.AIZ:	.ASCII	<00>
016006	045	101	110		.ASCII	/#AH/
016011	117	123	124		.ASCII	/OST/
016014	040	115	105		.ASCII	/ ME/
016017	115	117	122		.ASCII	/MOR/
016022	131	040	120		.ASCII	/Y P/
016025	101	122	111		.ASCII	/ARI/
016030	124	131	040		.ASCII	/TY /
016033	105	122	122		.ASCII	/ERR/
016036	000	000		P.AJA:	.ASCII	<00><00>
016040	045	101	103		.ASCII	/#AC/
016043	117	115	115		.ASCII	/OMM/
016046	101	116	104		.ASCII	/AND/
016051	040	124	111		.ASCII	/ TI/
016054	115	117	125		.ASCII	/MOU/
016057	124	040	117		.ASCII	/T O/
016062	122	040	122		.ASCII	/R R/
016065	105	124	122		.ASCII	/ETR/
016070	131	040	114		.ASCII	/Y L/
016073	111	115	111		.ASCII	/IMI/
016076	124	040	105		.ASCII	/T E/
016101	130	103	105		.ASCII	/XCE/
016104	105	104	105		.ASCII	/EDE/
016107	104	000	000		.ASCII	/D/<00><00>

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)SEQ 0097
Page 97

016112	045	101	123	P.AJB:	.ASCII	/AS/
016115	105	122	111		.ASCII	/ERI/
016120	101	114	111		.ASCII	/ALI/
016123	132	105	122		.ASCII	/ZER/
016126	057	104	105		.ASCII	<57>/DE/
016131	123	105	122		.ASCII	/SER/
016134	111	101	114		.ASCII	/IAL/
016137	111	132	105		.ASCII	/IZE/
016142	122	040	117		.ASCII	/R O/
016145	126	105	122		.ASCII	/VER/
016150	122	125	116		.ASCII	/RUN/
016153	040	117	122		.ASCII	/ OR/
016156	040	125	116		.ASCII	/ UN/
016161	104	105	122		.ASCII	/DER/
016164	122	125	116		.ASCII	/RUN/
016167	000				.ASCII	<00>
016170	045	101	105	P.AJC:	.ASCII	/AE/
016173	104	103	040		.ASCII	/DC /
016176	105	122	122		.ASCII	/ERR/
016201	000				.ASCII	<00>
016202	045	101	111	P.AJD:	.ASCII	/AI/
016205	116	103	117		.ASCII	/NCO/
016210	116	123	111		.ASCII	/NSI/
016213	123	124	105		.ASCII	/STE/
016216	116	124	040		.ASCII	/NT /
016221	111	116	124		.ASCII	/INT/
016224	105	122	116		.ASCII	/ERN/
016227	101	114	040		.ASCII	/AL /
016232	104	101	124		.ASCII	/DAT/
016235	101	040	123		.ASCII	/A S/
016240	124	122	125		.ASCII	/TRU/
016243	103	124	125		.ASCII	/CTU/
016246	122	105	000		.ASCII	/RE/<00>
016251	000				.ASCII	<00>
016252	045	101	104	P.AJE:	.ASCII	/AD/
016255	122	111	126		.ASCII	/RIV/
016260	105	040	103		.ASCII	/E C/
016263	117	115	115		.ASCII	/OMM/
016266	101	116	104		.ASCII	/AND/
016271	040	124	111		.ASCII	/ TI/
016274	115	105	117		.ASCII	/MEO/
016277	125	124	040		.ASCII	/UT /
016302	050	116	117		.ASCII	/(NO/
016305	040	122	105		.ASCII	/ RE/
016310	123	120	117		.ASCII	/SPO/
016313	116	123	105		.ASCII	/NSE/
016316	040	117	122		.ASCII	/ OR/
016321	040	123	105		.ASCII	/ SE/
016324	105	113	040		.ASCII	/EK /
016327	111	116	103		.ASCII	/INC/
016332	117	115	120		.ASCII	/OMP/
016335	114	105	124		.ASCII	/LET/
016340	105	051	000		.ASCII	/E)/<00>

016343	000				.ASCII	<00>
016344	045	101	103	P.AJF:	.ASCII	/AC/
016347	124	114	122		.ASCII	/TLR/
016352	040	104	105		.ASCII	/DE/
016355	124	105	103		.ASCII	/TEC/
016360	124	105	104		.ASCII	/TED/
016363	040	124	122		.ASCII	/TR/
016366	101	116	123		.ASCII	/ANS/
016371	115	111	123		.ASCII	/MIS/
016374	123	111	117		.ASCII	/SIO/
016377	116	040	117		.ASCII	/N O/
016402	122	040	120		.ASCII	/R P/
016405	122	117	124		.ASCII	/ROT/
016410	117	103	117		.ASCII	/OCO/
016413	114	040	105		.ASCII	/L E/
016416	122	122	000		.ASCII	/RR/<00>
016421	000				.ASCII	<00>
016422	045	101	120	P.AJG:	.ASCII	/AP/
016425	117	123	111		.ASCII	/OSI/
016430	124	111	117		.ASCII	/TIO/
016433	116	040	105		.ASCII	/N E/
016436	122	122	040		.ASCII	/RR /
016441	050	115	111		.ASCII	/(MI/
016444	123	055	123		.ASCII	/S-S/
016447	105	105	113		.ASCII	/EEK/
016452	051	000			.ASCII	/)/<00>
016454	045	101	114	P.AJH:	.ASCII	/AL/
016457	117	123	124		.ASCII	/OST/
016462	040	122	105		.ASCII	/RE/
016465	101	104	057		.ASCII	/AD/<57>
016470	127	122	111		.ASCII	/WRI/
016473	124	105	040		.ASCII	/TE /
016476	122	105	101		.ASCII	/REA/
016501	104	131	040		.ASCII	/DY /
016504	104	125	122		.ASCII	/DUR/
016507	111	116	107		.ASCII	/ING/
016512	057	102	105		.ASCII	<57>/BE/
016515	124	127	105		.ASCII	/TWE/
016520	105	116	040		.ASCII	/EN /
016523	124	122	101		.ASCII	/TRA/
016526	116	123	106		.ASCII	/NSF/
016531	105	122	123		.ASCII	/ERS/
016534	000	000			.ASCII	<00><00>
016536	045	101	104	P.AJI:	.ASCII	/AD/
016541	122	111	126		.ASCII	/RIV/
016544	105	040	103		.ASCII	/E C/
016547	114	117	103		.ASCII	/LOC/
016552	113	040	104		.ASCII	/K D/
016555	122	117	120		.ASCII	/ROP/
016560	117	125	124		.ASCII	/OUT/
016563	000				.ASCII	<00>
016564	045	101	114	P.AJJ:	.ASCII	/AL/
016567	117	123	124		.ASCII	/OST/

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

016572	040	122	105	.ASCII	/ RE/
016575	103	105	111	.ASCII	/CEI/
016600	126	105	122	.ASCII	/VER/
016603	040	122	105	.ASCII	/ RE/
016606	101	104	131	.ASCII	/ADY/
016611	040	102	105	.ASCII	/ BE/
016614	124	127	105	.ASCII	/TWE/
016617	105	116	040	.ASCII	/EN /
016622	123	105	103	.ASCII	/SEC/
016625	124	117	122	.ASCII	/TOR/
016630	123	000		.ASCII	/S/<00>
016632	045	101	104	P.AJK: .ASCII	/AD/
016635	122	111	126	.ASCII	/RIV/
016640	105	040	104	.ASCII	/E D/
016643	105	124	105	.ASCII	/ETE/
016646	103	124	105	.ASCII	/CTE/
016651	104	040	105	.ASCII	/D E/
016654	122	122	000	.ASCII	/RR/<00>
016657	000			.ASCII	<00>
016660	045	101	103	P.AJL: .ASCII	/AC/
016663	124	114	122	.ASCII	/TLR/
016666	040	104	105	.ASCII	/ DE/
016671	124	105	103	.ASCII	/TEC/
016674	124	105	104	.ASCII	/TED/
016677	040	120	125	.ASCII	/ PU/
016702	114	123	105	.ASCII	/LSE/
016705	040	117	122	.ASCII	/ OR/
016710	040	123	124	.ASCII	/ ST/
016713	101	124	105	.ASCII	/ATE/
016716	040	120	101	.ASCII	/ PA/
016721	122	111	124	.ASCII	/RIT/
016724	131	040	105	.ASCII	/Y E/
016727	122	122	000	.ASCII	/RR/<00>
016732	045	116	045	P.AJM: .ASCII	/N#/
016735	101	040	174	.ASCII	/A /<174>
016740	040	102	141	.ASCII	/ Be/
016743	144	040	102	.ASCII	/d B/
016746	154	157	143	.ASCII	/loc/
016751	153	040	125	.ASCII	/k U/
016754	156	162	145	.ASCII	/nre/
016757	160	157	162	.ASCII	/por/
016762	164	145	144	.ASCII	/ted/
016765	000			.ASCII	<00>
016766	045	116	045	P.AJN: .ASCII	/N#/
016771	101	040	174	.ASCII	/A /<174>
016774	040	105	162	.ASCII	/ Er/
016777	162	040	114	.ASCII	/r L/
017002	157	147	040	.ASCII	/og /
017005	107	145	156	.ASCII	/Gen/
017010	145	162	141	.ASCII	/era/
017013	164	145	144	.ASCII	/ted/
017016	000	000		.ASCII	<00><00>
017020	045	116	045	P.AJO: .ASCII	/N#/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK#USER2:(DIETZ.RDRX)ZRQACO.BL1;82 (35)

017023	101	040	174	.ASCII	/A /<174>
017026	040	123	145	.ASCII	/ Se/
017031	162	151	157	.ASCII	/rio/
017034	165	163	040	.ASCII	/us /
017037	105	170	143	.ASCII	/Exc/
017042	145	160	164	.ASCII	/ept/
017045	151	157	156	.ASCII	/ion/
017050	000	000		.ASCII	<00><00>
017052	045	116	045	P.AJP: .ASCII	/sNs/
017055	101	174	040	.ASCII	/A/<174>/ /
017060	105	156	141	.ASCII	/Ena/
017063	142	154	145	.ASCII	/ble/
017066	040	101	164	.ASCII	/ At/
017071	164	145	156	.ASCII	/ten/
017074	164	151	157	.ASCII	/tio/
017077	156	040	115	.ASCII	/n M/
017102	145	163	163	.ASCII	/ess/
017105	141	147	145	.ASCII	/age/
017110	163	000		.ASCII	/e/<00>
017112	045	116	045	P.AJQ: .ASCII	/sNs/
017115	101	174	040	.ASCII	/A/<174>/ /
017120	105	156	141	.ASCII	/Ena/
017123	142	154	145	.ASCII	/ble/
017126	040	115	151	.ASCII	/ Mi/
017131	163	143	145	.ASCII	/ece/
017134	154	154	141	.ASCII	/lla/
017137	156	145	157	.ASCII	/neo/
017142	165	163	040	.ASCII	/us /
017145	105	162	162	.ASCII	/Err/
017150	040	114	157	.ASCII	/ Lo/
017153	147	040	115	.ASCII	/g M/
017156	145	163	163	.ASCII	/ess/
017161	141	147	145	.ASCII	/age/
017164	163	000		.ASCII	/e/<00>
017166	045	116	045	P.AJR: .ASCII	/sNs/
017171	101	174	040	.ASCII	/A/<174>/ /
017174	105	156	141	.ASCII	/Ena/
017177	142	154	145	.ASCII	/ble/
017202	040	117	164	.ASCII	/ Ot/
017205	150	145	162	.ASCII	/her/
017210	040	110	157	.ASCII	/ Ho/
017213	163	164	163	.ASCII	/sta/
017216	040	105	162	.ASCII	/ Er/
017221	162	040	114	.ASCII	/r L/
017224	157	147	040	.ASCII	/og /
017227	115	145	163	.ASCII	/Mes/
017232	163	141	147	.ASCII	/eag/
017235	145	163	000	.ASCII	/ee/<00>
017240	045	116	045	P.AJS: .ASCII	/sNs/
017243	101	174	040	.ASCII	/A/<174>/ /
017246	105	156	141	.ASCII	/Ena/
017251	142	154	145	.ASCII	/ble/
017254	040	124	150	.ASCII	/ Th/

017257	151	163	040	.ASCII	/is /
017262	110	157	163	.ASCII	/Hos/
017265	164	163	040	.ASCII	/ts /
017270	105	162	162	.ASCII	/Err/
017273	040	114	157	.ASCII	/ Lo/
017276	147	040	115	.ASCII	/g M/
017301	145	163	163	.ASCII	/ess/
017304	141	147	145	.ASCII	/age/
017307	163	000	000	.ASCII	/s/<00><00>
017312	045	116	045	P.AJT: .ASCII	/sNs/
017315	101	174	040	.ASCII	/A/<174>/ /
017320	103	164	154	.ASCII	/Ctl/
017323	162	040	111	.ASCII	/r I/
017326	156	151	164	.ASCII	/nit/
017331	151	141	164	.ASCII	/iat/
017334	145	144	040	.ASCII	/ed /
017337	102	141	144	.ASCII	/Bad/
017342	040	102	154	.ASCII	/ Bl/
017345	157	143	153	.ASCII	/ock/
017350	040	122	160	.ASCII	/ Rp/
017353	154	143	155	.ASCII	/lcm/
017356	156	164	000	.ASCII	/nt/<00>
017361	000			.ASCII	<00>
017362	045	116	045	P.AJU: .ASCII	/sNs/
017365	101	174	040	.ASCII	/A/<174>/ /
017370	123	150	141	.ASCII	/Sha/
017373	144	157	167	.ASCII	/dow/
017376	151	156	147	.ASCII	/ing/
017401	000			.ASCII	<00>
017402	045	116	045	P.AJV: .ASCII	/sNs/
017405	101	174	040	.ASCII	/A/<174>/ /
017410	065	067	066	.ASCII	/576/
017413	040	102	171	.ASCII	/ By/
017416	164	145	040	.ASCII	/te /
017421	123	145	143	.ASCII	/Sec/
017424	164	157	162	.ASCII	/tor/
017427	163	000	000	.ASCII	/s/<00><00>
017432	045	116	045	P.AJW: .ASCII	/sNs/
017435	101	174	040	.ASCII	/A/<174>/ /
017440	103	157	155	.ASCII	/Com/
017443	160	141	162	.ASCII	/par/
017446	145	040	122	.ASCII	/e R/
017451	145	141	144	.ASCII	/ead/
017454	163	000		.ASCII	/s/<00>
017456	045	116	045	P.AJX: .ASCII	/sNs/
017461	101	174	040	.ASCII	/A/<174>/ /
017464	103	157	155	.ASCII	/Com/
017467	160	141	162	.ASCII	/par/
017472	145	040	127	.ASCII	/e W/
017475	162	151	164	.ASCII	/rit/
017500	145	163	000	.ASCII	/es/<00>
017503	000			.ASCII	<00>
017504	045	116	045	P.AJY: .ASCII	/sNs/

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)SEQ 0102
Page 102

017507	101	174	040	.ASCII	/A/<174>/ /
017512	103	164	154	.ASCII	/Ct1/
017515	162	040	111	.ASCII	/r I/
017520	156	151	164	.ASCII	/nit/
017523	151	141	164	.ASCII	/iat/
017526	145	144	040	.ASCII	/ed /
017531	102	141	144	.ASCII	/Bad/
017534	040	102	154	.ASCII	/ B1/
017537	157	143	153	.ASCII	/ock/
017542	040	122	160	.ASCII	/ Rp/
017545	154	143	155	.ASCII	/lcm/
017550	156	164	000	.ASCII	/nt/<00>
017553	000			.ASCII	<00>
017554	045	116	045	P.AJZ: .ASCII	/sNs/
017557	101	174	040	.ASCII	/A/<174>/ /
017562	111	156	141	.ASCII	/Ina/
017565	143	164	151	.ASCII	/cti/
017570	166	145	040	.ASCII	/ve /
017573	123	150	141	.ASCII	/Sha/
017576	144	157	167	.ASCII	/dow/
017601	040	123	145	.ASCII	/ Se/
017604	164	040	125	.ASCII	/t U/
017607	156	151	164	.ASCII	/nit/
017612	000	000		.ASCII	<00><00>
017614	045	116	045	P.AKA: .ASCII	/sNs/
017617	101	174	040	.ASCII	/A/<174>/ /
017622	122	145	155	.ASCII	/Rem/
017625	157	166	141	.ASCII	/ova/
017630	142	154	145	.ASCII	/ble/
017633	040	115	145	.ASCII	/ Me/
017636	144	151	141	.ASCII	/dia/
017641	000			.ASCII	<00>
017642	045	116	045	P.AKB: .ASCII	/sNs/
017645	101	174	040	.ASCII	/A/<174>/ /
017650	123	165	160	.ASCII	/Sup/
017653	160	162	145	.ASCII	/pre/
017656	163	163	040	.ASCII	/ss /
017661	103	141	143	.ASCII	/Cac/
017664	150	151	156	.ASCII	/hin/
017667	147	040	050	.ASCII	/g (/
017672	150	151	147	.ASCII	/hig/
017675	150	040	163	.ASCII	/h s/
017700	160	145	145	.ASCII	/pee/
017703	144	051	000	.ASCII	/d)/<00>
017706	045	116	045	P.AKC: .ASCII	/sNs/
017711	101	174	040	.ASCII	/A/<174>/ /
017714	123	165	160	.ASCII	/Sup/
017717	160	162	145	.ASCII	/pre/
017722	163	163	040	.ASCII	/ss /
017725	103	141	143	.ASCII	/Cac/
017730	150	151	156	.ASCII	/hin/
017733	147	040	050	.ASCII	/g (/
017736	154	157	167	.ASCII	/low/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0103
Page 103
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

017741	040	163	160	.ASCII	/ sp/
017744	145	145	144	.ASCII	/eed/
017747	051	000	000	.ASCII	/)/<00><00>
017752	045	116	045	P.AKD: .ASCII	/sNs/
017755	101	174	040	.ASCII	/A/<174>/ /
017760	127	162	151	.ASCII	/Wri/
017763	164	145	055	.ASCII	/te-/
017766	142	141	143	.ASCII	/bac/
017771	153	040	050	.ASCII	/k (/
017774	156	157	156	.ASCII	/non/
017777	055	166	157	.ASCII	/-vo/
020002	154	141	164	.ASCII	/lat/
020005	151	154	145	.ASCII	/ile/
020010	051	000		.ASCII	/)/<00>
020012	045	116	045	P.AKE: .ASCII	/sNs/
020015	101	174	040	.ASCII	/A/<174>/ /
020020	127	162	151	.ASCII	/Wri/
020023	164	145	040	.ASCII	/te /
020026	120	162	157	.ASCII	/Pro/
020031	164	145	143	.ASCII	/tec/
020034	164	040	050	.ASCII	/t (/
020037	150	141	162	.ASCII	/har/
020042	144	167	141	.ASCII	/dwa/
020045	162	145	051	.ASCII	/re)/
020050	000	000		.ASCII	<00><00>
020052	045	116	045	P.AKF: .ASCII	/sNs/
020055	101	174	040	.ASCII	/A/<174>/ /
020060	127	162	151	.ASCII	/Wri/
020063	164	145	040	.ASCII	/te /
020066	120	162	157	.ASCII	/Pro/
020071	164	145	143	.ASCII	/tec/
020074	164	040	050	.ASCII	/t (/
020077	163	157	146	.ASCII	/sof/
020102	164	167	141	.ASCII	/twa/
020105	162	145	040	.ASCII	/re /
020110	157	162	040	.ASCII	/or /
020113	166	157	154	.ASCII	/vol/
020116	165	155	145	.ASCII	/ume/
020121	051	000	000	.ASCII	/)/<00><00>
020124	045	116	045	P.AKG: .ASCII	/sNs/
020127	101	174	040	.ASCII	/A/<174>/ /
020132	065	067	066	.ASCII	/576/
020135	040	102	171	.ASCII	/ By/
020140	164	145	040	.ASCII	/te /
020143	123	145	143	.ASCII	/Sec/
020146	164	157	162	.ASCII	/tor/
020151	163	000	000	.ASCII	/s/<00><00>
020154	045	101	040	P.AKH: .ASCII	/sA /
020157	055	040	123	.ASCII	/- S/
020162	125	103	103	.ASCII	/UCC/
020165	105	123	123	.ASCII	/ESS/
020170	000	000		.ASCII	<00><00>
020172	045	101	040	P.AKI: .ASCII	/sA /

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

020175	055	040	111	.ASCII	/- I/
020200	116	126	101	.ASCII	/NVA/
020203	114	111	104	.ASCII	/LID/
020206	040	103	117	.ASCII	/ CO/
020211	115	115	101	.ASCII	/MMA/
020214	116	104	050	.ASCII	/ND(/
020217	123	105	122	.ASCII	/SER/
020222	126	105	122	.ASCII	/VER/
020225	040	156	157	.ASCII	/ no/
020230	156	111	104	.ASCII	/nID/
020233	114	105	040	.ASCII	/LE /
020236	157	162	040	.ASCII	/or /
020241	156	157	040	.ASCII	/no /
020244	155	145	144	.ASCII	/med/
020247	151	141	040	.ASCII	/ia /
020252	151	146	040	.ASCII	/if /
020255	105	130	055	.ASCII	/EX-/
020260	114	103	055	.ASCII	/LC-/
020263	120	122	107	.ASCII	/PRG/
020266	040	143	155	.ASCII	/ cm/
020271	144	051	000	.ASCII	/d)/<00>
020274	045	101	040	P.AKJ: .ASCII	/#A /
020277	055	040	116	.ASCII	/- N/
020302	117	040	122	.ASCII	/O R/
020305	105	107	111	.ASCII	/EGI/
020310	117	116	040	.ASCII	/ON /
020313	101	126	101	.ASCII	/AVA/
020316	111	114	101	.ASCII	/ILA/
020321	102	114	105	.ASCII	/BLE/
020324	000	000		.ASCII	<00><00>
020326	045	101	040	P.AKK: .ASCII	/#A /
020331	055	040	116	.ASCII	/- N/
020334	117	040	122	.ASCII	/O R/
020337	105	107	111	.ASCII	/EGI/
020342	117	116	040	.ASCII	/ON /
020345	123	125	111	.ASCII	/SUI/
020350	124	101	102	.ASCII	/TAB/
020353	114	105	000	.ASCII	/LE/<00>
020356	045	101	040	P.AKL: .ASCII	/#A /
020361	055	040	120	.ASCII	/- P/
020364	122	117	107	.ASCII	/ROG/
020367	122	101	115	.ASCII	/RAM/
020372	040	116	117	.ASCII	/ NO/
020375	124	040	113	.ASCII	/T K/
020400	116	117	127	.ASCII	/NOW/
020403	116	040	050	.ASCII	/N (/
020406	116	117	040	.ASCII	/NO /
020411	123	125	103	.ASCII	/SUC/
020414	110	040	120	.ASCII	/H P/
020417	122	117	107	.ASCII	/ROG/
020422	122	101	115	.ASCII	/RAM/
020425	040	117	116	.ASCII	/ ON/
020430	040	115	105	.ASCII	/ ME/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

020433	104	111	101
020436	051	000	
020440	045	101	040
020443	055	040	114
020446	117	101	104
020451	040	106	101
020454	111	114	125
020457	122	105	040
020462	050	111	116
020465	120	125	124
020470	040	105	122
020473	122	040	127
020476	110	111	114
020501	105	040	114
020504	117	101	104
020507	111	116	107
020512	040	120	122
020515	117	107	122
020520	101	115	051
020523	000		
020524	045	101	040
020527	055	040	123
020532	124	101	116
020535	104	101	114
020540	117	116	105
020543	040	050	123
020546	124	101	116
020551	104	101	114
020554	117	116	105
020557	040	115	117
020562	104	111	106
020565	111	105	122
020570	040	116	117
020573	124	040	123
020576	120	105	103
020601	111	106	111
020604	105	104	040
020607	106	117	122
020612	040	123	124
020615	101	116	104
020620	040	101	114
020623	117	116	105
020626	040	120	122
020631	107	056	051
020634	000	000	
020636	045	116	045
020641	101	174	040
020644	117	156	145
020647	040	123	145
020652	162	166	145
020655	162	040	141
020660	164	040	141
020663	040	124	151

P.AKM: .ASCII /DIA/
 .ASCII /)/<00>
 .ASCII /#A /
 .ASCII /- L/
 .ASCII /OAD/
 .ASCII / FA/
 .ASCII /ILU/
 .ASCII /RE /
 .ASCII / (IN/
 .ASCII /PUT/
 .ASCII / ER/
 .ASCII /R W/
 .ASCII /HIL/
 .ASCII /E L/
 .ASCII /OAD/
 .ASCII /ING/
 .ASCII / PR/
 .ASCII /OGR/
 .ASCII /AM)/
 .ASCII <00>
 P.AKN: .ASCII /#A /
 .ASCII /- S/
 .ASCII /TAN/
 .ASCII /DAL/
 .ASCII /ONE/
 .ASCII / (S/
 .ASCII /TAN/
 .ASCII /DAL/
 .ASCII /ONE/
 .ASCII / MO/
 .ASCII /DIF/
 .ASCII /IER/
 .ASCII / NO/
 .ASCII /T S/
 .ASCII /PEC/
 .ASCII /IFI/
 .ASCII /ED /
 .ASCII /FOR/
 .ASCII / ST/
 .ASCII /AND/
 .ASCII / AL/
 .ASCII /ONE/
 .ASCII / PR/
 .ASCII /G.)/
 .ASCII <00><00>
 P.AKO: .ASCII /#N#/
 .ASCII /A/<174>/ /
 .ASCII /One/
 .ASCII / Se/
 .ASCII /rve/
 .ASCII /r a/
 .ASCII /t a/
 .ASCII / Ti/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

020666	155	145	000		.ASCII	/me/<00>
020671	000				.ASCII	<00>
020672	045	116	045	P.AKP:	.ASCII	/sNs/
020675	101	174	040		.ASCII	/A/<174>/ /
020700	103	157	156		.ASCII	/Con/
020703	164	141	151		.ASCII	/tai/
020706	156	163	040		.ASCII	/ns /
020711	114	157	143		.ASCII	/Loc/
020714	141	154	040		.ASCII	/al /
020717	115	145	144		.ASCII	/Med/
020722	151	141	000		.ASCII	/ia/<00>
020725	000				.ASCII	<00>
020726	045	116	045	P.AKQ:	.ASCII	/sNs/
020731	101	174	040		.ASCII	/A/<174>/ /
020734	105	170	145		.ASCII	/Exe/
020737	143	165	164		.ASCII	/cut/
020742	145	040	114		.ASCII	/e L/
020745	157	143	141		.ASCII	/oca/
020750	154	040	120		.ASCII	/l P/
020753	162	147	040		.ASCII	/rg /
020756	143	155	144		.ASCII	/cmd/
020761	040	151	163		.ASCII	/ is/
020764	040	125	116		.ASCII	/ UN/
020767	123	125	120		.ASCII	/SUP/
020772	120	117	122		.ASCII	/POR/
020775	124	105	104		.ASCII	/TED/
021000	000	000			.ASCII	<00><00>
021002	045	116	045	P.AKR:	.ASCII	/sNs/
021005	101	174	040		.ASCII	/A/<174>/ /
021010	103	165	162		.ASCII	/Cur/
021013	162	145	156		.ASCII	/ren/
021016	164	154	171		.ASCII	/tly/
021021	040	151	156		.ASCII	/ in/
021024	040	101	143		.ASCII	/ Ac/
021027	164	151	166		.ASCII	/tiv/
021032	145	040	123		.ASCII	/e S/
021035	164	141	164		.ASCII	/tat/
021040	145	000			.ASCII	/e/<00>
021042	045	116	045	P.AKS:	.ASCII	/sNs/
021045	101	174	040		.ASCII	/A/<174>/ /
021050	123	164	141		.ASCII	/Ste/
021053	156	144	141		.ASCII	/nda/
021056	154	157	156		.ASCII	/lon/
021061	145	040	120		.ASCII	/e P/
021064	162	147	000		.ASCII	/rg/<00>
021067	000				.ASCII	<00>
021070	045	116	045	P.AKT:	.ASCII	/sNs/
021073	101	174	040		.ASCII	/A/<174>/ /
021076	116	145	145		.ASCII	/Nee/
021101	144	163	040		.ASCII	/de /
021104	157	166	145		.ASCII	/ove/
021107	162	154	141		.ASCII	/rla/
021112	171	000			.ASCII	/y/<00>

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)SEQ 0107
Page 107

021114	045	116	045	P.AKU:	.ASCII	/sNs/
021117	101	174	040		.ASCII	/A/<174>/ /
021122	116	145	145		.ASCII	/Nee/
021125	144	163	040		.ASCII	/ds /
021130	127	162	151		.ASCII	/Wri/
021133	164	145	141		.ASCII	/tea/
021136	142	154	145		.ASCII	/ble/
021141	057	122	145		.ASCII	<57>/Re/
021144	141	144	141		.ASCII	/ada/
021147	142	154	145		.ASCII	/ble/
021152	040	117	166		.ASCII	/ Ov/
021155	145	162	154		.ASCII	/erl/
021160	141	171	000		.ASCII	/ay/<00>
021163	000				.ASCII	<00>
021164	045	116	045	P.AKV:	.ASCII	/sNs/
021167	101	174	040		.ASCII	/A/<174>/ /
021172	125	163	145		.ASCII	/Use/
021175	163	040	123		.ASCII	/s S/
021200	164	144	040		.ASCII	/td /
021203	104	165	160		.ASCII	/Dup/
021206	040	104	151		.ASCII	/ Di/
021211	141	154	157		.ASCII	/alo/
021214	147	073	040		.ASCII	/g; /
021217	122	105	103		.ASCII	/REC/
021222	057	123	105		.ASCII	<57>/SE/
021225	116	104	057		.ASCII	/ND/<57>
021230	122	105	103		.ASCII	/REC/
021233	000				.ASCII	<00>
021234	045	116	045	P.AKW:	.ASCII	/sNs/
021237	101	011	052		.ASCII	/A/<11>/*/
021242	052	040	121		.ASCII	/* Q/
021245	125	105	123		.ASCII	/UES/
021250	124	111	117		.ASCII	/TIO/
021253	116	000	000		.ASCII	/N/<00><00>
021256	045	116	045	P.AKX:	.ASCII	/sNs/
021261	101	011	052		.ASCII	/A/<11>/*/
021264	052	040	104		.ASCII	/* D/
021267	105	106	101		.ASCII	/EFA/
021272	125	114	124		.ASCII	/ULT/
021275	040	121	125		.ASCII	/ QU/
021300	105	123	124		.ASCII	/EST/
021303	111	117	116		.ASCII	/ION/
021306	000	000			.ASCII	<00><00>
021310	045	116	045	P.AKY:	.ASCII	/sNs/
021313	101	011	052		.ASCII	/A/<11>/*/
021316	052	040	111		.ASCII	/* I/
021321	116	106	117		.ASCII	/NFO/
021324	122	115	101		.ASCII	/RMA/
021327	124	111	117		.ASCII	/TIO/
021332	116	000			.ASCII	/N/<00>
021334	045	116	045	P.AKZ:	.ASCII	/sNs/
021337	101	011	052		.ASCII	/A/<11>/*/
021342	052	040	124		.ASCII	/* T/

ZRQAM1
V01.2 RD/RX EXERCISER
PROTECTION TABLE

021345	105	122	115		.ASCII	/ERM/
021350	111	116	101		.ASCII	/INA/
021353	124	111	117		.ASCII	/TIO/
021356	116	000			.ASCII	/N/<00>
021360	045	116	045	P.ALA:	.ASCII	/NS/
021363	101	011	052		.ASCII	/A/<11>/*/
021366	052	040	106		.ASCII	/* F/
021371	101	124	101		.ASCII	/ATA/
021374	114	040	105		.ASCII	/L E/
021377	122	122	000		.ASCII	/RR/<00>
021402	045	116	045	P.ALB:	.ASCII	/NS/
021405	101	011	052		.ASCII	/A/<11>/*/
021410	052	040	123		.ASCII	/* S/
021413	120	105	103		.ASCII	/PEC/
021416	111	101	114		.ASCII	/IAL/
021421	000				.ASCII	<00>
021422	045	116	045	P.ALC:	.ASCII	/NS/
021425	101	040	055		.ASCII	/A -/
021430	040	111	114		.ASCII	/ IL/
021433	114	105	107		.ASCII	/LEG/
021436	101	114	040		.ASCII	/AL /
021441	125	116	111		.ASCII	/UNI/
021444	124	040	116		.ASCII	/T N/
021447	125	115	102		.ASCII	/UMB/
021452	105	122	000		.ASCII	/ER/<00>
021455	000				.ASCII	<00>
021456	045	116	045	P.ALD:	.ASCII	/NS/
021461	101	040	055		.ASCII	/A -/
021464	040	111	114		.ASCII	/ IL/
021467	114	105	107		.ASCII	/LEG/
021472	101	114	040		.ASCII	/AL /
021475	120	110	131		.ASCII	/PHY/
021500	123	111	103		.ASCII	/SIC/
021503	101	114	040		.ASCII	/AL /
021506	117	122	040		.ASCII	/OR /
021511	122	105	114		.ASCII	/REL/
021514	101	124	111		.ASCII	/ATI/
021517	126	105	040		.ASCII	/VE /
021522	102	114	117		.ASCII	/BLO/
021525	103	113	040		.ASCII	/CK /
021530	116	125	115		.ASCII	/NUM/
021533	102	105	122		.ASCII	/BER/
021536	000	000			.ASCII	<00><00>
021540	045	116	045	P.ALE:	.ASCII	/NS/
021543	101	040	055		.ASCII	/A -/
021546	040	104	105		.ASCII	/ DE/
021551	126	111	103		.ASCII	/VIC/
021554	105	040	105		.ASCII	/E E/
021557	122	122	000		.ASCII	/RR/<00>
021562	045	116	045	P.ALF:	.ASCII	/NS/
021565	101	040	055		.ASCII	/A -/
021570	040	132	105		.ASCII	/ ZE/
021573	122	117	040		.ASCII	/RO /

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

021576	114	105	116	.ASCII	/LEN/
021601	107	110	124	.ASCII	/GHT/
021604	040	115	105	.ASCII	/ ME/
021607	123	123	101	.ASCII	/SSA/
021612	107	105	000	.ASCII	/GE/<00>
021615	000			.ASCII	<00>
021616	045	116	045	P.ALG:	.ASCII /#N#/
021621	101	040	055		.ASCII /A -/
021624	055	040	101		.ASCII /- A/
021627	123	103	111		.ASCII /SCI/
021632	111	040	111		.ASCII /I I/
021635	116	106	117		.ASCII /NFO/
021640	122	115	101		.ASCII /RMA/
021643	124	111	117		.ASCII /TIO/
021646	116	040	000		.ASCII /N /<00>
021651	000				.ASCII <00>
021652	045	116	045	P.ALH:	.ASCII /#N#/
021655	101	040	055		.ASCII /A -/
021660	055	040	116		.ASCII /- N/
021663	117	116	055		.ASCII /ON-/
021666	101	103	123		.ASCII /ACS/
021671	111	111	040		.ASCII /II /
021674	111	116	106		.ASCII /INF/
021677	117	122	115		.ASCII /ORM/
021702	101	124	111		.ASCII /ATI/
021705	117	116	000		.ASCII /ON/<00>
021710	045	116	045	P.ALI:	.ASCII /#N#/
021713	101	040	055		.ASCII /A -/
021716	055	040	124		.ASCII /- T/
021721	105	122	115		.ASCII /ERM/
021724	111	116	101		.ASCII /INA/
021727	124	111	117		.ASCII /TIO/
021732	116	040	115		.ASCII /N M/
021735	105	123	123		.ASCII /ESS/
021740	101	107	105		.ASCII /AGE/
021743	000				.ASCII <00>
021744	045	116	045	P.ALJ:	.ASCII /#N#/
021747	101	040	055		.ASCII /A -/
021752	055	040	123		.ASCII /- S/
021755	125	103	103		.ASCII /UCC/
021760	105	123	123		.ASCII /ESS/
021763	057	106	101		.ASCII <57>/FA/
021766	111	114	125		.ASCII /ILU/
021771	122	105	040		.ASCII /RE /
021774	103	117	104		.ASCII /COD/
021777	105	000	000		.ASCII /E/<00><00>
022002	045	116	045	P.ALK:	.ASCII /#N#/
022005	101	040	055		.ASCII /A -/
022010	055	040	123		.ASCII /- S/
022013	105	116	104		.ASCII /END/
022016	040	102	111		.ASCII / BI/
022021	116	101	122		.ASCII /NAR/
022024	131	040	104		.ASCII /Y D/

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRGACO.BL1;82 (35)ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE

022027	101	124	101		.ASCII	/ATA/
022032	000	000			.ASCII	<00><00>
022034	045	116	045	P.ALL:	.ASCII	/#N#/
022037	101	040	055		.ASCII	/A -/
022042	055	040	123		.ASCII	/- S/
022045	105	116	104		.ASCII	/END/
022050	040	125	116		.ASCII	/ UN/
022053	111	124	040		.ASCII	/IT /
022056	116	125	115		.ASCII	/NUM/
022061	102	105	122		.ASCII	/BER/
022064	054	040	122		.ASCII	/, R/
022067	105	114	101		.ASCII	/ELA/
022072	124	111	126		.ASCII	/TIV/
022075	105	040	104		.ASCII	/E D/
022100	102	116	000		.ASCII	/BN/<00>
022103	000				.ASCII	<00>
022104	045	116	045	P.ALM:	.ASCII	/#N#/
022107	101	040	055		.ASCII	/A -/
022112	055	040	123		.ASCII	/- S/
022115	105	116	104		.ASCII	/END/
022120	040	125	116		.ASCII	/ UN/
022123	111	124	040		.ASCII	/IT /
022126	116	125	115		.ASCII	/NUM/
022131	102	105	122		.ASCII	/BER/
022134	054	040	122		.ASCII	/, R/
022137	105	114	101		.ASCII	/ELA/
022142	124	111	126		.ASCII	/TIV/
022145	105	040	104		.ASCII	/E D/
022150	102	116	054		.ASCII	/BN,/
022153	040	127	122		.ASCII	/ WR/
022156	111	124	105		.ASCII	/ITE/
022161	040	120	101		.ASCII	/ PA/
022164	124	124	105		.ASCII	/TTE/
022167	122	116	000		.ASCII	/RN/<00>
022172	045	116	045	P.ALN:	.ASCII	/#N#/
022175	101	040	055		.ASCII	/A -/
022200	055	040	123		.ASCII	/- S/
022203	105	116	104		.ASCII	/END/
022206	040	125	116		.ASCII	/ UN/
022211	111	124	040		.ASCII	/IT /
022214	116	125	115		.ASCII	/NUM/
022217	102	105	122		.ASCII	/BER/
022222	054	040	120		.ASCII	/, P/
022225	110	131	123		.ASCII	/HYS/
022230	111	103	101		.ASCII	/ICA/
022233	114	040	102		.ASCII	/L B/
022236	114	117	103		.ASCII	/LOC/
022241	113	040	116		.ASCII	/K N/
022244	125	115	102		.ASCII	/UMB/
022247	105	122	000		.ASCII	/ER/<00>
022252	045	116	045	P.ALO:	.ASCII	/#N#/
022255	101	040	055		.ASCII	/A -/
022260	055	040	123		.ASCII	/- S/

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

022263	105	116	104	.ASCII	/END/	
022266	040	125	116	.ASCII	/ UN/	
022271	111	124	040	.ASCII	/IT /	
022274	116	125	115	.ASCII	/NUM/	
022277	102	105	122	.ASCII	/BER/	
022302	054	040	114	.ASCII	/ . L/	
022305	117	107	111	.ASCII	/OGI/	
022310	103	101	114	.ASCII	/CAL/	
022313	040	040	102	.ASCII	/ B/	
022316	114	117	103	.ASCII	/LOC/	
022321	113	040	116	.ASCII	/K N/	
022324	125	115	102	.ASCII	/UMB/	
022327	105	122	040	.ASCII	/ER /	
022332	000	000		.ASCII	<00><00>	
022334	045	101	103	P.ALQ:	.ASCII	/#AC/
022337	124	114	122	.ASCII	/TLR/	
022342	040	124	111	.ASCII	/ TI/	
022345	115	105	117	.ASCII	/MEO/	
022350	125	124	000	.ASCII	/UT/<00>	
022353	000			.ASCII	<00>	
022354	045	101	105	P.ALQ:	.ASCII	/#AE/
022357	116	126	105	.ASCII	/NVE/	
022362	114	117	120	.ASCII	/LOP/	
022365	105	057	120	.ASCII	/E/<57>/P/	
022370	101	103	113	.ASCII	/ACK/	
022373	105	124	040	.ASCII	/ET /	
022376	122	105	101	.ASCII	/REA/	
022401	104	040	105	.ASCII	/D E/	
022404	122	122	040	.ASCII	/RR /	
022407	050	120	101	.ASCII	/(PA/	
022412	122	111	124	.ASCII	/RIT/	
022415	131	040	117	.ASCII	/Y O/	
022420	122	040	124	.ASCII	/R T/	
022423	111	115	105	.ASCII	/IME/	
022426	117	125	124	.ASCII	/OUT/	
022431	051	000	000	P.ALS:	.ASCII	/)/<00><00>
022434	045	101	105	.ASCII	/#AE/	
022437	116	126	105	.ASCII	/NVE/	
022442	114	117	120	.ASCII	/LOP/	
022445	105	057	120	.ASCII	/E/<57>/P/	
022450	101	103	113	.ASCII	/ACK/	
022453	105	124	040	.ASCII	/ET /	
022456	127	122	111	.ASCII	/WRI/	
022461	124	105	040	.ASCII	/TE /	
022464	105	122	122	.ASCII	/ERR/	
022467	040	050	120	.ASCII	/(P/	
022472	101	122	111	.ASCII	/ARI/	
022475	124	131	040	.ASCII	/TY /	
022500	117	122	040	.ASCII	/OR /	
022503	124	111	115	.ASCII	/TIM/	
022506	105	117	125	.ASCII	/EQU/	
022511	124	051	000	.ASCII	/T)/<00>	
022514	045	101	103	P.ALT:	.ASCII	/#AC/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0112
Page 112
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

022517	124	114	122	.ASCII	/TLR/	
022522	040	122	117	.ASCII	/ RO/	
022525	115	040	101	.ASCII	/M A/	
022530	116	104	040	.ASCII	/ND /	
022533	122	101	115	.ASCII	/RAM/	
022536	040	120	101	.ASCII	/ PA/	
022541	122	111	124	.ASCII	/RIT/	
022544	131	040	105	.ASCII	/Y E/	
022547	122	122	000	.ASCII	/RR/<00>	
022552	045	101	103	P.ALU:	.ASCII	/#AC/
022555	124	114	122	.ASCII	/TLR/	
022560	040	122	101	.ASCII	/ RA/	
022563	115	040	120	.ASCII	/M P/	
022566	101	122	111	.ASCII	/ARI/	
022571	124	131	040	.ASCII	/TY /	
022574	105	122	122	.ASCII	/ERR/	
022577	000			.ASCII	<00>	
022600	045	101	103	P.ALV:	.ASCII	/#AC/
022603	124	114	122	.ASCII	/TLR/	
022606	040	122	117	.ASCII	/ RO/	
022611	115	040	120	.ASCII	/M P/	
022614	101	122	111	.ASCII	/ARI/	
022617	124	131	040	.ASCII	/TY /	
022622	105	122	122	.ASCII	/ERR/	
022625	000			.ASCII	<00>	
022626	045	101	122	P.ALW:	.ASCII	/#AR/
022631	111	116	107	.ASCII	/ING/	
022634	040	122	105	.ASCII	/ RE/	
022637	101	104	040	.ASCII	/AD /	
022642	105	122	122	.ASCII	/ERR/	
022645	040	050	120	.ASCII	/ (P/	
022650	101	122	111	.ASCII	/ARI/	
022653	124	131	040	.ASCII	/TY /	
022656	117	122	040	.ASCII	/OR /	
022661	124	111	115	.ASCII	/TIM/	
022664	105	117	125	.ASCII	/EQU/	
022667	124	051	000	.ASCII	/T)/<00>	
022672	045	101	122	P.ALX:	.ASCII	/#AR/
022675	111	116	107	.ASCII	/ING/	
022700	040	127	122	.ASCII	/ WR/	
022703	111	124	105	.ASCII	/ITE/	
022706	040	105	122	.ASCII	/ ER/	
022711	122	040	050	.ASCII	/R (/	
022714	120	101	122	.ASCII	/PAR/	
022717	111	124	131	.ASCII	/ITY/	
022722	040	117	122	.ASCII	/ OR/	
022725	040	124	111	.ASCII	/ TI/	
022730	115	105	117	.ASCII	/MEO/	
022733	125	124	051	.ASCII	/UT)/	
022736	000	000		.ASCII	<00><00>	
022740	111	116	124	P.ALY:	.ASCII	/INT/
022743	105	122	122	.ASCII	/ERR/	
022746	125	120	124	.ASCII	/UPT/	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0113
Page 113
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

022751	040	115	101	.ASCII	/ MA/	
022754	123	124	105	.ASCII	/STE/	
022757	122	040	106	.ASCII	/R F/	
022762	101	111	114	.ASCII	/AIL/	
022765	125	122	105	.ASCII	/URE/	
022770	000	000		.ASCII	<00><00>	
022772	045	101	110	P.ALZ:	.ASCII	/AH/
022775	117	123	124	.ASCII	/OST/	
023000	040	101	103	.ASCII	/ AC/	
023003	103	105	123	.ASCII	/CES/	
023006	123	040	124	.ASCII	/S T/	
023011	111	115	105	.ASCII	/IME/	
023014	117	125	124	.ASCII	/OUT/	
023017	040	050	110	.ASCII	/ (H/	
023022	111	107	110	.ASCII	/IGH/	
023025	105	122	040	.ASCII	/ER /	
023030	114	105	126	.ASCII	/LEV/	
023033	105	114	040	.ASCII	/EL /	
023036	120	122	117	.ASCII	/PRO/	
023041	124	117	103	.ASCII	/TOC/	
023044	117	114	040	.ASCII	/OL /	
023047	104	105	120	.ASCII	/DEP/	
023052	105	116	104	.ASCII	/END/	
023055	105	116	124	.ASCII	/ENT/	
023060	051	000		.ASCII	/)/<00>	
023062	045	101	103	P.AMA:	.ASCII	/AC/
023065	122	105	104	.ASCII	/RED/	
023070	111	124	040	.ASCII	/IT /	
023073	114	111	115	.ASCII	/LIM/	
023076	111	124	040	.ASCII	/IT /	
023101	105	130	103	.ASCII	/EXC/	
023104	105	105	104	.ASCII	/EED/	
023107	105	104	000	.ASCII	/ED/<00>	
023112	045	101	121	P.AMB:	.ASCII	/AQ/
023115	055	102	125	.ASCII	/-BU/	
023120	123	040	115	.ASCII	/S M/	
023123	101	123	124	.ASCII	/AST/	
023126	105	122	040	.ASCII	/ER /	
023131	105	122	122	.ASCII	/ERR/	
023134	000	000		.ASCII	<00><00>	
023136	045	101	103	P.AMC:	.ASCII	/AC/
023141	124	114	122	.ASCII	/TLR/	
023144	040	106	101	.ASCII	/ FA/	
023147	124	101	114	.ASCII	/TAL/	
023152	040	105	122	.ASCII	/ ER/	
023155	122	000	000	.ASCII	/R/<00><00>	
023160	045	101	111	P.AMD:	.ASCII	/AI/
023163	116	123	124	.ASCII	/NST/	
023166	122	125	103	.ASCII	/RUC/	
023171	124	111	117	.ASCII	/TIO/	
023174	116	040	114	.ASCII	/N L/	
023177	117	117	120	.ASCII	/OOP/	
023202	040	124	111	.ASCII	/ TI/	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 B119-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

023205	115	105	117		.ASCII	/MEO/
023210	125	124	000		.ASCII	/UT/<00>
023213	000				.ASCII	<00>
023214	045	101	111	P.AME:	.ASCII	/#AI/
023217	114	114	105		.ASCII	/LLE/
023222	107	101	114		.ASCII	/GAL/
023225	040	126	111		.ASCII	/VI/
023230	122	124	125		.ASCII	/RTU/
023233	101	114	040		.ASCII	/AL /
023236	103	111	122		.ASCII	/CIR/
023241	103	125	111		.ASCII	/CUI/
023244	124	040	111		.ASCII	/T I/
023247	104	000	000		.ASCII	/D/<00><00>
023252	045	101	111	P.AMF:	.ASCII	/#AI/
023255	116	124	105		.ASCII	/NTE/
023260	122	122	125		.ASCII	/RRU/
023263	120	124	040		.ASCII	/PT /
023266	126	105	103		.ASCII	/VEC/
023271	124	117	122		.ASCII	/TOR/
023274	040	111	114		.ASCII	/ IL/
023277	114	105	107		.ASCII	/LEG/
023302	101	114	000		.ASCII	/AL/<00>
023305	000				.ASCII	<00>
023306	045	101	115	P.AMG:	.ASCII	/#AM/
023311	101	111	116		.ASCII	/AIN/
023314	124	105	116		.ASCII	/TEN/
023317	101	116	103		.ASCII	/ANC/
023322	105	040	122		.ASCII	/E R/
023325	105	101	104		.ASCII	/EAD/
023330	057	127	122		.ASCII	<57>/WR/
023333	111	124	105		.ASCII	/ITE/
023336	040	111	116		.ASCII	/ IN/
023341	126	101	114		.ASCII	/VAL/
023344	111	104	040		.ASCII	/ID /
023347	122	105	107		.ASCII	/REG/
023352	111	117	116		.ASCII	/ION/
023355	040	111	104		.ASCII	/ ID/
023360	105	116	124		.ASCII	/ENT/
023363	111	106	111		.ASCII	/IFI/
023366	105	122	000		.ASCII	/ER/<00>
023371	000				.ASCII	<00>
023372	045	101	115	P.AMH:	.ASCII	/#AM/
023375	101	111	116		.ASCII	/AIN/
023400	124	105	116		.ASCII	/TEN/
023403	101	116	103		.ASCII	/ANC/
023406	105	040	127		.ASCII	/E W/
023411	122	111	124		.ASCII	/RIT/
023414	105	040	114		.ASCII	/E L/
023417	117	101	104		.ASCII	/OAD/
023422	040	124	117		.ASCII	/ TO/
023425	040	116	117		.ASCII	/ NO/
023430	116	055	114		.ASCII	/N-L/
023433	117	101	104		.ASCII	/OAD/

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)SEQ 0115
Page 115

023436	101	102	114	.ASCII	/ABL/	
023441	105	040	103	.ASCII	/E C/	
023444	124	114	122	.ASCII	/TLR/	
023447	000			.ASCII	<00>	
023450	045	101	103	P.AMI:	.ASCII	/#AC/
023453	124	114	122	.ASCII	/TLR/	
023456	040	122	101	.ASCII	/ RA/	
023461	115	040	105	.ASCII	/M E/	
023464	122	122	040	.ASCII	/RR /	
023467	050	116	117	.ASCII	/(NO/	
023472	116	055	120	.ASCII	/N-P/	
023475	101	122	111	.ASCII	/ARI/	
023500	124	131	051	.ASCII	/TY)/	
023503	000			.ASCII	<00>	
023504	045	101	111	P.AMJ:	.ASCII	/#AI/
023507	116	111	124	.ASCII	/NIT/	
023512	040	123	105	.ASCII	/ SE/	
023515	121	125	105	.ASCII	/QUE/	
023520	116	103	105	.ASCII	/NCE/	
023523	040	105	122	.ASCII	/ ER/	
023526	122	000		.ASCII	/R/<00>	
023530	045	101	110	P.AMK:	.ASCII	/#AH/
023533	111	107	110	.ASCII	/IGH/	
023536	105	122	040	.ASCII	/ER /	
023541	114	105	126	.ASCII	/LEV/	
023544	105	114	040	.ASCII	/EL /	
023547	120	122	117	.ASCII	/PRO/	
023552	124	117	103	.ASCII	/TOC/	
023555	117	114	040	.ASCII	/OL /	
023560	111	116	103	.ASCII	/INC/	
023563	117	115	120	.ASCII	/OMP/	
023566	101	124	111	.ASCII	/ATI/	
023571	102	111	114	.ASCII	/BIL/	
023574	111	124	131	.ASCII	/ITY/	
023577	040	105	122	.ASCII	/ ER/	
023602	122	000		.ASCII	/R/<00>	
023604	045	101	120	P.AML:	.ASCII	/#AP/
023607	125	122	107	.ASCII	/URG/	
023612	105	057	120	.ASCII	/E/<57>/P/	
023615	117	114	114	.ASCII	/OLL/	
023620	040	110	101	.ASCII	/ HA/	
023623	122	104	127	.ASCII	/RDW/	
023626	101	122	105	.ASCII	/ARE/	
023631	040	106	101	.ASCII	/ FA/	
023634	111	114	125	.ASCII	/ILU/	
023637	122	105	000	.ASCII	/RE/<00>	
023642	045	101	115	P.AMM:	.ASCII	/#AM/
023645	101	120	120	.ASCII	/APP/	
023650	111	116	107	.ASCII	/ING/	
023653	040	122	105	.ASCII	/ RE/	
023656	107	111	123	.ASCII	/GIS/	
023661	124	105	122	.ASCII	/TER/	
023664	040	122	105	.ASCII	/ RE/	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

023667	101	104	040	.ASCII	/AD /
023672	106	101	111	.ASCII	/FAI/
023675	114	125	122	.ASCII	/LUR/
023700	105	040	050	.ASCII	/E (/
023703	120	101	122	.ASCII	/PAR/
023706	111	124	131	.ASCII	/ITY/
023711	040	117	122	.ASCII	/ OR/
023714	040	124	111	.ASCII	/ TI/
023717	115	105	117	.ASCII	/MEO/
023722	125	124	051	.ASCII	/UT)/
023725	000			.ASCII	<00>
023726	022334'			P.ALP: .WORD	P.ALQ
023730	022354'			.WORD	P.ALR
023732	022434'			.WORD	P.ALS
023734	022514'			.WORD	P.ALT
023736	022552'			.WORD	P.ALU
023740	022600'			.WORD	P.ALV
023742	022626'			.WORD	P.ALW
023744	022672'			.WORD	P.ALX
023746	022740'			.WORD	P.ALY
023750	022772'			.WORD	P.ALZ
023752	023062'			.WORD	P.AMA
023754	023112'			.WORD	P.AMB
023756	023136'			.WORD	P.AMC
023760	023160'			.WORD	P.AMD
023762	023214'			.WORD	P.AME
023764	023252'			.WORD	P.AMF
023766	023306'			.WORD	P.AMG
023770	023372'			.WORD	P.AMH
023772	023450'			.WORD	P.AMI
023774	023504'			.WORD	P.AMJ
023776	023530'			.WORD	P.AMK
024000	023604'			.WORD	P.AML
024002	023642'			.WORD	P.AMM
024004	045	101	124	P.AMO: .ASCII	/MAT/
024007	061	061	040	.ASCII	/11 /
024012	103	120	125	.ASCII	/CPU/
024015	040	106	101	.ASCII	/ FA/
024020	111	114	125	.ASCII	/ILU/
024023	122	105	000	.ASCII	/RE/<00>
024026	045	101	116	P.AMP: .ASCII	/MAN/
024031	117	116	055	.ASCII	/ON-/
024034	120	101	122	.ASCII	/PAR/
024037	111	124	131	.ASCII	/ITY/
024042	040	122	101	.ASCII	/ RA/
024045	115	040	105	.ASCII	/M E/
024050	122	122	000	.ASCII	/RR/<00>
024053	000			.ASCII	<00>
024054	045	101	123	P.AMQ: .ASCII	/MAS/
024057	124	101	124	.ASCII	/TAT/
024062	105	040	115	.ASCII	/E M/
024065	101	103	110	.ASCII	/ACH/
024070	111	116	105	.ASCII	/INE/

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

024073	040	106	101	.ASCII	/ FA/	
024076	111	114	125	.ASCII	/ILU/	
024101	122	105	040	.ASCII	/RE /	
024104	055	040	124	.ASCII	/- T/	
024107	061	061	040	.ASCII	/11 /	
024112	101	104	104	.ASCII	/ADD/	
024115	122	105	123	.ASCII	/RES/	
024120	123	040	122	.ASCII	/S R/	
024123	105	107	111	.ASCII	/EGI/	
024126	123	124	105	.ASCII	/STE/	
024131	122	000	000	.ASCII	/R/<00><00>	
024134	045	101	123	P.AMR:	.ASCII	/#AS/
024137	124	101	124	.ASCII	/TAT/	
024142	105	040	115	.ASCII	/E M/	
024145	101	103	110	.ASCII	/ACH/	
024150	111	116	105	.ASCII	/INE/	
024153	040	106	101	.ASCII	/ FA/	
024156	111	114	125	.ASCII	/ILU/	
024161	122	105	040	.ASCII	/RE /	
024164	055	040	121	.ASCII	/- Q/	
024167	055	102	125	.ASCII	/-BU/	
024172	123	040	101	.ASCII	/S A/	
024175	104	104	122	.ASCII	/DDR/	
024200	105	123	123	.ASCII	/ESS/	
024203	040	122	105	.ASCII	/ RE/	
024206	107	111	123	.ASCII	/GIS/	
024211	124	105	122	.ASCII	/TER/	
024214	000	000		.ASCII	<00><00>	
024216	045	101	123	P.AMS:	.ASCII	/#AS/
024221	124	101	124	.ASCII	/TAT/	
024224	105	040	115	.ASCII	/E M/	
024227	101	103	110	.ASCII	/ACH/	
024232	111	116	105	.ASCII	/INE/	
024235	040	106	101	.ASCII	/ FA/	
024240	111	114	125	.ASCII	/ILU/	
024243	122	105	040	.ASCII	/RE /	
024246	055	040	103	.ASCII	/- C/	
024251	122	103	040	.ASCII	/RC /	
024254	122	105	107	.ASCII	/REG/	
024257	111	123	124	.ASCII	/IST/	
024262	105	122	000	.ASCII	/ER/<00>	
024265	000			.ASCII	<00>	
024266	045	101	123	P.AMT:	.ASCII	/#AS/
024271	124	101	124	.ASCII	/TAT/	
024274	105	040	115	.ASCII	/E M/	
024277	101	103	110	.ASCII	/ACH/	
024302	111	116	105	.ASCII	/INE/	
024305	040	106	101	.ASCII	/ FA/	
024310	111	114	125	.ASCII	/ILU/	
024313	122	105	040	.ASCII	/RE /	
024316	055	040	123	.ASCII	/- S/	
024321	105	122	111	.ASCII	/ERI/	
024324	101	114	111	.ASCII	/ALI/	

5-Dec-1983 10:27:14

VAX-11 Bliss-16 V3-555

5-Dec-1983 10:27:04

DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

ZRQAM1 RD/RX EXERCISER
V01.2 PROTECTION TABLE

024327	132	105	122
024332	057	104	105
024335	123	105	122
024340	111	101	114
024343	111	132	105
024346	122	040	122
024351	105	107	111
024354	123	124	105
024357	122	000	000
024362	045	101	123
024365	124	101	124
024370	105	040	115
024373	101	103	110
024376	111	116	105
024401	040	106	101
024404	111	114	125
024407	122	105	040
024412	055	040	127
024415	122	117	116
024420	107	040	110
024423	101	122	104
024426	127	101	122
024431	105	040	126
024434	105	122	123
024437	111	117	116
024442	000	000	
024444	024004'		
024446	024026'		
024450	024054'		
024452	024134'		
024454	024216'		
024456	024266'		
024460	024362'		
024462	045	101	040
024465	045	117	066
024470	000	000	
024472	045	101	157
024475	143	164	040
024500	045	117	064
024503	000		
024504	045	123	064
024507	000		
024510	045	116	000
024513	000		
024514	045	101	040
024517	055	040	000
024522	045	101	052
024525	040	000	000
024530	000000C		
024532	172150		
024534	000154		

```

.ASCII /ZER/
.ASCII <57>/DE/
.ASCII /SER/
.ASCII /IAL/
.ASCII /IZE/
.ASCII /R R/
.ASCII /EGI/
.ASCII /STE/
P.AMU: .ASCII /R/<00><00>
.ASCII /#AS/
.ASCII /TAT/
.ASCII /E M/
.ASCII /ACH/
.ASCII /INE/
.ASCII / FA/
.ASCII /ILU/
.ASCII /RE /
.ASCII /- W/
.ASCII /RON/
.ASCII /G H/
.ASCII /ARD/
.ASCII /WAR/
.ASCII /E V/
.ASCII /ERS/
.ASCII /ION/
P.AMN: .ASCII <00><00>
.WORD P.AMO
.WORD P.AMP
.WORD P.AMQ
.WORD P.AMR
.WORD P.AMS
.WORD P.AMT
.WORD P.AMU
P.AMV: .ASCII /#A /
.ASCII /#06/
.ASCII <00><00>
P.AMW: .ASCII /#Ao/
.ASCII /ct /
.ASCII /#04/
P.AMX: .ASCII <00>
.ASCII /#S4/
P.AMY: .ASCII <00>
.ASCII /#N/<00>
P.AMZ: .ASCII <00>
.ASCII /#A /
.ASCII /- /<00>
P.ANA: .ASCII /#A#/
.ASCII / /<00><00>
L$HWLEN: .WORD <<L$NDHW-L$HWLEN>/2>
HWPT.IP.ADDR: .WORD -5630
HWPT.VECTOR:

```

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0119
Page 119
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

024536	000004	.WORD	154
		HWPT.BR.LEVEL::	
024540	100034	.WORD	4
		HWPT.DISK::	
024542	000000	.WORD	-77744
		HWPT.S.TRK::	
024544	177777	.WORD	0
		HWPT.E.TRK::	
024546		.WORD	-1
024550	000000C	L\$NDHW::.BLKW	1
		L\$SWLEN::	
024552	000040	.WORD	<<L\$NDSW-L\$SWLEN>/2>
		SWP.ERROR::	
024554	000000	.WORD	40
		SWP.XFER::	
024556	001002	.WORD	0
		SWP.FLAGS::	
024560	000000	.WORD	1002
		SWP.DPAT::	
024562	000142	.WORD	0
		SWP.RAT::	
024564	000013	.WORD	142
		DUPROUND::	
024566	000020	.WORD	13
		SWP.UCNT::	
024570		.WORD	20
		SWP.UDPAT::	
024630		.BLKW	20
024632	000000	L\$NDSW::.BLKW	1
024634	177777	L\$PROT::.WORD	0
024636	000006	.WORD	-1
		.WORD	6

000000		.PSECT	\$FFF\$,	RO
000000		CST::.BLKW	37	
000076		CST.ADDR::		
		.BLKW	1	
000100		DCT::.BLKW	11	
000122		DCT.ADDR::		
		.BLKW	1	
000124		RDRX.ADDR::		
		.BLKW	1	
000126		IRDRX.ADDR::		
		.BLKW	1	
000130		ICOM.ADDR::		
		.BLKW	1	
000132		ICST.ADDR::		
		.BLKW	1	
000134		IDCT.ADDR::		
		.BLKW	1	
000136	000020	RDM.CNT::		

ZRQAM1
V01.2RD/RX EXERCISER
PROTECTION TABLE5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)SEQ 0120
Page 120

000140		.WORD	20
000200		RANDOM::.BLKW	20
000200	001	TRK.SGN::	
000201	001	.BYTE	1
000202	001	.BYTE	1
000203	001	.BYTE	1
000204		.BYTE	1
001206		DUPPKT::.BLKW	401
001226		BST::.BLKW	10
001566		TALLY::.BLKW	160
001570		T.ADDR::.BLKW	1
		C.ERR.TBL::	
001572		.BLKW	1
		MSCP.PKT::	
003252		.BLKW	630
		IPKT.ADDR::	
003254		.BLKW	1
		PKT.USE::	
003270		.BLKW	6
003570		RETPKT::.BLKW	140
003574		RP.USE::.BLKW	2
		RP.INDX::	
003576		.BLKW	1
		RP.ADDR::	
003600		.BLKW	1
		ELOG.PKT::	
005230		.BLKW	614
		BUFF.ADDR::	
005250		.BLKW	10
		BUFF.OWN::	
005260		.BLKW	4
005264		IODQ::.BLKW	2
		IODQ.IN::	
005266		.BLKW	1
		IODQ.OUT::	
005270		.BLKW	1
		ENTRY.REASON::	
005271		.BLKB	1
		EOP.FLAG::	
005272		.BLKB	1
		DUP.FLAGS::	
005274		.BLKW	1
005276		CCTLR::.BLKW	1
005300		CDISK::.BLKW	1
005302		CUOFF::.BLKW	1
		CTLR.CNT::	
005304		.BLKW	1
005310		DUR::.BLKW	2
		QIO::.BLKB	1
		.EVEN	
005312		FREE.MEM.ADDR::	
		.BLKW	1
005314		BYTS.PER.QIO::	

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

005316	ST.CODE::	.BLKW	1
005320	SB.CODE::	.BLKW	1
005322	STEP::	.BLKW	1
005324	OF.RC::	.BLKW	1
005326	SA.REG::	.BLKW	1
005330	CMD.TIME::	.BLKW	1
005332	NEX::	.BLKW	1
005334	CRN.LOW::	.BLKW	1
005336	CRN.HIGH::	.BLKW	1
005340	P.INDEX::	.BLKW	1
005342	S.DUPPKT::	.BLKW	1
005344	S.PATTERN::	.BLKW	1
005346	CREDIT.BAL::	.BLKW	1
005350	INIT.OCCURED::	.BLKB	1
005351	NXT.PKT.2USE::	.BLKB	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

100000	BIT15--	-100000
040000	BIT14--	40000
020000	BIT13--	20000
010000	BIT12--	10000
004000	BIT11--	4000
002000	BIT10--	2000
001000	BIT09--	1000
000400	BIT08--	400
000200	BIT07--	200
000100	BIT06--	100
000040	BIT05--	40
000020	BIT04--	20
000010	BIT03--	10
000004	BIT02--	4
000002	BIT01--	2
000001	BIT00--	1
001000	BIT9--	1000
000400	BIT8--	400
000200	BIT7--	200

F10

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

SEQ 0122
Page 122

000100	BIT6--	100
000040	BIT5--	40
000020	BIT4--	20
000010	BIT3--	10
000004	BIT2--	4
000002	BIT1--	2
000001	BIT0--	1
000040	EF.START--	40
000037	EF.RESTART--	37
000036	EF.CONTINUE--	36
000035	EF.NEW--	35
000034	EF.PWR--	34
000340	PRI07--	340
000300	PRI06--	300
000240	PRI05--	240
000200	PRI04--	200
000140	PRI03--	140
000100	PRI02--	100
000040	PRI01--	40
000000	PRI00--	0
000004	EVL--	4
000010	LOT--	10
000020	ADR--	20
000040	IDU--	40
000100	ISR--	100
000200	UAM--	200
000400	BOE--	400
001000	PNT--	1000
002000	PRI--	2000
004000	IXE--	4000
010000	IBE--	10000
020000	IER--	20000
040000	LOE--	40000
100000	HOE--	-100000
000126'	L\$ERRTBL--	ERRTYP
024552'	L\$SW--	L\$SWLEN*2
024532'	L\$HW--	L\$HWLEN*2
000011'	L\$DEPO--	L\$REV*1
000136'	HWQ1--	P.AAA
000152'	HWQ2--	P.AAB
000162'	HWQ3--	P.AAC
000174'	HWQ4--	P.AAD
000220'	HWQ6--	P.AAE
000236'	HWQ7--	P.AAF
000332'	HWQ8--	P.AAG
000404'	HWQ9--	P.AAH
000504'	HWQ10--	P.AAI
000576'	HWQ11--	P.AAJ
000630'	SWQ1--	P.AAK
000652'	SWQ2--	P.AAL
000734'	SWQ4--	P.AAM
000756'	SWQ7--	P.AAN
001022'	SWQ9--	P.AAO

G10

ZRQAM1
V01.2

RD/RX EXERCISER
PROTECTION TABLE

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0123
Page 123
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (35)

001070'	SWQ10--	P.AAP
001134'	SWQ11--	P.AAQ
001166'	SWQ12--	P.AAR
001264'	SWQ13--	P.AAS
001342'	SWQ14--	P.AAT
001360'	SWQ15--	P.AAU
001430'	SWQ17--	P.AAV
001524'	SWQ19--	P.AAW
001546'	SWQ22--	P.AAX
001624'	SWM1--	P.AAY
001714'	NULL--	P.AAZ
001716'	DBM5--	P.ABA
001744'	DBM12--	P.ABB
002022'	DBM18--	P.ABC
002070'	DBM19--	P.ABD
002154'	DBM20--	P.ABE
002232'	DBM21--	P.ABF
002314'	DBM23--	P.ABG
002350'	DBM25--	P.ABH
002414'	DBM26--	P.ABI
002444'	DBM29--	P.ABJ
002512'	DBM107--	P.ABK
002550'	DBM108--	P.ABL
002626'	DBM109--	P.ABM
002706'	DBM110--	P.ABN
002746'	DBM112--	P.ABO
003004'	DU.MSG--	P.ABP
003514'	DU.RSN--	P.ABQ
003544'	MSG.01--	P.ACD
003575'	MSG.02--	P.ACE
003632'	MSG.03--	P.ACF
003664'	RPT1--	P.ACG
003750'	RPT2--	P.ACH
004014'	RPT3--	P.ACI
004100'	RPT4--	P.ACJ
004144'	RPT5--	P.ACK
004232'	RPT6--	P.ACL
004276'	RPT7--	P.ACM
004320'	RPT8--	P.ACN
004342'	RPT9--	P.ACO
004370'	RPT10--	P.ACP
004422'	RPT11--	P.ACQ
004510'	RPT12--	P.ACR
004556'	RPT13--	P.ACS
004656'	RPT14--	P.ACT
004754'	RPT15--	P.ACU
005054'	RPT16--	P.ACV
005136'	RPT17--	P.ACW
005160'	RPT18--	P.ACX
005242'	RPT19--	P.ACY
005264'	EGS.01--	P.ACZ
005304'	EGS.02--	P.ADA
005376'	EGD.10--	P.ADB

005430'	EGD.11--	P.ADC
005454'	EGD.12--	P.ADD
005502'	EGD.13--	P.ADE
005530'	EGD.14--	P.ADF
005550'	EGD.15--	P.ADG
005566'	EGD.16--	P.ADH
005616'	EGD.17--	P.ADI
005634'	EGD.18--	P.ADJ
005652'	EGD.19--	P.ADK
005670'	EGD.20--	P.ADL
005750'	EGD.21--	P.ADM
006052'	EGD.22--	P.ADN
006112'	EGD.23--	P.ADO
006154'	EGD.24--	P.ADP
006220'	EGH.30--	P.ADQ
006244'	EBS.01--	P.ADR
006306'	EBD.10--	P.ADS
006346'	EBD.12--	P.ADT
006412'	EBD.13--	P.ADU
006442'	EBD.14--	P.ADV
006500'	EBD.18--	P.ADW
006534'	EBD.19--	P.ADX
006614'	EH.0--	P.ADY
006652'	EH.1--	P.ADZ
006710'	EH.2--	P.AEA
006750'	EH.3--	P.AEB
007006'	EH.4--	P.AEC
007032'	EH.5--	P.AED
007062'	EH.6--	P.AEE
007112'	EH.7--	P.AEF
007142'	EH.8--	P.AEG
007176'	EH.9--	P.AEH
007226'	EH.10--	P.AEI
007256'	EH.12--	P.AEJ
007314'	EH.13--	P.AEK
010016'	ERR.COD--	P.AEL
010052'	ELG.00--	P.AFA
010364'	ELG.FMT--	P.AFB
010376'	EX.BDR--	P.AFH
010466'	EX.BDW--	P.AFI
010554'	EX.LBR--	P.AFJ
010626'	EX.LBW--	P.AFK
010700'	EX.RBN--	P.AFL
010764'	EX.CBR--	P.AFM
011034'	EX.CBW--	P.AFN
011104'	XX13--	P.AFO
011126'	XX14--	P.AFP
011142'	XX15--	P.AFQ
011164'	XX16--	P.AFR
011212'	XX17--	P.AFS
011230'	XX18--	P.AFT
011240'	XX19--	P.AFU
011252'	XX20--	P.AFV

011266'	XX21--	P.AFW
011334'	XX22--	P.AFX
011376'	XX23--	P.AFY
011432'	XX24--	P.AFZ
011472'	XX25--	P.AGA
011542'	XX26--	P.AGB
011616'	XX27--	P.AGC
011660'	XX29--	P.AGD
011704'	XX30--	P.AGE
011732'	XX31--	P.AGF
011762'	XX32--	P.AGG
012010'	XX33--	P.AGH
012046'	XX34--	P.AGI
012116'	XX35--	P.AGJ
012144'	XX37--	P.AGK
012170'	XX38--	P.AGL
012234'	XX39--	P.AGM
012254'	XX40--	P.AGN
012274'	XX41--	P.AGO
012324'	XX42--	P.AGP
012366'	EB.DCT--	P.AGQ
012434'	EB.COM1--	P.AGR
012516'	EB.PKT--	P.AGS
012560'	EB.RAL--	P.AGT
012620'	EB.ADDR--	P.AGU
012662'	EBNEX1--	P.AGV
012736'	EB.NEX2--	P.AGW
013034'	EBNEX3--	P.AGX
013130'	EB.SA--	P.AGY
013166'	CER.01--	P.AGZ
013232'	CER.02--	P.AHA
013306'	EX.SEQ--	P.AHB
013326'	EX.CRD--	P.AHC
013356'	EX.MTN--	P.AHD
013376'	EX.DGM--	P.AHE
013414'	EX.RD--	P.AHF
013424'	EX.WRT--	P.AHG
013434'	EX.ACC--	P.AHH
013446'	EX.ONL--	P.AHI
013460'	EX.SCC--	P.AHJ
013504'	EX.GDS--	P.AHK
013526'	EX.ESP--	P.AHL
013556'	EX.ELP--	P.AHM
013602'	EX.SDD--	P.AHN
013616'	EX.RCD--	P.AHO
013636'	EX.ABP--	P.AHP
013646'	SC.SDI--	P.AHQ
013672'	SC.CON--	P.AHR
013714'	SC.DUP--	P.AHS
013744'	SC.ONL--	P.AHT
013766'	SC.SON--	P.AHU
014006'	SC.UNK--	P.AHV
014060'	SC.VOL--	P.AHW

014140'	SC.IOP--	P.AHX
014206'	SC.DIS--	P.AHY
014300'	SC.FER--	P.AHZ
014364'	SC.FE2--	P.AIA
014440'	SC.ISH--	P.AIB
014520'	SC.IS2--	P.AIC
014576'	SC.DST--	P.AID
014652'	SC.DS2--	P.AIE
014724'	SC.ECC--	P.AIF
015004'	SC.ECD--	P.AIG
015034'	SC.RCT--	P.AIH
015054'	SC.FUL--	P.AII
015130'	SC.576--	P.AIJ
015204'	SC.FCT--	P.AIK
015252'	SC.EC1--	P.AIL
015300'	SC.EC2--	P.AIM
015326'	SC.EC3--	P.AIN
015356'	SC.EC4--	P.AIO
015404'	SC.EC5--	P.AIP
015432'	SC.EC6--	P.AIQ
015460'	SC.EC7--	P.AIR
015510'	SC.EC8--	P.AIS
015540'	SC.EC9--	P.AIT
015600'	SC.SMP--	P.AIU
015640'	SC.MMP--	P.AIV
015700'	SC.ODA--	P.AIW
015730'	SC.ODB--	P.AIX
015752'	SC.NXM--	P.AIY
016006'	SC.PAR--	P.AIZ
016040'	SC.CTO--	P.AJA
016112'	SC.SDS--	P.AJB
016170'	SC.EDC--	P.AJC
016202'	SC.IDS--	P.AJD
016252'	SC.SRT--	P.AJE
016344'	SC.SRI--	P.AJF
016422'	SC.POE--	P.AJG
016454'	SC.RDY--	P.AJH
016536'	SC.CLK--	P.AJI
016564'	SC.RSP--	P.AJJ
016632'	SC.SUR--	P.AJK
016660'	SC.PSP--	P.AJL
016732'	F.1--	P.AJM
016766'	F.2--	P.AJN
017020'	F.3--	P.AJO
017052'	F.4--	P.AJP
017112'	F.5--	P.AJQ
017166'	F.6--	P.AJR
017240'	F.7--	P.AJS
017312'	F.8--	P.AJT
017362'	F.9--	P.AJU
017402'	F.10--	P.AJV
017432'	F.11--	P.AJW
017456'	F.12--	P.AJX

017504'	F.13--	P.AJY
017554'	F.14--	P.AJZ
017614'	F.15--	P.AKA
017642'	F.16--	P.AKB
017706'	F.17--	P.AKC
017752'	F.18--	P.AKD
020012'	F.19--	P.AKE
020052'	F.20--	P.AKF
020124'	F.21--	P.AKG
020154'	EBH.30--	P.AKH
020172'	EBH.44--	P.AKI
020274'	EBH.45--	P.AKJ
020326'	EBH.46--	P.AKK
020356'	EBH.47--	P.AKL
020440'	EBH.48--	P.AKM
020524'	EBH.49--	P.AKN
020636'	DF.0--	P.AKO
020672'	DF.1--	P.AKP
020726'	DF.2--	P.AKQ
021002'	DF.3--	P.AKR
021042'	DF.4--	P.AKS
021070'	DF.5--	P.AKT
021114'	DF.6--	P.AKU
021164'	DF.7--	P.AKV
021234'	T.QUE--	P.AKW
021256'	T.DEF--	P.AKX
021310'	T.INF--	P.AKY
021334'	T.TER--	P.AKZ
021360'	T.FAT--	P.ALA
021402'	T.SPL--	P.ALB
021422'	E.UNT--	P.ALC
021456'	E.BLK--	P.ALD
021540'	E.DEV--	P.ALE
021562'	E.ZER--	P.ALF
021616'	M.ASC--	P.ALG
021652'	M.BIN--	P.ALH
021710'	M.TER--	P.ALI
021744'	M.COD--	P.ALJ
022002'	M.DAT--	P.ALK
022034'	M.UR--	P.ALL
022104'	M.URP--	P.ALM
022172'	M.UP--	P.ALN
022252'	M.UL--	P.ALO
023726'	CNTR.ERR--	P.ALP
024444'	RDRX.ERR--	P.AMN
024462'	EX.WRD--	P.AMV
024472'	EX.OP--	P.AMW
024504'	SPACE4--	P.AMX
024510'	CRLF--	P.AMY
024514'	DASH--	P.AMZ
024522'	ASTERISK--	P.ANA
024532'	DFPTBL--	L\$HWLEN+2
024552'	SFPTBL--	L\$SWLEN+2

PSECT SUMMARY

Psect Name	Words	Attributes
\$CODE\$	5328	RO , I , LCL, REL, CON
\$FFF\$	1397	RO , I , LCL, REL, CON

LIBRARY STATISTICS

File	----- Symbols -----		Percent	Blocks Read
	Total	Loaded		
DISK\$USER2:[DIETZ.RDRX]ZRQACO.L16;14	400	173	43	50

COMMAND QUALIFIERS

BLISS /PDP11 ZRQACO.BL1/LIST=ZRQACO.LI1/OBJECT=ZRQACO.OB1/SOURCE=PAGE:53

```

: 3442 module ZRQAM2 (
: 3443
: 3444 #title 'RD/RX EXERCISER'
: 3445         ident = 'V01.2',
: 3446         addressing_mode (absolute),
: 3447         environment (noeis)
: 3448     ) =
: 3449
: 3450 begin
: 3451
: 3452 #sbttl 'DECLARATIONS'
: 3453
: 3454 library 'ZRQACO.L16';           ! RDRX EXERCISER GLOBAL LIBRARY
: 3455
: 3456 require 'BLSMAC.REQ';         ! DIAGNOSTIC SUPERVISOR LIBRARY
: 4947
: 4948 forward routine
: 4949     NEX_TRAP : L$ISR novalue,
: 4950     EMS_01  : novalue,
: 4951     EMS_21  : novalue,
: 4952     EMS_13  : novalue,
: 4953     EMS_DUP : novalue,
: 4954     EMS_BLK : novalue,
: 4955     EMSCMD  : NOVALUE,
: 4956     SET_CPAR : novalue,
: 4957     SET_UPAR : novalue,
: 4958     EMS_DBN : novalue;
: 4959 external
: 4960     CST : blockvector [MAX_CTLR, CST_LEN, word] field (CST_FIELDS),
: 4961         ! RUN-TIME CONTROLLER STATUS TABLES
: 4962     CST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 4963         ! CONTROLLER STATUS TABLE ADDRESS OF "CURRENT" CONTROLLER
: 4964     DCT : blockvector [MAX_CTLR, DCT_LEN, word] field (DCT_FIELDS),
: 4965         ! DRIVER CONTROLLER TABLES
: 4966     DCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 4967         ! ADDRESS OF "CURRENT" DRIVER CONTROLLER TABLE
: 4968     RDRX_ADDR : ref rdx field (RC_REG),
: 4969         ! DEVICE ADDRESS OF "CURRENT" CONTROLLER
: 4970     IRDRX_ADDR : ref rdx field (RC_REG),
: 4971         ! DEVICE ADDRESS OF INTERRUPTING CONTROLLER
: 4972     ICOM_ADDR : ref block [COMM_LEN, word] field (COM_FIELDS),
: 4973         ! ADDRESS OF INTERRUPTING CONTROLLER'S COMMUNICATION AREA
: 4974     ICST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 4975         ! ADDRESS OF INTERRUPTING CONTROLLER'S CST
: 4976     IDCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 4977         ! ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 4978     RDM_CNT : word,           ! NUMBER OF RANDOM NUMBERS \ KEEP
: 4979
: 4980     RANDOM : vector [RDM_LEN, word],       ! RANDOM NUMBER TABLE / TOGETHER
: 4981
: 4982     TRK_SGN : vector [MAX_UNITS, byte, signed],           ! CURRENT TRACK DIRECTION
: 4983
: 4984     DUPPKT : BLOCK [257, WORD] field (DP_FIELDS),

```

```

: 4985          ! BUFFER CONTAINING DUP INFORMATION FROM RECEIVE AND SEND COMMANDS
: 4986  BST : blockvector [MAX_UNITS, 2, WORD],
: 4987          ! CONATIN LBNS(LO & HI FIELDS) FOR SEQUENTIAL I/O TRANSFER FOR EACH UNIT
: 4988  TALLY : vector [MAX_UNITS * TALLY_LEN, word] field (T_FIELDS),
: 4989          ! STATISTICS TABLES
: 4990  T_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 4991          ! ADDRESS OF STATISTICS TABLE (TALLY) FOR CURRENT UNIT
: 4992  C_ERR_TBL : blockvector [MAX_CTLR, C_ERR_LEN, word] field (C_ERR_FIELDS),
: 4993          ! STATISTICS TABLE FOR CONTROLLER ERRORS
: 4994  MSCP_PKT : blockvector [PKT_CNT, PKT_LEN, word] field (PKT_FIELDS),
: 4995          ! MSCP PACKET POOL
: 4996  IPKT_ADDR : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 4997          ! ADDRESS OF AN MSCP PACKET (INTERRUPT PROCESSING)
: 4998  PKT_USE : vector [PKT_CNT, byte, signed],
: 4999          ! MSCP PACKET POOL ALLOCATION TABLE
: 5000  RETPKT : blockvector [RP_CNT, RP_LEN, word] field (RP_FIELDS),
: 5001          ! RETURN PACKET POOL
: 5002  RP_USE : vector [RP_CNT, byte, signed],
: 5003          ! RETURN PACKET POOL ALLOCATION TABLE
: 5004  RP_INDX : word,          ! CURRENT RETURN PACKET INDEX
: 5005  RP_ADDR : ref block [RP_LEN, word] field (RP_FIELDS),
: 5006          ! CURRENT RETURN PACKET ADDRESS
: 5007  ELOG_PKT : blockvector [EP_CNT, EP_LEN, word] field (EP_FIELDS),
: 5008          ! ERROR-LOG PACKET SAVE AREA
: 5009  BUFF_ADDR : vector [MAX_BUF_CNT],          ! TABLE OF I/O BUFFER DESCRIPTORS
: 5010  BUFF_OWN : vector [MAX_BUF_CNT, byte, signed],          ! I/O BUFFER OWNERSHIP (CONTROLLER NUMBER)
: 5011  IODQ : vector [IODQ_LEN, byte],
: 5012          ! I/O DONE QUEUE - CIRCULAR QUEUE OF RETPKT INDECES
: 5013  IODQ_IN : word,          ! I/O DONE QUEUE IN POINTER
: 5014  IODQ_OUT : word,          ! I/O DONE QUEUE OUT POINTER
: 5015  ENTRY_REASON : byte,          ! CURRENT OPERATOR COMMAND
: 5016  EOP_FLAG : byte,          ! END-OF-PASS FLAG
: 5017  DUP_FLAGS : WORD,          ! DUP FLAGS
: 5018  CTLR : word,          ! NUMBER OF "CURRENT" CONTROLLER
: 5019  CDISK : word,          ! CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 5020  CUOFF : word,          ! CURRENT UNIT CST OFFSET
: 5021  CTLR_CNT : word,          ! TOTAL NUMBER OF CONFIGURED CONTROLLERS
: 5022  DUR : vector [MAX_UNITS, byte],          ! DROP UNIT REASON
: 5023  QIO : vector [MAX_CTLR, byte],          ! NUMBER OF OUTSTANDING QIOS PER CONTROLLER
: 5024  FREE_MEM_ADDR,          ! START OF FREE MEMORY
: 5025  BYTS_PER_QIO : word,          ! SIZE (BYTES) OF AN I/O BUFFER
: 5026  ST_CODE : word,          ! CURRENT STATUS CODE
: 5027  SB_CODE : word,          ! CURRENT SUB-CODE
: 5028  STEP : word,          ! CURRENT STEP IN HARD_INIT
: 5029  OF_RC : signed word,          ! OFFSET (0 OR 2) TO READ IP OR SA
: 5030  SA_REG : word,          ! STORAGE FOR SA REGISTER READS AND WRITES
: 5031  CMD_TIME : word,          ! COMMAND TIMEOUT VALUE (IN SECONDS)
: 5032  NEX : word,          ! NON-EXISTENT MEMORY TRAP INDICATOR
: 5033  CRN_LOW : word,          ! COMMAND REF NUMBER OF LAST COMMAND SENT
: 5034  CRN_HIGH : word,          ! COMMAND REF NUMBER (HI ORDER)
: 5035  P_INDEX : signed word,          ! CURRENT message PACKET INDEX
: 5036  S_DUPPKT : WORD,          ! DBN BYTE COUNTER
: 5037  S_PATTERN : WORD,          ! THE PATTERN WRITTEN TO DBN'S

```

ZRQAM2
V01.2RD/RX EXERCISER
DECLARATIONS5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 B11es-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1:82 (36)SEQ 0131
Page 131

```

: 5038 CREDIT_BAL : word,          : CREDIT BALANCE
: 5039 INIT_OCCURED : BYTE,      : INDICATES IF EXERCISER FINISHED INIT SEQUENCE
: 5040 NXT_PKT_2USE : byte,      : POINTER TO NEXT ENTRY IN PKT_USE TABLE
: 5041 DBM5,
: 5042 DBM107,
: 5043 DU_MSG,
: 5044 DU_RSN : vector [12],
: 5045 ERR_COD : vector [14],
: 5046 ELG_FMT : vector [5],
: 5047 !
: 5048 HWQ1,
: 5049 HWQ2,
: 5050 HWQ3,
: 5051 HWQ4,
: 5052 HWQ6,
: 5053 HWQ7,
: 5054 HWQ8,
: 5055 HWQ9,
: 5056 HWQ10,
: 5057 HWQ11,
: 5058 SWQ1,
: 5059 SWQ2,
: 5060 SWQ4,
: 5061 SWQ7,
: 5062 SWQ9,
: 5063 SWQ10,
: 5064 SWQ11,
: 5065 SWQ12,
: 5066 SWQ13,
: 5067 SWQ14,
: 5068 SWQ15,
: 5069 SWQ17,
: 5070 SWQ19,
: 5071 SWQ22,
: 5072
: 5073 SWM1,
: 5074 NULL,
: 5075 MSG_01,
: 5076 MSG_02,
: 5077 MSG_03,
: 5078 RPT1,
: 5079 RPT2,
: 5080 RPT3,
: 5081 RPT4,
: 5082 RPT5,
: 5083 RPT6,
: 5084 RPT7,
: 5085 RPT8,
: 5086 RPT9,
: 5087 RPT10,
: 5088 RPT11,
: 5089 RPT12,
: 5090 RPT13,

```

:	5091	RPT14.
:	5092	RPT15.
:	5093	RPT16.
:	5094	RPT17.
:	5095	RPT18.
:	5096	RPT19.
:	5097	EGS_01.
:	5098	EGS_02.
:	5099	EGD_10.
:	5100	EGD_11.
:	5101	EGD_12.
:	5102	EGD_13.
:	5103	EGD_14.
:	5104	EGD_15.
:	5105	EGD_16.
:	5106	EGD_17.
:	5107	EGD_18.
:	5108	EGD_19.
:	5109	EGD_20.
:	5110	EGD_21.
:	5111	EGD_22.
:	5112	EGD_23.
:	5113	EGD_24.
:	5114	EGH_30.
:	5115	EH_0.
:	5116	EH_1.
:	5117	EH_2.
:	5118	EH_3.
:	5119	EH_4.
:	5120	EH_5.
:	5121	EH_6.
:	5122	EH_7.
:	5123	EH_8.
:	5124	EH_9.
:	5125	EH_10.
:	5126	EH_12.
:	5127	EH_13.
:	5128	
:	5129	EBS_01.
:	5130	EBD_10.
:	5131	EBD_12.
:	5132	EBD_13.
:	5133	EBD_14.
:	5134	EBD_18.
:	5135	EBD_19.
:	5136	ELG_00.
:	5137	EX_BDR.
:	5138	EX_BDW.
:	5139	EX_LBR.
:	5140	EX_LBW.
:	5141	EX_RBN.
:	5142	EX_CBR.
:	5143	EX_CBW.

ZRQAM2
V01.2

RD/RX EXERCISER
DECLARATIONS

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0133
Page 133
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (36)

: 5144 XX13.
: 5145 XX14.
: 5146 XX15.
: 5147 XX16.
: 5148 XX17.
: 5149 XX18.
: 5150 XX19.
: 5151 XX20.
: 5152 XX21.
: 5153 XX22.
: 5154 XX23.
: 5155 XX24.
: 5156 XX25.
: 5157 XX26.
: 5158 XX27.
: 5159 XX29.
: 5160 XX30.
: 5161 XX31.
: 5162 XX32.
: 5163 XX33.
: 5164 XX34.
: 5165 XX35.
: 5166 !XX36.
: 5167 XX37.
: 5168 XX38.
: 5169 XX39.
: 5170 XX40.
: 5171 XX41.
: 5172 XX42.
: 5173 EB_DCT.
: 5174 EB_COMM.
: 5175 EB_PKT.
: 5176 EB_RAL.
: 5177 EB_ADDR.
: 5178 EBNEX1.
: 5179 EB_NEX2.
: 5180 EBNEX3.
: 5181 EB_SA.
: 5182 CER_01.
: 5183 CER_02.
: 5184 EX_SEQ.
: 5185 EX_CRD.
: 5186 EX_MTN.
: 5187 EX_DGM.
: 5188 EX_RD.
: 5189 EX_WRT.
: 5190 EX_ACC.
: 5191 EX_ONL.
: 5192 EX_SCC.
: 5193 EX_GDS.
: 5194 EX_ESP.
: 5195 EX_ELP.
: 5196 EX_SDD.

E11

ZRQAM2
V01.2

RD/RX EXERCISER
DECLARATIONS

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (36)

SEQ 0134

Page 134

:	5197	EX_RCD.
:	5198	EX_ABP.
:	5199	SC_SDI.
:	5200	SC_CON.
:	5201	SC_DUP.
:	5202	SC_ONL.
:	5203	SC_SON.
:	5204	SC_UNK.
:	5205	SC_VOL.
:	5206	SC_IOP.
:	5207	SC_DIS.
:	5208	SC_FER.
:	5209	SC_FE2.
:	5210	SC_ISH.
:	5211	SC_IS2.
:	5212	SC_DST.
:	5213	SC_DS2.
:	5214	SC_ECC.
:	5215	SC_ECD.
:	5216	SC_RCT.
:	5217	SC_FUL.
:	5218	SC_576.
:	5219	SC_FCT.
:	5220	SC_EC1.
:	5221	SC_EC2.
:	5222	SC_EC3.
:	5223	SC_EC4.
:	5224	SC_EC5.
:	5225	SC_EC6.
:	5226	SC_EC7.
:	5227	SC_EC8.
:	5228	SC_EC9.
:	5229	SC_SWP.
:	5230	SC_HMP.
:	5231	SC_ODA.
:	5232	SC_ODB.
:	5233	SC_NXM.
:	5234	SC_PAR.
:	5235	SC_CTO.
:	5236	SC_SDS.
:	5237	SC_EDC.
:	5238	SC_IDS.
:	5239	SC_SRT.
:	5240	SC_SRI.
:	5241	SC_POE.
:	5242	SC_RDY.
:	5243	SC_CLK.
:	5244	SC_RSP.
:	5245	SC_SUR.
:	5246	SC_PSP.
:	5247	F_1.
:	5248	F_2.
:	5249	F_3.

ZRQAM2
V01.2RD/RX EXERCISER
DECLARATIONS5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (36)SEQ 0135
Page 135

```
: 5250 F_4,  
: 5251 F_5,  
: 5252 F_6,  
: 5253 F_7,  
: 5254 F_8,  
: 5255 F_9,  
: 5256 F_10,  
: 5257 F_11,  
: 5258 F_12,  
: 5259 F_13,  
: 5260 F_14,  
: 5261 F_15,  
: 5262 F_16,  
: 5263 F_17,  
: 5264 F_18,  
: 5265 F_19,  
: 5266 F_20,  
: 5267 F_21,  
: 5268 EBH_30,  
: 5269 EBH_44,  
: 5270 EBH_45,  
: 5271 EBH_46,  
: 5272 EBH_47,  
: 5273 EBH_48,  
: 5274 EBH_49,  
: 5275 df_0,  
: 5276 df_1,  
: 5277 df_2,  
: 5278 df_3,  
: 5279 df_4,  
: 5280 df_5,  
: 5281 df_6,  
: 5282 df_7,  
: 5283 T_QUE,  
: 5284 T_DEF,  
: 5285 T_INF,  
: 5286 T_TER,  
: 5287 T_FAT,  
: 5288 T_SPL,  
: 5289 E_UNT,  
: 5290 E_BLK,  
: 5291 E_DEV,  
: 5292 E_ZER,  
: 5293 M_ASC,  
: 5294 M_BIN,  
: 5295 M_TER,  
: 5296 M_COD,  
: 5297 M_DAT,  
: 5298 M_UR,  
: 5299 M_URP,  
: 5300 M_UP,  
: 5301 M_UL,  
: 5302 CNTR_ERR : vector [23],
```

ZRQAM2
V01.2RD/RX EXERCISER
DECLARATIONS5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (36)SEQ 0136
Page 136

```

: 5303 RDRX_ERR : vector [7],
: 5304 EX_WRD,
: 5305 EX_OP,
: 5306 SPACE4,
: 5307 CRLF,
: 5308 DASH,
: 5309 ASTERISK,
: 5310 SWP_FLAGS : word,
: 5311 L$HMEM,
: 5312 L$LUN,
: 5313 L$UNIT;
:
: 5314
: 5315
: 5316 own
: 5317 TBL_SUC : vector [17] initial (NULL, SC_SDI, SC_CON, NULL, SC_DUP, NULL, NULL,
: 5318 NULL, SC_ONL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, SC_SON),
: 5319 TBL_OFI : vector [9] initial (SC_UNK, SC_VOL, SC_IOP, NULL, SC_DUP, NULL, NULL,
: 5320 NULL, SC_DIS),
: 5321 TBL_MFE : vector [11] initial (SC_FER, NULL, SC_ISH, SC_DST, SC_EC9, SC_576,
: 5322 SC_FCT, SC_ECC, SC_RCT, SC_FUL, SC_EC1),
: 5323 TBL_WPT : vector [3] initial (NULL, SC_SWP, SC_HWP),
: 5324 TBL_DAT : vector [16] initial (SC_FE2, NULL, SC_IS2, SC_DS2, SC_EC9, NULL, NULL,
: 5325 SC_ECD, SC_EC1, SC_EC2, SC_EC3, SC_EC4, SC_EC5, SC_EC6, SC_EC7, SC_EC8),
: 5326 TBL_HST : vector [5] initial (NULL, SC_ODA, SC_ODB, SC_NXM, SC_PAR),
: 5327 TBL_CNT : vector [4] initial (SC_CTO, SC_SDS, SC_EDC, SC_IDS),
: 5328 TBL_DRV : vector [9] initial (NULL, SC_SRT, SC_SRI, SC_POE, SC_RDY, SC_CLK, SC_RSP,
: 5329 SC_SUR, SC_PSP);

```

H11

ZRQAM2
V01.2

RD/RX EXERCISER
TYPE AND DESCRIPTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (37)

SEQ 0137
Page 137

```
: 5330 #sbttl 'TYPE AND DESCRIPTION'  
: 5331  
: 5332 EQUALS;  
: 5333  
: 5334 DEVTYP (#asciz'RQDX1');      ! NAME OF DEVICE SUPPORTED BY PROGRAM  
: 5335 DESCRIPT (#asciz'RD/RX EXERCISER'); ! TEST DESCRIPTION
```

ZRQAM2
V01.2RD/RX EXERCISER
HARDWARE PARAMETER CODING SECTION5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1:82 (38)SEQ 0138
Page 138

```

: 5336 #sbttl 'HARDWARE PARAMETER CODING SECTION'
: 5337
: 5338 !+
: 5339 ! THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
: 5340 ! THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: 5341 ! MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: 5342 ! INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: 5343 ! MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: 5344 ! WITH THE OPERATOR.
: 5345 !-
: 5346
: 5347 BGNHRD;
: 5348
: 5349 GPRMA (HWQ1, 0, 0, #0'160000', #0'177777', YES, 1);      ! IP ADDRESS
: 5350 GPRMA (HWQ2, 2, 0, #0'4', #0'774', YES, 1);           ! VECTOR
: 5351 GPRMD (HWQ3, 4, 0, #0'377', #0'0', #0'7', YES, 1);    ! BR LEVEL
: 5352 GPRMD (HWQ4, 6, D, #0'3', #decimal'0', #decimal'3', YES, 1); ! RDRX DRIVE NUMBER
: 5353 GPRML (HWQ10, 6, #0'000010', YES, 1);                 ! run dup exerciser
: 5354 XFERF (NODU);
: 5355 GPRML (HWQ11, 6, #0'000020', YES, 1);                 ! WRITE TO DBN'S
: 5356 #L (NODU);
: 5357 GPRMD (HWQ6, 8, D, #0'177777', #decimal'0', #0'177777', YES, 1); ! STARTING LBN
: 5358 GPRMD (HWQ7, 10, D, #0'177777', GP$ATLO (8), #0'177777', YES, 1); ! ENDING LBN
: 5359 GPRML (HWQ8, 6, #0'100000', NO, 1);                   ! EXER ON CUST DATA AREA
: 5360 XFERF (HMDONE);                                       ! NO - DONE
: 5361 GPRML (HWQ9, 6, #0'100000', NO, 1);                   ! ** WARNING / CONFIRM
: 5362 #L (HMDONE);
: 5363
: 5364 ENDHRD;

```

```

: 5365 #sbttl 'SOFTWARE PARAMETER CODING SECTION'
: 5366
: 5367 !*
: 5368 ! THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
: 5369 ! THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: 5370 ! MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: 5371 ! INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: 5372 ! MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: 5373 ! WITH THE OPERATOR.
: 5374 !-
: 5375
: 5376 BGNSFT;
: 5377 !

: 5378 !GPRML (SWQ16, 4, SWF_TRC, YES, 1);           ! ENABLE DIAGNOSTIC TRACE
: 5379 GPRMD (SWQ1, 0, D, #o'177777', 0, 65535, YES, 1); ! ERROR LIMIT
: 5380 GPRMD (SWQ2, 2, D, #o'177777', 0, 99, YES, 1);    ! TRANSFER LIMIT
: 5381 GPRMD (SWQ22, 10, D, #o'177777', 0, 144, YES, 1); ! NUMBER OF DBN'S WRITTEN AT ONE TIME
: 5382 GPRML (SWQ15, 4, SWF_CST, YES, 1);             ! CLEAR STATISTICAL TABLES ?
: 5383 !GPRML (SWQ20, 4, SWF_FER, YES, 1);             ! REWRITE BLOCKS WHEN "FORCED ERROR" BIT SET?
: 5384 !GPRML (SWQ21, 4, SWF_HOE, YES, 1);            ! HALT ON HARD/SOFT ERRORS WITH 'HOE' FLAG SET?
: 5385 GPRML (SWQ4, 4, SWF_UNT, YES, 1);             ! RANDOM UNIT MODE ?
: 5386 XFERF (SW2);                                   ! IF ANSWER FALSE NO SENSE IN ASK RATIO OF RDs TO RXs
: 5387 GPRMD (SWQ17, 8, D, #o'177777', 0, 100, YES, 1); ! PERCENT OF RD OPERATIONS
: 5388 #L (SW2);
: 5389 GPRML (SWQ19, 4, SWF_BLK, YES, 1);             ! RANDOM BLOCK MODE?
: 5390 GPRML (SWQ7, 4, SWF_CRC, YES, 1);             ! READ-COMPARES AT CONTROLLER ?
: 5391 DISPLAY (SWM1);                                ! REMAINING QUESTIONS ONLY APPLY ...
: 5392 GPRML (SWQ9, 4, SWF_CWC, YES, 1);             ! WRITE-COMPARES AT CONTROLLER ?
: 5393 XFERF (SW3);                                   ! IF NO, DO NEXT QUESTION
: 5394 XFER (SW4);
: 5395 #L (SW3);
: 5396 GPRML (SWQ10, 4, SWF_HMC, YES, 1);             ! CHECK WRITES AT HOST BY READING ?
: 5397 #L (SW4);
: 5398 GPRML (SWQ11, 4, SWF_UDP, YES, 1);             ! USER-DEFINED DATA PATTERN ?
: 5399 XFERF (SW5);                                   ! IF NO, DO NEXT QUESTION
: 5400 XFER (SW6);
: 5401 #L (SW5);
: 5402 GPRMD (SWQ12, 6, D, #o'177777', 0, DP_CNT, YES, 1); ! SELECT PRE-DEFINED DATA PATTERN
: 5403 XFER (SW7);                                     ! DONE
: 5404 #L (SW6);
: 5405 GPRMD (SWQ13, 12, D, #o'177777', 1, MAX_UDP_CNT, YES, 1); ! NO. OF WORDS IN USER DATA PATTERN
: 5406 GPRMD (SWQ14, 14, D, #o'177777', 0, #o'177777', NO, 12); ! PATTERN VALUES
: 5407 #L (SW7);
: 5408
: 5409 ENDSFT;

```

```

: 5410 #sbttl 'REPORT CODING SECTION'
: 5411
: 5412 !+
: 5413 ! THE REPORT CODING SECTION CONTAINS THE
: 5414 ! "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
: 5415 !-
: 5416
: 5417 BGNRPT;
: 5418                                     ! PRINTS MSCP DATA
: 5419 PRINTS (RPT1);
: 5420 PRINTS (RPT2);
: 5421 PRINTS (RPT3);
: 5422 PRINTS (RPT4);
: 5423 PRINTS (RPT5);
: 5424 PRINTS (RPT6);
: 5425
: 5426 incr CTLR from 0 to MAX_CTLR - 1 do
: 5427     begin
: 5428     SET_CPAR (.CTLR);
: 5429
: 5430     incr DISK from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5431     begin
: 5432     SET_UPAR (.DISK);
: 5433
: 5434     if (.CST_ADDR [.DISK, D_TYPE] eq1 RX_50) and                !RX 50
: 5435     (.CST_ADDR [.DISK, D_PRES] eq1 PRESENT)
: 5436     then
: 5437     PRINTS (RPT7, .L#LUN, .CST_ADDR [.DISK, D_DISK_NUM]);
: 5438
: 5439     if (.CST_ADDR [.DISK, D_TYPE] eq1 RD_51) and                !RD51
: 5440     (.CST_ADDR [.DISK, D_PRES] eq1 PRESENT)
: 5441     then
: 5442     PRINTS (RPT8, .L#LUN, .CST_ADDR [.DISK, D_DISK_NUM]);
: 5443
: 5444     if (.CST_ADDR [.DISK, D_TYPE] eq1 RD_52) and                !RD52
: 5445     (.CST_ADDR [.DISK, D_PRES] eq1 PRESENT)
: 5446     then
: 5447     PRINTS (RPT17, .L#LUN, .CST_ADDR [.DISK, D_DISK_NUM]);
: 5448
: 5449     if (.CST_ADDR [.DISK, D_TYPE] neq RD_52) and                !RD52
: 5450     (.CST_ADDR [.DISK, D_TYPE] neq RD_51) and
: 5451     (.CST_ADDR [.DISK, D_TYPE] neq RX_50) and
: 5452     (.CST_ADDR [.DISK, D_PRES] eq1 PRESENT)
: 5453     then
: 5454     PRINTS (RPT19, .L#LUN, .CST_ADDR [.DISK, D_DISK_NUM]);
: 5455
: 5456
: 5457     if .CST_ADDR [.DISK, D_PRES] eq1 PRESENT
: 5458     then
: 5459     begin
: 5460     PRINTS (RPT9,
: 5461     .T_ADDR [TOT_READS_HI], .T_ADDR [TOT_READS_LO],
: 5462     .T_ADDR [TOT_BYT_RED], .T_ADDR [TOT_BYT_RED_HI], .T_ADDR [TOT_BYT_RED_LO]);

```

```

: P 5463 PRINTS (RPT9,
: P 5464 .T_ADDR [TOT_WRITES_HI], .T_ADDR [TOT_WRITES_LO],
: 5465 .T_ADDR [MTOT_BYT_WRT], .T_ADDR [TOT_BYT_WRT_HI], .T_ADDR [TOT_BYT_WRT_LO]);
: P 5466 PRINTS (RPT10,
: P 5467 .T_ADDR [ERR_HRD_SEK], .T_ADDR [ERR_HRD_DAT], .T_ADDR [ERR_HRD_DRV], .T_ADDR [ERR_HRD_HST],
: 5468 .T_ADDR [ERR_SFT_SEK], .T_ADDR [ERR_SFT_DAT], .T_ADDR [ERR_SFT_DRV], .T_ADDR [ERR_SFT_HST]);
: 5469 end;
: 5470
: 5471 end;
: 5472
: 5473 if .CST [.CTRL, STATE] eq1 PRESENT
: 5474 then
: 5475 begin
: 5476 PRINTS (RPT11);
: 5477 PRINTS (RPT12, .C_ERR_TBL [.CTRL, C_ERR_HRD], .C_ERR_TBL [.CTRL, C_ERR_SFT]);
: 5478 end;
: 5479
: 5480 PRINTS (CRLF);
: 5481
: 5482 end;
: 5483 begin
: 5484 prints(crlf);
: 5485 PRINTS(RPT13);!†
: 5486 PRINTS(RPT14);
: 5487 PRINTS(RPT15);
: 5488 INCR CTRL FROM 0 TO MAX_CTRL-1 DO
: 5489 BEGIN
: 5490 SET_CPAR(.CTRL);
: 5491 INCR DISK FROM (0+OF_UN) TO (3*UNIT_SIZE+OF_UN) BY UNIT_SIZE DO
: 5492 BEGIN
: 5493 SET_UPAR(.DISK);
: 5494 IF .CST_ADDR[.DISK, D_TYPE] EQLU RD_51 and .CST_ADDR [.DISK, D_PRES] eq1 PRESENT
: 5495 THEN
: P 5496 PRINTS (RPT16,
: P 5497 .L$LUN, .CST_ADDR [.DISK, D_DISK_NUM],
: 5498 .T_ADDR [T_DBN_RD], .T_ADDR [T_BLK_RD], .T_ADDR [T_DBN_WT], .T_ADDR [T_BLK_WT]);
: 5499
: 5500 IF .CST_ADDR[.DISK, D_TYPE] EQLU RD_52 and .CST_ADDR [.DISK, D_PRES] eq1 PRESENT
: 5501 THEN
: P 5502 PRINTS (RPT18,
: P 5503 .L$LUN, .CST_ADDR [.DISK, D_DISK_NUM],
: 5504 .T_ADDR [T_DBN_RD], .T_ADDR [T_BLK_RD], .T_ADDR [T_DBN_WT], .T_ADDR [T_BLK_WT]);
: 5505 END;
: 5506 END;
: 5507 end;
: 5508
: 5509 PRINTS (CRLF);
: 5510
: 5511 ENDRPT;

```

! PRINTS DUP DATA

.TITLE ZRQAM2 RD/RX EXERCISER
.IDENT /V01.2/


```

          .ENABL  AMA
000000          .PSECT  $CODE$,  RO
000000      122      121      104      L$DVTYP::
          .ASCII  /RQD/
          .ASCII  /X1/<00>
000003      130      061      000      .BLKB  2
000006
000010      122      104      057      L$DESC::
000013      122      130      040      .ASCII  /RD/<57>
000016      105      130      105      .ASCII  /RX /
000021      122      103      111      .ASCII  /EXE/
000024      123      105      122      .ASCII  /RCI/
000027      000          .ASCII  /SER/
000030          .ASCII  <00>
000032      000000C      .BLKB  2
          L$HRDLN::
000034      000031      GP$1::  .WORD  <<<L$NDHRD-L$HRDLN>/2>-1>
000036      000000G      .WORD  31
000040      160000      .WORD  HWQ1
000042      177777      .WORD  -20000
000044      001031      GP$2::  .WORD  -1
000046      000000G      .WORD  1031
000050      000004      .WORD  HWQ2
000052      000774      .WORD  4
000054      002032      GP$3::  .WORD  774
000056      000000G      .WORD  2032
000060      000377      .WORD  HWQ3
000062      000000      .WORD  377
000064      000007      .WORD  0
000066      003052      GP$4::  .WORD  7
000070      000000G      .WORD  3052
000072      000003      .WORD  HWQ4
000074      000000      .WORD  3
000076      000003      .WORD  0
000100      003130      GP$5::  .WORD  3
000102      000000G      .WORD  3130
000104      000010      .WORD  HWQ10
000106      000000C      .WORD  10
000110      003130      $NODU:  .WORD  <<<<$LNODU-$NODU>*400>+4>+40>
000112      000000G      GP$6::  .WORD  3130
000114      000020      .WORD  HWQ11
000116      001004      .WORD  20
000120      004052      $LNODU:  .WORD  1004
000122      000000G      GP$7::  .WORD  4052
000124      177777      .WORD  HWQ6
000126      000000      .WORD  -1
000130      177777      .WORD  0
000132      005452      GP$8::  .WORD  -1
000134      000000G      .WORD  5452
000136      177777      .WORD  HWQ7
000140      000004      .WORD  -1
000142      177777      .WORD  4
          .WORD  -1

```

N11

ZRQAM2 RD/RX EXERCISER
V01.2 REPORT CODING SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555

DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (40)

SEQ 0143
Page 143

000144	000001		.WORD	1
000146	003120	GP#9::	.WORD	3120
000150	000000G		.WORD	HWQ8
000152	100000		.WORD	-100000
000154	000000C	\$HWDONE:	.WORD	<<<<\$LHWDONE-\$HWDONE>*400>+4>+40>
000156	003120	GP#10::	.WORD	3120
000160	000000G		.WORD	HWQ9
000162	100000		.WORD	-100000
000164	001004	\$LHWDONE:	.WORD	1004
000166		L\$NDHRD::	.BLKW	1
000170	000000C	L\$SFTLN::	.WORD	<<<<L\$NDSFT-L\$SFTLN>/2>-1>
000172	000052	GP#11::	.WORD	52
000174	000000G		.WORD	SWQ1
000176	177777		.WORD	-1
000200	000000		.WORD	0
000202	177777		.WORD	-1
000204	001052	GP#12::	.WORD	1052
000206	000000G		.WORD	SWQ2
000210	177777		.WORD	-1
000212	000000		.WORD	0
000214	000143		.WORD	143
000216	005052	GP#13::	.WORD	5052
000220	000000G		.WORD	SWQ22
000222	177777		.WORD	-1
000224	000000		.WORD	0
000226	000220		.WORD	220
000230	002130	GP#14::	.WORD	2130
000232	000000G		.WORD	SWQ15
000234	000200		.WORD	200
000236	002130	GP#15::	.WORD	2130
000240	000000G		.WORD	SWQ4
000242	000002		.WORD	2
000244	000000C	\$SW2:	.WORD	<<<<\$LSW2-\$SW2>*400>+4>+40>
000246	004052	GP#16::	.WORD	4052
000250	000000G		.WORD	SWQ17
000252	177777		.WORD	-1
000254	000000		.WORD	0
000256	000144		.WORD	144
000260	001004	\$LSW2:	.WORD	1004
000262	002130	GP#17::	.WORD	2130
000264	000000G		.WORD	SWQ19
000266	001000		.WORD	1000
000270	002130	GP#18::	.WORD	2130
000272	000000G		.WORD	SWQ7
000274	000004		.WORD	4
000276	000003	GP#DISP::	.WORD	3
000300	000000G		.WORD	SWM1
000302	002130	GP#19::	.WORD	2130
000304	000000G		.WORD	SWQ9

ZRQAM2
V01.2

RD/RX EXERCISER
REPORT CODING SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0144
Page 144
VAX-11 Bliss-16 V3-555
DISK\$USER2:([DIETZ.RDRX]ZRQACO.BL1;82 (40))

000306 000020
000310 000000C
000312 000000C
000314 001004
000316 002130
000320 000000G
000322 000040
000324 001004
000326 002130
000330 000000G
000332 000100
000334 000000C
000336 000000C
000340 001004
000342 003052
000344 000000G
000346 177777
000350 000000
000352 000025
000354 000000C
000356 001004
000360 006052
000362 000000G
000364 177777
000366 000001
000370 000020
000372 007222
000374 000000G
000376 177777
000400 000000
000402 177777
000404 000006
000406 001004
000410

.WORD 20
\$SW3: .WORD <<<<\$LSW3-\$SW3>*400>*4>*40>
\$SW4: .WORD <<<\$LSW4-\$SW4>*400>*4>
\$LSW3: .WORD 1004
GP\$20:: .WORD 2130
.WORD SWQ10
.WORD 40
\$LSW4: .WORD 1004
GP\$21:: .WORD 2130
.WORD SWQ11
.WORD 100
\$SW5: .WORD <<<<\$LSW5-\$SW5>*400>*4>*40>
\$SW6: .WORD <<<\$LSW6-\$SW6>*400>*4>
\$LSW5: .WORD 1004
GP\$22:: .WORD 3052
.WORD SWQ12
.WORD -1
.WORD 0
.WORD 25
\$SW7: .WORD <<<\$LSW7-\$SW7>*400>*4>
\$LSW6: .WORD 1004
GP\$23:: .WORD 6052
.WORD SWQ13
.WORD -1
.WORD 1
.WORD 20
GP\$24:: .WORD 7222
.WORD SWQ14
.WORD -1
.WORD 0
.WORD -1
.WORD 6
\$LSW7: .WORD 1004
L\$NDSFT: .BLKW 1

000000
000000 000000G
000002 000000G
000004 000000G
000006 000000G
000010 000000G
000012 000000G
000014 000000G
000016 000000G
000020 000000G
000022 000000G
000024 000000G
000026 000000G
000030 000000G
000032 000000G

.PSECT \$OWN\$, D
TBL.SUC: .WORD NULL
.WORD SC.SDI
.WORD SC.CON
.WORD NULL
.WORD SC.DUP
.WORD NULL
.WORD NULL
.WORD NULL
.WORD SC.ONL
.WORD NULL
.WORD NULL
.WORD NULL
.WORD NULL
.WORD NULL
.WORD NULL

ZRQAM2
V01.2

RD/RX EXERCISER
REPORT CODING SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (40)

000034	000000G		.WORD	NULL
000036	000000G		.WORD	NULL
000040	000000G		.WORD	SC.SON
000042	000000G	TBL.OFL:	.WORD	SC.UNK
000044	000000G		.WORD	SC.VOL
000046	000000G		.WORD	SC.IOP
000050	000000G		.WORD	NULL
000052	000000G		.WORD	SC.DUP
000054	000000G		.WORD	NULL
000056	000000G		.WORD	NULL
000060	000000G		.WORD	NULL
000062	000000G		.WORD	SC.DIS
000064	000000G	TBL.MFE:	.WORD	SC.FER
000066	000000G		.WORD	NULL
000070	000000G		.WORD	SC.ISH
000072	000000G		.WORD	SC.DST
000074	000000G		.WORD	SC.EC9
000076	000000G		.WORD	SC.576
000100	000000G		.WORD	SC.FCT
000102	000000G		.WORD	SC.ECC
000104	000000G		.WORD	SC.RCT
000106	000000G		.WORD	SC.FUL
000110	000000G		.WORD	SC.EC1
000112	000000G	TBL.WPT:	.WORD	NULL
000114	000000G		.WORD	SC.SWP
000116	000000G		.WORD	SC.HWP
000120	000000G	TBL.DAT:	.WORD	SC.FE2
000122	000000G		.WORD	NULL
000124	000000G		.WORD	SC.IS2
000126	000000G		.WORD	SC.DS2
000130	000000G		.WORD	SC.EC9
000132	000000G		.WORD	NULL
000134	000000G		.WORD	NULL
000136	000000G		.WORD	SC.ECD
000140	000000G		.WORD	SC.EC1
000142	000000G		.WORD	SC.EC2
000144	000000G		.WORD	SC.EC3
000146	000000G		.WORD	SC.EC4
000150	000000G		.WORD	SC.EC5
000152	000000G		.WORD	SC.EC6
000154	000000G		.WORD	SC.EC7
000156	000000G		.WORD	SC.EC8
000160	000000G	TBL.HST:	.WORD	NULL
000162	000000G		.WORD	SC.ODA
000164	000000G		.WORD	SC.ODB
000166	000000G		.WORD	SC.NXM
000170	000000G		.WORD	SC.PAR
000172	000000G	TBL.CNT:	.WORD	SC.CTO
000174	000000G		.WORD	SC.SDS
000176	000000G		.WORD	SC.EDC
000200	000000G		.WORD	SC.IDS
000202	000000G	TBL.DRV:	.WORD	NULL
000204	000000G		.WORD	SC.SRT

ZRQAM2
V01.2RD/RX EXERCISER
REPORT CODING SECTION5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:([DIETZ.RDRX])ZRQACO.BL1;82 (40)SEQ 0146
Page 146000206 000000G
000210 000000G
000212 000000G
000214 000000G
000216 000000G
000220 000000G
000222 000000G.WORD SC.SRI
.WORD SC.POE
.WORD SC.RDY
.WORD SC.CLK
.WORD SC.RSP
.WORD SC.SUR
.WORD SC.PSP.GLOBL CST, CST.ADDR, DCT, DCT.ADDR, RDRX.ADDR
.GLOBL IRDRX.ADDR, ICOM.ADDR, ICST.ADDR
.GLOBL IDCT.ADDR, RDM.CNT, RANDOM, TRK.SGN
.GLOBL DUPPKT, BST, TALLY, T.ADDR, C.ERR.TBL
.GLOBL MSCP.PKT, IPKT.ADDR, PKT.USE, RETPKT
.GLOBL RP.USE, RP.INDX, RP.ADDR, ELOG.PKT
.GLOBL BUFF.ADDR, BUFF.OWN, IODQ, IODQ.IN
.GLOBL IODQ.OUT, ENTRY.REASON, EOP.FLAG
.GLOBL DUP.FLAGS, CCTLR, CDISK, CUOFF
.GLOBL CTLR.CNT, DUR, QIO, FREE.MEM.ADDR
.GLOBL BYTS.PER.QIO, ST.CODE, SB.CODE
.GLOBL STEP, OF.RC, SA.REG, CMD.TIME
.GLOBL NEX, CRN.LOW, CRN.HIGH, P.INDEX
.GLOBL S.DUPPKT, S.PATTERN, CREDIT.BAL
.GLOBL INIT.OCCURED, NXT.PKT.USE, DBM5
.GLOBL DBM107, DU.MSG, DU.RSN, ERR.COD
.GLOBL ELG.FMT, HWQ1, HWQ2, HWQ3, HWQ4
.GLOBL HWQ6, HWQ7, HWQ8, HWQ9, HWQ10
.GLOBL HWQ11, SWQ1, SWQ2, SWQ4, SWQ7
.GLOBL SWQ9, SWQ10, SWQ11, SWQ12, SWQ13
.GLOBL SWQ14, SWQ15, SWQ17, SWQ19, SWQ22
.GLOBL SWM1, NULL, MSG.01, MSG.02, MSG.03
.GLOBL RPT1, RPT2, RPT3, RPT4, RPT5, RPT6
.GLOBL RPT7, RPT8, RPT9, RPT10, RPT11
.GLOBL RPT12, RPT13, RPT14, RPT15, RPT16
.GLOBL RPT17, RPT18, RPT19, EGS.01, EGS.02
.GLOBL EGD.10, EGD.11, EGD.12, EGD.13
.GLOBL EGD.14, EGD.15, EGD.16, EGD.17
.GLOBL EGD.18, EGD.19, EGD.20, EGD.21
.GLOBL EGD.22, EGD.23, EGD.24, EGM.30
.GLOBL EH.0, EH.1, EH.2, EH.3, EH.4, EH.5
.GLOBL EH.6, EH.7, EH.8, EH.9, EH.10
.GLOBL EH.12, EH.13, EBS.01, EBD.10, EBD.12
.GLOBL EBD.13, EBD.14, EBD.18, EBD.19
.GLOBL ELG.00, EX.BDR, EX.BDW, EX.LBR
.GLOBL EX.LBW, EX.RBN, EX.CBR, EX.CBW
.GLOBL XX13, XX14, XX15, XX16, XX17, XX18
.GLOBL XX19, XX20, XX21, XX22, XX23, XX24
.GLOBL XX25, XX26, XX27, XX29, XX30, XX31
.GLOBL XX32, XX33, XX34, XX35, XX37, XX38
.GLOBL XX39, XX40, XX41, XX42, EB.DCT
.GLOBL EB.COMM, EB.PKT, EB.RAL, EB.ADDR
.GLOBL EB.NEX1, EB.NEX2, EB.NEX3, EB.SA
.GLOBL CER.01, CER.02, EX.SEQ, EX.CRD

```

.GLOBL EX.MTN, EX.DGM, EX.RD, EX.WRT
.GLOBL EX.ACC, EX.ONL, EX.SCC, EX.GDS
.GLOBL EX.ESP, EX.ELP, EX.SDD, EX.RCD
.GLOBL EX.ABP, SC.SDI, SC.CON, SC.DUP
.GLOBL SC.ONL, SC.SON, SC.UNK, SC.VOL
.GLOBL SC.IOP, SC.DIS, SC.FER, SC.FE2
.GLOBL SC.ISH, SC.IS2, SC.DST, SC.DS2
.GLOBL SC.ECC, SC.ECD, SC.RCT, SC.FUL
.GLOBL SC.576, SC.FCT, SC.EC1, SC.EC2
.GLOBL SC.EC3, SC.EC4, SC.EC5, SC.EC6
.GLOBL SC.EC7, SC.EC8, SC.EC9, SC.SWP
.GLOBL SC.HWP, SC.ODA, SC.OOB, SC.NXM
.GLOBL SC.PAR, SC.CTO, SC.SDS, SC.EDC
.GLOBL SC.IDS, SC.SRT, SC.SRI, SC.POE
.GLOBL SC.RDY, SC.CLK, SC.RSP, SC.SUR
.GLOBL SC.PSP, F.1, F.2, F.3, F.4, F.5
.GLOBL F.6, F.7, F.8, F.9, F.10, F.11
.GLOBL F.12, F.13, F.14, F.15, F.16, F.17
.GLOBL F.18, F.19, F.20, F.21, EBH.30
.GLOBL EBH.44, EBH.45, EBH.46, EBH.47
.GLOBL EBH.48, EBH.49, DF.0, DF.1, DF.2
.GLOBL DF.3, DF.4, DF.5, DF.6, DF.7, T.QUE
.GLOBL T.DEF, T.INF, T.TER, T.FAT, T.SPL
.GLOBL E.UNT, E.BLK, E.DEV, E.ZER, M.ASC
.GLOBL M.BIN, M.TER, M.COD, M.DAT, M.UR
.GLOBL M.URP, M.UP, M.UL, CNTR.ERR, RDRX.ERR
.GLOBL EX.WRD, EX.OP, SPACE4, CRLF, DASH
.GLOBL ASTERISK, SWP.FLAGS, L$HMEM, L$LUN
.GLOBL L$UNIT

```

```

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001
001000
000400
000200
000100
000040
000020

```

```

BIT15-- -100000
BIT14-- 40000
BIT13-- 20000
BIT12-- 10000
BIT11-- 4000
BIT10-- 2000
BIT09-- 1000
BIT08-- 400
BIT07-- 200
BIT06-- 100
BIT05-- 40
BIT04-- 20
BIT03-- 10
BIT02-- 4
BIT01-- 2
BIT00-- 1
BIT9-- 1000
BIT8-- 400
BIT7-- 200
BIT6-- 100
BIT5-- 40
BIT4-- 20

```

ZRQAM2
V01.2

RD/RX EXERCISER
REPORT CODING SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0148
Page 148
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (40)

000010	BIT3==	10
000004	BIT2==	4
000002	BIT1==	2
000001	BIT0==	1
000040	EF.START==	40
000037	EF.RESTART==	37
000036	EF.CONTINUE==	36
000035	EF.NEW==	35
000034	EF.PWR==	34
000340	PRI07==	340
000300	PRI06==	300
000240	PRI05==	240
000200	PRI04==	200
000140	PRI03==	140
000100	PRI02==	100
000040	PRI01==	40
000000	PRI00==	0
000004	EVL==	4
000010	LOT==	10
000020	ADR==	20
000040	IDU==	40
000100	ISR==	100
000200	UAM==	200
000400	BOE==	400
001000	PNT==	1000
002000	PRI==	2000
004000	IXE==	4000
010000	IBE==	10000
020000	IER==	20000
040000	LOE==	40000
100000	HOE==	-100000
000034'	L\$HARD==	L\$HRDLN+2
000172'	L\$SOFT==	L\$SFTLN+2

000412

.SBTTL LRPT REPORT CODING SECTION
.PSECT \$CODE\$, RO

000000	004137	000000G	LRPT:	JSR	R1,\$SAVE4	:	5409
000004	012746	000000G		MOV	#RPT1,-(SP)	:	5419
000010	012746	000001		MOV	#1,-(SP)		
000014	010600			MOV	SP,R0	: SP,*	
000016	104416			TRAP	16		
000020	012716	000000G		MOV	#RPT2,(SP)	:	5420
000024	012746	000001		MOV	#1,-(SP)		
000030	010600			MOV	SP,R0	: SP,*	
000032	104416			TRAP	16		
000034	012716	000000G		MOV	#RPT3,(SP)	:	5421
000040	012746	000001		MOV	#1,-(SP)		
000044	010600			MOV	SP,R0	: SP,*	
000046	104416			TRAP	16		
000050	012716	000000G		MOV	#RPT4,(SP)	:	5422
000054	012746	000001		MOV	#1,-(SP)		

ZRQAM2	RD/RX EXERCISER		5-Dec-1983 10:27:14	VAX-11 Blues-16 V3-555	SEQ 0149
V01.2	REPORT CODING SECTION		5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (40)	Page 149
000060	010600		MOV	SP,R0	; SP,*
000062	104416		TRAP	16	
000064	012716	000000G	MOV	#RPT5,(SP)	
000070	012746	000001	MOV	#1,-(SP)	
000074	010600		MOV	SP,R0	; SP,*
000076	104416		TRAP	16	
000100	012716	000000G	MOV	#RPT6,(SP)	
000104	012746	000001	MOV	#1,-(SP)	
000110	010600		MOV	SP,R0	; SP,*
000112	104416		TRAP	16	
000114	005002		CLR	R2	; CTLR
000116	010216		MOV	R2,(SP)	; CTLR,*
000120	004737	000000V	JSR	PC,SET.CPAR	
000124	012703	000003	MOV	#3,R3	; *,DISK
000130	010316		MOV	R3,(SP)	; DISK,*
000132	004737	000000V	JSR	PC,SET.UPAR	
000136	010301		MOV	R3,R1	; DISK,*
000140	006301		ASL	R1	
000142	010100		MOV	R1,R0	
000144	063700	000000G	ADD	CST.ADDR,R0	
000150	132710	000034	BITB	#34,(R0)	
000154	001020		BNE	3#	
000156	032710	040000	BIT	#40000,(R0)	
000162	001415		BEQ	3#	
000164	111016		MOVB	(R0),(SP)	
000166	042716	177774	BIC	#177774,(SP)	
000172	013746	000000G	MOV	L#LUN,-(SP)	
000176	012746	000000G	MOV	#RPT7,-(SP)	
000202	012746	000003	MOV	#3,-(SP)	
000206	010600		MOV	SP,R0	; SP,*
000210	104416		TRAP	16	
000212	062706	000006	ADD	#6,SP	
000216	010100		MOV	R1,R0	
000220	063700	000000G	ADD	CST.ADDR,R0	
000224	111004		MOVB	(R0),R4	
000226	042704	177743	BIC	#177743,R4	
000232	020427	000004	CMP	R4,#4	
000236	001020		BNE	4#	
000240	032710	040000	BIT	#40000,(R0)	
000244	001415		BEQ	4#	
000246	111016		MOVB	(R0),(SP)	
000250	042716	177774	BIC	#177774,(SP)	
000254	013746	000000G	MOV	L#LUN,-(SP)	
000260	012746	000000G	MOV	#RPT8,-(SP)	
000264	012746	000003	MOV	#3,-(SP)	
000270	010600		MOV	SP,R0	; SP,*
000272	104416		TRAP	16	
000274	062706	000006	ADD	#6,SP	
000300	010100		MOV	R1,R0	
000302	063700	000000G	ADD	CST.ADDR,R0	
000306	111004		MOVB	(R0),R4	
000310	042704	177743	BIC	#177743,R4	
000314	020427	000010	CMP	R4,#10	

H12

ZRQAM2 V01.2	RD/RX EXERCISER REPORT CODING SECTION	5-Dec-1983 10:27:14 5-Dec-1983 10:27:04	VAX-11 Bliss-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (40)	SEQ 0150 Page 150		
000320	001020		BNE	5\$		
000322	032710	040000	BIT	#40000,(R0)	:	5445
000326	001415		BEQ	5\$		
000330	111016		MOVB	(R0),(SP)	:	5447
000332	042716	177774	BIC	#177774,(SP)		
000336	013746	000000G	MOV	L\$LUN,-(SP)		
000342	012746	000000G	MOV	#RPT17,-(SP)		
000346	012746	000003	MOV	#3,-(SP)		
000352	010600		MOV	SP,R0	: SP,*	
000354	104416		TRAP	16		
000356	062706	000006	ADD	#6,SP		
000362	010100		MOV	R1,R0	:	5449
000364	063700	000000G	ADD	CST.ADDR,R0		
000370	111004		MOVB	(R0),R4		
000372	042704	177743	BIC	#177743,R4		
000376	020427	000010	CMP	R4,#10		
000402	001431		BEQ	6\$		
000404	111004		MOVB	(R0),R4	:	5450
000406	042704	177743	BIC	#177743,R4		
000412	020427	000004	CMP	R4,#4		
000416	001423		BEQ	6\$		
000420	132710	000034	BITB	#34,(R0)	:	5451
000424	001420		BEQ	6\$		
000426	032710	040000	BIT	#40000,(R0)	:	5452
000432	001415		BEQ	6\$		
000434	111016		MOVB	(R0),(SP)	:	5454
000436	042716	177774	BIC	#177774,(SP)		
000442	013746	000000G	MOV	L\$LUN,-(SP)		
000446	012746	000000G	MOV	#RPT19,-(SP)		
000452	012746	000003	MOV	#3,-(SP)		
000456	010600		MOV	SP,R0	: SP,*	
000460	104416		TRAP	16		
000462	062706	000006	ADD	#6,SP		
000466	010100		MOV	R1,R0	:	5457
000470	063700	000000G	ADD	CST.ADDR,R0		
000474	032710	040000	BIT	#40000,(R0)		
000500	001506		BEQ	7\$		
000502	013700	000000G	MOV	T.ADDR,R0	:	5462
000506	016016	000032	MOV	32(R0),(SP)		
000512	016046	000034	MOV	34(R0),-(SP)		
000516	016046	000036	MOV	36(R0),-(SP)		
000522	016046	000016	MOV	16(R0),-(SP)		
000526	016046	000020	MOV	20(R0),-(SP)		
000532	012746	000000G	MOV	#RPT9,-(SP)		
000536	012746	000006	MOV	#6,-(SP)		
000542	010600		MOV	SP,R0	: SP,*	
000544	104416		TRAP	16		
000546	013700	000000G	MOV	T.ADDR,R0	:	5465
000552	016016	000040	MOV	40(R0),(SP)		
000556	016046	000042	MOV	42(R0),-(SP)		
000562	016046	000044	MOV	44(R0),-(SP)		
000566	016046	000024	MOV	24(R0),-(SP)		
000572	016046	000026	MOV	26(R0),-(SP)		

ZRQAM2
V01.2

RD/RX EXERCISER
REPORT CODING SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0151
Page 151
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (40)

000576	012746	000000G		MOV	#RPT9,-(SP)		
000602	012746	000006		MOV	#6,-(SP)		
000606	010600			MOV	SP,R0	; SP,*	
000610	104416			TRAP	16		
000612	013700	000000G		MOV	T.ADDR,R0	:	5468
000616	005016			CLR	(SP)		
000620	116016	000067		MOVB	67(R0),(SP)		
000624	005046			CLR	-(SP)		
000626	116016	000066		MOVB	66(R0),(SP)		
000632	005046			CLR	-(SP)		
000634	116016	000065		MOVB	65(R0),(SP)		
000640	005046			CLR	-(SP)		
000642	116016	000064		MOVB	64(R0),(SP)		
000646	005046			CLR	-(SP)		
000650	116016	000063		MOVB	63(R0),(SP)		
000654	005046			CLR	-(SP)		
000656	116016	000062		MOVB	62(R0),(SP)		
000662	005046			CLR	-(SP)		
000664	116016	000061		MOVB	61(R0),(SP)		
000670	005046			CLR	-(SP)		
000672	116016	000060		MOVB	60(R0),(SP)		
000676	012746	000000G		MOV	#RPT10,-(SP)		
000702	012746	000011		MOV	#11,-(SP)		
000706	010600			MOV	SP,R0	; SP,*	
000710	104416			TRAP	16		
000712	062706	000052		ADD	#52,SP	:	5459
000716	062703	000007	7#:	ADD	#7,R3	; *,DISK	5430
000722	020327	000030		CMP	R3,#30	; DISK,*	
000726	003002			BGT	8#		
000730	000137	000542'		JMP	2#		
000734	010216		8#:	MOV	R2,(SP)	; CTRL,*	5473
000736	012746	000076		MOV	#76,-(SP)		
000742	004737	000000G		JSR	PC,BL#MUL		
000746	005726			TST	(SP)+		
000750	005760	000002G		TST	CST+2(R0)		
000754	100026			BPL	9#		
000756	012716	000000G		MOV	#RPT11,(SP)	:	5476
000762	012746	000001		MOV	#1,-(SP)		
000766	010600			MOV	SP,R0	; SP,*	
000770	104416			TRAP	16		
000772	010200			MOV	R2,R0	; CTRL,*	5477
000774	006300			ASL	R0		
000776	005016			CLR	(SP)		
001000	116016	000001G		MOVB	C.ERR.TBL+1(R0),(SP)		
001004	005046			CLR	-(SP)		
001006	116016	000000G		MOVB	C.ERR.TBL(R0),(SP)		
001012	012746	000000G		MOV	#RPT12,-(SP)		
001016	012746	000003		MOV	#3,-(SP)		
001022	010600			MOV	SP,R0	; SP,*	
001024	104416			TRAP	16		
001026	062706	000010		ADD	#10,SP	:	5475
001032	012716	000000G	9#:	MOV	#CRLF,(SP)	:	5480
001036	012746	000001		MOV	#1,-(SP)		

ZRQAM2	RD/RX EXERCISER	REPORT CODING SECTION	5-Dec-1983 10:27:14	VAX-11 Bliss-16 V3-555	SEQ 0152
V01.2			5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (40)	Page 152
001042	010600		MOV SP,R0	; SP,*	
001044	104416		TRAP 16		
001046	005726		TST (SP)+		
001050	005202		INC R2	; CTLR	5427
001052	000243		.WORD CLV!CLC		5426
001054	003002		BGT 10\$		
001056	000137	000530'	JMP 1\$		
001062	012716	000000G	10\$: MOV #CRLF,(SP)		5484
001066	012746	000001	MOV #1,-(SP)		
001072	010600		MOV SP,R0	; SP,*	
001074	104416		TRAP 16		
001076	012716	000000G	MOV #RPT13,(SP)		5485
001102	012746	000001	MOV #1,-(SP)		
001106	010600		MOV SP,R0	; SP,*	
001110	104416		TRAP 16		
001112	012716	000000G	MOV #RPT14,(SP)		5486
001116	012746	000001	MOV #1,-(SP)		
001122	010600		MOV SP,R0	; SP,*	
001124	104416		TRAP 16		
001126	012716	000000G	MOV #RPT15,(SP)		5487
001132	012746	000001	MOV #1,-(SP)		
001136	010600		MOV SP,R0	; SP,*	
001140	104416		TRAP 16		
001142	005003		CLR R3	; CTLR	5488
001144	010316		11\$: MOV R3,(SP)	; CTLR,*	5490
001146	004737	000000V	JSR PC,SET.CPAR		
001152	012702	000003	MOV #3,R2	; *.DISK	5491
001156	010216		12\$: MOV R2,(SP)	; DISK,*	5493
001160	004737	000000V	JSR PC,SET.UPAR		
001164	010201		MOV R2,R1	; DISK,*	5494
001166	006301		ASL R1		
001170	010100		MOV R1,R0		
001172	063700	000000G	ADD CST.ADDR,R0		
001176	111004		MOVB (R0),R4		
001200	042704	177743	BIC #177743,R4		
001204	020427	000004	CMP R4,#4		
001210	001032		BNE 13\$		
001212	032710	040000	BIT #40000,(R0)		
001216	001427		BEQ 13\$		
001220	013704	000000G	MOV T.ADDR,R4		5498
001224	016416	000050	MOV 50(R4),(SP)		
001230	016446	000052	MOV 52(R4),-(SP)		
001234	016446	000054	MOV 54(R4),-(SP)		
001240	016446	000056	MOV 56(R4),-(SP)		
001244	111046		MOVB (R0),-(SP)		
001246	042716	177774	BIC #177774,(SP)		
001252	013746	000000G	MOV L#LUN,-(SP)		
001256	012746	000000G	MOV #RPT16,-(SP)		
001262	012746	000007	MOV #7,-(SP)		
001266	010600		MOV SP,R0	; SP,*	
001270	104416		TRAP 16		
001272	062706	000016	ADD #16,SP		

001276	010100		13#:	MOV	R1,R0			
001300	063700	000000G		ADD	CST,ADDR,R0	:		5500
001304	111001			MOVB	(R0),R1			
001306	042701	177743		BIC	#177743,R1			
001312	020127	000010		CMP	R1,#10			
001316	001032			BNE	14#			
001320	032710	040000		BIT	#40000,(R0)			
001324	001427			BEQ	14#			
001326	013701	000000G		MOV	T,ADDR,R1	:		5504
001332	016116	000050		MOV	50(R1),(SP)			
001336	016146	000052		MOV	52(R1),-(SP)			
001342	016146	000054		MOV	54(R1),-(SP)			
001346	016146	000056		MOV	56(R1),-(SP)			
001352	111046			MOVB	(R0),-(SP)			
001354	042716	177774		BIC	#177774,(SP)			
001360	013746	000000G		MOV	L#LUN,-(SP)			
001364	012746	000000G		MOV	#RPT18,-(SP)			
001370	012746	000007		MOV	#7,-(SP)			
001374	010600			MOV	SP,R0	:	SP,*	
001376	104416			TRAP	16			
001400	062706	000016		ADD	#16,SP			
001404	062702	000007	14#:	ADD	#7,R2	:	*,DISK	5491
001410	020227	000030		CMP	R2,#30	:	DISK,*	
001414	003660			BLE	12#			
001416	005203			INC	R3	:	CTLR	5488
001420	000243			.WORD	CLV!CLC			
001422	003650			BLE	11#			
001424	012716	000000G		MOV	#CRLF,(SP)	:		5509
001430	012746	000001		MOV	#1,-(SP)			
001434	010600			MOV	SP,R0	:	SP,*	
001436	104416			TRAP	16			
001440	062706	000030		ADD	#30,SP	:		5409
001444	000207			RTS	PC			

; Routine Size: 403 words, Routine Base: \$CODE\$ + 0412
 ; Maximum stack depth per invocation: 35 words

000000	004737	000412'		.SBTTL	L#RPT REPORT CODING SECTION			
000004	104425		L#RPT::	JSR	PC,LRPT	:		5509
000006	000207			TRAP	25			
				RTS	PC			

; Routine Size: 4 words, Routine Base: \$CODE\$ + 2060
 ; Maximum stack depth per invocation: 2 words

```

: 5512 #sbttl 'INITIALIZE SECTION'
: 5513
: 5514 BGNINIT;
: 5515
: 5516 local
: 5517     DELAY_MULT : word,
: 5518     FLAG : byte,
: 5519     TEMP : word,
: 5520     HWPT_REF : ref block [HWPT_LEN, word] field (HWP_FIELDS),
: 5521     CLEAR_TABLES : byte;
: 5522
: 5523 SETPRI (PRI07);           ! PRIORITY 7 - NO INTERRUPTS ALLOWED DURING INIT
: 5524
: 5525 if READEF (EF_NEW)      ! IS THIS A NEW PASS?
: 5526 then
: 5527     begin
: 5528     ENTRY_REASON = NEW_PASS;
: 5529
: 5530     if (.SWP_FLAGS and SWF_CST) neq SWF_CST
: 5531     then
: 5532         CLEAR_TABLES = FALSE
: 5533     else
: 5534         CLEAR_TABLES = TRUE;
: 5535
: 5536     end;
: 5537
: 5538 if READEF (EF_START)     ! IS THIS A START?
: 5539 then
: 5540     begin
: 5541     BRESET;
: 5542     ENTRY_REASON = START;
: 5543     CLEAR_TABLES = TRUE;
: 5544     INIT_OCCURED = FALSE;
: 5545     end;
: 5546
: 5547 if READEF (EF_RESTART)  ! IS THIS A RESTART?
: 5548 then
: 5549     begin
: 5550     ENTRY_REASON = RESTART;
: 5551     CLEAR_TABLES = TRUE;
: 5552     end;
: 5553
: 5554 if READEF (EF_CONTINUE) ! IS THIS A CONTINUE?
: 5555 then
: 5556     begin
: 5557     ENTRY_REASON = CONT;
: 5558
: 5559     if (.SWP_FLAGS and SWF_CST) neq SWF_CST
: 5560     then
: 5561         CLEAR_TABLES = FALSE
: 5562     else
: 5563         CLEAR_TABLES = TRUE;
: 5564

```

```

:      5565      end;
:      5566
:      5567      if READEF (EF_PWR)
:      5568      then
:      5569          begin
:      5570              ENTRY_REASON = PWR_FAIL;
:      5571              CLEAR_TABLES = TRUE;
:      5572              INIT_OCCURED = FALSE;
:      5573              PRINTF (MSG_01);
:      5574
:      5575              incr COUNT from 0 to 60 do
:      5576                  begin
:      5577                      DELAY_MULT = 333;
:      5578                      DELAY (.DELAY_MULT);
:      5579                      BREAK;
:      5580                  end;
:      5581
:      5582          end;
:      5583
:      5584      !SETVEC (O_TVEC, O_BRK, PRI07);
:      5585
:      5586      !+
:      5587      ! MAKE SURE THAT NOT MORE THAN MAX_UNITS HAVE BEEN SPECIFIED.
:      5588      ! IF THERE ARE TOO MANY, NOTIFY USER AND RETURN TO SUPERVISOR.
:      5589      ! (DIAGNOSTIC IS ABORTED).
:      5590      !-
:      5591
:      5592      if .L$UNIT gtru MAX_UNITS
:      5593      then
:      5594          begin
:      5595              ERRSF (1, EGS_01, EMS_01);
:      5596              DOCLN;
:      5597          end;
:      5598
:      5599      !+
:      5600      ! THE FOLLOWING CODE IS EXECUTED FOR ALL ENTRY REASONS EXCEPT NEW_PASS.
:      5601      ! ALL RUN-TIME CONTROLLER STATUS TABLES (CST@) ARE CLEARED TO 0, THEN
:      5602      ! LOADED WITH CONFIGURATION DATA FROM THE HARDWARE P-TABLES.
:      5603      !-
:      5604
:      5605      if (.ENTRY_REASON neq NEW_PASS)
:      5606      then
:      5607          begin
:      5608
:      5609              incr COUNT from 0 to ((MAX_CTLR * CST_LEN * 2) - 2) by 2 do
:      5610                  (CST + .COUNT) = 0;
:      5611
:      5612              incr UNIT from 0 to (.L$UNIT - 1) do
:      5613                  ! LOOP THROUGH ALL UNITS
:      5614                  if GPHARD (.UNIT, HWPT_REF) neq 0
:      5615                  then
:      5616                      ! IF HWP TABLE FOUND
:      5617                      begin
:      5618                          FLAG = NOT_FOUND;

```



```

: 5671 CST [.CTLR, .TEMP, D_PRES] = PRESENT;
: 5672 CST [.CTLR, .TEMP + 1, D_BEGO] = .HWPT_REF [HWP_BEG_TRK];
: 5673 ! CST [.CTLR, .TEMP + 2, D_BEGO] = .HWPT_REF [HWP_BEG_TRK];
: 5674 ! CST [.CTLR, .TEMP + 3, D_ENDO] = .HWPT_REF [HWP_END_TRK];
: 5675 ! CST [.CTLR, .TEMP + 4, D_ENDO] = .HWPT_REF [HWP_END_TRK];
: 5676 ! CST [.CTLR, .TEMP + 5, D_DBN] = 0;
: 5677 ! CST [.CTLR, .TEMP + 5, NODUPMEDIA] = NOT(.HWPT_REF [HWP_DUPEX]);
: 5678 ! CST [.CTLR, .TEMP + 5, DUPWRITE] = (.HWPT_REF [HWP_DUPWT]);
: 5679 ! CST [.CTLR, .TEMP + 6, D_COUNT] = 0;
: 5680
: 5681 exitloop;
: 5682 end; ! IF EMPTY CST FOUND
: 5683
: 5684 if .FLAG eq1 NOT_FOUND ! IF NO EMPTY CST FOUND
: 5685 then
: 5686 begin
: 5687 PRINTF (CER_02, MAX_CTLR); ! "MORE THAN X DIFFERENT IP ADDRESSES."
: 5688 DUR [.UNIT] = DU_CONF; ! CONFIGURATION ERROR
: 5689 DODU (.UNIT); ! DROP UNIT
: 5690 end;
: 5691
: 5692 end; ! IF NO IP ADDR MATCH IN CST
: 5693
: 5694 end; ! IF GPHARD RETURNS A HWP TABLE
: 5695
: 5696 ! CONFIGURATON CHECK FOR LEGAL RDRX UNIT MIX BECAUSE WE HAVE DIFFERENT
: 5697 ! DRIVES : THE RDS1, AND RX50.
: 5698 ! (NEEDED?)
: 5699 !
: 5700 end; ! END OF "NON NEW_PASS" INITIALIZATION
: 5701
: 5702 if .ENTRY_REASON eq1 NEW_PASS
: 5703 then
: 5704 begin
: 5705
: 5706 incr UNIT from 0 to (.L$UNIT - 1) do
: 5707 GPHARD (.UNIT, HWPT_REF); ! DUMMY GPHARDs FOR NEW PASS
: 5708
: 5709 incr CTLR from 0 to (MAX_CTLR - 1) do
: 5710 incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5711 (CST [.CTLR, .OFFSET, D_STAT] = OFFLINE); ! START EACH UNIT AS OFFLINE
: 5712
: 5713
: 5714
: 5715 end;
: 5716
: 5717 if (.ENTRY_REASON eq1 START)
: 5718 then
: 5719 begin
: 5720 CTLR_CNT = 0; ! NUMBER OF CONFIGURED CONTROLLERS
: 5721 CRN_LOW = CRN_HIGH = 0; ! INITIALIZE COMMAND REF NUMBER
: 5722 incr CTLR from 0 to (MAX_CTLR - 1) do
: 5723

```


ZRQAM2
V01.2

RD/RX EXERCISER
INITIALIZE SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 B1:es-16 V3-555
DISK\$USER2:[DIETZ,RDRX]ZRQACO.BL1;82 (41)

SEQ 0158
Page 158

```

: 5724         if .CST [.CTLR, IP_ADDR] neq 0      ! IF CONTROLLER IS PRESENT
: 5725         then
: 5726             CTLR_CNT = .CTLR_CNT + 1;        ! INCREMENT CONTROLLER COUNT
: 5727
: 5728         MEMORY (FREE_MEM_ADDR);              ! GET START OF FREE MEMORY
: 5729
: 5730         end;                                ! END OF "START" INITIALIZATION
: 5731
: 5732         !+
: 5733         ! CLEAR STATISTICS TABLES
: 5734         !-
: 5735
: 5736         incr UNITS from 0 to MAX_UNITS - 1 do ! CLEAR CURRENT STATISTICS
: 5737             incr COUNT from 0 to TALLY_CLEAR - 1 do
: 5738                 TALLY [.UNITS * TALLY_LEN + .COUNT] = 0;
: 5739
: 5740         if .CLEAR_TABLES                      ! IF CLEAR TABLES ON EVERY PASS
: 5741         then
: 5742             incr UNITS from 0 to MAX_UNITS - 1 do
: 5743                 incr COUNT from TALLY_CLEAR to TALLY_LEN - 1 do      ! INITIALIZE TOTALS
: 5744                     TALLY [.UNITS * TALLY_LEN + .COUNT] = 0;      !
: 5745
: 5746         if .CLEAR_TABLES
: 5747         then
: 5748             incr CTLR from 0 to MAX_CTLR - 1 do
: 5749                 begin
: 5750                     C_ERR_TBL [.CTLR, C_ERR_HRD] = 0;      ! INITIALIZE CONTROLLER ERRORS
: 5751                     C_ERR_TBL [.CTLR, C_ERR_SFT] = 0;      !
: 5752                 end;
: 5753
: 5754         SETPRI (PRI00);                        ! SET PROGRAM PRIORITY TO 0
: 5755
: 5756         ENDINIT;

```

.GLOBL L\$DLY

000000	004137	000000G	LINIT:	.SBTTL	LINIT INITIALIZE SECTION		
000004	162706	000016		JSR	R1, \$SAVES	:	5511
000010	012700	000340		SUB	#16, SP	:	
000014	104441			MOV	#340, R0	:	5523
000016	012700	000035		TRAP	41	:	
000022	104447			MOV	#35, R0	:	5525
000024	103012			TRAP	47	:	
000026	112737	000005 000000G		BHIS	2#	:	
000034	105737	000000G		MOVB	#5, ENTRY.REASON	:	5528
000040	100402			TSTB	SMP.FLAGS	:	5530
000042	105016			BMI	1#	:	
000044	000402			CLRB	(SP)	:	CLEAR.TABLES
000046	112716	000001	1#:	BR	2#	:	5530
000052	012700	000040	2#:	MOVB	#1, (SP)	:	*.CLEAR.TABLES
				MOV	#40, R0	:	5538

ZRQAM2
V01.2

RD/RX EXERCISER
INITIALIZE SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0159
Page 159
VAX-11 Bliss-16 V3-555
DISK\$USER2:(DIETZ.RDRX)ZRQACO.BL1;82 (41)

000056	104447			TRAP	47			
000060	103010			BHIS	3#			
000062	104433			TRAP	33			
000064	112737	000001	000000G	MOVB	#1,ENTRY.REASON	:		5540
000072	112716	000001		MOVB	#1,(SP)	:	*,CLEAR.TABLES	5542
000076	105037	000000G		CLRB	INIT.OCCURED	:		5543
000102	012700	000037		MOV	#37,R0	:		5544
000106	104447			TRAP	47	:		5547
000110	103005			BHIS	4#			
000112	112737	000002	000000G	MOVB	#2,ENTRY.REASON	:		5550
000120	112716	000001		MOVB	#1,(SP)	:	*,CLEAR.TABLES	5551
000124	012700	000036		MOV	#36,R0	:		5554
000130	104447			TRAP	47	:		
000132	103012			BHIS	6#			
000134	112737	000003	000000G	MOVB	#3,ENTRY.REASON	:		5557
000142	105737	000000G		TSTB	SWP.FLAGS	:		5559
000146	100402			BMI	5#			
000150	105016			CLRB	(SP)	:	CLEAR.TABLES	5561
000152	000402			BR	6#	:		5559
000154	112716	000001		MOVB	#1,(SP)	:	*,CLEAR.TABLES	5563
000160	012700	000034		MOV	#34,R0	:		5567
000164	104447			TRAP	47	:		
000166	103040			BHIS	12#			
000170	112737	000004	000000G	MOVB	#4,ENTRY.REASON	:		5570
000176	112716	000001		MOVB	#1,(SP)	:	*,CLEAR.TABLES	5571
000202	105037	000000G		CLRB	INIT.OCCURED	:		5572
000206	012746	000000G		MOV	#MSG.01,-(SP)	:		5573
000212	012746	000001		MOV	#1,-(SP)	:		
000216	010600			MOV	SP,R0	:	SP,*	
000220	104417			TRAP	17			
000222	012702	000075		MOV	#75,R2	:	*,COUNT	5575
000226	012703	000515		MOV	#515,R3	:	*,DELAY.MULT	5577
000232	010301			MOV	R3,R1	:	DELAY.MULT,##TMP2	5578
000234	001411			BEQ	11#			
000236	013700	000000G		MOV	L#DLY,R0	:	*,##TMP1	
000242	001404			BEQ	10#			
000244	005066	000020		CLR	20(SP)	:	##TMP	
000250	005300			DEC	R0	:	##TMP1	
000252	001374			BNE	9#			
000254	005301			DEC	R1	:	##TMP2	
000256	000766			BR	8#			
000260	104422			TRAP	22			
000262	005302			DEC	R2	:	COUNT	5575
000264	001360			BNE	7#			
000266	022626			CMP	(SP)+,(SP)+	:		5569
000270	023727	000000G	000004	CMP	L#UNIT,#4	:		5592
000276	101405			BLOS	13#			
000300	104454			TRAP	54	:		5595
000302	000001			.WORD	1			
000304	000000G			.WORD	EGS.01			
000306	000000V			.WORD	EMS.01			
000310	104444			TRAP	44			
000312	123727	000000G	000005	CMPB	ENTRY.REASON,#5	:		5605

ZRQAM2	RD/RX EXERCISER						
V01.2	INITIALIZE SECTION						
				5-Dec-1983 10:27:14	VAX-11 Bliss-16 V3-555	SEQ 0160	
				5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (41)	Page 160	
000320	001002			BNE	14\$		
000322	000137	003624'		JMP	35\$		
000326	005000		14\$:	CLR	R0	; COUNT	5609
000330	005060	000000G	15\$:	CLR	CST(R0)	; *(COUNT)	5610
000334	062700	000002		ADD	#2,R0	; *,COUNT	5609
000340	020027	000074		CMP	R0,#74	; COUNT,*	
000344	003771			BLE	15\$		
000346	013766	000000G 000012		MOV	L\$UNIT,12(SP)		5612
000354	005005			CLR	R5	; UNIT	
000356	000137	003602'		JMP	33\$		
000362	010500		16\$:	MOV	R5,R0	; UNIT,*	5614
000364	104442			TRAP	42		
000366	010066	000010		MOV	R0,10(SP)	; *,HWPT.REF	
000372	001002			BNE	17\$		
000374	000137	003600'		JMP	32\$		
000400	105004		17\$:	CLRB	R4	; FLAG	5617
000402	005066	000002		CLR	2(SP)	; CTLR	5619
000406	016646	000002	18\$:	MOV	2(SP),-(SP)	; CTLR,*	5621
000412	012746	000076		MOV	#76, -(SP)		
000416	004737	000000G		JSR	PC,BL\$MUL		
000422	022626			CMP	(SP)+,(SP)+		
000424	026076	000000G 000010		CMP	CST(R0),#10(SP)	; *,HWPT.REF	
000432	001174			BNE	23\$		
000434	012766	000001 000004		MOV	#1,4(SP)		5641
000442	112704	000001		MOVB	#1,R4	; *,FLAG	
000446	012700	000006		MOV	#6,R0		5624
000452	066600	000010		ADD	10(SP),R0	; HWPT.REF,*	
000456	010066	000006		MOV	R0,6(SP)		
000462	111046			MOVB	(R0),-(SP)		
000464	042716	177774		BIC	#177774,(SP)		
000470	012746	000007		MOV	#7, -(SP)		
000474	004737	000000G		JSR	PC,BL\$MUL		
000500	010003			MOV	R0,R3		
000502	005726			TST	(SP)+		
000504	016616	000004		MOV	4(SP),(SP)	; CTLR,*	
000510	012746	000037		MOV	#37, -(SP)		
000514	004737	000000G		JSR	PC,BL\$MUL		
000520	010002			MOV	R0,R2		
000522	022626			CMP	(SP)+,(SP)+		
000524	060300			ADD	R3,R0		
000526	006300			ASL	R0		
000530	032760	040000 000006G		BIT	#40000,CST+6(R0)		
000536	001106			BNE	22\$		
000540	010301			MOV	R3,R1	; *,TEMP	5627
000542	062701	000003		ADD	#3,R1	; *,TEMP	
000546	010200			MOV	R2,R0		5628
000550	060100			ADD	R1,R0	; TEMP,*	
000552	006300			ASL	R0		
000554	062700	000000G		ADD	#CST,R0		
000560	017610	000006		MOV	#6(SP),(R0)		
000564	010503			MOV	R5,R3	; UNIT,*	5629
000566	000303			SWAB	R3		
000570	042703	170377		BIC	#170377,R3		

ZRQAM2
V01.2 RD/RX EXERCISER
INITIALIZE SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0161
Page 161
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (41)

000574	042710	007400		BIC	#7400,(R0)		
000600	050310			BIS	R3,(R0)		
000602	042710	010000		BIC	#10000,(R0)	:	5630
000606	052710	040000		BIS	#40000,(R0)	:	5631
000612	010200			MOV	R2,R0	:	5632
000614	060100			ADD	R1,R0	: TEMP,*	
000616	006300			ASL	R0		
000620	016603	000010		MOV	10(SP),R3	: HWPT.REF,*	
000624	016360	000010	000002G	MOV	10(R3),CST+2(R0)	: *(HWPT.REF),*	
000632	010200			MOV	R2,R0	:	5634
000634	060100			ADD	R1,R0	: TEMP,*	
000636	006300			ASL	R0		
000640	016360	000012	000006G	MOV	12(R3),CST+6(R0)	: *(HWPT.REF),*	
000646	010200			MOV	R2,R0	:	5636
000650	060100			ADD	R1,R0	: TEMP,*	
000652	006300			ASL	R0		
000654	062700	000012G		ADD	#CST+12,R0		
000660	105010			CLRB	(R0)		
000662	005003			CLR	R3	:	5637
000664	132776	000010	000006	BITB	#10,#6(SP)		
000672	001401			BEQ	19#		
000674	005203			INC	R3		
000676	005103		19#:	COM	R3		
000700	042710	100000		BIC	#100000,(R0)		
000704	032703	000001		BIT	#1,R3		
000710	001402			BEQ	20#		
000712	052710	100000		BIS	#100000,(R0)		
000716	117603	000006	20#:	MOVB	#6(SP),R3	:	5638
000722	042710	010000		BIC	#10000,(R0)		
000726	032703	000020		BIT	#20,R3		
000732	001402			BEQ	21#		
000734	052710	010000		BIS	#10000,(R0)		
000740	010200		21#:	MOV	R2,R0	:	5639
000742	060100			ADD	R1,R0	: TEMP,*	
000744	006300			ASL	R0		
000746	005060	000014G		CLR	CST+14(R0)		
000752	000432			BR	24#	:	5626
000754	017646	000010	22#:	MOV	#10(SP),-(SP)	: HWPT.REF,*	5646
000760	117646	000010		MOVB	#10(SP),-(SP)		
000764	042716	177774		BIC	#177774,(SP)		
000770	012746	000000G		MOV	#CER.01, -(SP)		
000774	012746	000003		MOV	#3, -(SP)		
001000	010600			MOV	SP,R0	: SP,*	
001002	104417			TRAP	17		
001004	062706	000010		ADD	#10,SP		
001010	112765	000001	000000G	MOVB	#1,DUR(R5)	: *,*(UNIT)	5648
001016	010500			MOV	R5,R0	: UNIT,*	5649
001020	104451			TRAP	51		
001022	000406			BR	24#	:	5645
001024	005266	000002	23#:	INC	2(SP)	: CTLR	5619
001030	000243			.WORD	CLV:CLC		
001032	003002			BGT	24#		

ZRQAM2	RD/RX EXERCISER	5-Dec-1983 10:27:14	VAX-11 Blues-16 V3-555	
V01.2	INITIALIZE SECTION	5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (41)	
001034	000137 002476'			
001040	105704	24\$:	JMP 18\$	
001042	001402		TSTB R4	; FLAG 5654
001044	000137 003600'		BEQ 25\$	
001050	005002	25\$:	JMP 32\$	
001052	010246	26\$:	CLR R2	; CTLR 5658
001054	012746 000076		MOV R2, -(SP)	; CTLR,* 5660
001060	004737 000000G		MOV #76, -(SP)	
001064	022626		JSR PC, BL\$MUL	
001066	005760 000000G		CMP (SP)+, (SP)+	
001072	001162		TST CST(R0)	
001074	112704 000001		BNE 30\$	
001100	017660 000010 000000G		MOVB #1, R4	; *,FLAG 5663
001106	016646 000010		MOV #10(SP), CST(R0)	; HWPT.REF,* 5664
001112	062716 000002		MOV 10(SP), -(SP)	; HWPT.REF,* 5665
001116	013603		ADD #2, (SP)	
001120	042703 177000		MOV #8(SP)+, R3	
001124	042760 000777 000002G		BIC #177000, R3	
001132	050360 000002G		BIC #777, CST+2(R0)	
001136	016603 000010		BIS R3, CST+2(R0)	
001142	116360 000004 000004G		MOV 10(SP), R3	; HWPT.REF,* 5666
001150	012700 000006		MOVB 4(R3), CST+4(R0)	; *(HWPT.REF),* 5667
001154	060300		MOV #6, R0	
001156	010066 000006		ADD R3, R0	; HWPT.REF,* 5668
001162	111046		MOV R0, 6(SP)	
001164	042716 177774		MOVB (R0), -(SP)	
001170	012746 000007		BIC #177774, (SP)	
001174	004737 000000G		MOV #7, -(SP)	
001200	005726		JSR PC, BL\$MUL	
001202	010001		TST (SP)+	
001204	062701 000003		MOV R0, R1	; *,TEMP 5668
001210	010216		ADD #3, R1	; *,TEMP
001212	012746 000037		MOV R2, (SP)	; CTLR,* 5668
001215	004737 000000G		MOV #37, -(SP)	
001222	010003		JSR PC, BL\$MUL	
001224	005726		MOV R0, R3	
001226	060100		TST (SP)+	
001230	006300		ADD R1, R0	; TEMP,* 5669
001232	062700 000000G		ASL R0	
001236	017610 000010		ADD #CST, R0	
001242	010516		MOV #10(SP), (R0)	
001244	000316		MOV R5, (SP)	; UNIT,* 5669
001246	042716 170377		SWAB (SP)	
001252	042710 007400		BIC #170377, (SP)	
001256	052610		BIC #7400, (R0)	
001260	042710 010000		BIS (SP)+, (R0)	
001264	052710 040000		BIC #10000, (R0)	; 5670
001270	010300		BIS #40000, (R0)	; 5671
001272	060100		MOV R3, R0	; 5672
001274	006300		ADD R1, R0	; TEMP,*
001276	016646 000010		ASL R0	
001302	062716 000010		MOV 10(SP), -(SP)	; HWPT.REF,*
001306	013660 000002G		ADD #10, (SP)	
			MOV #8(SP)+, CST+2(R0)	

H13

ZRQAM2 V01.2	RD/RX EXERCISER INITIALIZE SECTION	5-Dec-1983 10:27:14 5-Dec-1983 10:27:04	VAX-11 B11es-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (41)	SEQ 0163 Page 163
001312	010300		MOV R3,R0	5674
001314	060100		ADD R1,R0	; TEMP,*
001316	006300		ASL R0	; HWPT.REF,*
001320	016646	000010	MOV 10(SP),-(SP)	
001324	062716	000012	ADD #12,(SP)	
001330	013660	000006G	MOV @ (SP)+,CST+6(R0)	
001334	010300		MOV R3,R0	5676
001336	060100		ADD R1,R0	; TEMP,*
001340	006300		ASL R0	
001342	062700	000012G	ADD #CST+12,R0	
001346	105010		CLRB (R0)	
001350	005046		CLR -(SP)	5677
001352	131776	000010	BITB (PC),@10(SP)	
001356	001401		BEQ 27#	
001360	005216		INC (SP)	
001362	005116		COM (SP)	
001364	011646		MOV (SP),-(SP)	
001366	042710	100000	BIC #100000,(R0)	
001372	006026		ROR (SP)+	
001374	1030u2		BCC 28#	
001376	052710	100000	BIS #100000,(R0)	
001402	117616	000010	MOVB @10(SP),(SP)	5678
001406	042710	010000	BIC #10000,(R0)	
001412	032726	000020	BIT #20,(SP)+	
001416	001402		BEQ 29#	
001420	052710	010000	BIS #10000,(R0)	
001424	010300		MOV R3,R0	5679
001426	060100		ADD R1,R0	; TEMP,*
001430	006300		ASL R0	
001432	005060	000014G	CLR CST+14(R0)	
001436	000403		BR 31#	5662
001440	005202		INC R2	5658
001442	000243		.WORD CLV!CLC	
001444	003602		BLE 26#	
001446	105704		TSTB R4	; FLAG
001450	001017		BNE 32#	5684
001452	012746	000001	MOV #1,-(SP)	5687
001456	012746	000000G	MOV #CER.02,-(SP)	
001462	012746	000002	MOV #2,-(SP)	
001466	010600		MOV SP,R0	; SP,*
001470	104417		TRAP 17	
001472	112765	000001 000000G	MOVB #1,DUR(R5)	; *,*(UNIT)
001500	010500		MOV R5,R0	5688
001502	104451		TRAP 51	; UNIT,*
001504	062706	000006	ADD #6,SP	5686
001510	005205		INC R5	; UNIT
001512	020566	000012	32#: CMP R5,12(SP)	5612
001516	002002		33#: BGE 34#	; UNIT,*
001520	000137	002452'	JMP 16#	
001524	123727	000000G 000005	34#: CMPB ENTRY.REASON,#5	5702
001532	001035		BNE 40#	
001534	013701	000000G	35#: MOV L#UNIT,R1	5706

ZRQAM2	RD/RX EXERCISER								
V01.2	INITIALIZE SECTION								
001540	005002								
001542	000405								
001544	010200			36#:	MOV	R2,R0			
001546	104442				TRAP	42			5707
001550	010066	000010			MOV	R0,10(SP)			
001554	005202				INC	R2			
001556	020201			37#:	CMP	R2,R1			5706
001560	002771				BLT	36#			
001562	005003				CLR	R3			
001564	012701	000003		38#:	MOV	#3,R1			5709
001570	010300			39#:	MOV	R3,R0			5710
001572	060100				ADD	R1,R0			5711
001574	006300				ASL	R0			
001576	042760	020000	000000G		BIC	#20000,CST(R0)			
001604	062701	000007			ADD	#7,R1			5710
001610	020127	000030			CMP	R1,#30			
001614	003765				BLE	39#			
001616	062703	000037			ADD	#37,R3			5709
001622	000243				.WORD	CLV!CLC			
001624	003757				BLE	38#			
001626	123727	000000G	000001	40#:	CMPB	ENTRY.REASON,#1			5717
001634	001023				BNE	43#			
001636	005037	000000G			CLR	CTLR.CNT			5720
001642	005037	000000G			CLR	CRN.HIGH			5721
001646	005037	000000G			CLR	CRN.LOW			
001652	005000				CLR	R0			5722
001654	005760	000000G		41#:	TST	CST(R0)			5724
001660	001402				BEQ	42#			
001662	005237	000000G			INC	CTLR.CNT			5726
001666	062700	000076		42#:	ADD	#76,R0			5722
001672	000243				.WORD	CLV!CLC			
001674	003767				BLE	41#			
001676	104431				TRAP	31			5728
001700	010037	000000G			MOV	R0,FREE.MEM.ADDR			
001704	005001			43#:	CLR	R1			5736
001706	005003			44#:	CLR	R3			5737
001710	010300			45#:	MOV	R3,R0			5738
001712	060100				ADD	R1,R0			
001714	006300				ASL	R0			
001716	005060	000000G			CLR	TALLY(R0)			
001722	005203				INC	R3			5737
001724	020327	000006			CMP	R3,#6			
001730	003767				BLE	45#			
001732	062701	000034			ADD	#34,R1			5736
001736	020127	000124			CMP	R1,#124			
001742	003761				BLE	44#			
001744	032716	000001			BIT	#1,(SP)			5740
001750	001435				BEQ	49#			
001752	005001				CLR	R1			5742
001754	012703	000007		46#:	MOV	#7,R3			5743
001760	010300			47#:	MOV	R3,R0			5744

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (41)

SEQ 0164
Page 164

ZRQAM2 RD/RX EXERCISER
V01.2 INITIALIZE SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (41)

001762	060100		ADD	R1,R0	; UNITS,*	
001764	006300		ASL	R0		
001766	005060	000000G	CLR	TALLY(R0)		
001772	005203		INC	R3	; COUNT	5743
001774	020327	000033	CMP	R3,#33	; COUNT,*	
002000	003767		BLE	47#		
002002	062701	000034	ADD	#34,R1	; *,UNITS	5742
002006	020127	000124	CMP	R1,#124	; UNITS,*	
002012	003760		BLE	46#		
002014	032716	000001	BIT	#1,(SP)	; *,CLEAR.TABLES	5746
002020	001411		BEQ	49#		
002022	005000		CLR	R0	; CTLR	5748
002024	105060	000000G	CLRB	C.ERR.TBL(R0)	; *(CTLR)	5750
002030	105060	000001G	CLRB	C.ERR.TBL+1(R0)	; *(CTLR)	5751
002034	062700	000002	ADD	#2,R0	; *,CTLR	5748
002040	000243		.WORD	CLV!CLC		
002042	003770		BLE	48#		
002044	005000		CLR	R0		5754
002046	104441		TRAP	41		
002050	062706	000016	ADD	#16,SP		5511
002054	000207		RTS	PC		

; Routine Size: 535 words, Routine Base: \$CODE\$ + 2070
; Maximum stack depth per invocation: 19 words

000000	004737	002070'	.SBTTL	L\$INIT INITIALIZE SECTION		
000004	104411		L\$INIT::JSR	PC,LINIT		5754
000006	000207		TRAP	11		
			RTS	PC		

; Routine Size: 4 words, Routine Base: \$CODE\$ + 4146
; Maximum stack depth per invocation: 2 words


```

:      5757  #sbttl 'AUTODROP SECTION'
:      5758
:      5759  !+
:      5760  ! THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
:      5761  ! THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
:      5762  ! SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
:      5763  ! DROPPED FROM TESTING.
:      5764  !-
:      5765
:      5766  BGNAUTO;
:      5767
:      5768  !if (.SWP_FLAGS and SWF_TRC) eq1 SWF_TRC
:      5769  !then PRINTF (DBM3);
:      5770
:      5771  ENDAUTO;
: INFO#219      L1:5756
: Empty compound expression is illegal
: Error occurred expanding macro ENDAUTO, called from source
    
```

```

000000 000207          LAUTO: .SBTTL LAUTO AUTODROP SECTION
                                RTS    PC
                                ;
: Routine Size: 1 word,      Routine Base: $CODE$ + 4156
: Maximum stack depth per invocation: 0 words
    
```

```

000000 004737 004156'          L$AUTO: .SBTTL L$AUTO AUTODROP SECTION
000004 104461          :JSR      PC,LAUTO
000006 000207          TRAP    61
                                RTS    PC
                                ;
: Routine Size: 4 words,      Routine Base: $CODE$ + 4160
: Maximum stack depth per invocation: 2 words
    
```

ZRQAM2
V01.2

RD/RX EXERCISER
CLEANUP CODING SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (43)

```

: 5772 #sbttl 'CLEANUP CODING SECTION'
: 5773
: 5774 !+
: 5775 ! THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
: 5776 ! AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
: 5777 !-
: 5778
: 5779 BGNCLN;
: 5780
: 5781 incr CTLR from 0 to (MAX_CTLR - 1) do          ! FOR EACH CONTROLLER
: 5782
: 5783     if (RDRX_ADDR = .CST [.CTLR, IP_ADDR]) neq 0      ! IF CONTROLLER EXISTS
: 5784     then
: 5785         begin
: 5786         if .DCT [.CTLR, STAT] eq 1 ONLINE
: 5787         then
: 5788             incr COUNT from 1 to 10000 do
: 5789                 begin
: 5790                 DELAY (1);
: 5791                 if .DCT [.CTLR, CRING_CNT] eq 1 0      !WAIT TIL OUTSTANDING COMMANDS FINISHED
: 5792                 then EXITLOOP;
: 5793                 end;
: 5794                 WRT_RDRX (RCIP, RC_ALL, ALL_ONES);      ! WRITE IP TO STOP DEVICE
: 5795                 CLRVEC (.CST[.CTLR, VEC_ADDR]);        ! RETURN CONTROLLER'S TRAP VECTOR TO SUPERVISOR
: 5796                 end;
: 5797             DORPT;                                     !GIVE REPORT
: 5798         ENDCLN;

```

```

000000 004137 000000G          .SBTTL LCLEAN CLEANUP CODING SECTION
000004 005746          LCLEAN: JSR   R1,$SAVES          ;
000006 005005          TST   -(SP)
000010 010546          CLR   R5          ; CTLR
000012 012746 000076      1$:  MOV   R5,-(SP)          ; CTLR,*
000016 004737 000000G      MOV   #76,-(SP)
000022 010004          JSR   PC,BL$MUL
000024 022626          MOV   R0,R4
000026 016437 000000G 000000G  CMP   (SP)+,(SP)+
000034 001444          MOV   CST(R4),RDRX.ADDR
000036 010546          BEQ   8$
000040 012746 000022      MOV   R5,-(SP)          ; CTLR,*
000044 004737 000000G      MOV   #22,-(SP)
000050 022626          JSR   PC,BL$MUL
000052 005760 000000G      CMP   (SP)+,(SP)+
000056 100022          TST   DCT(R0)
000060 012703 023420      BPL   7$
000064 012702 000001      2$:  MOV   #23420,R3          ; *,COUNT
000070 001410          3$:  MOV   #1,R2          ; *,$$TMP2
000072 013701 000000G      BEQ   6$
000076 001403          MOV   L$DLY,R1          ; *,$$TMP1
000100 005016          4$:  BEQ   5$
000102 005301          CLR   (SP)          ; $$TMP
          DEC   R1          ; $$TMP1

```

M13

ZRQAM2
V01.2

RD/RX EXERCISER
CLEANUP CODING SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (43)

000104	001375		BNE	4\$			
000106	005302		DEC	R2		; \$\$TMP2	
000110	000767		BR	3\$			
000112	105760	000000G	6\$: TSTB	DCT(R0)			5791
000116	001402		BEQ	7\$			5792
000120	005303		DEC	R3		; COUNT	5788
000122	001360		BNE	2\$			
000124	012700	177777	7\$: MOV	#-1,R0		; *,RC.REG	5794
000130	010077	000000G	MOV	R0,@RDRX.ADDR		; RC.REG,*	
000134	016400	000002G	MOV	CST+2(R4),R0			5795
000140	042700	177000	BIC	#177000,R0			
000144	104436		TRAP	36			
000146	005205		8\$: INC	R5		; CTRL	5781
000150	000243			.WORD CLV!CLC			
000152	003716		BLE	1\$			
000154	104424		TRAP	24			5796
000156	005726		TST	(SP)+			5771
000160	000207		RTS	PC			

; Routine Size: 57 words, Routine Base: \$CODE\$ + 4170
; Maximum stack depth per invocation: 10 words

000000	004737	004170'		.SBTTL L\$CLEAN CLEANUP CODING SECTION			
			L\$CLEAN::				
			JSR	PC,L\$CLEAN			5797
000004	104412		TRAP	12			
000006	000207		RTS	PC			

; Routine Size: 4 words, Routine Base: \$CODE\$ + 4352
; Maximum stack depth per invocation: 2 words

```

: 5799 #sbttl 'DROP UNIT SECTION'
: 5800
: 5801 !+
: 5802 ! THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
: 5803 ! TO NO LONGER BE TESTED.
: 5804 !-
: 5805
: 5806 BGNDU;
: 5807
: 5808 local
: 5809     UNIT : word,                                ! NUMBER OF UNIT BEING DROPPED
: 5810     PRINT : byte initial (byte (FALSE));    ! NO PRINTING OF DROP UNIT MESSAGE
: 5811
: 5812 label
: 5813     SEARCH;
: 5814
: 5815 begin
: 5816
: 5817 register
: 5818     INPUT = 0;                                ! UNIT NUMBER APPEARS IN RO UPON ENTRY
: 5819
: 5820 UNIT = .INPUT;                                ! GET UNIT NUMBER
: 5821 end;                                          ! UNDECLARE REGISTER
: 5822
: 5823 if (.SWP_FLAGS and SWF_TRC) eq1 SWF_TRC
: 5824 then
: 5825     PRINTF (DBMS, .UNIT);
: 5826
: 5827 SEARCH :                                     ! BEGIN SEARCH BLOCK
: 5828 begin
: 5829
: 5830 incr CTLR from 0 to (MAX_CTLR - 1) do         ! FOR EACH CONTROLLER
: 5831
: 5832     incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do ! FOR EACH UNIT ENTRY IN CST
: 5833
: 5834     if (.CST [.CTLR, .OFFSET, D_UNIT] eq1 .UNIT) and ! IF UNIT MATCHES CST ENTRY
: 5835     (.CST [.CTLR, .OFFSET, D_PRES] eq1 PRESENT)
: 5836     then
: 5837         begin
: 5838
: 5839         if (.CST [.CTLR, .OFFSET, D_STAT] eq1 ONLINE) or ! IF UNIT IS STILL ALIVE
: 5840         (.DUR [.UNIT] eq1 DU_ONLINE) or
: 5841         (.DUR [.UNIT] eq1 DU_PROTECT)
: 5842         then
: 5843             begin
: 5844                 PRINT = TRUE;                    ! O.K. TO PRINT MESSAGE
: 5845
: 5846                 if (.CST [.CTLR, U_CNT] gtru 0) and
: 5847                 (.CST [.CTLR, .OFFSET, D_STAT] eq1 ONLINE)
: 5848                 then
: 5849                     CST [.CTLR, U_CNT] = .CST [.CTLR, U_CNT] - 1; ! DECREMENT UNIT COUNT
: 5850
: 5851                 if (.CST [.CTLR, U_CNT] eq1 0) and

```

ZRQAM2
V01.2

RD/RX EXERCISER
DROP UNIT SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (44)

SEQ 0170
Page 170

```

:      5852      (.CST [.CTRL, .OFFSET, D_STAT] eq1 ONLINE)
:      5853      then
:      5854      EOP_FLAG = TRUE;                ! DECLARE END OF PASS IF ALL UNITS OFFLINE
:      5855
:      5856      CST [.CTRL, .OFFSET, D_STAT] = OFFLINE;    ! MARK UNIT OFFLINE
:      5857      end;                            ! IF UNIT WAS STILL ALIVE
:      5858
:      5859      leave SEARCH;                    ! EXIT SEARCH BLOCK
:      5860      end;                            ! IF UNIT FOUND IN CST
:      5861
:      5862      end;
:      5863
:      5864      if .PRINT or                      ! IF OK TO PRINT MESSAGE
:      5865      (.DUR [.UNIT] eq1 DU_CONF) or
:      5866      (.DUR [.UNIT] eq1 DU_INIT) or
:      5867      (.DUR [.UNIT] eq1 DU_ONLINE) or
:      5868      (.DUR [.UNIT] eq1 DU_AV) or
:      5869      (.DUR [.UNIT] eq1 DU_PROTECT)
:      5870      then
:      5871      begin
:      5872      PRINTF (DU_MSG, .UNIT);
:      5873      PRINTF (.DU_RSN [.DUR [.UNIT]]);
:      5874      end;
:      5875
:      5876      ENDDU;

```

000000	004137	000000G	LDU:	.SBTTL	LDU DROP UNIT SECTION		
000004	024646			JSR	R1,\$SAVES	:	5798
000006	105004			CMP	-(SP),-(SP)		
000010	010003			CLRB	R4	: PRINT	
000012	032737	000001 000000G		MOV	R0,R3	: INPUT,UNIT	5820
000020	001411			BIT	#1,SWP.FLAGS	:	5823
000022	010346			BEQ	1#		
000024	012746	000000G		MOV	R3, -(SP)	: UNIT,*	5825
000030	012746	000002		MOV	#DBMS, -(SP)		
000034	010600			MOV	#2, -(SP)		
000036	104417			MOV	SP,R0	: SP,*	
000040	062706	000006		TRAP	17		
000044	005005		1#:	ADD	#6,SP		
000046	010546		2#:	CLR	R5	: CTRL	5830
000050	012746	000037		MOV	R5, -(SP)	: CTRL,*	5834
000054	004737	000000G		MOV	#37, -(SP)		
000060	010066	000006		JSR	PC,BL#MUL		
000064	012701	000003		MOV	R0,6(SP)		
000070	010100		3#:	MOV	#3,R1	: *,OFFSET	5832
000072	066600	000006		MOV	R1,R0	: OFFSET,*	5834
000076	006300			ADD	6(SP),R0		
000100	012766	000000G 000004		ASL	R0		
000106	060066	000004		MOV	#CST,4(SP)		
000112	010302			ADD	R0,4(SP)		
000114	017600	000004		MOV	R3,R2	: UNIT,*	
				MOV	#4(SP),R0		

C14

ZRQAM2
V01.2

RD/RX EXERCISER
DROP UNIT SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (44)

SEQ 0171
Page 171

000120	000300			SWAB	R0		
000122	042700	177760		BIC	#177760,R0		
000126	020002			CMP	R0,R2		
000130	001060			BNE	9#		
000132	032776	040000	000004	BIT	#40000,#4(SP)	:	5835
000140	001454			BEQ	9#		
000142	005002			CLR	R2	:	5839
000144	032776	020000	000004	BIT	#20000,#4(SP)		
000152	001402			BEQ	4#		
000154	005202			INC	R2		
000156	000410			BR	5#		
000160	126327	000000G	000007	4#:	CMPB	DUR(R3),#7	: *(UNIT),*
000166	001404				BEQ	5#	
000170	126327	000000G	000011		CMPB	DUR(R3),#11	: *(UNIT),*
000176	001033				BNE	8#	
000200	112704	000001		5#:	MOVB	#1,R4	: *,PRINT
000204	010516				MOV	R5,(SP)	: CTLR,*
000206	012746	000076			MOV	#76,-(SP)	
000212	004737	000000G			JSR	PC,BL\$MUL	
000216	005726				TST	(SP)+	
000220	062700	000004G			ADD	#CST+4,R0	
000224	105760	000001			TSTB	1(R0)	
000230	001404				BEQ	6#	
000232	006002				ROR	R2	:
000234	105660	000001			SBCB	1(R0)	:
000240	001007				BNE	7#	:
000242	032776	020000	000004	6#:	BIT	#20000,#4(SP)	:
000250	001403				BEQ	7#	:
000252	112737	000001	000000G		MOVB	#1,EOP.FLAG	:
000260	042776	020000	000004	7#:	BIC	#20000,#4(SP)	:
000266	022626			8#:	CMP	(SP)+,(SP)+	:
000270	000411				BR	10#	
000272	062701	000007		9#:	ADD	#7,R1	: *,OFFSET
000276	020127	000030			CMP	R1,#30	: OFFSET,*
000302	003672				BLE	3#	
000304	022626				CMP	(SP)+,(SP)+	
000306	005205				INC	R5	: CTLR
000310	000243				.WORD	CLV!CLC	
000312	003655				BLE	2#	
000314	006004			10#:	ROR	R4	: PRINT
000316	103424				BLO	11#	
000320	126327	000000G	000001		CMPB	DUR(R3),#1	: *(UNIT),*
000326	001420				BEQ	11#	
000330	126327	000000G	000002		CMPB	DUR(R3),#2	: *(UNIT),*
000336	001414				BEQ	11#	
000340	126327	000000G	000007		CMPB	DUR(R3),#7	: *(UNIT),*
000346	001410				BEQ	11#	
000350	126327	000000G	000013		CMPB	DUR(R3),#13	: *(UNIT),*
000356	001404				BEQ	11#	
000360	126327	000000G	000011		CMPB	DUR(R3),#11	: *(UNIT),*
000366	001023				BNE	12#	
000370	010346			11#:	MOV	R3,-(SP)	: UNIT,*

ZRQAM2 RD/RX EXERCISER
V01.2 DROP UNIT SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0172
Page 172
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (44)

000372	012746	000000G		MOV	#DU.MSG, -(SP)			
000376	012746	000002		MOV	#2, -(SP)			
000402	010600			MOV	SP, R0	:	SP, *	
000404	104417			TRAP	17			
000406	005000			CLR	R0	:		
000410	156300	000000G		BISB	DUR(R3), R0	:	*(UNIT), *	5873
000414	006300			ASL	R0			
000416	016016	000000G		MOV	DU.RSN(R0), (SP)			
000422	012746	000001		MOV	#1, -(SP)			
000426	010600			MOV	SP, R0	:	SP, *	
000430	104417			TRAP	17			
000432	062706	000010		ADD	#10, SP	:		5871
000436	022626		12:	CMP	(SP)+, (SP)+	:		5798
000440	000207			RTS	PC			

: Routine Size: 145 words, Routine Base: \$CODE\$ + 4362
: Maximum stack depth per invocation: 14 words

000000	004737	004362'		.SBTTL	L\$DU DROP UNIT SECTION			
000004	104453		L\$DU::	JSR	PC, LDU	:		5874
000006	000207			TRAP	53			
				RTS	PC			

: Routine Size: 4 words, Routine Base: \$CODE\$ + 5024
: Maximum stack depth per invocation: 2 words

ZRQAM2
V01.2

RD/RX EXERCISER
ADD UNIT SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (45)

```

: 5877 #sbttl 'ADD UNIT SECTION'
: 5878
: 5879 !+
: 5880 ! THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
: 5881 ! TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
: 5882 ! TO THE TEST CYCLE.
: 5883 !-
: 5884
: 5885 BGNAU;
: 5886
: 5887 local
: 5888     STINDX : word,
: 5889     ENDIDX : word;
: 5890
: 5891 register
: 5892     UNIT = 0;           ! UNIT NUMBER APPEARS IN RO UPON ENTRY
: 5893
: 5894 if (.SWP_FLAGS and SWF_CST) eq1 SWF_CST
: 5895 then
: 5896     begin               ! IF CLEAR STAT. TABLES TRUE....
: 5897     STINDX = .UNIT * TALLY_LEN; ! ZERO OUT
: 5898     ENDIDX = .STINDX + TALLY_LEN - 1; ! ADDED
: 5899
: 5900     incr COUNT from .STINDX to .ENDIDX do ! UNIT'S
: 5901     TALLY [.COUNT] = 0;           ! STATISTICS
: 5902
: 5903     end;
: 5904
: 5905 ENDAU;

```

000000	004137	000000G	LAU:	.SBTTL	LAU ADD UNIT SECTION		
000004	105737	000000G		JSR	R1,\$SAVE2	:	5876
000010	100023			TSTB	SWP.FLAGS	:	5894
000012	010046			BPL	3#		
000014	012746	000034		MOV	RO,-(SP)	: UNIT,*	5897
000020	004737	000000G		MOV	#34,-(SP)		
000024	010002			JSR	PC,BL#MUL		
000026	062702	000033		MOV	RO,R2	: STINDX,ENDIDX	5898
000032	010001			ADD	#33,R2	: *,ENDIDX	
000034	005301			MOV	RO,R1	: STINDX,COUNT	5900
000036	000404			DEC	R1	: COUNT	
000040	010100		1#:	BR	2#		
000042	006300			MOV	R1,RO	: COUNT,*	5901
000044	005060	000000G		ASL	RO		
000050	005201		2#:	CLR	TALLY(RO)		
000052	020102			INC	R1	: COUNT	5900
000054	003771			CMP	R1,R2	: COUNT,ENDIDX	
000056	022626			BLE	1#		
000060	000207		3#:	CMP	(SP)+,(SP)+		5896
				RTS	PC	:	5876

: Routine Size: 25 words, Routine Base: #CODE# + 5034

ZRQAM2 RD/RX EXERCISER
V01.2 ADD UNIT SECTION

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0174
Page 174
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (45)

; Maximum stack depth per invocation: 6 words

000000	004737	005034'					
000004	104452		L\$AU::	.SBTTL	L\$AU ADD UNIT SECTION		
000006	000207			JSR	PC,LAU	:	5903
				TRAP	52		
				RTS	PC		

; Routine Size: 4 words, Routine Base: \$CODE\$ + 5116
; Maximum stack depth per invocation: 2 words

ZRQAM2
V01.2

RD/RX EXERCISER
NON-EXISTENT MEMORY TRAP HANDLER

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (46)

SEQ 0175
Page 175

```

: 5906 .sbttl 'NON-EXISTENT MEMORY TRAP HANDLER'
: 5907
: 5908 !+
: 5909 ! THIS TRAP HANDLER IS VECTORED FROM LOCATION 4 FOR ALL UNIBUS TIMEOUT
: 5910 ! ERRORS, INDICATING THAT AN ATTEMPT WAS MADE TO REFERENCE A NON-EXISTENT
: 5911 ! MEMORY LOCATION. ITS MAIN PURPOSE IS TO SET A FLAG FOR THE RDRX
: 5912 ! REGISTER EXISTENCE TEST, INDICATING THE ABSENCE OF A DEVICE REGISTER.
: 5913 !-
: 5914
: 5915 BGNSRV (NEX_TRAP);
: 5916
: 5917 NEX = TRUE;                ! NEX TRAP OCCURRED
: 5918
: 5919 ENDSRV;
    
```

```

000000 012737 000001 000000G      .SBTTL NEX.TRAP NON-EXISTENT MEMORY TRAP HANDLER
                                NEX.TRAP::
000006 000002                    MOV     #1,NEX
                                RTI
                                ;
                                ;
    
```

5917
5915

```

: Routine Size: 4 words,      Routine Base: $CODE$ + 5126
: Maximum stack depth per invocation: 0 words
    
```

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (47)

```

: 5920 #sbttl 'GLOBAL ROUTINES'
: 5921
: 5922 global routine SET_CPAR (CTRL) : novalue =
: 5923
: 5924 !+
: 5925 ! THIS ROUTINE SETS UP THE COMMONLY-USED CONTROLLER-RELATED DATA ITEMS
: 5926 ! FOR THE GIVEN CONTROLLER NUMBER.
: 5927 !
: 5928 ! INPUTS:
: 5929 !     CTRL - CONTROLLER NUMBER
: 5930 !
: 5931 ! IMPLICIT OUTPUTS:
: 5932 !     CCTLR - CURRENT CONTROLLER NUMBER
: 5933 !     CST_ADDR - ADDRESS OF CONTROLLER'S STATUS TABLE
: 5934 !     DCT_ADDR - ADDRESS OF CONTROLLER'S DRIVER TABLE
: 5935 !     RDRX_ADDR - ADDRESS OF CONTROLLER'S IP REGISTER
: 5936 !-
: 5937
: 5938 begin
: 5939 CCTLR = .CTRL;           ! SET CURRENT CONTROLLER NUMBER
: 5940 CST_ADDR = CST + (.CTRL * CST_LEN * 2); ! CALCULATE ADDRESS OF CONTROLLER'S CST
: 5941 DCT_ADDR = DCT + (.CTRL * DCT_LEN * 2); ! CALCULATE ADDRESS OF CONTROLLER'S DCT
: 5942 RDRX_ADDR = .CST_ADDR [IP_ADDR];      ! GET CONTROLLER'S DEVICE ADDRESS
: 5943 end;

```

		.SBTTL	SET.CPAR GLOBAL ROUTINES	
000000	010146	SET.CPAR::		
000002	016601	000004	MOV R1, -(SP)	5922
000006	010137	000000G	MOV 4(SP), R1	5939
000012	010146		MOV R1, CCTLR	
000014	012746	000076	MOV R1, -(SP)	5940
000020	004737	000000G	MOV #76, -(SP)	
000024	062700	000000G	JSR PC, BL#MUL	
000030	010037	000000G	ADD #CST, R0	
000034	010116		MOV R0, CST_ADDR	
000036	012746	000022	MOV R1, (SP)	5941
000042	004737	000000G	MOV #22, -(SP)	
000046	062700	000000G	JSR PC, BL#MUL	
000052	010037	000000G	ADD #DCT, R0	
000056	017737	000000G 000000G	MOV R0, DCT_ADDR	
000064	062706	000006	MOV #CST_ADDR, RDRX_ADDR	5942
000070	012601		ADD #6, SP	5938
000072	000207		MOV (SP)+, R1	5922
			RTS PC	

; Routine Size: 30 words, Routine Base: #CODE# + 5136
; Maximum stack depth per invocation: 5 words

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (48)

SEQ 0177
Page 177

```

: 5944 global routine SET_UPAR (OFFSET) : novalue =
: 5945
: 5946 !!+
: 5947 !! THIS ROUTINE SETS UP THE COMMONLY-USED UNIT-RELATED DATA ITEMS FOR
: 5948 !! THE CURRENT CONTROLLER AND GIVEN CST OFFSET.
: 5949 !!
: 5950 !! INPUTS:
: 5951 !!     OFFSET - WORD OFFSET INTO CURRENT CONTROLLER'S CST WHICH
: 5952 !!             DESCRIBES A UNIT
: 5953 !!
: 5954 !! IMPLICIT INPUTS:
: 5955 !!     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 5956 !!
: 5957 !! IMPLICIT OUTPUTS:
: 5958 !!     CUOFF - CURRENT UNIT'S CST OFFSET
: 5959 !!     CDISK - CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 5960 !!     L$LUN - CURRENT UNIT NUMBER (DRS UNIT NUMBER)
: 5961 !!     T_ADDR - ADDRESS OF CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 5962 !!-
: 5963
: 5964 begin
: 5965 CUOFF = .OFFSET;
: 5966 CDISK = .CST_ADDR [.OFFSET, D_DISK_NUM];
: 5967 L$LUN = .CST_ADDR [.OFFSET, D_UNIT];
: 5968 T_ADDR = TALLY + (.L$LUN * TALLY_LEN * 2);
: 5969 end;

```

000000	010146		.SBTTL	SET_UPAR GLOBAL ROUTINES		
			SET_UPAR::			
000002	016637	000004	000000G	MOV R1, -(SP)	:	5944
000010	016600	000004		MOV 4(SP), CUOFF	:	5965
000014	006300			MOV 4(SP), R0	:	5966
000016	063700	000000G		ASL R0		
000022	111037	000000G		ADD CST_ADDR, R0		
000026	042737	177774	000000G	MOVB (R0), CDISK		
000034	011001			BIC #177774, CDISK		
000036	000301			MOV (R0), R1	:	5967
000040	042701	177760		SWAB R1		
000044	010137	000000G		BIC #177760, R1		
000050	010146			MOV R1, L\$LUN		
000052	012746	000070		MOV R1, -(SP)	:	5968
000056	004737	000000G		MOV #70, -(SP)		
000062	062700	000000G		JSR PC, BL\$MUL		
000066	010037	000000G		ADD #TALLY, R0		
000072	022626			MOV R0, T_ADDR		
000074	012601			CMP (SP)+, (SP)+	:	5964
000076	000207			MOV (SP)+, R1	:	5944
				RTS PC		

```

: Routine Size: 32 words, Routine Base: $CODE$ + 5232
: Maximum stack depth per invocation: 4 words
: 5970

```

```

: 5971
: 5972 global routine GET_PKT (CTRL) =
: 5973
: 5974 !+
: 5975 ! THIS ROUTINE SEARCHES THE MSCP PACKET POOL ALLOCATION TABLE (PKT_USE)
: 5976 ! FOR A FREE MSCP PACKET TO ALLOCATE TO THE GIVEN CONTROLLER. IF ONE IS
: 5977 ! FOUND, THE PACKET IS ZEROED OUT, AND THE PACKET INDEX IS RETURNED
: 5978 ! TO THE CALLER. OTHERWISE, A -1 IS RETURNED INDICATING NONE AVAILABLE.
: 5979 !
: 5980 ! INPUTS:
: 5981 ! CTRL - CONTROLLER NUMBER REQUESTING ALLOCATION
: 5982 !-
: 5983
: 5984 begin
: 5985
: 5986 local
: 5987 index : signed word initial (-1),
: 5988 RING_ADDR : word,
: 5989 PACKET_OWNED : byte,
: 5990 NEW_PKT : byte;
: 5991
: 5992 NEW_PKT = .NXT_PKT_2USE;          ! NEXT PACKET TO TRY
: 5993
: 5994 incr COUNT from 0 to (PKT_CNT - 1) do      ! FOR EACH ENTRY IN ALLOCATION TABLE
: 5995 begin
: 5996 PACKET_OWNED = FALSE;
: 5997
: 5998 if .PKT_USE [.NEW_PKT] lss 0      ! IF ENTRY INDICATES FREE PACKET
: 5999 then
: 6000 begin
: 6001 RING_ADDR = .DCT_ADDR [RR_BEG];      ! FIRST RESPONSE PACKET'S ADDRESS
: 6002
: 6003 incr I from 1 to (RRING_LEN + CRING_LEN) do ! FOR EACH PACKET ADDRESS
: 6004 if (.RING_ADDR eqs .MSCP_PKT [.NEW_PKT, PKT_LO]) and
: 6005 ((.RING_ADDR + 2) and ED_OWN) eq1 ED_OWN)
: 6006 then
: 6007 begin
: 6008 PACKET_OWNED = TRUE;          ! CHECK ADDRESS AND OWNERSHIP
: 6009 exitloop;                    ! PACKET OWNED BY CONTROLLER
: 6010 end
: 6011 else
: 6012 RING_ADDR = .RING_ADDR + 4;      ! ADDRESS OF NEXT PACKET IN RING
: 6013
: 6014 if not .PACKET_OWNED          ! IF NOT ALREADY USED
: 6015 then
: 6016 begin
: 6017 PKT_USE [.NEW_PKT] = .CTRL;      ! ALLOCATE PACKET TO CONTROLLER
: 6018 index = .NEW_PKT;
: 6019
: 6020 incr J from 2 to (PKT_LEN - 1) do      ! ZERO OUT PACKET
: 6021 MSCP_PKT [.NEW_PKT, .J, 0, 16, 0] = 0;
: 6022
: 6023 exitloop;                        ! DONE

```

```

:      6024          end;
:      6025
:      6026          end;
:      6027
:      6028          NEW_PKT = .NEW_PKT + 1;          ! TRY NEXT PACKET IN RING
:      6029
:      6030          if .NEW_PKT gequ PKT_CNT
:      6031          then
:      6032              NEW_PKT = 0;          ! IF BEYOND ALL PACKETS, START AT THE TOP
:      6033
:      6034          end;
:      6035
:      6036          if (.index geq 0) and          ! IF PACKET FOUND
:      6037              (.PKT_USE [.index] geq 0)
:      6038          then
:      6039              begin
:      6040                  MSCP_PKT [.index, MSGLEN] = SZ_GEN;          ! PACKET SIZE - ONLY ONLINE AND SCC CHANGE IT
:      6041                  MSCP_PKT [.index, CREDITS] = 1;          ! CREDIT SIZE
:      6042                  NXT_PKT_2USE = .NEW_PKT + 1;          ! NEXT PACKET TO ALLOCATE
:      6043
:      6044                  if NXT_PKT_2USE gequ PKT_CNT
:      6045                  then
:      6046                      NXT_PKT_2USE = 0;          ! CYCLE BACK TO BEGINNING IF AT END
:      6047
:      6048                  end;
:      6049
:      6050          return .index;
:      6051          end;

```

000000	004137	000000G	GET.PKT::	.SBTTL GET.PKT GLOBAL ROUTINES		
000004	162706	000006		JSR R1,\$SAVES	:	5972
000010	012704	177777		SUB #6,SP		
000014	113705	000000G		MOV #-1,R4	: *,INDEX	5984
000020	012766	000014 000004		MOVB NXT.PKT.2USE,R5	: *,NEW.PKT	5992
000026	105066	000002	1#:	MOV #14,4(SP)	: *,COUNT	5994
000032	005003			CLRB 2(SP)	: PACKET.OWNED	5996
000034	150503			CLR R3	:	5998
000036	105763	000000G		BISB R5,R3	: NEW.PKT.*	
000042	002076			TSTB PKT.USE(R3)		
000044	013700	000000G		BGE 7#		
000050	016016	000004		MOV DCT.ADDR,R0	:	6001
000054	010346			MOV 4(R0),(SP)	: *,RING.ADDR	
000056	012746	000104		MOV R3,-(SP)	:	6004
000062	004737	000000G		MOV #104,-(SP)		
000066	012702	000010		JSR PC,BL#MUL		
000072	027660	000004 000000G	2#:	MOV #10,R2	: *,I	6003
000100	001016			CMP #4(SP),MSCP.PKT(R0)	: RING.ADDR,*	6004
000102	012746	000002		BNE 3#		
000106	066616	000006		MOV #2,-(SP)	:	6005
000112	012601			ADD 6(SP),(SP)	: RING.ADDR,*	
				MOV (SP)+,R1		

ZRQAM2 V01.2	RD/RX EXERCISER GLOBAL ROUTINES		5-Dec-1983 10:27:14 5-Dec-1983 10:27:04	VAX-11 Bliss-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (49)	
000114	042701	077777	BIC	#77777,R1	
000120	020127	100000	CMP	R1,#-100000	
000124	001004		BNE	3#	
000126	112766	000001 000006	MOVB	#1,6(SP)	; *,PACKET.OWNED 6008
000134	000405		BR	4#	; 6007
000136	062766	000004 000004	3#: ADD	#4,4(SP)	; *,RING.ADDR 6012
000144	005302		DEC	R2	; I 6003
000146	001351		BNE	2#	
000150	032766	000001 000006	4#: BIT	#1,6(SP)	; *,PACKET.OWNED 6014
000156	001027		BNE	6#	
000160	116663	000030 000000G	MOVB	30(SP),PKT.USE(R3)	; CTLR,* 6017
000166	010304		MOV	R3,R4	; *,INDEX 6018
000170	010316		MOV	R3,(SP)	; 6021
000172	012746	000042	MOV	#42,-(SP)	
000176	004737	000000G	JSR	PC,BL\$MUL	
000202	005726		TST	(SP)+	
000204	012701	000002	MOV	#2,R1	; *,J 6020
000210	010003		5#: MOV	R0,R3	; 6021
000212	060103		ADD	R1,R3	; J,*
000214	006303		ASL	R3	
000216	005063	000000G	CLR	MSCP.PKT(R3)	
000222	005201		INC	R1	; J 6020
000224	020127	000041	CMP	R1,#41	; J,*
000230	003767		BLE	5#	
000232	022626		CMP	(SP)+,(SP)+	; 6016
000234	000411		BR	9#	
000236	022626		6#: CMP	(SP)+,(SP)+	; 6000
000240	105205		7#: INCB	R5	; NEW.PKT 6028
000242	120527	000014	CMPB	R5,#14	; NEW.PKT,* 6030
000246	103401		BLO	8#	
000250	105005		CLRB	R5	; NEW.PKT 6032
000252	005366	000004	8#: DEC	4(SP)	; COUNT 5994
000256	001263		BNE	1#	
000260	005704		9#: TST	R4	; INDEX 6036
000262	002434		BLT	11#	
000264	105764	000000G	TSTB	PKT.USE(R4)	; *(INDEX) 6037
000270	002431		BLT	11#	
000272	010446		MOV	R4,-(SP)	; INDEX,* 6040
000274	012746	000104	MOV	#104,-(SP)	
000300	004737	000000G	JSR	PC,BL\$MUL	
000304	012760	000040 000006G	MOV	#40,MSCP.PKT+6(R0)	
000312	142760	000017 000010G	BICB	#17,MSCP.PKT+10(R0)	; 6041
000320	152760	000001 000010G	BISB	#1,MSCP.PKT+10(R0)	
000326	005000		CLR	R0	; 6042
000330	150500		BISB	R5,R0	; NEW.PKT,*
000332	005200		INC	R0	
000334	110037	000000G	MOVB	R0,NXT.PKT.2USE	
000340	120027	000014	CMPB	R0,#14	; NXT.PKT.2USE,* 6044
000344	103402		BLO	10#	
000346	105037	000000G	CLRB	NXT.PKT.2USE	; 6046
000352	022626		10#: CMP	(SP)+,(SP)+	; 6039
000354	010400		11#: MOV	R4,R0	; INDEX,* 5984
000356	062706	000006	ADD	#6,SP	; 5972

M14

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (49)

SEQ 0181
Page 181

000362 000207

RTS PC

: Routine Size: 122 words, Routine Base: \$CODE\$ + 5332
: Maximum stack depth per invocation: 13 words

: 6052


```

: 6053 global routine PUT_PKT (index) : novalue =
: 6054
: 6055 !+
: 6056 ! THE MSCP PACKET DESIGNATED BY "INDEX" IS RETURNED TO THE POOL BY THIS
: 6057 ! ROUTINE.
: 6058 !-
: 6059
: 6060 begin
: 6061
: 6062 local
: 6063 RING_ADDR : word,
: 6064 OWNER : word;
: 6065
: 6066 RING_ADDR = .DCT_ADDR [RR_BEG]; ! ADDRESS IN FIRST RESPONSE RING
: 6067
: 6068 incr COUNT from 1 to (RRING_LEN + CRING_LEN) do ! FOR EACH ADDRESS IN THE RINGS
: 6069 begin
: 6070
: 6071 if .MSCP_PKT [.index, PKT_LO] eqla ..RING_ADDR ! IF ADDRESS MATCHES
: 6072 then
: 6073 begin
: 6074 OWNER = .RING_ADDR + 2; ! ADDRESS OF OWNERSHIP WORD
: 6075 .OWNER = ..OWNER and (not (ED_OWN)) and (not (ED_FLAG)); ! GIVE OWNERSHIP TO HOST
: 6076 end;
: 6077
: 6078 RING_ADDR = .RING_ADDR + 4; ! LOOK AT NEXT PACKET ADDRESS IN RING
: 6079 end;
: 6080
: 6081 PKT_USE [.index] = -1;
: 6082 end;

```

			.SBTTL	PUT.PKT GLOBAL ROUTINES	
000000	004137	000000G	PUT.PKT::		
			JSR	R1, \$SAVE4	:
000004	013700	000000G	MOV	DCT.ADDR, R0	:
000010	016003	000004	MOV	4(R0), R3	: *, RING.ADDR
000014	016601	000014	MOV	14(SP), R1	: INDEX, *
000020	010146		MOV	R1, -(SP)	
000022	012746	000104	MOV	#104, -(SP)	
000026	004737	000000G	JSR	PC, BL \$MUL	
000032	012702	000010	MOV	#10, R2	: *, COUNT
000036	026013	000000G	1\$: CMP	MSCP.PKT(R0), (R3)	: *, RING.ADDR
000042	001005		BNE	2\$	
000044	012704	000002	MOV	#2, R4	: *, OWNER
000050	060304		ADD	R3, R4	: RING.ADDR, OWNER
000052	042714	140000	BIC	#140000, (R4)	: *, OWNER
000056	062703	000004	2\$: ADD	#4, R3	: *, RING.ADDR
000062	005302		DEC	R2	: COUNT
000064	001364		BNE	1\$	
000066	112761	000377 000000G	MOVB	#377, PKT.USE(R1)	:
000074	022626		CMP	(SP)+, (SP)+	:
000076	000207		RTS	PC	:
					6053
					6066
					6071
					6068
					6071
					6074
					6075
					6078
					6068
					6081
					6060
					6053

B15

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1:82 (50)

SEQ 0183
Page 183

; Routine Size: 32 words, Routine Base: \$CODE\$ + 5716
; Maximum stack depth per invocation: 8 words

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (51)

```

: 6083 routine PUTA_PKT (CTRL) : novalue =
: 6084
: 6085 !+
: 6086 ! THIS ROUTINE DEALLOCATES ALL MSCP PACKETS WHICH HAVE BEEN ALLOCATED
: 6087 ! TO A PARTICULAR CONTROLLER.
: 6088 !
: 6089 ! INPUTS:
: 6090 ! CTRL - CONTROLLER NUMBER
: 6091 !-
: 6092
: 6093 incr COUNT from 0 to (PKT_CNT - 1) do ! FOR EACH ENTRY IN ALLOCATION TABLE
: 6094
: 6095 if .PKT_USE [.COUNT] eq1 .CTRL ! IF PACKET IS ALLOCATED TO GIVEN CONTROLLER
: 6096 then
: 6097 PKT_USE [.COUNT] = -1; ! DEALLOCATE IT

```

Address	Label	Code	Comment	Address
000000	010146	PUTA.PKT:	.SBTTL PUTA.PKT GLOBAL ROUTINES	
000002	005000	MOV	R1, -(SP)	6083
000004	116001	CLR	R0	6093
000010	020166	1\$: MOVB	PKT_USE(R0), R1	6095
000014	001003	CMP	R1, 4(SP)	
000016	112760	BNE	2\$	
000024	005200	2\$: MOVB	#377, PKT_USE(R0)	6097
000026	020027	INC	R0	6093
000032	003764	CMP	R0, #13	
000034	012601	BLE	1\$	
000036	000207	MOV	(SP)+, R1	6083
		RTS	PC	

: Routine Size: 16 words, Routine Base: \$CODE\$ + 6016
: Maximum stack depth per invocation: 2 words

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (52)

SEQ 0185
Page 185

```

: 6098 global routine GET_RETPKT (CTRL) =
: 6099
: 6100 !*
: 6101 ! THIS ROUTINE SEARCHES THE RETURN PACKET POOL ALLOCATION TABLE (RP_USE)
: 6102 ! FOR A FREE RETURN PACKET TO ALLOCATE TO THE GIVEN CONTROLLER. IF ONE IS
: 6103 ! FOUND, THE PACKET IS ZEROED OUT, AND THE PACKET INDEX IS RETURNED TO
: 6104 ! THE CALLER. OTHERWISE, A -1 IS RETURNED INDICATING NONE AVAILABLE.
: 6105 !
: 6106 ! INPUTS:
: 6107 ! CTRL - CONTROLLER NUMBER REQUESTING ALLOCATION
: 6108 !-
: 6109
: 6110 begin
: 6111
: 6112 local
: 6113 index : signed word initial (-1);          ! ASSUME NONE AVAILABLE
: 6114
: 6115 incr COUNT from 0 to (RP_CNT - 1) do      ! FOR EACH ENTRY IN TABLE
: 6116
: 6117 if .RP_USE [.COUNT] les 0                ! IF FREE RETPKT IS FOUND
: 6118 then
: 6119     begin
: 6120     RP_USE [.COUNT] = .CTRL;              ! ALLOCATE RETURN PACKET TO CONTROLLER
: 6121     index = .COUNT;
: 6122
: 6123     incr J from 0 to (RP_LEN - 1) do        ! ZERO OUT RETPKT
: 6124     RETPKT [.COUNT, .J, 0, 16, 0] = 0;
: 6125
: 6126     exitloop;                              ! DONE
: 6127     end;
: 6128
: 6129 return .index;                            ! RETURN PACKET INDEX (OR -1) TO CALLER
: 6130 end;

```

000000	004137	000000G	.SBTTL GET.RETPKT GLOBAL ROUTINES		
000004	012704	177777	GET.RETPKT::		
000010	005003		JSR R1,\$SAVE4	:	6098
000012	105763	000000G	MOV #-1,R4	:	* ,INDEX 6110
000016	002025		CLR R3	:	COUNT 6115
000020	116663	000014 000000G	1\$: TSTB RP.USE(R3)	:	* (COUNT) 6117
000026	010304		BGE 3\$		
000030	010346		MOVB 14(SP),RP.USE(R3)	:	CTRL,*(COUNT) 6120
000032	012746	000030	MOV R3,R4	:	COUNT,INDEX 6121
000036	004737	000000G	MOV R3,-(SP)	:	COUNT,* 6124
000042	022626		MOV #30,-(SP)		
000044	005002		JSR PC,BL\$MUL		
000046	010001		CMP (SP)+,(SP)+		
000050	060201		2\$: CLR R2	:	J 6123
000052	006301		MOV R0,R1	:	
000054	005061	000000G	ADD R2,R1	:	J,* 6124
			ASL R1		
			CLR RETPKT(R1)		

E15

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (52)

SEQ 0186
Page 186

000060	005202		INC	R2	:	J	6123
000062	020227	000027	CMP	R2,#27	:	J,*	
000066	003767		BLE	2\$			
000070	000404		BR	4\$			
000072	005203	3\$:	INC	R3	:	COUNT	6119
000074	020327	000003	CMP	R3,#3	:	COUNT,*	6115
000100	003744		BLE	1\$			
000102	010400	4\$:	MOV	R4,R0	:	INDEX,*	6110
000104	000207		RTS	PC	:		6098

: Routine Size: 35 words, Routine Base: \$CODE\$ + 6056
: Maximum stack depth per invocation: 8 words

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (53)

```

:      6131 global routine PUT_RETPKT (index) : novalue =
:      6132
:      6133 !+
:      6134 !      THE RETURN PACKET DESIGNATED BY "INDEX" IS RETURNED TO THE POOL BY THIS
:      6135 !      ROUTINE.
:      6136 !-
:      6137
:      6138      RP_USE [.index] = -1;

```

```

000000 016600 000002          .SBTTL PUT_RETPKT GLOBAL ROUTINES
                                PUT_RETPKT::
000004 112760 000377 000000G      MOV      2(SP),R0          ; INDEX,*      6138
000012 000207          MOVB     #377,RP.USE(R0)
                                RTS      PC          ;              6131

```

```

: Routine Size: 6 words,      Routine Base: $CODE$ + 6164
: Maximum stack depth per invocation: 0 words

```

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0188
Page 188
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (54)

```

: 6139 global routine GET_IO_BUFF (ADDR) : novalue =
: 6140
: 6141 !+
: 6142 ! THIS ROUTINE HANDLES THE ALLOCATION OF AN I/O BUFFER FROM THE BUFFER
: 6143 ! POOL.
: 6144 !
: 6145 ! INPUTS:
: 6146 ! ADDR - ADDRESS TO STORE THE 2-WORD BUFFER DESCRIPTOR
: 6147 !
: 6148 ! IMPLICIT INPUTS:
: 6149 ! CCTLR - CURRENT CONTROLLER NUMBER
: 6150 !
: 6151 ! OUTPUTS:
: 6152 ! THE ALLOCATED BUFFER'S DESCRIPTOR IS LOADED INTO THE TWO
: 6153 ! WORDS AT "ADDR" AND "ADDR + 2". OTHERWISE, A ZERO IS RETURNED
: 6154 ! AT "ADDR" IF NO BUFFERS ARE AVAILABLE.
: 6155 !-
: 6156
: 6157 begin
: 6158 .ADDR = 0; ! ASSUME FAILURE
: 6159
: 6160 incr COUNT from 0 to (GIO_PER_CTLR * MAX_CTLR - 1) do ! FOR EACH ENTRY IN BUFFER TABLE
: 6161
: 6162 if .BUFF_OWN [.COUNT] les 0 ! IF BUFFER IS FREE
: 6163 then
: 6164 begin
: 6165 .BUFF_OWN [.COUNT] = .CCTLR; ! ALLOCATE BUFFER TO CONTROLLER
: 6166 .ADDR = .BUFF_ADDR [.COUNT]; ! RETURN BUFFER DESCRIPTOR
: 6167 exitloop; ! DONE
: 6168 end;
: 6169
: 6170 end; ! ROUTINE GET_IO_BUFF

```

000000	010146		.SBTTL	GET.IO.BUFF GLOBAL ROUTINES		
			GET.IO.BUFF::			
000002	005076	000004	MOV	R1, -(SP)	:	6139
000006	005001		CLR	@4(SP)	:	6158
000010	105761	000000G	CLR	R1	:	6160
000014	002011		1\$: TSTB	BUFF.OWN(R1)	:	6162
000016	113761	000000G 000000G	BGE	2\$:	
000024	010100		MOVB	CCTLR, BUFF.OWN(R1)	:	6165
000026	006300		MOV	R1, R0	:	6166
000030	016076	000000G 000004	ASL	R0	:	
000036	000404		MOV	BUFF.ADDR(R0), @4(SP)	:	
000040	005201		BR	3\$:	6164
000042	020127	000007	2\$: INC	R1	:	6160
000046	003760		2\$: CMP	R1, #7	:	
000050	012601		BLE	1\$:	
000052	000207		3\$: MOV	(SP)+, R1	:	6139
			RTS	PC	:	

; Routine Size: 22 words, Routine Base: \$CODE\$ + 6200

H15

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 B11es-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (54)

SEQ 0189
Page 189

; Maximum stack depth per invocation: 2 words

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0190
Page 190
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (55)

```

: 6171 global routine PUT_IO_BUFF (ADDR) : novalue =
: 6172
: 6173 !+
: 6174 ! THIS ROUTINE HANDLES THE DEALLOCATION OF AN I/O BUFFER, RETURNING IT
: 6175 ! TO THE BUFFER POOL.
: 6176 !
: 6177 ! INPUTS:
: 6178 ! ADDR - ADDRESS OF THE 2-WORD BUFFER DESCRIPTOR TO BE
: 6179 ! DEALLOCATED
: 6180 !-
: 6181
: 6182 incr COUNT from 0 to (QIO_PER_CTLR * MAX_CTLR - 1) do ! FOR EACH ENTRY IN BUFFER TABLE
: 6183
: 6184 if .BUFF_ADDR [.COUNT] eqa ..ADDR ! IF THIS IS THE BUFFER'S ENTRY
: 6185 then
: 6186 begin
: 6187 BUFF_OWN [.COUNT] = -1; ! DEALLOCATE BUFFER
: 6188 exitloop; ! DONE
: 6189 end;

```

		.SBTTL	PUT.IO.BUFF GLOBAL ROUTINES			
000000	010146	PUT.IO.BUFF::				
000002	005001	MOV	R1, -(SP)	:	6171	
000004	010100	CLR	R1	:	6182	
000006	006300	1\$: MOV	R1, R0	:	6184	
000010	026076	ASL	R0			
000016	001004	000000G 000004	CMP	BUFF_ADDR(R0), #4(SP)	:	*, ADDR
000020	112761	000377 000000G	BNE	2\$		
000026	000404		MOVB	#377, BUFF_OWN(R1)	:	*, *(COUNT)
000030	005201		BR	3\$:	
000032	020127	000007	2\$: INC	R1	:	COUNT
000036	003762		CMP	R1, #7	:	COUNT, *
000040	012601		BLE	1\$		
000042	000207		3\$: MOV	(SP)+, R1	:	
			RTS	PC	:	6171

```

: Routine Size: 18 words, Routine Base: $CODE$ + 6254
: Maximum stack depth per invocation: 2 words

```

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (56)

```

:      6190 global routine PUTA_BUFF : novalue =
:      6191
:      6192 !+
:      6193 ! THIS ROUTINE DEALLOCATES ALL I/O BUFFERS WHICH HAVE BEEN ALLOCATED TO
:      6194 ! THE CURRENT CONTROLLER (CCTLR).
:      6195 !-
:      6196
:      6197     incr COUNT from 0 to (QIO_PER_CTLR * MAX_CTLR - 1) do           ! FOR EACH ENTRY IN BUFFER TABLE
:      6198
:      6199     if .BUFF_OWN [.COUNT] eq1 .CCTLR           ! IF THIS BUFFER IS ALLOCATED TO THE CURRENT CONTROLLER
:      6200     then
:      6201         BUFF_OWN [.COUNT] = -1;           ! DEALLOCATE IT

```

```

000000 010146          .SBTTL PUTA.BUFF GLOBAL ROUTINES
000002 005000          PUTA.BUFF::
000004 116001 000000G 1$:   MOV     R1, -(SP)           ; COUNT      6190
000010 020137 000000G   CLR     R0                    ;             6197
000014 001003          MOVB    BUFF_OWN(R0),R1       ; *(COUNT),* 6199
000016 112760 000377 000000G   CMP     R1,CCTLR
000024 005200          BNE     2$
000026 020027 000007 2$:   MOVB    #377,BUFF_OWN(R0)   ; *,*(COUNT) 6201
000032 003764          INC     R0                    ; COUNT      6197
000034 012601          CMP     R0,#7
000036 000207          BLE     1$
                                MOV     (SP)+,R1
                                RTS     PC

```

```

: Routine Size: 16 words,      Routine Base: $CODE$ + 6320
: Maximum stack depth per invocation: 2 words

```

```

: 6202 global routine OUT_IODQ =
: 6203
: 6204 !+
: 6205 ! THIS ROUTINE RETURNS TO THE CALLER THE NEXT RETPKT INDEX TO BE
: 6206 ! PROCESSED FROM THE I/O DONE QUEUE (IODQ). THE "OUT" POINTER TO THE
: 6207 ! QUEUE IS ALSO UPDATED.
: 6208 !
: 6209 ! INPUTS:
: 6210 !     NONE
: 6211 !
: 6212 ! OUTPUTS:
: 6213 !     THE INDEX OF THE NEXT RETPKT TO BE PROCESSED.
: 6214 !-
: 6215
: 6216 begin
: 6217
: 6218 local
: 6219     index : word;
: 6220
: 6221     index = .IODQ [.IODQ_OUT];           ! GET NEXT RETPKT INDEX
: 6222     IODQ_OUT = .IODQ_OUT + 1;         ! ADVANCE "OUT" POINTER
: 6223
: 6224     if .IODQ_OUT gequ IODQ_LEN        ! IF BEYOND END OF QUEUE
: 6225     then
: 6226         IODQ_OUT = 0;                 ! SET POINTER TO BEGINNING OF QUEUE
: 6227
: 6228     return .index;                    ! RETURN INDEX TO CALLER
: 6229 end;

```

Address	Offset	Hex	Assembly	Comment	Label
000000	013700	000000G	.SBTTL	OUT.IODQ GLOBAL ROUTINES	
			OUT.IODQ::		
000004	116000	000000G	MOV	IODQ.OUT,RO	6221
000010	042700	177400	MOVB	IODQ(RO),RO	*.INDEX
000014	005237	000000G	BIC	#177400,RO	*.INDEX
000020	023727	000000G 000004	INC	IODQ.OUT	6222
000026	103402		CMP	IODQ.OUT,#4	6224
000030	005037	000000G	BLO	1#	
000034	000207		CLR	IODQ.OUT	6226
		1#:	RTS	PC	6202

```

: Routine Size: 15 words,      Routine Base: $CODE$ + 6360
: Maximum stack depth per invocation: 0 words

```

```

: 6230 global routine IN_IODQ (index) : novalue =
: 6231
: 6232 !+
: 6233 ! THIS ROUTINE INSERTS A RETURN PACKET INDEX INTO THE I/O DONE QUEUE, AND
: 6234 ! UPDATES THE IODQ_IN POINTER.
: 6235 !-
: 6236
: 6237 if ((.IODQ_IN + 1) eql .IODQ_OUT) or
: 6238 (.IODQ_IN - (IODQ_LEN - 1) eql .IODQ_OUT)
: 6239 then
: 6240 return
: 6241 else
: 6242 begin
: 6243 IODQ [.IODQ_IN] = .index; ! LOAD INDEX INTO QUEUE
: 6244 IODQ_IN = .IODQ_IN + 1; ! ADVANCE "IN" POINTER
: 6245
: 6246 if .IODQ_IN gequ IODQ_LEN ! IF BEYOND END OF QUEUE
: 6247 then
: 6248 IODQ_IN = 0; ! CYCLE BACK TO BEGINNING OF QUEUE
: 6249
: 6250 end; ! IF IODQ IS NOT FULL

```

```

000000 010146 .SBTTL IN.IODQ GLOBAL ROUTINES
IN.IODQ:
000002 013701 000000G MOV R1, -(SP) ; 6230
000006 010100 MOV IODQ.IN, R1 ; 6237
000010 005200 MOV R1, R0
000012 020037 000000G INC R0
000016 001421 CMP R0, IODQ.OUT
000020 010100 BEQ 1$ ;
000022 162700 000003 MOV R1, R0 ; 6238
000026 020037 000000G SUB #3, R0
000032 001413 CMP R0, IODQ.OUT
000034 116661 000004 000000G BEQ 1$ ; 6240
000042 005237 000000G MOVB 4(SP), IODQ(R1) ; INDEX, # 6243
000046 023727 000000G 000004 INC IODQ.IN ; 6244
000054 103402 BLO 1$ ; 6246
000056 005037 000000G CLR IODQ.IN ;
000062 012601 1$: MOV (SP)+, R1 ; 6248
000064 000207 RTS PC ; 6230

```

; Routine Size: 27 words, Routine Base: \$CODE\$ + 6416
; Maximum stack depth per invocation: 2 words

```

: 6251 global routine DROP_CTLR (CTLR, REASON) : novalue =
: 6252
: 6253 !+
: 6254 ! THIS ROUTINE DROPS ALL UNITS ASSOCIATED WITH THE CONTROLLER DESIGNATED
: 6255 ! BY "CTLR". THE REASON FOR DROPPING THE DEVICE IS LOADED INTO THE DUR
: 6256 ! VECTOR FOR EACH ATTACHED UNIT. THIS DATA IS THEN USED BY THE DROP UNIT
: 6257 ! SECTION.
: 6258 !-
: 6259
: 6260 begin
: 6261
: 6262 local
: 6263 UNIT;
: 6264
: 6265 incr N from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do ! FOR EACH UNIT IN CST
: 6266
: 6267 if .CST [.CTLR, .N, D_PRES] eq1 PRESENT ! IF UNIT IS CONFIGURED
: 6268 then
: 6269 begin
: 6270 UNIT = .CST [.CTLR, .N, D_UNIT]; ! GET DRS UNIT NUMBER
: 6271 DUR [.UNIT] = .REASON; ! SET REASON FOR DROPPING UNIT
: 6272 DODU (.UNIT); ! DROP UNIT
: 6273 end;
: 6274
: 6275 end;

```

Address	Label	Code	Operation	Comments	Line No.
000000	004137	000000G	.SBTTL DROP_CTLR GLOBAL ROUTINES		
000004	016646	000014	DROP_CTLR:: JSR R1,\$SAVE3		6251
000010	012746	000037	MOV 14(SP),-(SP)	; CTLR,*	6267
000014	004737	000000G	MOV #37,-(SP)		
000020	010003		JSR PC,BL\$MUL		
000022	012702	000003	MOV R0,R3		
000026	010300		MOV #3,R2	; *,N	6265
000030	060200		MOV R3,R0	; N,*	6267
000032	006300		ADD R2,R0		
000034	032760	040000 000000G	ASL R0		
000042	001412		BIT #40000,CST(R0)		
000044	016001	000000G	BEQ 2\$		
000050	000301		MOV CST(R0),R1	; *,UNIT	6270
000052	042701	177760	SWAB R1	; UNIT	
000056	116661	000016 000000G	BIC #177760,R1	; *,UNIT	
000064	010100		MOVB 16(SP),DUR(R1)	; REASON,*(UNIT)	6271
000066	104451		MOV R1,R0	; UNIT,*	6272
000070	062702	000007	TRAP 51		
000074	020227	000030	2\$: ADD #7,R2	; *,N	6265
000100	003752		CMP R2,#30	; N,*	
000102	022626		BLE 1\$		
000104	000207		CMP (SP)+,(SP)+		6260
			RTS PC		6251

; Routine Size: 35 words, Routine Base: \$CODE\$ + 6504

N15

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (59)

SEQ 0195
Page 195

; Maximum stack depth per invocation: 8 words

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1:82 (60)

SEQ 0196
Page 196

```

: 6276 global routine DRV_CTLERR (CTRL) : novalue *
: 6277
: 6278 :+
: 6279 : THIS ROUTINE IS CALLED BY DRV_TIMCHK AND FATAL_ERROR WHENEVER AN
: 6280 : UNRECOVERABLE CONTROLLER ERROR HAS BEEN DETECTED. ITS PURPOSE IS TO
: 6281 : CLEAN UP ALL CONTROLLER-RELATED DATA IN THE "DRIVER" PORTION OF THE
: 6282 : PROGRAM. THIS INCLUDES MARKING THE CONTROLLER OFFLINE, CLEARING THE
: 6283 : C-RING COUNT, AND DEALLOCATING MSCP PACKETS DESCRIBED IN THE RESPONSE
: 6284 : RING.
: 6285 :
: 6286 : INPUTS:
: 6287 : CTRL - DYING CONTROLLER NUMBER
: 6288 :-
: 6289
: 6290 begin
: 6291
: 6292 local
: 6293 D_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS); ! CONTROLLER'S DCT ADDRESS
: 6294
: 6295 D_ADDR = DCT * (.CTRL * DCT_LEN * 2); ! GET CONTROLLER'S DCT ADDR
: 6296 D_ADDR [WORD0] = OFFLINE; ! MARK DCT OFFLINE AND CLEAR CRING_CNT
: 6297 PUTA_PKT (.CTRL); ! RELEASE ALL PACKETS ALLOCATED TO CONTROLLER
: 6298 DROP_CTRL (.CTRL, DU_CFATAL); ! DROP ALL UNITS ON THE CONTROLLER
: 6299 end; ! ROUTINE DRV_CTLERR
    
```

000000	010146		.SBTTL DRV_CTLERR GLOBAL ROUTINES		
			DRV_CTLERR::		
000002	016601	000004	MOV R1, -(SP)	:	6276
000006	010146		MOV 4(SP), R1	:	6295
000010	012746	000022	MOV R1, -(SP)	:	
000014	004737	000000G	MOV #22, -(SP)	:	
000020	062700	000000G	JSR PC, BL\$MUL	:	
000024	005010		ADD #DCT, R0	:	
000026	010116		CLR (R0)	:	6296
000030	004737	006016'	MOV R1, (SP)	:	6297
000034	010116		JSR PC, PUTA_PKT	:	
000036	012746	000006	MOV R1, (SP)	:	6298
000042	004737	006504'	MOV #6, -(SP)	:	
000046	062706	000006	JSR PC, DROP_CTRL	:	
000052	012601		ADD #5, SP	:	6290
000054	000207		MOV (SP)+, R1	:	6276
			RTS PC	:	

: Routine Size: 23 words, Routine Base: \$CODE\$ + 6612
: Maximum stack depth per invocation: 5 words

ZRQAM2
V01.2RD/RX EXERCISER
GLOBAL ROUTINES5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK:USER2:[DIETZ.RDRX]ZRQACO.9L1;82 (61)SEQ 0197
Page 197

```

6300 global routine SEND (index) =
6301
6302 !+
6303 !
6304 !   IF THE CURRENT RDRX IS ONLINE AND ITS CRING IS NOT FULL, THEN THIS
6305 !   ROUTINE "SENDS" A COMMAND TO THE RDRX BY LOADING THE PACKET
6306 !   DESCRIPTOR OF AN MSCP PACKET INTO THE COMMAND RING AND READING THE
6307 !   DEVICE'S IP REGISTER.  IF THE
6308 !   CURRENT RDRX IS NOT ONLINE, THEN A FAILURE INDICATION IS RETURNED TO
6309 !   THE CALLER, AND NO ACTION IS TAKEN.
6310
6311 !   INPUTS:
6312 !       INDEX - INDEX OF MSCP PACKET CONTAINING THE COMMAND TO
6313 !               BE SENT
6314
6315 !   IMPLICIT INPUTS:
6316 !       CCTRL - CURRENT CONTROLLER NUMBER
6317 !       DCT_ADDR - ADDRESS OF CURRENT CONTROLLER'S DCT
6318 !-
6319 begin
6320
6321 local
6322     SLOT_ADDR,
6323     TEMP : word;
6324
6325 if ((.DCT_ADDR [STAT] eql ONLINE) and
6326     (.DCT_ADDR [CRING_CNT] lssu CRING_LEN)) or
6327     (.MSCP_PKT [.index, OPCODE] eql OP_SCC)
6328 then
6329
6330     if (not ((.MSCP_PKT [.index, OPCODE] eql OP_ACC) or
6331             (.MSCP_PKT [.index, OPCODE] eql OP_ONL) or
6332             (.MSCP_PKT [.index, OPCODE] eql OP_RD) or
6333             (.MSCP_PKT [.index, OPCODE] eql OP_SCC) or
6334             (.MSCP_PKT [.index, OPCODE] eql OP_WRT) OR
6335             (.MSCP_PKT [.INDEX, OPCODE] EQL OP_SDD) OR
6336             (.MSCP_PKT [.INDEX, OPCODE] EQL OP_RCD) OR
6337             (.MSCP_PKT [.INDEX, OPCODE] EQL OP_GDS) OR
6338             (.MSCP_PKT [.INDEX, OPCODE] EQL OP_ELP) OR
6339             (.MSCP_PKT [.INDEX, OPCODE] EQL OP_ABT) OR
6340             (.MSCP_PKT [.INDEX, OPCODE] EQL OP_ESP) ) )
6341     then
6342         begin
6343             PRINTF (DBM107, .MSCP_PKT [.index, OPCODE]);
6344             return FAILURE;
6345         end
6346     else
6347         begin
6348
6349             do
6350                 BREAK
6351             until ((.MSCP_PKT [.index, CMD_TYPE] eql IMM_CMD) and
6352                 (.CREDIT_BAL gequ 1)) or

```

```

! IF DEVICE IS ONLINE AND
! ITS CRING IS NOT FULL
! OR IT IS A SET-CTRL-CHAR COMMAND

```

```

! LOOP TILL CREDIT BALANCE POSITIVE

```



```

:      6353          (.CREDIT_BAL gtru 1);
:      6354
:      6355          MSCP_PKT [.index, CRN_LO] = (CRN_LOW = .CRN_LOW + 1);          ! ASSIGN CMD REF NUM
:      6356
:      6357          if .CRN_LOW eq 0
:      6358          then CRN_HIGH = .CRN_HIGH + 1;
:      6359
:      6360          MSCP_PKT [.index, CRN_HI] = .CRN_HIGH;
:      6361          SLOT_ADDR = .DCT_ADDR [CR_NEXT];          ! ADDR OF NEXT COMMAND SLOT
:      6362
:      6363          DO BREAK
:      6364          UNTIL ((.SLOT_ADDR + 2) and ED_OWN) eq 0);
:      6365
:      6366          SETPRI (PRI07);
:      6367
:      6368          .SLOT_ADDR = .MSCP_PKT [.index, PKT_LO];          ! LOAD BUFF DESC (LO) INTO COMMAND SLOT
:      6369          SLOT_ADDR = .SLOT_ADDR + 2;          ! ADVANCE TO NEXT WORD
:      6370          .SLOT_ADDR = .MSCP_PKT [.index, PKT_HI];          ! LOAD BUFF DESC (HI) INTO COMMAND SLOT
:      6371          .SLOT_ADDR = ..SLOT_ADDR and (not (ED_FLAG));          ! CLEAR INTERRUPT FLAG IN CASE SET
:      6372          .SLOT_ADDR = ..SLOT_ADDR or ED_OWN;          ! GIVE OWNERSHIP TO CONTROLLER
:      6373          SLOT_ADDR = .SLOT_ADDR + 2;          ! ADVANCE TO NEXT COMMAND SLOT
:      6374
:      6375          if .SLOT_ADDR gtra .DCT_ADDR [CR_END]          ! IF BEYOND END OF CRING
:      6376          then
:      6377              SLOT_ADDR = .DCT_ADDR [CR_BEG];          ! CYCLE BACK TO BEGINNING
:      6378
:      6379          DCT_ADDR [CR_NEXT] = .SLOT_ADDR;          ! RESTORE CR_NEXT POINTER IN DCT
:      6380          DCT_ADDR [CRING_CNT] = .DCT_ADDR [CRING_CNT] + 1;          ! INCR # OF COMMANDS IN CRING
:      6381          if (.MSCP_PKT [.index, CONNID] eq CID_MSCP)          ! if MSCP command
:      6382          then (CREDIT_BAL = .CREDIT_BAL - 1);          ! DECREMENT CREDIT BALANCE
:      6383          TEMP = .RDRX_ADDR [RCIP, RC_ALL];          ! READ IP TO FORCE PORT TO POLL
:      6384          SETPRI (PRI00);          ! LOWER PRIORITY
:      6385          return SUCCESS;
:      6386          end
:      6387
:      6388          else
:      6389              return FAILURE;          ! IF DEVICE IS NOT ONLINE
:      6390
:      6391          end;          ! ROUTINE-SEND

```

000000	004137	000000G	SEND::	.SBTTL	SEND GLOBAL ROUTINES		
000004	005746			JSR	R1,\$SAVE2	:	6300
000006	005777	000000G		TST	-(SP)	:	
000012	100004			TST	@DCT.ADDR	:	6325
000014	127727	000000G 000004		BPL	1#	:	
000022	103413			CHPB	@DCT.ADDR,#4	:	6326
000024	016646	000012	1#:	BLO	2#	:	
000030	012746	000104		MOV	12(SP),-(SP)	:	INDEX,#
000034	004737	000000G		MOV	@104, -(SP)	:	6327
000040	022626			JSR	PC,BL#MUL	:	
000042	126027	000022G 000004		CHP	(SP)+,(SP)+	:	
				CHPB	MSCP.PKT+22(R0),#4	:	

E16

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0199
Page 199
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (61)

000050	001170			BNE	10\$		
000052	016646	000012	2\$:	MOV	12(SP), -(SP)	:	INDEX,* 6330
000056	012746	000104		MOV	#104, -(SP)		
000062	004737	000000G		JSR	PC, BL\$MUL		
000066	010002			MOV	R0, R2		
000070	022626			CMP	(SP)+, (SP)+		
000072	005000			CLR	R0		
000074	156200	000022G		BISB	MSCP.PKT+22(R2), R0		
000100	020027	000020		CMP	R0, #20		
000104	001450			BEQ	3\$		
000106	020027	000011		CMP	R0, #11	:	6331
000112	001445			BEQ	3\$		
000114	020027	000041		CMP	R0, #41	:	6332
000120	001442			BEQ	3\$		
000122	020027	000004		CMP	R0, #4	:	6333
000126	001437			BEQ	3\$		
000130	020027	000042		CMP	R0, #42	:	6334
000134	001434			BEQ	3\$		
000136	020027	000004		CMP	R0, #4	:	6335
000142	001431			BEQ	3\$		
000144	020027	000005		CMP	R0, #5	:	6336
000150	001426			BEQ	3\$		
000152	020027	000001		CMP	R0, #1	:	6337
000156	001423			BEQ	3\$		
000160	020027	000003		CMP	R0, #3	:	6338
000164	001420			BEQ	3\$		
000166	020027	000006		CMP	R0, #6	:	6339
000172	001415			BEQ	3\$		
000174	020027	000002		CMP	R0, #2	:	6340
000200	001412			BEQ	3\$		
000202	010046			MOV	R0, -(SP)	:	6343
000204	012746	000000G		MOV	#DBM107, -(SP)		
000210	012746	000002		MOV	#2, -(SP)		
000214	010600			MOV	SP, R0	:	SP,*
000216	104417			TRAP	17		
000220	062706	000006		ADD	#6, SP	:	6342
000224	000502			BR	10\$:	6330
000226	104422		3\$:	TRAP	22	:	6349
000230	005762	000004G		TST	MSCP.PKT+4(R2)	:	6351
000234	001003			BNE	4\$		
000236	005737	000000G		TST	CREDIT.BAL	:	6352
000242	001004			BNE	5\$		
000244	023727	000000G 000001	4\$:	CMP	CREDIT.BAL, #1	:	6353
000252	101765			BLOS	3\$		
000254	013700	000000G	5\$:	MOV	CRN.LOW, R0	:	6355
000260	005200			INC	R0		
000262	010037	000000G		MOV	R0, CRN.LOW		
000266	010062	000012G		MOV	R0, MSCP.PKT+12(R2)		
000272	001002			BNE	6\$:	6357
000274	005237	000000G		INC	CRN.HIGH	:	6358
000300	013762	000000G 000014G	6\$:	MOV	CRN.HIGH, MSCP.PKT+14(R2)	:	6360
000306	013700	000000G		MOV	DCT.ADDR, R0	:	6361
000312	016001	000020		MOV	20(R0), R1	:	*.SLOT.ADDR

F16

ZRQAM2	RD/RX EXERCISER							
V01.2	GLOBAL ROUTINES							
000316	104422			7\$:	TRAP	22		
000320	032761	100000	000002		BIT	#-100000,2(R1)		6363
000326	001373				BNE	7\$		6364
000330	012700	000340			MOV	#340,R0		
000334	104441				TRAP	41		6366
000336	016221	000000G			MOV	MSCP.PKT(R2),(R1)+		6368
000342	016211	000002G			MOV	MSCP.PKT+2(R2),(R1)		6370
000346	042711	040000			BIC	#40000,(R1)		6371
000352	052721	100000			BIS	#100000,(R1)+		6372
000356	013700	000000G			MOV	DCT.ADDR,R0		6375
000362	020160	000012			CMP	R1,12(R0)		
000366	101402				BLOS	8\$		
000370	016001	000010			MOV	10(R0),R1		6377
000374	010160	000020		8\$:	MOV	R1,20(R0)		6379
000400	105210				INCB	(R0)		6380
000402	105762	000011G			TSTB	MSCP.PKT+11(R2)		6381
000406	001002				BNE	9\$		
000410	005337	000000G			DEC	CREDIT.BAL		6382
000414	017716	000000G		9\$:	MOV	#RDRX.ADDR,(SP)		6383
000420	005000				CLR	R0		6384
000422	104441				TRAP	41		
000424	012700	000001			MOV	#1,R0		6330
000430	000401				BR	11\$		6319
000432	005000			10\$:	CLR	R0		
000434	005726			11\$:	TST	(SP)+		
000436	000207				RTS	PC		6300

: Routine Size: 144 words, Routine Base: \$CODE\$ + 6670
 : Maximum stack depth per invocation: 9 words

G16

ZRQAM2
V01.2

RD/RX EXERCISER
GLOBAL ROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (62)

SEQ 0201
Page 201

```

:      6392 global routine WAIT : novalue =
:      6393
:      6394 !+
:      6395 !   THE PURPOSE OF THIS ROUTINE IS TO KILL TIME UNTIL AN RDRX INTERRUPT
:      6396 !   RESULTS IN A RETURN PACKET INDEX BEING DEPOSITED INTO THE I/O DONE
:      6397 !   QUEUE (IODQ).
:      6398 !-
:      6399 do
:      6400     BREAK                               ! BREAK FOR ACT
:      6401 until (.IODQ_IN neq .IODQ_OUT);
    
```

```

000000 104422          .SBTTL  WAIT GLOBAL ROUTINES
000000          WAIT::
000002 023737 000000G 000000G          1$: TRAP 22          : 6399
000010 001773          CMP  IODQ.IN,IODQ.OUT      : 6401
000012 000207          BEQ  1$
                                RTS  PC          : 6392
    
```

```

: Routine Size: 6 words,      Routine Base: $CODE$ + 7330
: Maximum stack depth per invocation: 2 words
    
```

```

6402 global routine MODULAS (LO_LIMIT, HI_LIMIT) =
6403
6404 !+
6405 ! THE PURPOSE OF THIS ROUTINE IS TO GET A RANDOM NUMBER
6406 ! BETWEEN LO AND HI LIMITS. THIS SHOULD WORK FOR A 16 BIT
6407 ! WORD. THE "MOD"FUNC ONLY WORKS ON 15 BITS.
6408 !-
6409 begin
6410 OWN X : word; ! VARIABLE FOR RANDOM WORD TABLE
6411 LOCAL ANSWER : unsigned word, ! FINAL ANSWER
6412 SAVESZ : UNSIGNED WORD, ! SAVES THE SIZE OF WINDOW
6413 SIZE : unsigned word; ! SIZE OF WINDOW
6414
6415 X = .X + 1;
6416 IF .X GEQ RDM_LEN
6417 THEN X = 0; ! KEEP ROTATING RANDOM NUMBERS BEING USED
6418
6419 SIZE = .HI_LIMIT - .LO_LIMIT;
6420 SAVESZ = .HI_LIMIT - .LO_LIMIT;
6421 IF (.SIZE lequ #0'077777') ! IF 15TH BIT NOT SET
6422 THEN ANSWER = ((.RANDOM [.X] and #0'077777') MOD (.SIZE + 1)) ! ONLY 15 BIT WORD SO TAKE RANDOM SAMPLE
6423 ELSE ! 16 BIT WORD
6424 begin
6425 SIZE = .SIZE + -1; ! MAKES SIZE A 15 BIT LENGHT OR DIVIDE IN HALF
6426 ANSWER = (.RANDOM [.X] and #0'077777') MOD (.SIZE + 1); ! GETS 15 BIT RANDOM #
6427 ANSWER = .ANSWER + 1; ! BUILD UP TO REGULAR SIZE
6428 ANSWER = .ANSWER + (.RANDOM [.X + 1] and 1); ! RANDOMLY FILL 1ST BIT
6429 if (.ANSWER gtru SAVESZ)
6430 then ANSWER = .SAVESZ; ! IT IS POSSIBLE TO BE 1 LARGER THAN SIZE SO CHECK
6431 end; ! AND MAKE CORRECTION
6432 return .ANSWER;
6433 end; ! end of MODULAS ROUTINE
    
```

007344

X: .BLKW 1

		.SBTTL MODULAS GLOBAL ROUTINES		
000000	004137	000000G	MODULAS::	
000004	005746		JSR R1, \$SAVE2	6402
000006	005237	007344'	TST -(SP)	
000012	023727	007344' 000020	INC X	6415
000020	002402		CMP X, #20	6416
000022	005037	007344'	BLT 1#	
000026	016600	000012	CLR X	6417
000032	166600	000014	1#:	6419
000036	010001		MOV 12(SP), R0	
000040	010016		SUB 14(SP), R0	
000042	013700	007344'	MOV R0, R1	
000046	006300		MOV R0, (SP)	6420
000050	020127	077777	MOV X, R0	6422
000054	101011		ASL R0	
			CMP R1, #77777	
			BHI 2#	6421

I16

ZRQAM2	RD/RX EXERCISER		5-Dec-1983 10:27:14	VAX-11 Bliss-16 V3-555	SEQ 0203
V01.2	GLOBAL ROUTINES		5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (63)	Page 203
000056	016046	000000G			
000062	042716	100000			6422
000066	010146				
000070	005216				
000072	004737	000000G			
000076	000431				
000100	006201		2\$:		6421
000102	016046	000000G			6425
000106	042716	100000			6426
000112	010146				
000114	005216				
000116	004737	000000G			
000122	006300				
000124	013701	007344'			
000130	006301				
000132	116102	000002G			
000136	042702	177776			
000142	060200				
000144	012701	000004			
000150	060601				
000152	020001				
000154	101402				
000156	016600	000004			
000162	062706	000006	3\$:		6427
000166	000207				6428

: Routine Size: 60 words, Routine Base: \$CODE\$ + 7346
 : Maximum stack depth per invocation: 7 words

```

: 6434 %sbttl 'ERROR MESSAGE SUBROUTINES'
: 6435
: 6436 routine EMS_SA : novalue =
: 6437
: 6438 !+
: 6439 ! THIS ROUTINE PRINTS (EXTENDED) THE GLOBAL DATUM "SA_REG" WHICH CONTAINS
: 6440 ! THE CONTENTS OF THE SA REGISTER.
: 6441 !-
: 6442
: 6443 if .SA_REG eql %o'177777' ! IF CONTROLLER TIME-OUT
: 6444 then
: 6445 begin
: 6446 PRINTX (CRLF);
: 6447 PRINTX (ASTERISK);
: 6448 PRINTX (.CNTR_ERR [0]);
: 6449 end
: 6450 else
: 6451
: 6452 if (.SA_REG and %o'003777') lequ 22 ! IF GENERIC CONTROLLER ERROR
: 6453 then
: 6454 begin
: 6455 PRINTX (CRLF);
: 6456 PRINTX (ASTERISK);
: 6457 PRINTX (.CNTR_ERR [.SA_REG and %o'003777']);
: 6458 end
: 6459 else
: 6460
: 6461 if ((.SA_REG and %o'003777') - 400) lequ 6 ! IF RDRX SPECIFIC CONTROLLER ERROR
: 6462 then
: 6463 begin
: 6464 PRINTX (CRLF);
: 6465 PRINTX (ASTERISK);
: 6466 PRINTX (.RDRX_ERR [(.SA_REG and %o'003777') - 400]);
: 6467 end
: 6468 else
: 6469 PRINTX (XX14, .SA_REG); ! JUST PRINT CONTENTS OF SA

```

000000	010146		.SBTTL	EMS_SA ERROR MESSAGE SUBROUTINES	
000002	013701	000000G	EMS_SA: MOV	R1, -(SP)	6436
000006	020127	177777	MOV	SA_REG, R1	6443
000012	001023		CMP	R1, #-1	
000014	012746	000000G	BNE	1#	
000020	012746	000001	MOV	#CRLF, -(SP)	6446
000024	010600		MOV	#1, -(SP)	
000026	104415		MOV	SP, R0	: SP, *
000030	012716	000000G	TRAP	15	
000034	012746	000001	MOV	#ASTERISK, (SP)	: 6447
000040	010600		MOV	#1, -(SP)	
000042	104415		MOV	SP, R0	: SP, *
000044	013716	000000G	TRAP	15	
000050	012746	000001	MOV	CNTR_ERR, (SP)	: 6448
			MOV	#1, -(SP)	

000054	010600		MOV	SP,R0	; SP,*	
000056	104415		TRAP	15		
000060	000475		BR	3#		
000062	010100	1#:	MOV	R1,R0		6445
000064	042700	174000	BIC	#174000,R0		6452
000070	020027	000026	CMP	R0,#26		
000074	101030		BHI	2#		
000076	012746	000000G	MOV	#CRLF,-(SP)		6455
000102	012746	000001	MOV	#1,-(SP)		
000106	010600		MOV	SP,R0	; SP,*	
000110	104415		TRAP	15		
000112	012716	000000G	MOV	#ASTERISK,(SP)		6456
000116	012746	000001	MOV	#1,-(SP)		
000122	010600		MOV	SP,R0	; SP,*	
000124	104415		TRAP	15		
000126	013700	000000G	MOV	SA.REG,R0		6457
000132	042700	174000	BIC	#174000,R0		
000136	006300		ASL	R0		
000140	016016	000000G	MOV	CNTR.ERR(R0),(SP)		
000144	012746	000001	MOV	#1,-(SP)		
000150	010600		MOV	SP,R0	; SP,*	
000152	104415		TRAP	15		
000154	000437		BR	3#		6454
000156	010100	2#:	MOV	R1,R0		6461
000160	042700	174000	BIC	#174000,R0		
000164	162700	000620	SUB	#620,R0		
000170	020027	000006	CMP	R0,#6		
000174	101031		BHI	4#		
000176	012746	000000G	MOV	#CRLF,-(SP)		6464
000202	012746	000001	MOV	#1,-(SP)		
000206	010600		MOV	SP,R0	; SP,*	
000210	104415		TRAP	15		
000212	012716	000000G	MOV	#ASTERISK,(SP)		6465
000216	012746	000001	MOV	#1,-(SP)		
000222	010600		MOV	SP,R0	; SP,*	
000224	104415		TRAP	15		
000226	013700	000000G	MOV	SA.REG,R0		6466
000232	042700	174000	BIC	#174000,R0		
000236	006300		ASL	R0		
000240	016016	176340G	MOV	RDRX.ERR-1440(R0),(SP)		
000244	012746	000001	MOV	#1,-(SP)		
000250	010600		MOV	SP,R0	; SP,*	
000252	104415		TRAP	15		
000254	005726	3#:	TST	(SP)+		6463
000256	000407		BR	5#		6461
000260	010146	4#:	MOV	R1,-(SP)		6469
000262	012746	000000G	MOV	#XX14,-(SP)		
000266	012746	000002	MOV	#2,-(SP)		
000272	010600		MOV	SP,R0	; SP,*	
000274	104415		TRAP	15		
000276	062706	000006	5#:	ADD	#6,SP	6443
000302	012601		MOV	(SP)+,R1		6436
000304	000207		RTS	PC		

L16

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (64)

SEQ 0206
Page 206

; Routine Size: 99 words, Routine Base: \$CODE\$ + 7536
; Maximum stack depth per invocation: 7 words

```

: 6470 routine EMS_SBC : novalue =
: 6471
: 6472 !+
: 6473 ! THIS ROUTINE PRINTS THE GLOBAL DATUM "SB_CODE" (SUB-CODE) IF
: 6474 ! EITHER THE STATUS CODE (ST_CODE) OR THE SUB-CODE IS NON-ZERO. (A
: 6475 ! NON-ZERO SUB-CODE ALWAYS HAS SIGNIFICANCE, WHEREAS A ZERO SUB-CODE ONLY
: 6476 ! HAS MEANING WITH A NON-ZERO STATUS CODE).
: 6477 !-
: 6478
: 6479 begin
: 6480
: 6481 if (.ST_CODE or .SB_CODE) neq 0 ! PRINT SUB-CODE ONLY ON ERROR
: 6482 then
: 6483 begin
: 6484 PRINTB (XX16); ! SUB-CODE :
: 6485
: 6486 case .ST_CODE from ST_SUC to ST_DRV of
: 6487 set
: 6488
: 6489 [ST_SUC]: if .SB_CODE lequ 16 ! SUCCESS SUB-CODES
: 6490 then PRINTB (.TBL_SUC [.SB_CODE]);
: 6491
: 6492 [ST_CMD]: PRINTB (EX_OP, .SB_CODE / 8); ! INVALID COMMAND
: 6493
: 6494 [ST_ABO]: ; ! COMMAND ABORTED
: 6495
: 6496 [ST_OFL]: if .SB_CODE lequ 8 ! UNIT OFFLINE
: 6497 then PRINTB (.TBL_OFL [.SB_CODE]);
: 6498
: 6499
: 6500
: 6501 [ST_AVL]: ; ! UNIT AVAILABLE
: 6502
: 6503 [ST_MFE]: if .SB_CODE lequ 10 ! MEDIA FORMAT ERROR
: 6504 then PRINTB (.TBL_MFE [.SB_CODE]);
: 6505
: 6506
: 6507 [ST_WPT]: if (.SB_CODE / 128) lequ 2 ! WRITE PROTECTED
: 6508 then PRINTB (.TBL_WPT [(.SB_CODE / 128)]);
: 6509
: 6510
: 6511 [ST_CMP]: ; ! COMPARE ERROR
: 6512
: 6513 [ST_DAT]: if .SB_CODE lequ 15 ! DATA ERROR
: 6514 then PRINTB (.TBL_DAT [.SB_CODE]);
: 6515
: 6516
: 6517 [ST_HST]: if .SB_CODE lequ 4 ! HOST ACCESS ERROR
: 6518 then PRINTB (.TBL_HST [.SB_CODE]);
: 6519
: 6520
: 6521 [ST_CNT]: if .SB_CODE lequ 3 ! CONTROLLER ERROR
: 6522 then

```

B1

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0208
Page 208
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (65)

```

:      6523          PRINTB (.TBL_CNT [.SB_CODE]);
:      6524
:      6525          [ST_DRV]:      if .SB_CODE lequ 8          ! DRIVE ERROR
:      6526          then
:      6527          PRINTB (.TBL_DRV [.SB_CODE]);
:      6528
:      6529          [outrange]:    PRINTB (EX_OP, .SB_CODE);      ! JUST PRINT SUB-CODE IF NO MATCH
:      6530          tes;
:      6531
:      6532          end;
:      6533
:      6534          end;

```

000000	013700	000000G		.SBTTL	EMS.SBC ERROR MESSAGE SUBROUTINES	
000004	053700	000000G		EMS.SBC:MOV	ST.CODE,RO	6481
000010	001001			BIS	SB.CODE,RO	
000012	000207			BNE	1#	
000014	012746	000000G		RTS	PC	
000020	012746	000001		1#: MOV	@XX16,-(SP)	6484
000024	010600			MOV	@1,-(SP)	
000026	104414			MOV	SP,RO	: SP,+
000030	013700	000000G		TRAP	14	
000034	020027	000013		MOV	ST.CODE,RO	6486
000040	101003			CMP	RO,#13	
000042	006300			BHI	3#	
000044	066007	000000'		ASL	RO	
000050	013716	000000G		ADD	P.AAA(RO),PC	: Case dispatch
000054	012746	000000G		3#: MOV	SB.CODE,(SP)	6529
000060	012746	000002		MOV	@EX.OP,-(SP)	
000064	010600			MOV	@2,-(SP)	
000066	104414			MOV	SP,RO	: SP,+
000070	022626			TRAP	14	
000072	000435			CMP	(SP)+,(SP)+	
000074	023727	000000G	000020	BR	6#	6486
000102	101165			4#: CMP	SB.CODE,#20	6489
000104	013700	000000G		BHI	14#	
000110	006300			MOV	SB.CODE,RO	6491
000112	016016	000000'		ASL	RO	
000116	012746	000001		MOV	TBL.SUC(RO),(SP)	
000122	010600			MOV	@1,-(SP)	
000124	104414			MOV	SP,RO	: SP,+
000126	000565			TRAP	14	
000130	013716	000000G		BR	15#	
000134	012746	000010		5#: MOV	SB.CODE,(SP)	6493
000140	004737	000000G		MOV	@10,-(SP)	
000144	010016			JSR	PC,BL#DIV	
000146	012746	000000G		MOV	RO,(SP)	
000152	012746	000002		MOV	@EX.OP,-(SP)	
000156	010600			MOV	@2,-(SP)	
000160	104414			MOV	SP,RO	: SP,+
000162	062706	000006		TRAP	14	
				ADD	@6,SP	

ZRQAM2
V01.2RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (65)SEQ 0209
Page 209

000166	000546		6#:	BR	16#	:	6486
000170	023727	000000G 000010	7#:	CMP	SB.CODE,#10	:	6497
000176	101142			BHI	16#	:	
000200	013700	000000G		MOV	SB.CODE,RO	:	6499
000204	006300			ASL	RO		
000206	016016	000042'		MOV	TBL.OFL(RO),(SP)		
000212	012746	000001		MOV	#1,-(SP)		
000216	010600			MOV	SP,RO	: SP,*	
000220	104414			TRAP	14		
000222	000527			BR	15#		
000224	023727	000000G 000012	8#:	CMP	SB.CODE,#12	:	6503
000232	101124			BHI	16#	:	
000234	013700	000000G		MOV	SB.CODE,RO	:	6505
000240	006300			ASL	RO		
000242	016016	000064'		MOV	TBL.MFE(RO),(SP)		
000246	012746	000001		MOV	#1,-(SP)		
000252	010600			MOV	SP,RO	: SP,*	
000254	104414			TRAP	14		
000256	000511			BR	15#		
000260	013716	000000G	9#:	MOV	SB.CODE,(SP)	:	6507
000264	012746	000200		MOV	#200,-(SP)		
000270	004737	000000G		JSR	PC,BL\$DIV		
000274	005726			TST	(SP),*		
000276	020027	000002		CMP	RO,#2		
000302	101100			BHI	16#	:	
000304	006300			ASL	RO		6509
000306	016016	000112'		MOV	TBL.WPT(RO),(SP)		
000312	012746	000001		MOV	#1,-(SP)		
000316	010600			MOV	SP,RO	: SP,*	
000320	104414			TRAP	14		
000322	000467			BR	15#		
000324	023727	000000G 000017	10#:	CMP	SB.CODE,#17	:	6513
000332	101064			BHI	16#	:	
000334	013700	000000G		MOV	SB.CODE,RO	:	6515
000340	006300			ASL	RO		
000342	016016	000120'		MOV	TBL.DAT(RO),(SP)		
000346	012746	000001		MOV	#1,-(SP)		
000352	010600			MOV	SP,RO	: SP,*	
000354	104414			TRAP	14		
000356	000451			BR	15#		
000360	023727	000000G 000004	11#:	CMP	SB.CODE,#4	:	6517
000366	101046			BHI	16#	:	
000370	013700	000000G		MOV	SB.CODE,RO	:	6519
000374	006300			ASL	RO		
000376	016016	000160'		MOV	TBL.HST(RO),(SP)		
000402	012746	000001		MOV	#1,-(SP)		
000406	010600			MOV	SP,RO	: SP,*	
000410	104414			TRAP	14		
000412	000433			BR	15#		
000414	023727	000000G 000003	12#:	CMP	SB.CODE,#3	:	6521
000422	101030			BHI	16#	:	
000424	013700	000000G		MOV	SB.CODE,RO	:	6523
000430	006300			ASL	RO		

D1

000432	016016	000172'		MOV	TBL.CNT(R0),(SP)		
000436	012746	000001		MOV	#1,-(SP)		
000442	010600			MOV	SP,R0	; SP,*	
000444	104414			TRAP	14		
000446	000415			BR	15#		
000450	023727	000000G 000010	13#:	CMP	SB.CODE,#10		6525
000456	101012		14#:	BHI	16#		
000460	013700	000000G		MOV	SB.CODE,R0		6527
000464	006300			ASL	R0		
000466	016016	000202'		MOV	TBL.DRV(R0),(SP)		
000472	012746	000001		MOV	#1,-(SP)		
000476	010600			MOV	SP,R0	; SP,*	
000500	104414			TRAP	14		
000502	005726		15#:	TST	(SP)+		
000504	022626		16#:	CMP	(SP)+,(SP)+		6483
000506	000207			RTS	PC		6470

; Routine Size: 164 words, Routine Base: \$CODE\$ + 10044
 ; Maximum stack depth per invocation: 7 words

000000				.PSECT	\$PLIT\$, R0, D		
		P.AAA:					
		2#:		.WORD	24	; CASE Table for EMS.SBC+0044	6486
000000	000024			.WORD	60	; [4#]	
000002	000060			.WORD	434	; [5#]	
000004	000434			.WORD	120	; [16#]	
000006	000120			.WORD	434	; [7#]	
000010	000434			.WORD	154	; [16#]	
000012	000154			.WORD	210	; [8#]	
000014	000210			.WORD	434	; [9#]	
000016	000434			.WORD	254	; [16#]	
000020	000254			.WORD	310	; [10#]	
000022	000310			.WORD	344	; [11#]	
000024	000344			.WORD	400	; [12#]	
000026	000400			.WORD		; [13#]	

```

: 6535 GLOBAL routine EMSCMD : novalue =
: 6536
: 6537 !!+
: 6538 ! THIS ROUTINE PRINTS THE ENTIRE RETURN PACKET INCLUDING OPCODE,
: 6539 ! STATUS, SUB-STATUS, MODIFIERS OR FLAGS, AND ETC.
: 6540 ! THESE FIELDS ARE "TRANSLATED" INTO ENGLISH TEXT IF POSSIBLE
: 6541 ! RATHER THAN PRINTED AS RAW NUMBERS.
: 6542 !
: 6543 ! IMPLICIT INPUTS:
: 6544 ! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 6545 !-
: 6546 begin
: 6547
: 6548 OWN
: 6549 EBH_TB1 : VECTOR [7] INITIAL (EBH_30,EBH_44,EBH_45,
: 6550 EBH_46,EBH_47,EBH_48,EBH_49);
: 6551
: 6552 ! TABLE OF BASIC, HARD ERROR MESSAGE ADDRESSES, INDEXED BY STATUS CODE
: 6553
: 6554 PRINTB (XX13, .CDISK); ! "DISK XXX"
: 6555 !PRINTB (XX36, .CRN_LOW); ! EXPECTED CRN : XXXXXX
: 6556 PRINTX (XX35, .RP_ADDR [CRF_HI], .RP_ADDR [CRF_LO]); ! RECEIVED CRN : XXXXXX
: 6557 printx (xx29); ! "message type:"
: 6558 SELECTU (.RP_ADDR [MESTYP]) OF
: 6559 SET
: 6560 [#o'0']: PRINTX (EX_SEQ); ! "SEQUENTIAL"
: 6561 [#o'1']: PRINTX (EX_DGM); ! "DATAGRAM"
: 6562 [#o'2']: PRINTX (EX_CRD); ! "CREDIT NOTIFICATION" PACKET TYPE
: 6563 [#o'15']: PRINTX (EX_MTN); ! "MAINTENANCE"
: 6564 [OTHERWISE]: PRINTX (XX37, .RP_ADDR [MESTYP]); ! UNKOWN MESSAGE TYPE
: 6565 TES;
: 6566
: 6567 PRINTB (XX17); ! "COMMAND: "
: 6568
: 6569 SELECTU (.RP_ADDR [conid]) OF
: 6570 SET
: 6571 [#o'2']:
: 6572 BEGIN
: 6573 PRINTB (XX18); ! PRINTS -DUP-
: 6574 SELECTU (.RP_ADDR [ENDCOD]) OF
: 6575 SET
: 6576 [#o'201']: PRINTB (EX_GDS);
: 6577 [#o'202']: PRINTB (EX_ESP); ! PRINTS A COMMAND
: 6578 [#o'203']: PRINTB (EX_ELP);
: 6579 [#o'205']: PRINTB (EX_RCD);
: 6580 [#o'204']: PRINTB (EX_SDD);
: 6581 [#o'206']: PRINTB (EX_ABP);
: 6582 [OTHERWISE]: PRINTB (EX_OP, .RP_ADDR [ENDCOD]); ! PRINT ENDCODE VALUE
: 6583 TES;
: 6584 printb (xx15); ! "status:"
: 6585 IF (.RP_ADDR [STSCOD] GEQU 0) AND (.RP_ADDR [STSCOD] LEQU 7) ! IF STATUS CODE IS WITHIN RANGE
: 6586 THEN PRINTB (.EBH_TB1 [.RP_ADDR [STSCOD]]) ! PRINTB APPROPRIATE MESSAGE
: 6587 ELSE PRINTB (ex_op, .RP_ADDR [STSCOD]); ! JUST PRINT STATUS CODE

```

```

: 6588
: 6589 IF .RP_ADDR [ENDCOD] EQL #0'204' or
: 6590 .RP_ADDR [ENDCOD] EQL #0'205' ! IF A SEND DATA OR RECEIVE DATA COMMAND THEN
: 6591 then
: 6592 begin
: 6593 PRINTX (XX25, .RP_ADDR [BCNT_LO]); ! FOR ANY "ACTUAL # OF BYTES TRANSFERRED: XXXXX."
: 6594 PRINTX (XX26, .RP_ADDR [BUFF_1], .RP_ADDR [BUFF_0]); ! "I/O BUFFER DESCRIPTOR: XXXXXX XXXXXX"
: 6595 EMS_DUP (); ! prints contents of dup packet
: 6596 end;
: 6597 IF .RP_ADDR [ENDCOD] EQL #0'201'
: 6598 then
: 6599 begin
: 6600 if BIT_TST (RP_ADDR [9, 8, 1, 0], 1)
: 6601 then PRINTB (df_0);
: 6602 if BIT_TST (RP_ADDR [9, 9, 1, 0], 1)
: 6603 then PRINTB (df_1);
: 6604 if BIT_TST (RP_ADDR [9, 10, 1, 0], 1)
: 6605 then PRINTB (df_2);
: 6606 if BIT_TST (RP_ADDR [9, 11, 1, 0], 1)
: 6607 then PRINTB (df_3);
: 6608 end;
: 6609
: 6610 IF .RP_ADDR [ENDCOD] EQL #0'203' ! IF A GET DUST STATUS OR EXEC. LOCAL PRG COMMAND TH
: 6611 then
: 6612 begin
: 6613 if BIT_TST (RP_ADDR [9, 8, 1, 0], 1)
: 6614 then PRINTB (df_4);
: 6615 if BIT_TST (RP_ADDR [9, 9, 1, 0], 1)
: 6616 then PRINTB (df_5);
: 6617 if BIT_TST (RP_ADDR [9, 10, 1, 0], 1)
: 6618 then PRINTB (df_6);
: 6619 if BIT_TST (RP_ADDR [9, 11, 1, 0], 1)
: 6620 then PRINTB (df_7);
: 6621 end;
: 6622 PRINTX (XX23, .CST_ADDR [.CUOFF + 5, D_DBN], .CST_ADDR [.CUOFF + 5, D_DBN]); ! "DBN: XXXXXX."
: 6623 END;
: 6624 [#0'0']:
: 6625 BEGIN
: 6626 PRINTB (XX19); !PRINTS -MSCP- !MSC
: 6627 SELECTU (.RP_ADDR [ENDCOD]) OF
: 6628 SET
: 6629 [#0'204']: PRINTB (EX_SCC);
: 6630 [#0'211']: PRINTB (EX_ONL);
: 6631 [#0'220']: PRINTB (EX_ACC);
: 6632 [#0'241']: PRINTB (EX_RD); ! PRINTS THE COMMAND
: 6633 [#0'242']: PRINTB (EX_WRT);
: 6634 [OTHERWISE]: PRINTB (EX_OP, .RP_ADDR [ENDCOD]); ! PRINT ENDCODE VALUE
: 6635 YES;
: 6636 if .RP_ADDR [CMDMOD] eq1 MD_CMP THEN PRINTB (XX20); ! PRINTS THE MODIFIER IF NECESSARY
: 6637
: 6638 PRINTB (XX15); ! STATUS:
: 6639 if (.ST_CODE gtru 0) and ! IF STATUS CODE IS WITHIN RANGE
: 6640 (.ST_CODE lequ 11)

```

```

:      6641      then
:      6642          PRINTB (.ERR_COD [.ST_CODE - 1])          ! PRINTB APPROPRIATE MESSAGE
:      6643      else
:      6644
:      6645          if .ST_CODE eq1 ST_DIA
:      6646              then
:      6647                  PRINTB (.ERR_COD [11])          ! MESSAGE FROM INTERNAL DIAGNOSTICS
:      6648              else
:      6649                  PRINTB (EX_OP, .ST_CODE);          ! JUST PRINT STATUS CODE WHEN NO MATCH
:      6650
:      6651      EMS_SBC ();          ! PRINTS STATUS SUB-CODE
:      6652
:      6653      IF .RP_ADDR [ENDCOD] EQLU %o'220' OR
:
:      6654          .RP_ADDR [ENDCOD] EQLU %o'241' OR
:      6655          .RP_ADDR [ENDCOD] EQLU %o'242'
:      6656      THEN
:
:      6657          begin
:      6658              printX (XX24, .rp_addr [CBCNT_LO]);          ! "BYTE COUNT IN COMMAND: XXXXXXXX"
:      6659              PRINTX (XX25, .RP_ADDR [BCNT_LO]);          ! FOR ANY "ACTUAL # OF BYTES TRANSFERRED: XXXXX."
:      6660              PRINTX (XX26, .RP_ADDR [BUFF_1], .RP_ADDR [BUFF_0]); ! "I/O BUFFER DESCRIPTOR: XXXXXX XXXXXX"
:      6661              if BIT_TST (RP_ADDR [FLAGS], EF_0)          ! IF BAD BLOCK REPORTED
:      6662                  then
:      6663                      PRINTB (XX21, .RP_ADDR [BBLK_HI], .RP_ADDR [BBLK_LO]) ! "BAD BLOCK REPORTED: XXXXXX."
:      6664                  else
:      6665                      printX (XX22, .RP_ADDR [LBN_LO], .RP_ADDR [LBN_HI], .RP_ADDR [LBN_LO]); ! "LBN: XXXX
:
:      6666              PRINTB (XX41);
:      6667              if BIT_TST (RP_ADDR [FLAGS], EF_1)          ! IF BAD BLOCK UNREPORTED
:      6668                  then PRINTB (F_1);
:      6669              if BIT_TST (RP_ADDR [FLAGS], EF_2)          ! IF ERROR LOG GENERATED
:      6670                  then PRINTB (F_2);
:      6671              if BIT_TST (RP_ADDR [FLAGS], EF_3)          ! IF SERIOUS EXCEPTION
:      6672                  then PRINTB (F_3);
:      6673              END;
:      6674      IF .RP_ADDR [ENDCOD] EQLU %o'204'
:
:      6675          THEN
:      6676              begin
:      6677                  PRINTB (XX39);
:      6678                  if BIT_TST (RP_ADDR [BCNT_HI], EF_4)          ! IF
:      6679                      then PRINTB (F_4);
:      6680                  if BIT_TST (RP_ADDR [BCNT_HI], EF_5)          ! IF
:      6681                      then PRINTB (F_5);
:      6682                  if BIT_TST (RP_ADDR [BCNT_HI], EF_6)          ! IF
:      6683                      then PRINTB (F_6);
:      6684                  if BIT_TST (RP_ADDR [BCNT_HI], EF_7)          ! IF
:      6685                      then PRINTB (F_7);
:      6686                  if BIT_TST (RP_ADDR [BCNT_HI], EF_8)          ! IF
:      6687                      then PRINTB (F_8);
:      6688                  if BIT_TST (RP_ADDR [BCNT_HI], EF_9)          ! IF
:      6689                      then PRINTB (F_9);
:      6690                  if BIT_TST (RP_ADDR [BCNT_HI], EF_10)          ! IF
:      6691                      then PRINTB (F_10);
:      6692              end;
:      6693      IF .RP_ADDR [ENDCOD] EQLU %o'211'

```



```

:      6694      THEN                                     ! MSCP ONLINE comman
d
:      6695          begin
:      6696              PRINTB (XX40);
:      6697              if BIT_TST (RP_ADDR [BCNT_HI], EF_11) ! IF
:      6698                  then PRINTB (F_11);
:      6699              if BIT_TST (RP_ADDR [BCNT_HI], EF_12) ! IF
:      6700                  then PRINTB (F_12);
:      6701              if BIT_TST (RP_ADDR [BCNT_HI], EF_13) ! IF
:      6702                  then PRINTB (F_13);
:      6703              if BIT_TST (RP_ADDR [BCNT_HI], EF_14) ! IF
:      6704                  then PRINTB (F_14);
:      6705              if BIT_TST (RP_ADDR [BCNT_HI], EF_15) ! IF
:      6706                  then PRINTB (F_15);
:      6707              if BIT_TST (RP_ADDR [BCNT_HI], EF_16) ! IF
:      6708                  then PRINTB (F_16);
:      6709              if BIT_TST (RP_ADDR [BCNT_HI], EF_17) ! IF
:      6710                  then PRINTB (F_17);
:      6711              if BIT_TST (RP_ADDR [BCNT_HI], EF_18) ! IF
:      6712                  then PRINTB (F_18);
:      6713              if BIT_TST (RP_ADDR [BCNT_HI], EF_19) ! IF
:      6714                  then PRINTB (F_19);
:      6715              if BIT_TST (RP_ADDR [BCNT_HI], EF_20) ! IF
:      6716                  then PRINTB (F_20);
:      6717              if BIT_TST (RP_ADDR [BCNT_HI], EF_21) ! IF
:      6718                  then PRINTB (F_21);
:      6719          end;
:      6720      END;
:      6721      tes;                                     !+
:      6722          !-                                     !-
:      6723          !-                                     !-
:      6724
:      6725      !PRINTX (XX27);                         ! "CONTENTS OF PACKET;"
:      6726      !EMS_BLK (.RP_ADDR, PKT_LEN);          ! PRINT BLOCK OF WORDS AS LONG AS A MESSAGE PACKET INCASE A PACKET IS USED
:      6727          ! INSTEAD OF A RETURN PACKET
:      6728      !PRINTX (XX42);                         ! PRINTS CONTENTS OF DUP I/O PACKET
:      6729      !EMS_BLK (DUPPKT, 3);
:      6730      PRINTB (CRLF);                          ! PRINTS CARRIAGE RETURN AFTER ERROR MESSAGE
:      6731      end;                                     ! ROUTINE EMSCMD

```

010554	000000G	EBH.TB1:	.PSECT	#CODE#	RO
010554	000000G		.WORD	EBH.30	
010556	000000G		.WORD	EBH.44	
010560	000000G		.WORD	EBH.45	
010562	000000G		.WORD	EBH.46	
010564	000000G		.WORD	EBH.47	
010566	000000G		.WORD	EBH.48	
010570	000000G		.WORD	EBH.49	

```

000000 004137 000000G      .SBTTL  EMSCMD ERROR MESSAGE SUBROUTINES
EMSCMD::JSR      R1,#SAVE3                                     ;

```

ZRQAM2	RD/RX EXERCISER	5-Dec-1983 10:27:14	VAX-11 Bliss-16 V3-555	SEQ 0215
V01.2	ERROR MESSAGE SUBROUTINES	5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)	Page 215
000004	013746	000000G	MOV CDISK,-(SP)	6554
000010	012746	000000G	MOV #XX13,-(SP)	
000014	012746	000002	MOV #2,-(SP)	
000020	010600		MOV SP,R0	; SP,*
000022	104414		TRAP 14	
000024	013700	000000G	MOV RP,ADDR,R0	6556
000030	016016	000004	MOV 4(R0),(SP)	
000034	016046	000006	MOV 6(R0),-(SP)	
000040	012746	000000G	MOV #XX35,-(SP)	
000044	012746	000003	MOV #3,-(SP)	
000050	010600		MOV SP,R0	; SP,*
000052	104415		TRAP 15	
000054	012716	000000G	MOV #XX29,(SP)	6557
000060	012746	000001	MOV #1,-(SP)	
000064	010600		MOV SP,R0	; SP,*
000066	104415		TRAP 15	
000070	013700	000000G	MOV RP,ADDR,R0	6558
000074	116002	000002	MOVB 2(R0),R2	
000100	006202		ASR R2	
000102	006202		ASR R2	
000104	006202		ASR R2	
000106	006202		ASR R2	
000110	042702	177760	BIC #177760,R2	
000114	012701	177777	MOV #-1,R1	
000120	005702		TST R2	
000122	001010		BNE 1#	
000124	005001		CLR R1	
000126	012716	000000G	MOV #EX.SEQ,(SP)	6560
000132	012746	000001	MOV #1,-(SP)	
000136	010600		MOV SP,R0	; SP,*
000140	104415		TRAP 15	
000142	005726		TST (SP)+	
000144	020227	000001	1#: CMP R2,#1	6558
000150	001010		BNE 2#	
000152	005001		CLR R1	
000154	012716	000000G	MOV #EX.DGM,(SP)	6561
000160	012746	000001	MOV #1,-(SP)	
000164	010600		MOV SP,R0	; SP,*
000166	104415		TRAP 15	
000170	005726		TST (SP)+	
000172	020227	000002	2#: CMP R2,#2	6558
000176	001010		BNE 3#	
000200	005001		CLR R1	
000202	012716	000000G	MOV #EX.CRD,(SP)	6562
000206	012746	000001	MOV #1,-(SP)	
000212	010600		MOV SP,R0	; SP,*
000214	104415		TRAP 15	
000216	005726		TST (SP)+	
000220	020227	000015	3#: CMP R2,#15	6558
000224	001010		BNE 4#	
000226	005001		CLR R1	
000230	012716	000000G	MOV #EX.MTN,(SP)	6563
000234	012746	000001	MOV #1,-(SP)	

000240	010600		MOV	SP,R0	; SP,*	
000242	104415		TRAP	15		
000244	005726		TST	(SP)+		
000246	005701	4#:	TST	R1		6558
000250	001422		BEQ	5#		
000252	013700	000000G	MOV	RP.ADDR,R0		6564
000256	116001	000002	MOVB	2(R0),R1		
000262	006201		ASR	R1		
000264	006201		ASR	R1		
000266	006201		ASR	R1		
000270	006201		ASR	R1		
000272	042701	177760	BIC	#177760,R1		
000276	010116		MOV	R1,(SP)		
000300	012746	000000G	MOV	#XX37,-(SP)		
000304	012746	000002	MOV	#2,-(SP)		
000310	010600		MOV	SP,R0	; SP,*	
000312	104415		TRAP	15		
000314	022626		CMP	(SP)+,(SP)+		
000316	012716	000000G	MOV	#XX17,(SP)		6567
000322	012746	000001	MOV	#1,-(SP)		
000326	010600		MOV	SP,R0	; SP,*	
000330	104414		TRAP	14		
000332	013700	000000G	MOV	RP.ADDR,R0		6569
000336	005003		CLR	R3		
000340	156003	000003	BISB	3(R0),R3		
000344	020327	000002	CMP	R3,#2		
000350	001402		BEQ	6#		
000352	000137	012256'	JMP	26#		
000356	012716	000000G	MOV	#XX18,(SP)		6573
000362	012746	000001	MOV	#1,-(SP)		
000366	010600		MOV	SP,R0	; SP,*	
000370	104414		TRAP	14		
000372	013700	000000G	MOV	RP.ADDR,R0		6574
000376	005002		CLR	R2		
000400	156002	000014	BISB	14(R0),R2		
000404	012701	177777	MOV	#-1,R1		
000410	020227	000201	CMP	R2,#201		
000414	001010		BNE	7#		
000416	005001		CLR	R1		
000420	012716	000000G	MOV	#EX.GDS,(SP)		6576
000424	012746	000001	MOV	#1,-(SP)		
000430	010600		MOV	SP,R0	; SP,*	
000432	104414		TRAP	14		
000434	005726		TST	(SP)+		
000436	020227	000202	CMP	R2,#202		6574
000442	001010		BNE	8#		
000444	005001		CLR	R1		
000446	012716	000000G	MOV	#EX.ESP,(SP)		6577
000452	012746	000001	MOV	#1,-(SP)		
000456	010600		MOV	SP,R0	; SP,*	
000460	104414		TRAP	14		
000462	005726		TST	(SP)+		
000464	020227	000203	8#:	CMP	R2,#203	6574

K1

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)

SEQ 0217
Page 217

000470	001010		BNE	9:			
000472	005001		CLR		R1		
000474	012716	000000G	MOV		#EX.ELP,(SP)	:	6578
000500	012746	000001	MOV		#1,-(SP)		
000504	010600		MOV		SP,R0	: SP,*	
000506	104414		TRAP		14		
000510	005726		TST		(SP)+		
000512	020227	000205	9:	CMP	R2,#205	:	6574
000516	001010		BNE		10:		
000520	005001		CLR		R1		
000522	012716	000000G	MOV		#EX.RCD,(SP)	:	6579
000526	012746	000001	MOV		#1,-(SP)		
000532	010600		MOV		SP,R0	: SP,*	
000534	104414		TRAP		14		
000536	005726		TST		(SP)+		
000540	020227	000204	10:	CMP	R2,#204	:	6574
000544	001010		BNE		11:		
000546	005001		CLR		R1		
000550	012716	000000G	MOV		#EX.SDD,(SP)	:	6580
000554	012746	000001	MOV		#1,-(SP)		
000560	010600		MOV		SP,R0	: SP,*	
000562	104414		TRAP		14		
000564	005726		TST		(SP)+		
000566	020227	000206	11:	CMP	R2,#206	:	6574
000572	001010		BNE		12:		
000574	005001		CLR		R1		
000576	012716	000000G	MOV		#EX.ABP,(SP)	:	6581
000602	012746	000001	MOV		#1,-(SP)		
000606	010600		MOV		SP,R0	: SP,*	
000610	104414		TRAP		14		
000612	005726		TST		(SP)+		
000614	005701		12:	TST	R1	:	6574
000616	001414		BEQ		13:		
000620	013700	000000G	MOV		RP.ADDR,R0	:	6582
000624	005016		CLR		(SP)		
000626	116016	000014	MOVB		14(R0),(SP)		
000632	012746	000000G	MOV		#EX.OP,-(SP)		
000636	012746	000002	MOV		#2,-(SP)		
000642	010600		MOV		SP,R0	: SP,*	
000644	104414		TRAP		14		
000646	022626		CMP		(SP)+,(SP)+		
000650	012716	000000G	13:	MOV	#XX15,(SP)	:	6584
000654	012746	000001	MOV		#1,-(SP)		
000660	010600		MOV		SP,R0	: SP,*	
000662	104414		TRAP		14		
000664	013700	000000G	MOV		RP.ADDR,R0	:	6585
000670	116000	000016	MOVB		16(R0),R0		
000674	042700	177740	BIC		#177740,R0		
000700	020027	000007	CMF		R0,#7		
000704	101010		BHI		14:		
000706	006300		ASL		R0	:	6586
000710	016016	010554'	MOV		EBH.TB1(R0),(SP)		
000714	012746	000001	MOV		#1,-(SP)		

ZRQAM2
V01.2RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)SEQ 0218
Page 218

000720	010600			MOV	SP,R0			; SP,*	
000722	104414			TRAP	14				
000724	000410			BR	15\$				6585
000726	010016		14\$:	MOV	R0,(SP)				6587
000730	012746	000000G		MOV	#EX.OP,-(SP)				
000734	012746	000002		MOV	#2,-(SP)				
000740	010600			MOV	SP,R0			; SP,*	
000742	104414			TRAP	14				
000744	005726			TST	(SP)+				
000746	013700	000000G	15\$:	MOV	RP.ADDR,R0				6589
000752	126027	000014	000204	CMPB	14(R0),#204				
000760	001404			BEQ	16\$				
000762	126027	000014	000205	CMPB	14(R0),#205				6590
000770	001032			BNE	17\$				
000772	013700	000000G	16\$:	MOV	RP.ADDR,R0				6593
000776	016016	000020		MOV	20(R0),(SP)				
001002	012746	000000G		MOV	#XX25,-(SP)				
001006	012746	000002		MOV	#2,-(SP)				
001012	010600			MOV	SP,R0			; SP,*	
001014	104415			TRAP	15				
001016	013700	000000G		MOV	RP.ADDR,R0				6594
001022	016016	000024		MOV	24(R0),(SP)				
001026	016046	000026		MOV	26(R0),-(SP)				
001032	012746	000000G		MOV	#XX26,-(SP)				
001036	012746	000003		MOV	#3,-(SP)				
001042	010600			MOV	SP,R0			; SP,*	
001044	104415			TRAP	15				
001046	004737	000000V		JSR	PC,EMS.DUP				6595
001052	062706	000012		ADD	#12,SP				6592
001056	013700	000000G	17\$:	MOV	RP.ADDR,R0				6597
001062	126027	000014	000201	CMPB	14(R0),#201				
001070	001062			BNE	21\$				
001072	032760	000400	000022	BIT	#400,22(R0)				6600
001100	001407			BEQ	18\$				
001102	012716	000000G		MOV	#DF.0,(SP)				6601
001106	012746	000001		MOV	#1,-(SP)				
001112	010600			MOV	SP,R0			; SP,*	
001114	104414			TRAP	14				
001116	005726			TST	(SP)+				
001120	013700	000000G	18\$:	MOV	RP.ADDR,R0				6602
001124	032760	001000	000022	BIT	#1000,22(R0)				
001132	001407			BEQ	19\$				
001134	012716	000000G		MOV	#DF.1,(SP)				6603
001140	012746	000001		MOV	#1,-(SP)				
001144	010600			MOV	SP,R0			; SP,*	
001146	104414			TRAP	14				
001150	005726			TST	(SP)+				
001152	013700	000000G	19\$:	MOV	RP.ADDR,R0				6604
001156	032760	002000	000022	BIT	#2000,22(R0)				
001164	001407			BEQ	20\$				
001166	012716	000000G		MOV	#DF.2,(SP)				6605
001172	012746	000001		MOV	#1,-(SP)				
001176	010600			MOV	SP,R0			; SP,*	

M1

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)

SEQ 0219
Page 219

001200	104414			TRAP	14		
001202	005726			TST	(SP)+		
001204	013700	000000G	20#:	MOV	RP.ADDR,RO	:	6606
001210	032760	004000	000022	BIT	#4000,22(RO)		
001216	001407			BEQ	21#		
001220	012716	000000G		MOV	#DF.3,(SP)	:	6607
001224	012746	000001		MOV	#1,-(SP)		
001230	010600			MOV	SP,RO	: SP,*	
001232	104414			TRAP	14		
001234	005726			TST	(SP)+		
001236	013700	000000G	21#:	MOV	RP.ADDR,RO	:	6610
001242	126027	000014	000203	CMPB	14(RO),#203		
001250	001062			BNE	25#		
001252	032760	000400	000022	BIT	#400,22(RO)	:	6613
001260	001407			BEQ	22#		
001262	012716	000000G		MOV	#DF.4,(SP)	:	6614
001266	012746	000001		MOV	#1,-(SP)		
001272	010600			MOV	SP,RO	: SP,*	
001274	104414			TRAP	14		
001276	005726			TST	(SP)+		
001300	013700	000000G	22#:	MOV	RP.ADDR,RO	:	6615
001304	032760	001000	000022	BIT	#1000,22(RO)		
001312	001407			BEQ	23#		
001314	012716	000000G		MOV	#DF.5,(SP)	:	6616
001320	012746	000001		MOV	#1,-(SP)		
001324	010600			MOV	SP,RO	: SP,*	
001326	104414			TRAP	14		
001330	005726			TST	(SP)+		
001332	013700	000000G	23#:	MOV	RP.ADDR,RO	:	6617
001336	032760	002000	000022	BIT	#2000,22(RO)		
001344	001407			BEQ	24#		
001346	012716	000000G		MOV	#DF.6,(SP)	:	6618
001352	012746	000001		MOV	#1,-(SP)		
001356	010600			MOV	SP,RO	: SP,*	
001360	104414			TRAP	14		
001362	005726			TST	(SP)+		
001364	013700	000000G	24#:	MOV	RP.ADDR,RO	:	6619
001370	032760	004000	000022	BIT	#4000,22(RO)		
001376	001407			BEQ	25#		
001400	012716	000000G		MOV	#DF.7,(SP)	:	6620
001404	012746	000001		MOV	#1,-(SP)		
001410	010600			MOV	SP,RO	: SP,*	
001412	104414			TRAP	14		
001414	005726			TST	(SP)+		
001416	013700	000000G	25#:	MOV	CUOFF,RO	:	6622
001422	006300			ASL	RO		
001424	063700	000000G		ADD	CST.ADDR,RO		
001430	005016			CLR	(SP)		
001432	116016	000012		MOVB	12(RO),(SP)		
001436	005046			CLR	-(SP)		
001440	116016	000012		MOVB	12(RO),(SP)		
001444	012746	000000G		MOV	#XX23,-(SP)		
001450	012746	000003		MOV	#3,-(SP)		

N1

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)

SEQ 0220
Page 220

001454	010600		MOV	SP,R0	; SP,*	
001456	104415		TRAP	15		
001460	062706	000014	ADD	#14,SP		
001464	005703		TST	R3		6572
001466	001402	26#:	BEQ	27#		6569
001470	000137	014322'	JMP	66#		
001474	012716	000000G	MOV	#XX19,(SP)		
001500	012746	000001	MOV	#1,-(SP)		6626
001504	010600		MOV	SP,R0	; SP,*	
001506	104414		TRAP	14		
001510	013700	000000G	MOV	RP.ADDR,R0		6627
001514	005002		CLR	R2		
001516	156002	000014	BISB	14(R0),R2		
001522	012701	177777	MOV	#-1,R1		
001526	020227	000204	CMP	R2,#204		
001532	001010		BNE	28#		
001534	005001		CLR	R1		
001536	012716	000000G	MOV	#EX.SCC,(SP)		6629
001542	012746	000001	MOV	#1,-(SP)		
001546	010600		MOV	SP,R0	; SP,*	
001550	104414		TRAP	14		
001552	005726		TST	(SP)+		
001554	020227	000211	CMP	R2,#211		6627
001560	001010		BNE	29#		
001562	005001		CLR	R1		
001564	012716	000000G	MOV	#EX.ONL,(SP)		6630
001570	012746	000001	MOV	#1,-(SP)		
001574	010600		MOV	SP,R0	; SP,*	
001576	104414		TRAP	14		
001600	005726		TST	(SP)+		
001602	020227	000220	CMP	R2,#220		6627
001606	001010		BNE	30#		
001610	005001		CLR	R1		
001612	012716	000000G	MOV	#EX.ACC,(SP)		6631
001616	012746	000001	MOV	#1,-(SP)		
001622	010600		MOV	SP,R0	; SP,*	
001624	104414		TRAP	14		
001626	005726		TST	(SP)+		
001630	020227	000241	CMP	R2,#241		6627
001634	001010		BNE	31#		
001636	005001		CLR	R1		
001640	012716	000000G	MOV	#EX.RD,(SP)		6632
001644	012746	000001	MOV	#1,-(SP)		
001650	010600		MOV	SP,R0	; SP,*	
001652	104414		TRAP	14		
001654	005726		TST	(SP)+		
001656	020227	000242	CMP	R2,#242		6627
001662	001010		BNE	32#		
001664	005001		CLR	R1		
001666	012716	000000G	MOV	#EX.WRT,(SP)		6633
001672	012746	000001	MOV	#1,-(SP)		
001676	010600		MOV	SP,R0	; SP,*	
001700	104414		TRAP	14		

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0221
Page 221
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)

001702	005726			TST	(SP),		
001704	005701			TST	R1	:	
001706	001414		32:	BEQ	33:	:	6627
001710	013700	000000G		MOV	RP,ADDR,RO	:	
001714	005016			CLR	(SP)	:	6634
001716	116016	000014		MOVB	14(RO),(SP)		
001722	012746	000000G		MOV	#EX.OP,-(SP)		
001726	012746	000002		MOV	#2,-(SP)		
001732	010600			MOV	SP,RO	: SP, *	
001734	104414			TRAP	14		
001736	022626			CMP	(SP), (SP),		
001740	013700	000000G		MOV	RP,ADDR,RO	:	
001744	026027	000012	040000	CMP	12(RO), #40000		6636
001752	001007			BNE	34:		
001754	012716	000000G		MOV	#XX20,(SP)		
001760	012746	000001		MOV	#1,-(SP)		
001764	010600			MOV	SP,RO	: SP, *	
001766	104414			TRAP	14		
001770	005726			TST	(SP),		
001772	012716	000000G		MOV	#XX15,(SP)	:	6638
001776	012746	000001		MOV	#1,-(SP)		
002002	010600			MOV	SP,RO	: SP, *	
002004	104414			TRAP	14		
002006	013700	000000G		MOV	ST.CODE,RO	:	6639
002012	001413			BEQ	35:		
002014	020027	000013		CMP	RO,#13	:	6640
002020	101010			BHI	35:		
002022	006300			ASL	RO	:	6642
002024	016016	177776G		MOV	ERR.COD-2(RO),(SP)		
002030	012746	000001		MOV	#1,-(SP)		
002034	010600			MOV	SP,RO	: SP, *	
002036	104414			TRAP	14		
002040	000422			BR	37:	:	6639
002042	020027	000037		CMP	RO,#37	:	6645
002046	001007			BNE	36:		
002050	013716	000026G		MOV	ERR.COD+26,(SP)	:	6647
002054	012746	000001		MOV	#1,-(SP)		
002060	010600			MOV	SP,RO	: SP, *	
002062	104414			TRAP	14		
002064	000410			BR	37:	:	6645
002066	010016			MOV	RO,(SP)	:	6649
002070	012746	000000G		MOV	#EX.OP,-(SP)		
002074	012746	000002		MOV	#2,-(SP)		
002100	010600			MOV	SP,RO	: SP, *	
002102	104414			TRAP	14		
002104	005726			TST	(SP),		
002106	004737	010044'		JSR	PC,EMS.SBC	:	6651
002112	013700	000000G		MOV	RP,ADDR,RO	:	6653
002116	116000	000014		MOVB	14(RO),RO		
002122	042700	177400		BIC	#177400,RO		
002126	020027	000220		CMP	RO,#220		
002132	001406			BEQ	38:		
002134	020027	000241		CMP	RO,#241	:	6654

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0222
Page 222
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)

002140	001403		BEQ	38:			
002142	020027	000242	CMP		RO,#242	:	6655
002146	001154		BNE		44:		
002150	013700	000000G	MOV	38:	RP.ADDR,RO	:	6658
002154	016016	000044	MOV		44(RO),(SP)		
002160	012746	000000G	MOV		#XX24,-(SP)		
002164	012746	000002	MOV		#2,-(SP)		
002170	010600		MOV		SP,RO	: SP,*	
002172	104415		TRAP		15		
002174	013700	000000G	MOV		RP.ADDR,RO	:	6659
002200	016016	000020	MOV		20(RO),(SP)		
002204	012746	000000G	MOV		#XX25,-(SP)		
002210	012746	000002	MOV		#2,-(SP)		
002214	010600		MOV		SP,RO	: SP,*	
002216	104415		TRAP		15		
002220	013700	000000G	MOV		RP.ADDR,RO	:	6660
002224	016016	000024	MOV		24(RO),(SP)		
002230	016046	000026	MOV		26(RO),(SP)		
002234	012746	000000G	MOV		#XX26,-(SP)		
002240	012746	000003	MOV		#3,-(SP)		
002244	010600		MOV		SP,RO	: SP,*	
002246	104415		TRAP		15		
002250	013700	000000G	MOV		RP.ADDR,RO	:	6661
002254	005760	000014	TST		14(RO)		
002260	100013		BPL		39:		
002262	016016	000040	MOV		40(RO),(SP)		6663
002266	016046	000042	MOV		42(RO),(SP)		
002272	012746	000000G	MOV		#XX21,-(SP)		
002276	012746	000003	MOV		#3,-(SP)		
002302	010600		MOV		SP,RO	: SP,*	
002304	104414		TRAP		14		
002306	000415		BR		40:		6661
002310	016016	000050	MOV	39:	50(RO),(SP)	:	6665
002314	016046	000052	MOV		52(RO),(SP)		
002320	016046	000050	MOV		50(RO),(SP)		
002324	012746	000000G	MOV		#XX22,-(SP)		
002330	012746	000004	MOV		#4,-(SP)		
002334	010600		MOV		SP,RO	: SP,*	
002336	104415		TRAP		15		
002340	005726		TST		(SP)+		
002342	012716	000000G	MOV	40:	#XX41,(SP)	:	6666
002346	012746	000001	MOV		#1,-(SP)		
002352	010600		MOV		SP,RO	: SP,*	
002354	104414		TRAP		14		
002356	013700	000000G	MOV		RP.ADDR,RO	:	6667
002362	032760	040000 000014	BIT		#40000,14(RO)		
002370	001407		BEQ		41:		
002372	012716	000000G	MOV		#F.1,(SP)	:	6668
002376	012746	000001	MOV		#1,-(SP)		
002402	010600		MOV		SP,RO	: SP,*	
002404	104414		TRAP		14		
002406	005726		TST		(SP)+		
002410	013700	000000G	MOV	41:	RP.ADDR,RO	:	6669

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0223
Page 223
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)

002414	032760	020000	000014		BIT	#20000,14(R0)		
002422	001407				BEQ	42#		
002424	012716	000000G			MOV	#F.2,(SP)	:	6670
002430	012746	000001			MOV	#1,-(SP)		
002434	010600				MOV	SP,R0	: SP,*	
002436	104414				TRAP	14		
002440	005726				TST	(SP)+		
002442	013700	000000G		42#:	MOV	RP.ADDR,R0	:	6671
002446	032760	010000	000014		BIT	#10000,14(R0)		
002454	001407				BEQ	43#		
002456	012716	000000G			MOV	#F.3,(SP)	:	6672
002462	012746	000001			MOV	#1,-(SP)		
002466	010600				MOV	SP,R0	: SP,*	
002470	104414				TRAP	14		
002472	005726				TST	(SP)+		
002474	062706	000026		43#:	ADD	#26,SP	:	6657
002500	013700	000000G		44#:	MOV	RP.ADDR,R0	:	6674
002504	126027	000014	000204		CMPB	14(R0),#204		
002512	001144				BNE	52#		
002514	012716	000000G			MOV	#XX39,(SP)	:	6677
002520	012746	000001			MOV	#1,-(SP)		
002524	010600				MOV	SP,R0	: SP,*	
002526	104414				TRAP	14		
002530	013700	000000G			MOV	RP.ADDR,R0	:	6678
002534	105760	000022			TSTB	22(R0)		
002540	100007				BPL	45#		
002542	012716	000000G			MOV	#F.4,(SP)	:	6679
002546	012746	000001			MOV	#1,-(SP)		
002552	010600				MOV	SP,R0	: SP,*	
002554	104414				TRAP	14		
002556	005726				TST	(SP)+		
002560	013700	000000G		45#:	MOV	RP.ADDR,R0	:	6680
002564	032760	000100	000022		BIT	#100,22(R0)		
002572	001407				BEQ	46#		
002574	012716	000000G			MOV	#F.5,(SP)	:	6681
002600	012746	000001			MOV	#1,-(SP)		
002604	010600				MOV	SP,R0	: SP,*	
002606	104414				TRAP	14		
002610	005726				TST	(SP)+		
002612	013700	000000G		46#:	MOV	RP.ADDR,R0	:	6682
002616	032760	000040	000022		BIT	#40,22(R0)		
002624	001407				BEQ	47#		
002626	012716	000000G			MOV	#F.6,(SP)	:	6683
002632	012746	000001			MOV	#1,-(SP)		
002636	010600				MOV	SP,R0	: SP,*	
002640	104414				TRAP	14		
002642	005726				TST	(SP)+		
002644	013700	000000G		47#:	MOV	RP.ADDR,R0	:	6684
002650	032760	000020	000022		BIT	#20,22(R0)		
002656	001407				BEQ	48#		
002660	012716	000000G			MOV	#F.7,(SP)	:	6685
002664	012746	000001			MOV	#1,-(SP)		
002670	010600				MOV	SP,R0	: SP,*	

E2

ZRQAM2	RD/RX EXERCISER	5-Dec-1983 10:27:14	VAX-11 Bliss-16 V3-555	SEQ 0224
V01.2	ERROR MESSAGE SUBROUTINES	5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)	Page 224
002672	104414			
002674	005726			
002676	013700	000000G		
002702	016000	000022	48:	6686
002706	042700	077777		
002712	020027	100000		
002716	001007			
002720	012716	000000G		
002724	012746	000001		6687
002730	010600			
002732	104414			
002734	005726			
002736	013700	000000G	49:	6688
002742	032760	000002 000022		
002750	001407			
002752	012716	000000G		
002756	012746	000001		6689
002762	010600			
002764	104414			
002766	005726			
002770	013700	000000G	50:	6690
002774	032760	000001 000022		
003002	001407			
003004	012716	000000G		
003010	012746	000001		6691
003014	010600			
003016	104414			
003020	005726			
003022	005726		51:	6676
003024	013700	000000G	52:	6693
003030	126027	000014 000211		
003036	001402			
003040	000137	014316'		
003044	012716	000000G	53:	6696
003050	012746	000001		
003054	010600			
003056	104414			
003060	013700	000000G		
003064	032760	000001 000022		6697
003072	001407			
003074	012716	000000G		
003100	012746	000001		6698
003104	010600			
003106	104414			
003110	005726			
003112	013700	000000G	54:	6699
003116	032760	000002 000022		
003124	001407			
003126	012716	000000G		
003132	012746	000001		6700
003136	010600			
003140	104414			
003142	005726			

ZRQAM2	RD/RX EXERCISER	5-Dec-1983 10:27:14	VAX-11 Bliss-16 V3-555	SEQ 0225
V01.2	ERROR MESSAGE SUBROUTINES	5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)	Page 225
003144	013700	000000G		
003150	016000	000022		
003154	042700	077777		
003160	020027	100000		
003164	001007			
003166	012716	000000G		
003172	012746	000001		
003176	010600			
003200	104414			
003202	005726			
003204	013700	000000G		
003210	032760	040000 000022		
003216	001407			
003220	012716	000000G		
003224	012746	000001		
003230	010600			
003232	104414			
003234	005726			
003236	013700	000000G		
003242	105760	000022		
003246	100007			
003250	012716	000000G		
003254	012746	000001		
003260	010600			
003262	104414			
003264	005726			
003266	013700	000000G		
003272	032760	004000 000022		
003300	001407			
003302	012716	000000G		
003306	012746	000001		
003312	010600			
003314	104414			
003316	005726			
003320	013700	000000G		
003324	032760	002000 000022		
003332	001407			
003334	012716	000000G		
003340	012746	000001		
003344	010600			
003346	104414			
003350	005726			
003352	013700	000000G		
003356	032760	000100 000022		
003364	001407			
003366	012716	000000G		
003372	012746	000001		
003376	010600			
003400	104414			
003402	005726			
003404	013700	000000G		
003410	032760	020000 000022		
003416	001407			

55#:	MOV	RP.ADDR,R0	:	6701
	MOV	22(R0),R0		
	BIC	#77777,R0		
	CMP	R0,#-100000		
	BNE	56#		
	MOV	#F.13,(SP)	:	6702
	MOV	#1,-(SP)		
	MOV	SP,R0	: SP,*	
	TRAP	14		
	TST	(SP)+		
56#:	MOV	RP.ADDR,R0	:	6703
	BIT	#40000,22(R0)		
	BEQ	57#		
	MOV	#F.14,(SP)	:	6704
	MOV	#1,-(SP)		
	MOV	SP,R0	: SP,*	
	TRAP	14		
	TST	(SP)+		
57#:	MOV	RP.ADDR,R0	:	6705
	TSTB	22(R0)		
	BPL	58#		
	MOV	#F.15,(SP)	:	6706
	MOV	#1,-(SP)		
	MOV	SP,R0	: SP,*	
	TRAP	14		
	TST	(SP)+		
58#:	MOV	RP.ADDR,R0	:	6707
	BIT	#4000,22(R0)		
	BEQ	59#		
	MOV	#F.16,(SP)	:	6708
	MOV	#1,-(SP)		
	MOV	SP,R0	: SP,*	
	TRAP	14		
	TST	(SP)+		
59#:	MOV	RP.ADDR,R0	:	6709
	BIT	#2000,22(R0)		
	BEQ	60#		
	MOV	#F.17,(SP)	:	6710
	MOV	#1,-(SP)		
	MOV	SP,R0	: SP,*	
	TRAP	14		
	TST	(SP)+		
60#:	MOV	RP.ADDR,R0	:	6711
	BIT	#100,22(R0)		
	BEQ	61#		
	MOV	#F.18,(SP)	:	6712
	MOV	#1,-(SP)		
	MOV	SP,R0	: SP,*	
	TRAP	14		
	TST	(SP)+		
61#:	MOV	RP.ADDR,R0	:	6713
	BIT	#20000,22(R0)		
	BEQ	62#		

ZRQAM2	RD/RX EXERCISER	5-Dec-1983 10:27:14	VAX-11 Bliss-16 V3-555	SEQ 0226
V01.2	ERROR MESSAGE SUBROUTINES	5-Dec-1983 10:27:04	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (66)	Page 226
003420	012716	000000G	MOV #F.19,(SP)	6714
003424	012746	000001	MOV #1,-(SP)	
003430	010600		MOV SP,R0	
003432	104414		TRAP 14	
003434	005726		TST (SP)+	
003436	013700	000000G	MOV RP.ADDR,R0	6715
003442	032760	010000 000022	BIT #10000,22(R0)	
003450	001407		BEQ 63\$	
003452	012716	000000G	MOV #F.20,(SP)	6716
003456	012746	000001	MOV #1,-(SP)	
003462	010600		MOV SP,R0	
003464	104414		TRAP 14	
003466	005726		TST (SP)+	
003470	013700	000000G	MOV RP.ADDR,R0	6717
003474	032760	000004 000022	BIT #4,22(R0)	
003502	001407		BEQ 64\$	
003504	012716	000000G	MOV #F.21,(SP)	6718
003510	012746	000001	MOV #1,-(SP)	
003514	010600		MOV SP,R0	
003516	104414		TRAP 14	
003520	005726		TST (SP)+	
003522	005726		TST (SP)+	6695
003524	062706	000006	ADD #6,SP	6625
003530	012716	000000G	MOV #CRLF,(SP)	6730
003534	012746	000001	MOV #1,-(SP)	
003540	010600		MOV SP,R0	
003542	104414		TRAP 14	
003544	062706	000022	ADD #22,SP	6546
003550	000207		RTS PC	6535

; Routine Size: 949 words, Routine Base: \$CODE\$ + 10572
; Maximum stack depth per invocation: 29 words

```

: 6732 !+
: 6733 GLOBAL ROUTINE EMS_DBN : NOVALUE =
: 6734 !+
: 6735 ! THIS ROUTINE PRINTS THE PRESENT DBN
: 6736 !
: 6737 ! IMPLICIT INPUTS:
: 6738 ! CST_ADDR - ADDRESS OF CONTROLLER STATUS TABLE
: 6739 !-
: 6740 BEGIN
: 6741 PRINTB (XX13, .CDISK); ! "DISK XXX"
: 6742 PRINTB (XX23, .CST_ADDR [.CUOFF + 5, D_DBN], .CST_ADDR [.CUOFF + 5, D_DBN]); ! "DBN: XXXXXX."
: 6743 PRINTB (XX32, .S_DUPPKT - 2); ! PRINTS THE BYTE COUNT
: 6744 PRINTB (XX33, .S_PATTERN); ! PRINTS THE PATTERN WRITTEN
: 6745 PRINTB (XX34, .(DUPPKT + .S_DUPPKT), .(DUPPKT + .S_DUPPKT)); ! PRINTS THE WORD READ
: 6746 EMS_BLK (DUPPKT +2, 256); ! PRINTS THE WHOLE BLOCK READ IN OCTAL
: 6747 END; ! ROUTINE EMS-DBN

```

Address	Offset	Label	Code	Comment	Line No.
000000	013746	000000G	EMS.DBN: .SBTTL	EMS.DBN ERROR MESSAGE SUBROUTINES	
000004	012746	000000G	MOV	CDISK, -(SP)	6741
000010	012746	000002	MOV	#XX13, -(SP)	
000014	010600		MOV	#2, -(SP)	
000016	104414		MOV	SP, R0	; SP, *
000020	013700	000000G	TRAP	14	
000024	006300		MOV	CUOFF, R0	; 6742
000026	063700	000000G	ASL	R0	
000032	005016		ADD	CST.ADDR, R0	
000034	116016	000012	CLR	(SP)	
000040	005046		MOVB	12(R0), (SP)	
000042	116016	000012	CLR	-(SP)	
000046	012746	000000G	MOVB	12(R0), (SP)	
000052	012746	000003	MOV	#XX23, -(SP)	
000056	010600		MOV	#3, -(SP)	
000060	104414		MOV	SP, R0	; SP, *
000062	013716	000000G	TRAP	14	
000066	162716	000002	MOV	S.DUPPKT, (SP)	; 6743
000072	012746	000000G	SUB	#2, (SP)	
000076	012746	000002	MOV	#XX32, -(SP)	
000102	010600		MOV	#2, -(SP)	
000104	104414		MOV	SP, R0	; SP, *
000106	013716	000000G	TRAP	14	
000112	012746	000000G	MOV	S.PATTERN, (SP)	; 6744
000116	012746	000002	MOV	#XX33, -(SP)	
000122	010600		MOV	#2, -(SP)	
000124	104414		MOV	SP, R0	; SP, *
000126	013700	000000G	TRAP	14	
000132	016016	000000G	MOV	S.DUPPKT, R0	; 6745
000136	011646		MOV	DUPPKT(R0), (SP)	
000140	012746	000000G	MOV	(SP), -(SP)	
000144	012746	000003	MOV	#XX34, -(SP)	
000150	010600		MOV	#3, -(SP)	
			MOV	SP, R0	; SP, *

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 B111-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (67)

000152	104414		TRAP	14		
000154	012716	000002G	MOV	#DUPPKT+2,(SP)	:	6746
000160	012746	000400	MOV	#400,-(SP)		
000164	004737	000000V	JSR	PC,EMS.BLK		
000170	062706	000034	ADD	#34,SP	:	6740
000174	000207		RTS	PC	:	6733

; Routine Size: 63 words, Routine Base: \$CODE\$ + 14344
; Maximum stack depth per invocation: 15 words

```

: 6748 ROUTINE EMS_DUP : NOVALUE =
: 6749 !+
: 6750 ! THIS ROUTINE PRINTS OUT THE LISTING OF THE DUP PACKETS ONLY ASSOCIATED WITH DUP RECEIVED DATA COMMAND
: 6751 ! AND TE SEND DATA COMMAND.
: 6752 !-
: 6753 begin
: 6754
: 6755 OWN
: 6756 MESSAGE TYPE : VECTOR [6] INITIAL (T_QUE, T_DEF, T_INF, T_TER, T_FAT, T_SPL),
: 6757 MSGNUMBERS : VECTOR [9] INITIAL (M_TER, M_BIN, M_COD, M_DAT, M_UR, M_URP, M_UP, M_UL, M_ASC),
: 6758 RCD_ERRORS : VECTOR [5] INITIAL (EBH_30, E_UNT, E_BLK, E_DEV, E_ZER);
: 6759 ! TABLE OF BASIC, HARD ERROR MESSAGE ADDRESSES, INDEXED BY STATUS CODE
: 6760
: 6761 PRINTB (XX29); ! PRINTS DUP MESSAGE TYPE
: 6762 if (.duppkt [duptype] lequ 6)
: 6763 then
: 6764 PRINTB (.MESSAGE TYPE [.DUPPKT [DUPTYPE] - 1])
: 6765 else
: 6766 printb (ex_op, .duppkt [duptype]);
: 6767
: 6768 PRINTB (XX30); ! PRINTS DUP MESSAGE NUMBER
: 6769 if (.duppkt [dupmsg] lssu 9)
: 6770 then
: 6771 PRINTB (.MSGNUMBERS [.DUPPKT [DUPMSG] - 1])
: 6772 else
: 6773 printb (ex_op, .duppkt [dupmsg]);
: 6774
: 6775 PRINTB (XX31); ! PRINTS DUP BUFFER 1 WHICH SOMETIMES IS AN
: 6776 IF (.DUPPKT [DUPBF1] LSSU 5) ! ERROR MESSAGE
: 6777 THEN
: 6778 PRINTB (.RCD_ERRORS [.DUPPKT [DUPBF1]])
: 6779 ELSE
: 6780 PRINTB (EX_WRD, .DUPPKT [DUPBF1]);
: 6781
: 6782 end;

```

```

014542 000000G MESSAGE TYPE:
: .WORD T.QUE
014544 000000G .WORD T.DEF
014546 000000G .WORD T.INF
014550 000000G .WORD T.TER
014552 000000G .WORD T.FAT
014554 000000G .WORD T.SPL
014556 000000G MSGNUMBERS:
: .WORD M.TER
014560 000000G .WORD M.BIN
014562 000000G .WORD M.COD
014564 000000G .WORD M.DAT
014566 000000G .WORD M.UR
014570 000000G .WORD M.URP
014572 000000G .WORD M.UP
014574 000000G .WORD M.UL

```


014576 000000G
 014600 000000G
 014602 000000G
 014604 000000G
 014606 000000G
 014610 000000G

.WORD M.ASC
 RCD.ERRORS:
 .WORD EBH.30
 .WORD E.UNT
 .WORD E.BLK
 .WORD E.DEV
 .WORD E.ZER

000000 012746 000000G
 000004 012746 000001
 000010 010600
 000012 104414
 000014 013700 000000G
 000020 006200
 000022 006200
 000024 006200
 000026 006200
 000030 000300
 000032 042700 177760
 000036 020027 000006
 000042 101010
 000044 006300
 000046 016016 014540'
 000052 012746 000001
 000056 010600
 000060 104414
 000062 000410
 000064 010016
 000066 012746 000000G
 000072 012746 000002
 000076 010600
 000100 104414
 000102 005726
 000104 012716 000000G
 000110 012746 000001
 000114 010600
 000116 104414
 000120 013700 000000G
 000124 042700 170000
 000130 020027 000011
 000134 103010
 000136 006300
 000140 016016 014554'
 000144 012746 000001
 000150 010600
 000152 104414
 000154 000410
 000156 010016
 000160 012746 000000G
 000164 012746 000002
 000170 010600

.SBTTL EMS.DUP ERROR MESSAGE SUBROUTINES
 EMS.DUP: MOV #XX29, -(SP) ;
 MOV #1, -(SP) ;
 MOV SP, RO ; SP,*
 TRAP 14 ;
 MOV DUPPKT, RO ;
 ASR RO ;
 ASR RO ;
 ASR RO ;
 ASR RO ;
 SWAB RO ;
 BIC #177760, RO ;
 CMP RO, #6 ;
 BHI 1# ;
 ASL RO ;
 MOV MESSAGE-2(RO), (SP) ;
 MOV #1, -(SP) ;
 MOV SP, RO ; SP,*
 TRAP 14 ;
 BR 2# ;
 1#: MOV RO, (SP) ;
 MOV #EX.OP, -(SP) ;
 MOV #2, -(SP) ;
 MOV SP, RO ; SP,*
 TRAP 14 ;
 TST (SP), ;
 2#: MOV #XX30, (SP) ;
 MOV #1, -(SP) ;
 MOV SP, RO ; SP,*
 TRAP 14 ;
 MOV DUPPKT, RO ;
 BIC #170000, RO ;
 CMP RO, #11 ;
 BHIS 3# ;
 ASL RO ;
 MOV MSGNUMBERS-2(RO), (SP) ;
 MOV #1, -(SP) ;
 MOV SP, RO ; SP,*
 TRAP 14 ;
 BR 4# ;
 3#: MOV RO, (SP) ;
 MOV #EX.OP, -(SP) ;
 MOV #2, -(SP) ;
 MOV SP, RO ; SP,*

6761
 6762
 6764
 6762
 6766
 6768
 6769
 6771
 6769
 6773

L2

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0231
Page 231
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (68)

000172	104414		TRAP	14		
000174	005726		TST	(SP)+		
000176	012716	000000G	MOV	#XX31,(SP)	:	6775
000202	012746	000001	MOV	#1,-(SP)		
000206	010600		MOV	SP,R0	: SP,*	
000210	104414		TRAP	14		
000212	013700	000002G	MOV	DUPPKT+2,R0	:	6776
000216	020027	000005	CMP	R0,#5		
000222	103010		BHIS	5#		
000224	006300		ASL	R0	:	6778
000226	016016	014600'	MOV	RCD.ERRORS(R0),(SP)		
000232	012746	000001	MOV	#1,-(SP)		
000236	010600		MOV	SP,R0	: SP,*	
000240	104414		TRAP	14		
000242	000410		BR	6#	:	6776
000244	010016		MOV	R0,(SP)	:	6780
000246	012746	000000G	MOV	#EX.WRD,-(SP)		
000252	012746	000002	MOV	#2,-(SP)		
000256	010600		MOV	SP,R0	: SP,*	
000260	104414		TRAP	14		
000262	005726		TST	(SP)+		
000264	062706	000016	ADD	#16,SP	:	6753
000270	000207		RTS	PC	:	6748

; Routine Size: 93 words, Routine Base: \$CODE\$ + 14612
; Maximum stack depth per invocation: 10 words

```

: 6783 global routine EMS_BLK (ADDR, LENGTH) : novalue =
: 6784
: 6785 !+
: 6786 ! THIS ROUTINE WILL PRINTX A BLOCK OF MEMORY WHICH IS "LENGTH" WORDS
: 6787 ! LONG STARTING AT ADDRESS "ADDR". PRINTING IS DONE IN OCTAL, 8 WORDS
: 6788 ! TO A LINE.
: 6789 !-
: 6790
: 6791 begin
: 6792
: 6793 literal
: 6794 MASK = %o'7';
: 6795 PRINTX (CRLF);
: 6796 incr COUNT from 1 to .LENGTH do          ! FOR EACH WORD TO PRINT
: 6797 begin
: 6798
: 6799 if ((.COUNT - 1) and MASK) eql 0        ! IF AT START OF A NEW LINE
: 6800 then
: 6801 PRINTX (SPACE4);                          ! PRINTX 4 SPACES
: 6802
: 6803 PRINTX (EX_WRD, ..ADDR);                  ! PRINTX A WORD
: 6804 ADDR = .ADDR + 2;                        ! ADVANCE TO NEXT ADDRESS
: 6805
: 6806 if (((.COUNT and MASK) eql 0) or      ! IF AT THE END OF A LINE OR
: 6807 (.COUNT eql .LENGTH))
: 6808 then
: 6809 PRINTX (CRLF);                            ! PRINTX <CR><LF>
: 6810
: 6811 end;
: 6812
: 6813 end;

```

Address	Offset	Label	Instruction	Comment	Line No.	
000000	010146	EMS.BLK::	.SBTTL EMS.BLK ERROR MESSAGE SUBROUTINES			
000002	012746	000000G	MOV R1, -(SP)		6783	
000006	012746	000001	MOV #CRLF, -(SP)		6795	
000012	010600		MOV #1, -(SP)			
000014	104415		MOV SP, R0	: SP, *		
000016	005001		TRAP 15			
000020	000445		CLR R1	: COUNT	6796	
000022	010100	1#:	BR 5#			
000024	005300		MOV R1, R0	: COUNT, *	6799	
000026	032700	000007	DEC R0			
000032	001007		BIT #7, R0			
000034	012716	000000G	BNE 2#			
000040	012746	000001	MOV #SPACE4, (SP)		6801	
000044	010600		MOV #1, -(SP)			
000046	104415		MOV SP, R0	: SP, *		
000050	005726		TRAP 15			
000052	017616	000012	TST (SP),			
000056	012746	000000G	2#:	MOV #12(SP), (SP)	: ADDR, *	6803
			MOV #EX.WRD, -(SP)			

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (69)

SEQ 0233
Page 233

000062	012746	000002		MOV	#2,-(SP)			
000066	010600			MOV	SP,R0		; SP,*	
000070	104415			TRAP	15			
000072	062766	000002	000016	ADD	#2,16(SP)		; *,ADDR	6804
000100	032701	000007		BIT	#7,R1		; *,COUNT	6806
000104	001403			BEQ	3#			
000106	020166	000014		CMP	R1,14(SP)		; COUNT,LENGTH	6807
000112	001007			BNE	4#			
000114	012716	000000G	3#:	MOV	#CRLF,(SP)			6809
000120	012746	000001		MOV	#1,-(SP)			
000124	010600			MOV	SP,R0		; SP,*	
000126	104415			TRAP	15			
000130	005726			TST	(SP)+			
000132	022626		4#:	CMP	(SP)+,(SP)+			6797
000134	005201		5#:	INC	R1		; COUNT	6796
000136	020166	000010		CMP	R1,10(SP)		; COUNT,LENGTH	
000142	003727			BLE	1#			
000144	022626			CMP	(SP)+,(SP)+			6791
000146	012601			MOV	(SP)+,R1			6783
000150	000207			RTS	PC			

: Routine Size: 53 words, Routine Base: \$CODE\$ + 15104
: Maximum stack depth per invocation: 8 words

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0234
Page 234
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (70)

```

:      6814 ROUTINE EMS_MAL : NOVALUE =
:      6815 !*
:      6816 !      THIS ROUTINE PRINTS ALL TABLES IN AN EFFORT TO FIND THE MISSING MESSAGE OR THE PROBLEM
:      6817 !-
:      6818 begin
:      6819
:      6820 PRINTB (EB_DCT,dct);           ! PRINT " DCT TABLE CONTENTS"
:      6821 EMS_BLK (DCT, DCT_LEN);    ! PRINT BLOCK OF WORDS
:      6822
:      6823 PRINTB (EB_COMM, .DCT [0, RR_BEG]); ! PRINT "COMMAND RING" AND STARTING ADDR
:      6824 EMS_BLK (.DCT [0, RR_BEG] - 4, COMM_LEN - 4); ! PRINT BLOCK OF COMMAND RING
:      6825
:      6826 PRINTB (EBNEX1, .DCT [0, RR_POLL]); !PRINT ADDR OF COMMAND OF NEXT RR POLL
:      6827
:      6828 PRINTB (EB_NEX2, ..DCT [0, RR_POLL] - 8); !PRINT ADDR OF PACKET TO BE POLLED
:      6829
:      6830 PRINTB (EBNEX3, .DCT [0, CR_POLL] - 8); !PRINT ADDR OF PACKET TO BE POLLED
:      6831 PRINTB (EB_PKT);           ! PRINTS "PACKETS IN MEMORY"
:      6832 incr COUNT from 0 to PKT_CNT - 1 do ! FOR EACH MSCP ENVELOPE
:      6833     begin
:      6834     EMS_BLK ((MSCP_PKT [.COUNT, 0,0,16,0]), PKT_LEN); !PRINTS CONTENTS OF PACKETS
:      6835     end;
:      6836 printb (EB_SA);           ! "Contents of SA Register"
:      6837 printb (ex_wrd, .(172150)); ! Contents of sa register
:      6838 end;

```

			.SBTTL	EMS.MAL ERROR MESSAGE SUBROUTINES	
000000	010146		EMS.MAL:MOV	R1,-(SP)	6814
000002	012746	000000G	MOV	#DCT,-(SP)	6820
000006	012746	000000G	MOV	#EB.DCT,-(SP)	
000012	012746	000002	MOV	#2,-(SP)	
000016	010600		MOV	SP,R0	: SP,*
000020	104414		TRAP	14	
000022	012716	000000G	MOV	#DCT,(SP)	: 6821
000026	012746	000011	MOV	#11,-(SP)	
000032	004737	015104'	JSR	PC,EMS.BLK	
000036	013716	000004G	MOV	DCT+4,(SP)	: 6823
000042	012746	000000G	MOV	#EB.COMM,-(SP)	
000046	012746	000002	MOV	#2,-(SP)	
000052	010600		MOV	SP,R0	: SP,*
000054	104414		TRAP	14	
000056	013716	000004G	MOV	DCT+4,(SP)	: 6824
000062	162716	000004	SUB	#4,(SP)	
000066	012746	000020	MOV	#20,-(SP)	
000072	004737	015104'	JSR	PC,EMS.BLK	
000076	013716	000014G	MOV	DCT+14,(SP)	: 6826
000102	012746	000000G	MOV	#EBNEX1,-(SP)	
000106	012746	000002	MOV	#2,-(SP)	
000112	010600		MOV	SP,R0	: SP,*
000114	104414		TRAP	14	
000116	017716	000014G	MOV	#DCT+14,(SP)	: 6828
000122	162716	000010	SUB	#10,(SP)	

C3

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0235
Page 235
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (70)

000126	012746	000000G		MOV	#EB.NEX2,-(SP)		
000132	012746	000002		MOV	#2,-(SP)		
000136	010600			MOV	SP,R0	; SP,*	
000140	104414			TRAP	14		
000142	013716	000016G		MOV	DCT+16,(SP)		6830
000146	162716	000010		SUB	#10,(SP)		
000152	012746	000000G		MOV	#EBNEX3,-(SP)		
000156	012746	000002		MOV	#2,-(SP)		
000162	010600			MOV	SP,R0	; SP,*	
000164	104414			TRAP	14		
000166	012716	000000G		MOV	#EB.PKT,(SP)		6831
000172	012746	000001		MOV	#1,-(SP)		
000176	010600			MOV	SP,R0	; SP,*	
000200	104414			TRAP	14		
000202	005001			CLR	R1	; COUNT	6832
000204	010116		14:	MOV	R1,(SP)	; COUNT,*	6834
000206	062716	000000G		ADD	#MSCP.PKT,(SP)		
000212	012746	000042		MOV	#42,-(SP)		
000216	004737	015104'		JSR	PC,EMS.BLK		
000222	005726			TST	(SP)*		6833
000224	062701	000104		ADD	#104,R1	; *.COUNT	6832
000230	020127	001354		CMP	R1,#1354	; COUNT,*	
000234	003763			BLE	14		
000236	012716	000000G		MOV	#EB.SA,(SP)		6836
000242	012746	000001		MOV	#1,-(SP)		
000246	010600			MOV	SP,R0	; SP,*	
000250	104414			TRAP	14		
000252	013716	120166		MOV	#120166,(SP)		6837
000256	012746	000000G		MOV	#EX.WRD,-(SP)		
000262	012746	000002		MOV	#2,-(SP)		
000266	010600			MOV	SP,R0	; SP,*	
000270	104414			TRAP	14		
000272	062706	000042		ADD	#42,SP		6818
000276	012601			MOV	(SP)*,R1		6814
000300	000207			RTS	PC		

; Routine Size: 97 words, Routine Base: \$CODE\$ + 15256
; Maximum stack depth per invocation: 20 words

; 6839

```
: 6840 !ROUTINE EMS_RAL : NOVALUE =
: 6841 !*
: 6842 !      THIS ROUTINE PRINTS ALL return packets IN AN EFFORT TO FIND THE MISSING packet OR THE PROBLEM
: 6843 !-
: 6844 !begin
: 6845 !PRINTB (EB_RAL);
: 6846 !incr COUNT from 0 to RP_CNT - 1 do
: 6847 !      begin
: 6848 !          EMS_BLK ((RETPKT + .COUNT * RP_LEN) , RP_LEN);           ! PRINT BLOCK OF WORDS
: 6849 !      end;
: 6850
: 6851 !end;
: 6852
```

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0237
Page 237
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (72)

```

: 6853 routine EMS_LBN : novalue =
: 6854
: 6855 !+
: 6856 ! THIS ROUTINE PRINTS (EXTENDED) ONE OF TWO BLOCK NUMBERS APPEARING IN
: 6857 ! THE CURRENT RETURN PACKET. NORMALLY, THE LBN FIELD IS PRINTED; THIS
: 6858 ! FIELD WAS COPIED INTO THE RETURN PACKET FROM THE ASSOCIATED COMMAND
: 6859 ! PACKET. HOWEVER, IF THE "FLAGS" FIELD OF THE CURRENT RETURN PACKET
: 6860 ! INDICATES "BAD BLOCK REPORTED", THEN THE "FIRST BAD BLOCK" FIELD IS
: 6861 ! PRINTED.
: 6862 !
: 6863 ! IMPLICIT INPUTS:
: 6864 ! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 6865 !-
: 6866
: 6867 if BIT_TST (RP_ADDR [FLAGS], EF_BBR) ! IF BAD BLOCK REPORTED
: 6868 then
: 6869 PRINTX (XX21, .RP_ADDR [BBLK_HI], .RP_ADDR [BBLK_LO]) ! "BAD BLOCK REPORTED: XXXXXX."
: 6870 else
: 6871 PRINTX (XX22, .RP_ADDR [LBN_LO], .RP_ADDR [LBN_HI], .RP_ADDR [LBN_LO]); ! "LBN: XXXXXX."

```

```

000000 013700 000000G .SBTTL EMS.LBN ERROR MESSAGE SUBROUTINES
000004 005760 000014 EMS.LBN:MOV RP_ADDR,R0 ; 6867
000010 100013 TST 14(R0) ;
000012 016046 000040 BPL 1# ;
000016 016046 000042 MOV 40(R0),-(SP) ; 6869
000022 012746 000000G MOV 42(R0),-(SP)
000026 012746 000003 MOV #XX21,-(SP)
000032 010600 MOV #3,-(SP)
000034 104415 MOV SP,R0 ; SP,*
000036 000415 TRAP 15 ;
000040 016046 000050 1#: MOV 50(R0),-(SP) ; 6867
000044 016046 000052 MOV 52(R0),-(SP) ; 6871
000050 016046 000050 MOV 50(R0),-(SP)
000054 012746 000000G MOV #XX22,-(SP)
000060 012746 000004 MOV #4,-(SP)
000064 010600 MOV SP,R0 ; SP,*
000066 104415 TRAP 15 ;
000070 005726 TST (SP)+
000072 062706 000010 2#: ADD #10,SP ; 6867
000076 000207 RTS PC ; 6853

```

```

: Routine Size: 32 words, Routine Base: #CODE# + 15560
: Maximum stack depth per invocation: 7 words

```



```

: 6872 global routine EMS_EL (index) : novalue =
: 6873
: 6874 !+
: 6875 ! THIS ROUTINE IS CALLED FROM 'SEQUEN' AND 'DATAGM' AND PRINTS THE CONTENTS OF THE
: 6876 ! ERROR-LOG PACKET
: 6877 !-
: 6878
: 6879 begin
: 6880
: 6881 local
: 6882 ELOG_ADDR : ref block [EP_LEN, word] field (EP_FIELDS),
: 6883 REASON : word,
: 6884 DISK_NUM : byte,
: 6885 ELOG_CODE : byte,
: 6886 ELOG_SUB : word;
: 6887
: 6888 ELOG_ADDR = ELOG_PKT + (.index * EP_LEN * 2);           ! ERROR LOG PACKET'S ADDRESS
: 6889 REASON = .ELOG_ADDR [EL_FORMAT];                       ! FORMAT
: 6890 DISK_NUM = .ELOG_ADDR [EL_DK_NUM];                     ! DISK NUMBER
: 6891 ELOG_CODE = .ELOG_ADDR [EL_CODE];                      ! CODE
: 6892 ELOG_SUB = .ELOG_ADDR [EL_SUBCODE];                   ! SUBCODE
: 6893 PRINTB (ELG_00);                                       ! ERROR-LOG MESSAGE RECEIVED
: 6894
: 6895 if (.REASON eq1 FORMAT_CNTR) or
: 6896     (.REASON eq1 FORMAT_HOST)
: 6897 then
: 6898     PRINTB (.ELG_FMT [.REASON])                          ! PRINT BASIC REASON
: 6899 else
: 6900     PRINTB (.ELG_FMT [.REASON], .DISK_NUM);             ! PRINT BASIC REASON WITH DISK NUMBER
: 6901
: 6902 if (.ELOG_CODE gtru 0) and
: 6903     (.ELOG_CODE lequ 11)
: 6904 then
: 6905     begin
: 6906     PRINTX (ASTERISK);
: 6907     PRINTX (.ERR_COD [.ELOG_CODE - 1]);                 ! CODE
: 6908     end
: 6909 else
: 6910
: 6911     if .ELOG_CODE eq1 ST_DIA                             ! MESSAGE FROM INTERNAL DIAGNOSTICS
: 6912     then
: 6913         begin
: 6914         PRINTX (ASTERISK);
: 6915         PRINTX (.ERR_COD [12]);
: 6916         end;
: 6917
: 6918     if (.ELOG_CODE eq1 ST_MFE) and
: 6919         (.ELOG_SUB lequ 10)
: 6920     then
: 6921         begin
: 6922         PRINTX (CRLF);
: 6923         PRINTX (ASTERISK);
: 6924         PRINTX (.TBL_MFE [.ELOG_SUB]);                   ! MEDIA FORMAT ERROR

```

```

:      6925         end;
:      6926
:      6927         if (.ELOG_CODE eq1 ST_DAT) and
:      6928             (.ELOG_SUB lequ 15)
:      6929         then
:      6930             begin
:      6931                 PRINTX (CRLF);
:      6932                 PRINTX (ASTERISK);
:      6933                 PRINTX (.TBL_DAT [.ELOG_SUB]);           ! DATA ERROR
:      6934             end;
:      6935
:      6936         if (.ELOG_CODE eq1 ST_HST) and
:      6937             (.ELOG_SUB lequ 4)
:      6938         then
:      6939             begin
:      6940                 PRINTX (CRLF);
:      6941                 PRINTX (ASTERISK);
:      6942                 PRINTX (.TBL_HST [.ELOG_SUB]);           ! HOST ACCESS ERROR
:      6943             end;
:      6944
:      6945         if (.ELOG_CODE eq1 ST_CNT) and
:      6946             (.ELOG_SUB lequ 3)
:      6947         then
:      6948             begin
:      6949                 PRINTX (CRLF);
:      6950                 PRINTX (ASTERISK);
:      6951                 PRINTX (.TBL_CNT [.ELOG_SUB]);           ! CONTROLLER ERROR
:      6952             end;
:      6953
:      6954         if (.ELOG_CODE eq1 ST_DRV) and
:      6955             (.ELOG_SUB lequ 8)
:      6956         then
:      6957             begin
:      6958                 PRINTX (CRLF);
:      6959                 PRINTX (ASTERISK);
:      6960                 PRINTX (.TBL_DRV [.ELOG_SUB]);           ! DRIVE ERROR
:      6961             end;
:      6962
:      6963         if .REASON eq1 FORMAT_XFER           ! IF DISK XFER INVOLVED
:      6964         then
:      6965
:      6966             if .ELOG_ADDR [EL_BLOCK_TYPE] eq1 TYPE_LBN           ! PRINT LBN OR RBN
:      6967             then
:      6968                 PRINTX (XX22, .ELOG_ADDR [EL_BLOCK], 0, .ELOG_ADDR [EL_BLOCK])
:      6969             else
:      6970                 PRINTX (EX_RBN, .ELOG_ADDR [EL_BLOCK], 0, .ELOG_ADDR [EL_BLOCK]);
:      6971
:      6972         PRINTX (CRLF);
:      6973         EMS_BLK ((.ELOG_ADDR + 2), ((.ELOG_ADDR [EL_MSGLEN] + 1) / 2) + 2); ! PRINTX CONTENTS OF PACKET
:      6974         ELOG_ADDR [EL_CONTENTS] = EMPTY;           ! DECLARE SAVE AREA FREE
:      6975         PRINTX (CRLF);
:      6976         end;

```

Address	Offset	OpCode	Comment	Label	OpCode	Comment	Address
000000	004137	000000G	EMS.EL::		.SBTTL	EMS.EL ERROR MESSAGE SUBROUTINES	
000004	005746		JSR		R1,#SAVE5		6872
000006	016646	000020	TST		-(SP)		
000012	012746	000102	MOV		20(SP),-(SP)	; INDEX,*	6888
000016	004737	000000G	MOV		#102,-(SP)		
000022	062700	000000G	JSR		PC,BL#MUL		
000026	010003		ADD		#ELOG.PKT,R0		
000030	005004		MOV		R0,R3	; *,ELOG.ADDR	
000032	156304	000016	CLR		R4	; REASON	6889
000036	116366	000012	BISB		16(R3),R4	; *(ELOG.ADDR),REASON	
000044	116300	000020	MOVB	000004	12(R3),4(SP)	; *(ELOG.ADDR),DISK.NUM	6890
000050	042700	177740	MOVB		20(R3),R0	; *(ELOG.ADDR),*	6891
000054	105002		BIC		#177740,R0		
000056	050002		CLRB		R2	; ELOG.CODE	
000060	016301	000020	BIS		R0,R2	; *,ELOG.CODE	
000064	006201		MOV		20(R3),R1	; *(ELOG.ADDR),ELOG.SUB	6892
000066	006201		ASR		R1	; ELOG.SUB	
000070	006201		ASR		R1	; ELOG.SUB	
000072	006201		ASR		R1	; ELOG.SUB	
000074	006201		ASR		R1	; ELOG.SUB	
000076	042701	174000	ASR		R1	; ELOG.SUB	
000102	012716	000000G	BIC		#174000,R1	; *,ELOG.SUB	
000106	012746	000001	MOV		#ELG.00,(SP)		6893
000112	010600		MOV		#1,-(SP)		
000114	104414		MOV		SP,R0	; SP,*	
000116	010405		TRAP		14		
000120	006305		MOV		R4,R5	; REASON,*	6898
000122	005704		ASL		R5		
000124	001403		TST		R4	; REASON	6895
000126	020427	000001	BEQ		1#		
000132	001007		CMP		R4,#1	; REASON,*	6896
000134	016516	000000G	BNE		2#		
000140	012746	000001	MOV	1#:	ELG.FMT(R5),(SP)		6898
000144	010600		MOV		#1,-(SP)		
000146	104414		MOV		SP,R0	; SP,*	
000150	000412		TRAP		14		
000152	005016		BR		3#		6895
000154	116616	000006	CLR	2#:	(SP)		6900
000160	016546	000000G	MOVB		6(SP),(SP)	; DISK.NUM,*	
000164	012746	000002	MOV		ELG.FMT(R5),-(SP)		
000170	010600		MOV		#2,-(SP)		
000172	104414		MOV		SP,R0	; SP,*	
000174	005726		TRAP		14		
000176	105702		TST		(SP),		
000200	001423		TSTB	3#:	R2	; ELOG.CODE	6902
000202	120227	000013	BEQ		4#		
000206	101020		CMPB		R2,#13	; ELOG.CODE,*	6903
000210	012716	000000G	BHI		4#		
000214	012746	000001	MOV		#ASTERISK,(SP)		6906
000220	010600		MOV		#1,-(SP)		
000222	104415		MOV		SP,R0	; SP,*	
			TRAP		15		

ZRQAM2
V01.2RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 B111-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (73)SEQ 0241
Page 241

000224	005000		CLR	R0					
000226	150200		BISB	R2,R0		:	ELOG.CODE,*		6907
000230	006300		ASL	R0		:			
000232	016016	177776G	MOV	ERR.COD-2(R0),(SP)		:			
000236	012746	000001	MOV	#1,-(SP)		:			
000242	010600		MOV	SP,R0		:	SP,*		
000244	104415		TRAP	15		:			
000246	000417		BR	5#		:			6905
000250	120227	000037	CMPB	R2,#37		:	ELOG.CODE,*		6911
000254	001015		BNE	6#		:			
000256	012716	000000G	MOV	#ASTERISK,(SP)		:			6914
000262	012746	000001	MOV	#1,-(SP)		:			
000266	010600		MOV	SP,R0		:	SP,*		
000270	104415		TRAP	15		:			
000272	013716	000030G	MOV	ERR.COD+30,(SP)		:			6915
000276	012746	000001	MOV	#1,-(SP)		:			
000302	010600		MOV	SP,R0		:	SP,*		
000304	104415		TRAP	15		:			
000306	022626		CMP	(SP)+,(SP)+		:			6913
000310	120227	000005	CMPB	R2,#5		:	ELOG.CODE,*		6918
000314	001031		BNE	7#		:			
000316	020127	000012	CMP	R1,#12		:	ELOG.SUB,*		6919
000322	101026		BHI	7#		:			
000324	012716	000000G	MOV	#CRLF,(SP)		:			6922
000330	012746	000001	MOV	#1,-(SP)		:			
000334	010600		MOV	SP,R0		:	SP,*		
000336	104415		TRAP	15		:			
000340	012716	000000G	MOV	#ASTERISK,(SP)		:			6923
000344	012746	000001	MOV	#1,-(SP)		:			
000350	010600		MOV	SP,R0		:	SP,*		
000352	104415		TRAP	15		:			
000354	010100		MOV	R1,R0		:	ELOG.SUB,*		6924
000356	006300		ASL	R0		:			
000360	016016	000064'	MOV	TBL.MFE(R0),(SP)		:			
000364	012746	000001	MOV	#1,-(SP)		:			
000370	010600		MOV	SP,R0		:	SP,*		
000372	104415		TRAP	15		:			
000374	062706	000006	ADD	#6,SP		:			6921
000400	120227	000010	CMPB	R2,#10		:	ELOG.CODE,*		6927
000404	001031		BNE	8#		:			
000406	020127	000017	CMP	R1,#17		:	ELOG.SUB,*		6928
000412	101026		BHI	8#		:			
000414	012716	000000G	MOV	#CRLF,(SP)		:			6931
000420	012746	000001	MOV	#1,-(SP)		:			
000424	010600		MOV	SP,R0		:	SP,*		
000426	104415		TRAP	15		:			
000430	012716	000000G	MOV	#ASTERISK,(SP)		:			6932
000434	012746	000001	MOV	#1,-(SP)		:			
000440	010600		MOV	SP,R0		:	SP,*		
000442	104415		TRAP	15		:			
000444	010100		MOV	R1,R0		:	ELOG.SUB,*		6933
000446	006300		ASL	R0		:			
000450	016016	000120'	MOV	TBL.DAT(R0),(SP)		:			

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0242
Page 242
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (73)

000454	012746	000001		MOV	#1,-(SP)			
000460	010600			MOV	SP,R0	; SP,*		
000462	104415			TRAP	15			
000464	062706	000006		ADD	#6,SP			6930
000470	120227	000011	8#:	CMPB	R2,#11	; ELOG.CODE,*		6936
000474	001031			BNE	9#			
000476	020127	000004		CMP	R1,#4	; ELOG.SUB,*		6937
000502	101026			BHI	9#			
000504	012716	000000G		MOV	#CRLF,(SP)			6940
000510	012746	000001		MOV	#1,-(SP)			
000514	010600			MOV	SP,R0	; SP,*		
000516	104415			TRAP	15			
000520	012716	000000G		MOV	#ASTERISK,(SP)			6941
000524	012746	000001		MOV	#1,-(SP)			
000530	010600			MOV	SP,R0	; SP,*		
000532	104415			TRAP	15			
000534	010100			MOV	R1,R0	; ELOG.SUB,*		6942
000536	006300			ASL	R0			
000540	016016	000160'		MOV	TBL.HST(R0),(SP)			
000544	012746	000001		MOV	#1,-(SP)			
000550	010600			MOV	SP,R0	; SP,*		
000552	104415			TRAP	15			
000554	062706	000006		ADD	#6,SP			6939
000560	120227	000012	9#:	CMPB	R2,#12	; ELOG.CODE,*		6945
000564	001031			BNE	10#			
000566	020127	000003		CMP	R1,#3	; ELOG.SUB,*		6946
000572	101026			BHI	10#			
000574	012716	000000G		MOV	#CRLF,(SP)			6949
000600	012746	000001		MOV	#1,-(SP)			
000604	010600			MOV	SP,R0	; SP,*		
000606	104415			TRAP	15			
000610	012716	000000G		MOV	#ASTERISK,(SP)			6950
000614	012746	000001		MOV	#1,-(SP)			
000620	010600			MOV	SP,R0	; SP,*		
000622	104415			TRAP	15			
000624	010100			MOV	R1,R0	; ELOG.SUB,*		6951
000626	006300			ASL	R0			
000630	016016	000172'		MOV	TBL.CNT(R0),(SP)			
000634	012746	000001		MOV	#1,-(SP)			
000640	010600			MOV	SP,R0	; SP,*		
000642	104415			TRAP	15			
000644	062706	000006		ADD	#6,SP			6948
000650	120227	000013	10#:	CMPB	R2,#13	; ELOG.CODE,*		6954
000654	001030			BNE	11#			
000656	020127	000010		CMP	R1,#10	; ELOG.SUB,*		6955
000662	101025			BHI	11#			
000664	012716	000000G		MOV	#CRLF,(SP)			6958
000670	012746	000001		MOV	#1,-(SP)			
000674	010600			MOV	SP,R0	; SP,*		
000676	104415			TRAP	15			
000700	012716	000000G		MOV	#ASTERISK,(SP)			6959
000704	012746	000001		MOV	#1,-(SP)			
000710	010600			MOV	SP,R0	; SP,*		

000712	104415			TRAP	15			
000714	006301			ASL	R1			6960
000716	016116	000202'		MOV	TBL.DRV(R1),(SP)			
000722	012746	000001		MOV	#1,-(SP)			
000726	010600			MOV	SP,R0		; SP,*	
000730	104415			TRAP	15			
000732	062706	000006		ADD	#6,SP			6957
000736	020427	000002	11#:	CMP	R4,#2		; REASON,*	6963
000742	001035			BNE	14#			
000744	032763	170000	000060	BIT	#170000,60(R3)		; *,*(ELOG.ADDR)	6966
000752	001014			BNE	12#			
000754	016316	000056		MOV	56(R3),(SP)		; *(ELOG.ADDR),*	6968
000760	005046			CLR	-(SP)			
000762	016346	000056		MOV	56(R3),-(SP)		; *(ELOG.ADDR),*	
000766	012746	000000G		MOV	#XX22,-(SP)			
000772	012746	000004		MOV	#4,-(SP)			
000776	010600			MOV	SP,R0		; SP,*	
001000	104415			TRAP	15			
001002	000413			BR	13#			6966
001004	016316	000056	12#:	MOV	56(R3),(SP)		; *(ELOG.ADDR),*	6970
001010	005046			CLR	-(SP)			
001012	016346	000056		MOV	56(R3),-(SP)		; *(ELOG.ADDR),*	
001016	012746	000000G		MOV	#EX.RBN,-(SP)			
001022	012746	000004		MOV	#4,-(SP)			
001026	010600			MOV	SP,R0		; SP,*	
001030	104415			TRAP	15			
001032	062706	000010	13#:	ADD	#10,SP			6966
001036	012716	000000G	14#:	MOV	#CRLF,(SP)			6972
001042	012746	000001		MOV	#1,-(SP)			
001046	010600			MOV	SP,R0		; SP,*	
001050	104415			TRAP	15			
001052	012716	000002		MOV	#2,(SP)			6973
001056	060316			ADD	R3,(SP)		; ELOG.ADDR,*	
001060	016346	000002		MOV	2(R3),-(SP)		; *(ELOG.ADDR),*	
001064	005216			INC	(SP)			
001066	012746	000002		MOV	#2,-(SP)			
001072	004737	000000G		JSR	PC,BL#DIV			
001076	010066	000002		MOV	R0,2(SP)			
001102	062766	000002	000002	ADD	#2,2(SP)			
001110	005726			TST	(SP)+			
001112	004737	015104'		JSR	PC,EMS.BLK			
001116	105063	000001		CLRB	1(R3)		; *(ELOG.ADDR)	6974
001122	012716	000000G		MOV	#CRLF,(SP)			6975
001126	012746	000001		MOV	#1,-(SP)			
001132	010600			MOV	SP,R0		; SP,*	
001134	104415			TRAP	15			
001136	062706	000020		ADD	#20,SP			6872
001142	000207			RTS	PC			

; Routine Size: 306 words, Routine Base: #CODE# + 15660
; Maximum stack depth per invocation: 17 words

```

: 6977 global routine EMS_CMP (ADDR) : novalue =
: 6978
: 6979 !*
: 6980 ! THIS ROUTINE IS CALLED FROM 'HOST_WRT_CHK' AND PRINTS RELEVANT DATA ON A HOST
: 6981 ! COMPARE ERROR
: 6982 !-
: 6983
: 6984 begin
: 6985
: 6986 local
: 6987     ORIG_ADDR : ref block [RP_LEN, word] field (RP_FIELDS);
: 6988
: 6989     ORIG_ADDR = .ADDR;                                ! ADDRESS OF THE WRITE RETPKT
: 6990     PRINTB (XX13, .CDISK);                            ! "DISK XXX"
: 6991     PRINTB (DASH);                                    !
: 6992     PRINTB (.ERR_COD [12]);                          ! " - HOST COMPARE ERROR"
: 6993     PRINTX (EX_LBW, .ORIG_ADDR [LBN_LO], .ORIG_ADDR [LBN_HI], .ORIG_ADDR [LBN_LO]); ! LBN (WRITE)
: 6994     PRINTX (EX_LBR, .RP_ADDR [LBN_LO], .RP_ADDR [LBN_HI], .RP_ADDR [LBN_LO]); ! LBN (READ)
: 6995     PRINTX (EX_CBW, .ORIG_ADDR [CBCNT_LO]);          ! BYTE COUNT (WRITE)
: 6996     PRINTX (XX25, .ORIG_ADDR [BCNT_LO]);            ! BYTE COUNT XMITTED (WRITE)
: 6997     PRINTX (EX_CBR, .RP_ADDR [CBCNT_LO]);          ! BYTE COUNT (READ);
: 6998     PRINTX (XX25, .RP_ADDR [BCNT_LO]);              ! BYTE COUNT XMITTED (READ)
: 6999     PRINTX (EX_BDW, .ORIG_ADDR [BUFF_1], .ORIG_ADDR [BUFF_0]); ! BUFFER ADDRESS (WRITE)
: 7000     PRINTX (EX_BDR, .RP_ADDR [BUFF_1], .RP_ADDR [BUFF_0]); ! BUFFER ADDRESS (READ)
: 7001 end;

```

Address	Offset	Label	Instruction	Comment	Line No.
000000	004137	000000G	.SBTTL EMS.CMP ERROR MESSAGE SUBROUTINES		
		EMS.CMP:	JSR R1, #SAVE2		6977
000004	016602	000010	MOV 10(SP), R2	! ADDR, ORIG_ADDR	6989
000010	013746	000000G	MOV CDISK, -(SP)	!	6990
000014	012746	000000G	MOV #XX13, -(SP)		
000020	012746	000002	MOV #2, -(SP)		
000024	010600		MOV SP, R0	! SP, *	
000026	104414		TRAP 14		
000030	012716	000000G	MOV #DASH, (SP)		6991
000034	012746	000001	MOV #1, -(SP)		
000040	010600		MOV SP, R0	! SP, *	
000042	104414		TRAP 14		
000044	013716	000030G	MOV ERR_COD+30, (SP)		6992
000050	012746	000001	MOV #1, -(SP)		
000054	010600		MOV SP, R0	! SP, *	
000056	104414		TRAP 14		
000060	016216	000050	MOV 50(R2), (SP)	! *(ORIG.ADDR), *	6993
000064	016246	000052	MOV 52(R2), -(SP)	! *(ORIG.ADDR), *	
000070	016246	000050	MOV 50(R2), -(SP)	! *(ORIG.ADDR), *	
000074	012746	000000G	MOV #EX.LBW, -(SP)		
000100	012746	000004	MOV #4, -(SP)		
000104	010600		MOV SP, R0	! SP, *	
000106	104415		TRAP 15		
000110	013701	000000G	MOV RP_ADDR, R1		6994
000114	016116	000050	MOV 50(R1), (SP)		

M3

ZRQAM2 RD/RX EXERCISER
V01.2 ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0245
Page 245
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (74)

000120	010100		MOV	R1,R0	; RP.ADDR,*	
000122	016046	000052	MOV	52(R0),-(SP)		
000126	016146	000050	MOV	50(R1),-(SP)		
000132	012746	000000G	MOV	#EX.LBR,-(SP)		
000136	012746	000004	MOV	#4,-(SP)		
000142	010600		MOV	SP,R0	; SP,*	
000144	104415		TRAP	15		
000146	016216	000044	MOV	44(R2),(SP)	; *(ORIG.ADDR),*	6995
000152	012746	000000G	MOV	#EX.CBW,-(SP)		
000156	012746	000002	MOV	#2,-(SP)		
000162	010600		MOV	SP,R0	; SP,*	
000164	104415		TRAP	15		
000166	016216	000020	MOV	20(R2),(SP)	; *(ORIG.ADDR),*	6996
000172	012746	000000G	MOV	#XX25,-(SP)		
000176	012746	000002	MOV	#2,-(SP)		
000202	010600		MOV	SP,R0	; SP,*	
000204	104415		TRAP	15		
000206	013700	000000G	MOV	RP.ADDR,R0		6997
000212	016016	000044	MOV	44(R0),(SP)		
000216	012746	000000G	MOV	#EX.CBR,-(SP)		
000222	012746	000002	MOV	#2,-(SP)		
000226	010600		MOV	SP,R0	; SP,*	
000230	104415		TRAP	15		
000232	013700	000000G	MOV	RP.ADDR,R0		6998
000236	016016	000020	MOV	20(R0),(SP)		
000242	012746	000000G	MOV	#XX25,-(SP)		
000246	012746	000002	MOV	#2,-(SP)		
000252	010600		MOV	SP,R0	; SP,*	
000254	104415		TRAP	15		
000256	016216	000024	MOV	24(R2),(SP)	; *(ORIG.ADDR),*	6999
000262	016246	000026	MOV	26(R2),-(SP)	; *(ORIG.ADDR),*	
000266	012746	000000G	MOV	#EX.BDW,-(SP)		
000272	012746	000003	MOV	#3,-(SP)		
000276	010600		MOV	SP,R0	; SP,*	
000300	104415		TRAP	15		
000302	013700	000000G	MOV	RP.ADDR,R0		7000
000306	016016	000024	MOV	24(R0),(SP)		
000312	016046	000026	MOV	26(R0),-(SP)		
000316	012746	000000G	MOV	#EX.BDR,-(SP)		
000322	012746	000003	MOV	#3,-(SP)		
000326	010600		MOV	SP,R0	; SP,*	
000330	104415		TRAP	15		
000332	062706	000066	ADD	#66,SP		6984
000336	000207		RTS	PC		6977

; Routine Size: 112 words, Routine Base: #CODE# + 17024
; Maximum stack depth per invocation: 32 words

ZRQAM2 RD/RX EXERCISER
V01.2 ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (75)

: 7002 BGNMSG (EMS_01);

000000	004737	000000V		.SBTTL	EMS.01 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS.01::	JSR	PC,M#EMS.01	7002
000006	000207			TRAP	23	
				RTS	PC	

: Routine Size: 4 words, Routine Base: \$CODE\$ + 17364
: Maximum stack depth per invocation: 2 words

: 7003 PRINTB (EBS_01, MAX_UNITS); ! "MORE THAN XX UNITS SPECIFIED"
: 7004 ENDMSG;

000000	012746	000004		.SBTTL	M#EMS.01 ERROR MESSAGE SUBROUTINES	
000004	012746	000000G	M#EMS.01:	MOV	#4,-(SP)	7003
000010	012746	000002		MOV	#EBS.01,-(SP)	
000014	010600			MOV	#2,-(SP)	
000016	104414			MOV	SP,R0	: SP,*
000020	062706	000006		TRAP	14	
000024	000207			ADD	#6,SP	
				RTS	PC	7002

: Routine Size: 11 words, Routine Base: \$CODE\$ + 17374
: Maximum stack depth per invocation: 5 words

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (76)

: 7005 BGNMSG (EMS_10);

000000	004737	000000V	EMS.10::	.SBTTL	EMS.10 ERROR MESSAGE SUBROUTINES		
000004	104423			JSR	PC,M#EMS.10	,	7005
000006	000207			TRAP	23		
				RTS	PC		

: Routine Size: 4 words, Routine Base: #CODE# + 17422
: Maximum stack depth per invocation: 2 words

: 7006 PRINTB (EBD_10, .RDRX_ADDR + .OF_RC); ! "NO RESPONSE AT ADDRESS XXXXXX"
: 7007 ENDMSG;

000000	013746	000000G	M#EMS.10:	.SBTTL	M#EMS.10 ERROR MESSAGE SUBROUTINES		
000004	063716	000000G		MOV	RDRX.ADDR, -(SP)	,	7006
000010	012746	000000G		ADD	OF_RC, (SP)		
000014	012746	000002		MOV	#EBD.10, -(SP)		
000020	010600			MOV	#2, -(SP)		
000022	104414			MOV	SP, R0	, SP, *	
000024	062706	000006		TRAP	14		
000030	000207			ADD	#6, SP	,	7005
				RTS	PC		

: Routine Size: 13 words, Routine Base: #CODE# + 17432
: Maximum stack depth per invocation: 5 words

: 7008 BGNMSG (EMS_12);

```

000000 004737 000000V      .SBTTL  EMS.12 ERROR MESSAGE SUBROUTINES
000004 104423      EMS.12::JSR  PC,M$EMS.12      ;
000006 000207      TRAP  23
      RTS  PC

```

: Routine Size: 4 words, Routine Base: \$CODE\$ + 17464
 : Maximum stack depth per invocation: 2 words

: 7009 PRINTB (EBD_12, .RDRX_ADDR); ! "INCORRECT BR LEVEL GIVEN FOR DEVICE XXXXXX"
 : 7010 ENDMSG;

```

000000 013746 000000G      .SBTTL  M$EMS.12 ERROR MESSAGE SUBROUTINES
      M$EMS.12:
000004 012746 000000G      MOV  RDRX.ADDR, -(SP)      ;
000010 012746 0000002      MOV  #EBD.12, -(SP)
000014 010600      MOV  #2, -(SP)
000016 104414      MOV  SP, R0      ; SP,*
000020 062706 0000006      TRAP  14
000024 000207      ADD  #6, SP      ;
      RTS  PC

```

: Routine Size: 11 words, Routine Base: \$CODE\$ + 17474
 : Maximum stack depth per invocation: 5 words

ZRQAM2 RD/RX EXERCISER
V01.2 ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0249
Page 249
VAX-11 B111-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (78)

: 7011 BGNMSG (EMS_13);

000000	004737	000000V		.SBTTL	EMS.13 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS.13::	JSR	PC,M#EMS.13	7011
000006	000207			TRAP	23	
				RTS	PC	

: Routine Size: 4 words, Routine Base: %CODE# + 17522
: Maximum stack depth per invocation: 2 words

:	7012	PRINTB (EBD_13, .STEP);	!	"STEP X READ ERROR"
:	7013	EMS_SA ();	!	PRINTX SA CONTENTS
:	7014	ENDMSG;		

000000	013746	000000G		.SBTTL	M#EMS.13 ERROR MESSAGE SUBROUTINES	
000004	012746	000000G	M#EMS.13:	MOV	STEP,-(SP)	7012
000010	012746	000002		MOV	#EBD.13,-(SP)	
000014	010600			MOV	#2,-(SP)	
000016	104414			MOV	SP,R0	; SP,4
000020	004737	007536'		TRAP	14	
000024	062706	000006		JSR	PC,EMS.SA	
000030	000207			ADD	#6,SP	
				RTS	PC	7011

: Routine Size: 13 words, Routine Base: %CODE# + 17532
: Maximum stack depth per invocation: 5 words

ZRQAM2 RD/RX EXERCISER
V01.2 ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (79)

: 7015 BGNMSG (EMS_14);

000000	004737	000000V		.SBTTL	EMS.14 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS.14::	JSR	PC,M\$EMS.14	7015
000006	000207			TRAP	23	
				RTS	PC	

: Routine Size: 4 words, Routine Base: \$CODE\$ + 17564
: Maximum stack depth per invocation: 2 words

:	7016	PRINTB (EBD_14, .IRDRX_ADDR);	!	"BAD SA CODE FROM DEVICE XXXXXX"
:	7017	EMS_SA ();	!	PRINTX SA REGISTER CONTENTS
:	7018	ENDMSG;		

000000	013746	000000G		.SBTTL	M\$EMS.14 ERROR MESSAGE SUBROUTINES	
			M\$EMS.14:	MOV	IRDRX_ADDR, -(SP)	7016
000004	012746	000000G		MOV	#EBD.14, -(SP)	
000010	012746	000002		MOV	#2, -(SP)	
000014	010600			MOV	SP, R0	: SP, *
000016	104414			TRAP	14	
000020	004737	007536'		JSR	PC, EMS_SA	7017
000024	062706	000006		ADD	#6, SP	7015
000030	000207			RTS	PC	

: Routine Size: 13 words, Routine Base: \$CODE\$ + 17574
: Maximum stack depth per invocation: 5 words

ZRQAM2 RD/RX EXERCISER
V01.2 ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0251
Page 251
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (80)

: 7019 BGNMSG (EMS_18);

000000	004737	000000V		.SBTTL	EMS.18 ERROR MESSAGE SUBROUTINES		
000004	104423		EMS.18::	JSR	PC,M#EMS.18	;	7019
000006	000207			TRAP	23		
				RTS	PC		

: Routine Size: 4 words, Routine Base: \$CODE\$ + 17626
: Maximum stack depth per invocation: 2 words

:	7020	PRINTB (EBD_18, .CDISK);	!	"DISK XXX WENT OFFLINE"
:	7021	EMSCMD ();	!	PRINTX RELEVANT RETPKT FIELDS
:	7022	ENDMSG;		

000000	013746	000000G		.SBTTL	M#EMS.18 ERROR MESSAGE SUBROUTINES		
000004	012746	000000G	M#EMS.18:	MOV	CDISK,-(SP)	;	7020
000010	012746	000002		MOV	#EBD.18,-(SP)		
000014	010600			MOV	#2,-(SP)		
000016	104414			MOV	SP,R0	;	SP,*
000020	004737	010572'		TRAP	14		
000024	062706	000006		JSR	PC,EMSCMD	;	7021
000030	000207			ADD	#6,SP	;	7019
				RTS	PC		

: Routine Size: 13 words, Routine Base: \$CODE\$ + 17636
: Maximum stack depth per invocation: 5 words

ZRGAM2
V01.2RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES5-Dec-1983 10:27:14
5-Dec-1983 10:27:04VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRGACO.BL1;82 (81)SEQ 0252
Page 252:
7023 BGNMSG (EMS_21);

000000	004737	000000V		.SBTTL	EMS.21 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS.21::	JSR	PC,M\$EMS.21	7023
000006	000207			TRAP	23	
				RTS	PC	

: Routine Size: 4 words, Routine Base: \$CODE\$ + 17670
 : Maximum stack depth per invocation: 2 words

:	7024	EMSCMD ();	:	CONTENTS OF RETURN PACKET
:	7025	!EMS_RAL ();	:	all return packets
:	7026	EMS_MAL ();	:	all message packets
:	7027	ENDMSG;		

000000	004737	010572'		.SBTTL	M\$EMS.21 ERROR MESSAGE SUBROUTINES	
000004	004737	015256'	M\$EMS.21:	JSR	PC,EMSCMD	7024
000010	000207			JSR	PC,EMS.MAL	7026
				RTS	PC	7023

: Routine Size: 5 words, Routine Base: \$CODE\$ + 17700
 : Maximum stack depth per invocation: 1 word

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

SEQ 0253
Page 253
VAX-11 Bliss-16 V3-555
DISK#USER2:(DIETZ.RDRX)ZRQACO.BL1;82 (82)

: 7028 BGNMSG (EMS_22)

000000	004737	000000V		.SBTTL	EMS.22 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS.22::	JSR	PC,M#EMS.22	7028
000006	000207			TRAP	23	
				RTS	PC	

: Routine Size: 4 words, Routine Base: \$CODE\$ + 17712
: Maximum stack depth per invocation: 2 words

: 7029 EMS_DBN (); ! contents of dup buffer
: 7030 ENDMSG;

000000	004737	014344'		.SBTTL	M#EMS.22 ERROR MESSAGE SUBROUTINES	
000004	000207		M#EMS.22::	JSR	PC,EMS.DBN	7029
				RTS	PC	7028

: Routine Size: 3 words, Routine Base: \$CODE\$ + 17722
: Maximum stack depth per invocation: 1 word

ZRQAM2 RD/RX EXERCISER
 V01.2 ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
 5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
 DISK#USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (83)

: 7031 BGNMSG (EMS_30);

```

000000 004737 000000V          .SBTTL EMS.30 ERROR MESSAGE SUBROUTINES
000004 104423          EMS.30::JSR PC,M#EMS.30 ;
000006 000207          TRAP 23 ;
          RTS PC ;
  
```

: Routine Size: 4 words, Routine Base: \$CODE\$ + 17730
 : Maximum stack depth per invocation: 2 words

: 7032 EMSCMD (); ! contents of return packet
 : 7033 ENDMSG;

```

000000 004737 010572'          .SBTTL M#EMS.30 ERROR MESSAGE SUBROUTINES
000004 000207          M#EMS.30: JSR PC,EMSCMD ;
          RTS PC ;
  
```

: Routine Size: 3 words, Routine Base: \$CODE\$ + 17740
 : Maximum stack depth per invocation: 1 word

: 7034
 : 7035 end
 : 7036
 : 7037 eludom

OTS external references

```

.GLOBL $SAVE5, $SAVE4, $SAVE3, $SAVE2
.GLOBL BL$DIV, BL$MOD, BL$MUL
  
```

PSECT SUMMARY

Psect Name	Words	Attributes
\$OWN\$	74	RW . D . LCL. REL. CON
\$CODE\$	4083	RO . I . LCL. REL. CON
\$PLIT\$	12	RO . D . LCL. REL. CON

LIBRARY STATISTICS

File	Total	Symbols Loaded	Percent	Blocks Read
DISK#USER2:[DIETZ.RDRX]ZRQACO.L16;14	400	296	74	82

J4

ZRQAM2
V01.2

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

5-Dec-1983 10:27:14
5-Dec-1983 10:27:04

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL1;82 (83)

SEQ 0255
Page 255

: Information: 1
: Warnings: 0
: Errors: 0

COMMAND QUALIFIERS

: BLISS /PDP11 ZRQACO.BL1/LIST=ZRQACO.LI1/OBJECT=ZRQACO.OB1/SOURCE=PAGE:53

: Size: 3922 code + 6972 data words
: Run Time: 03:47.9
: Elapsed Time: 34:26.9
: Memory Used: 695 pages
: Compilation Complete

```

: 0001 module ZRQAM3 (
: 0002
: 0003 $title 'RD/RX EXERCISER'
: 0004         ident = 'V01.2',
: 0005         addressing_mode (absolute),
: 0006         environment (noeis)
: 0007     ) =
: 0008
: 0009 begin
: 0010
: 0011 $sbttl 'DECLARATIONS'
: 0012
: 0013 library 'ZRQACO.L16';           ! RDRX EXERCISER GLOBAL LIBRARY
: 0014
: 0015 require 'BLSMAC.REQ';        ! DIAGNOSTIC SUPERVISOR LIBRARY
: 1506
: 1507 EQUALS;
: 1508
: 1509 forward routine                ! ROUTINES APPEAR IN THIS ORDER
: 1510     INIT_TEST : novalue,        !     INDENTATION IMPLIES CALLED SUBROUTINE
: 1511     DRIVER_INIT : novalue,
: 1512     CTLR_INIT : novalue,
: 1513     INI_CTLR_DAT : novalue,
: 1514     REG_EXIST,
: 1515     VEC_BR_TEST,
: 1516     INT_GEN,
: 1517     HARD_INIT,
: 1518     INI_RRING : novalue,
: 1519     SET_CTLR_CHAR,
: 1520     UNIT_INIT : novalue,
: 1521     DR_ERR : novalue,
: 1522     ACCESS : novalue,
: 1523     MULTI_DRIVE : novalue,
: 1524     MD_INIT : novalue,
: 1525     INIT_IO_BUFF : novalue,
: 1526     FATAL_ERROR : novalue,
: 1527     QIO_OK,
: 1528     QIO_OUT,
: 1529     QIO_GEN : novalue,
: 1530     GET_RANDOM : novalue,
: 1531     QIO_UNIT : novalue,
: 1532     QIO_FUNC : novalue,
: 1533     DUP : novalue,
: 1534     DUPWRTOBN : novalue,
: 1535     DUPREDOBN : novalue,
: 1536     DUPCOMMAND : novalue,
: 1537     DUPIDLE : novalue,
: 1538     QIO_LBN : novalue,
: 1539     QIO_SIZE : novalue,
: 1540     FILL_BUFF : novalue,
: 1541     PROC_RETPKT : novalue,
: 1542     DIO_RETPKT : novalue,
: 1543     DUP_COMPARE : novalue,

```

```

: 1544         IO_RETPKT : novalue,
: 1545         FSET_UPAR : novalue,
: 1546         HARD_ERROR : novalue,
: 1547         UPD_IO_TALLY : novalue,
: 1548         OVF_CHK : novalue,
: 1549         HOST_WRT_CHK,
: 1550         SWEEP : novalue,
: 1551         RPS_REM,
: 1552         DR_RETPKT : novalue,
: 1553     AZINTO : L$ISR novalue,
: 1554     AZINT : novalue,
: 1555 !         FATAL_ERROR : novalue,
: 1556         POLL_CRING : novalue,
: 1557         POLL_RRING : novalue,
: 1558         DUP_RSP : novalue,
: 1559         MSCP_RSP : novalue,
: 1560         SEQUEN : novalue,
: 1561         SOFT_ERROR : novalue,
: 1562         DATAGM : novalue;
: 1563 !         SOFT_ERROR : novalue;
: 1564
: 1565 external
: 1566     CST : blockvector [MAX_CTLR, CST_LEN, word] field (CST_FIELDS),
: 1567         ! RUN-TIME CONTROLLER STATUS TABLES
: 1568     CST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 1569         ! CONTROLLER STATUS TABLE ADDRESS OF "CURRENT" CONTROLLER
: 1570     DCT : blockvector [MAX_CTLR, DCT_LEN, word] field (DCT_FIELDS),
: 1571         ! DRIVER CONTROLLER TABLES
: 1572     DCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 1573         ! ADDRESS OF "CURRENT" DRIVER CONTROLLER TABLE
: 1574     RDRX_ADDR : ref rdx field (RC_REG),
: 1575         ! DEVICE ADDRESS OF "CURRENT" CONTROLLER
: 1576     IRDRX_ADDR : ref rdx field (RC_REG),
: 1577         ! DEVICE ADDRESS OF INTERRUPTING CONTROLLER
: 1578     ICOM_ADDR : ref block [COM_LEN, word] field (COM_FIELDS),
: 1579         ! ADDRESS OF INTERRUPTING CONTROLLER'S COMMUNICATION AREA
: 1580     ICST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 1581         ! ADDRESS OF INTERRUPTING CONTROLLER'S CST
: 1582     IDCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 1583         ! ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 1584     DUPPKT : BLOCK [257, WORD] field (DP_FIELDS),
: 1585         ! BUFFER CONTAINING DUP INFORMATION FROM RECEIVE AND SEND COMMANDS
: 1586     BST : blockvector [MAX_UNITS, 2, word],
: 1587         ! BLOCK SEQUENCE TABLE FOR SEQUENTIAL LBN (VS. RANDOM SEEK) MODE
: 1588     TALLY : vector [MAX_UNITS * TALLY_LEN, word] field (T_FIELDS),
: 1589         ! STATISTICS TABLES
: 1590     T_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 1591         ! ADDRESS OF STATISTICS TABLE (TALLY) FOR CURRENT UNIT
: 1592     C_ERR_TBL : blockvector [MAX_CTLR, C_ERR_LEN, word] field (C_ERR_FIELDS),
: 1593         ! STATISTICS TABLE FOR CONTROLLER ERRORS
: 1594     MSCP_PKT : blockvector [PKT_CNT, PKT_LEN, word] field (PKT_FIELDS),
: 1595         ! MSCP PACKET POOL
: 1596     IPKT_ADDR : ref block [PKT_LEN, word] field (PKT_FIELDS),

```

```

: 1597          : ADDRESS OF AN MSCP PACKET (INTERRUPT PROCESSING)
: 1598 PKT_USE : vector [PKT_CNT, byte, signed],
: 1599          : MSCP PACKET POOL ALLOCATION TABLE
: 1600 RETPKT : blockvector [RP_CNT, RP_LEN, word] field (RP_FIELDS),
: 1601          : RETURN PACKET POOL
: 1602 RP_USE : vector [RP_CNT, byte, signed],
: 1603          : RETURN PACKET POOL ALLOCATION TABLE
: 1604 RP_INDX : word,          : CURRENT RETURN PACKET INDEX
: 1605 RP_ADDR : ref block [RP_LEN, word] field (RP_FIELDS),
: 1606          : CURRENT RETURN PACKET ADDRESS
: 1607 RDM_CNT : word,          : NUMBER OF RANDOM NUMBERS \ KEEP
: 1608 RANDOM : vector [RDM_LEN, word],      : RANDOM NUMBER TABLE / TOGETHER
: 1609 TRK_SGN : vector [MAX_UNITS, byte, signed], : CURRENT TRACK DIRECTION
: 1610 ELOG_PKT : blockvector [EP_CNT, EP_LEN, word] field (EP_FIELDS),
: 1611          : ERROR-LOG PACKET SAVE AREA
: 1612 BUFF_ADDR : vector [MAX_BUF_CNT],      : TABLE OF I/O BUFFER DESCRIPTORS
: 1613 BUFF_OWN : vector [MAX_BUF_CNT, byte, signed], : I/O BUFFER OWNERSHIP (CONTROLLER NUMBER)
: 1614 IODQ : vector [IODQ_LEN, byte],
: 1615          : I/O DONE QUEUE - CIRCULAR QUEUE OF RETPKT INDECS
: 1616 IODQ_IN : word,          : I/O DONE QUEUE IN POINTER
: 1617 IODQ_OUT : word,        : I/O DONE QUEUE OUT POINTER
: 1618 ENTRY_REASON : byte,    : CURRENT OPERATOR COMMAND
: 1619 EOP_FLAG : byte,        : END-OF-PASS FLAG
: 1620 DUP_FLAGS : WORD,       : DUP FLAGS
: 1621 CCTRL : word,           : NUMBER OF "CURRENT" CONTROLLER
: 1622 CDISK : word,          : CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 1623 CUOFF : word,         : CURRENT UNIT CST OFFSET
: 1624 CTRL_CNT : word,      : TOTAL NUMBER OF CONFIGURED CONTROLLERS
: 1625 DUR : vector [MAX_UNITS, byte], : DROP UNIT REASON
: 1626 QIO : vector [MAX_CTRL, byte],  : NUMBER OF OUTSTANDING QIOS PER CONTROLLER
: 1627 FREE_MEM_ADDR, : START OF FREE MEMORY
: 1628 BYTS_PER_QIO : word,    : SIZE (BYTES) OF AN I/O BUFFER
: 1629 ST_CODE : word,        : CURRENT STATUS CODE
: 1630 SB_CODE : word,       : CURRENT SUB-CODE
: 1631 STEP : word,          : CURRENT STEP IN HARD INIT
: 1632 OF_RC : signed word,  : OFFSET (0 OR 2) TO READ IP OR SA
: 1633 SA_REG : word,        : STORAGE FOR SA REGISTER READS AND WRITES
: 1634 CMD_TIME : word,     : COMMAND TIMEOUT VALUE (IN SECONDS)
: 1635 NEX : word,           : NON-EXISTENT MEMORY TRAP INDICATOR
: 1636 CRN_LOW : word,       : COMMAND REF NUMBER OF LAST COMMAND SENT
: 1637 CRN_HIGH : word,     : COMMAND REF NUMBER OF LAST COMMAND SENT
: 1638 P_INDEX : signed word, : CURRENT message PACKET INDEX
: 1639 S_DUPPKT : WORD,      : DBN BYTE COUNTER
: 1640 S_PATTERN : WORD,    : THE PATTERN WRITTEN TO DBN'S
: 1641 CREDIT_BAL : word,   : CREDIT BALANCE
: 1642 INIT_OCCURED : byte,  : indicates if exerciser finished init sequence
: 1643 NXT_PKT_2USE : byte,  : POINTER TO NEXT ENTRY IN PKT_USE TABLE
: 1644 DBM12,
: 1645 DBM18,
: 1646 DBM19,
: 1647 DBM20,
: 1648 DBM21,
: 1649 !DBM22,

```

ZRQAM3
V01.2

RD/RX EXERCISER
DECLARATIONS

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (1)

```

: 1650 DBM23.
: 1651 DBM25.
: 1652 DBM26.
: 1653 DBM29.
: 1654 DBM108.
: 1655 DBM109.
: 1656 DBM110.
: 1657 DBM112.
: 1658 !der10.
: 1659 !der13.
: 1660 EH_0.
: 1661 EH_1.
: 1662 EH_2.
: 1663 EH_3.
: 1664 EH_4.
: 1665 EH_5.
: 1666 EH_6.
: 1667 EH_7.
: 1668 EH_8.
: 1669 EH_9.
: 1670 EH_10.
: 1671 EH_12.
: 1672 EH_13.
: 1673 MSG_02.
: 1674 MSG_03.
: 1675 EGS_02.
: 1676 EGD_10.
: 1677 EGD_11.
: 1678 EGD_12.
: 1679 EGD_13.
: 1680 EGD_14.
: 1681 EGD_15.
: 1682 EGD_16.
: 1683 EGD_17.
: 1684 EGD_18.
: 1685 EGD_19.
: 1686 EGD_20.
: 1687 EGD_21.
: 1688 EGD_22.
: 1689 EGD_23.
: 1690 EGD_24.
: 1691 EGH_30.
: 1692 EX_GDS.
: 1693 EX_ESP.
: 1694 EX_ELP.
: 1695 EX_RCD.
: 1696 EX_SDD.
: 1697 EX_ABP.
: 1698 EX_WRD.
: 1699
: 1700 CRLF.
: 1701 SWP_ERROR : word.
: 1702 SWP_XFER : word.

```

! HARD ERROR LIMIT FOR DROPPING UNIT
! TRANSFER LIMIT FOR DROPPING UNIT

```

: 1703 SWP_FLAGS : word, ! FLAGS (SEE DOCUMENTATION)
: 1704 dupound : word, ! ratio constant for dup exerciser
: 1705 SWP_RAT : word, ! RDS1 OPERATION RATIO
: 1706 SWP_DPAT : word, ! DATA PATTERN NUMBER
: 1707 SWP_UCNT : word, ! USER DATA PATTERN COUNT
: 1708 SWP_UDPAT : vector [MAX_UDP_CNT, word], ! USER DATA PATTERN
: 1709 L$LUN,
: 1710 L$UNIT;
:
: 1711
: 1712 psect
: 1713 own = $GGG$(read, nowrite, execute, local, concatenate);
: 1714
: 1715 own
: 1716 COMM_AREA : blockvector [MAX_CTLR, COMM_LEN, word] field (COM_FIELDS),
: 1717 ! COMMUNICATIONS AREA BETWEEN HOST AND AZTEC CONTROLLERS
: 1718
: 1719 DPST : vector [MAX_UNITS, byte], ! DATA PATTERN SEQUENCE TABLE
: 1720 MAX_LBN : vector [MAX_UNITS, word], ! LARGEST LBN ALLOWED
: 1721 STORAGE : vector [MAX_UNITS, word], ! DUMMY STORAGE
: 1722 INT_ADDR : vector [MAX_CTLR] initial (AZINT0 $(, AZINT1, AZINT2, AZINT3)$),
: 1723 ! INTERRUPT SERVICE ROUTINE ADDRESS TABLE
: 1724 ICTLR : word, ! INTERRUPTING CONTROLLING NUMBER
: 1725 MX1 : signed word, ! MSCP PKT INDEX FOR FIRST QIO
: 1726 MX2 : signed word, ! MSCP PKT INDEX FOR SECOND QIO
: 1727 MAD1 : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 1728 ! ADDRESS OF MSCP PACKET FOR FIRST QIO
: 1729 MAD2 : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 1730 ! ADDRESS OF MSCP PACKET FOR SECOND QIO
: 1731 LAST_PKT : blockvector [MAX_CTLR, LAST_PKT_LEN, word] field (LAST_PKT_FIELDS),
: 1732 ! SAVE AREA FOR INFO ABOUT LAST RESPONSE PACKET
: 1733 PAT02 : vector [2] initial (1, ! PATTERN 2
: 1734 %o'000000'),
: 1735 PAT03 : vector [2] initial (1, ! PATTERN 3
: 1736 %o'177777'),
: 1737 PAT04 : vector [2] initial (1, ! PATTERN 4
: 1738 %o'105613'),
: 1739 PAT05 : vector [2] initial (1, ! PATTERN 5
: 1740 %o'031463'),
: 1741 PAT06 : vector [2] initial (1, ! PATTERN 6
: 1742 %o'030221'),
: 1743 PAT07 : vector [17] initial (16, ! PATTERN 7
: 1744 %o'000001', %o'000003', %o'000007', %o'000017',
: 1745 %o'000037', %o'000077', %o'000177', %o'000377',
: 1746 %o'000777', %o'001777', %o'003777', %o'007777',
: 1747 %o'017777', %o'037777', %o'077777', %o'177777'),
: 1748 PAT08 : vector [17] initial (16, ! PATTERN 8
: 1749 %o'177776', %o'177774', %o'177770', %o'177760',
: 1750 %o'177740', %o'177700', %o'177600', %o'177400',
: 1751 %o'177000', %o'176000', %o'174000', %o'170000',
: 1752 %o'160000', %o'140000', %o'100000', %o'000000'),
: 1753 PAT09 : vector [17] initial (16, ! PATTERN 9
: 1754 rep 3 of (%o'000000'), rep 3 of (%o'177777'),
: 1755 rep 2 of (%o'000000'), rep 2 of (%o'177777'),

```

```

: 1756          %o'000000', %o'177777', %o'000000', %o'177777',
: 1757          %o'000000', %o'177777'),
: 1758 PAT10 : vector [2] initial (1,          ! PATTERN 10
: 1759          %o'133331'),
: 1760 PAT11 : vector [17] initial (16,        ! PATTERN 11
: 1761          rep 3 of (%o'052525'), rep 3 of (%o'125252'),
: 1762          rep 2 of (%o'052525'), rep 2 of (%o'125252'),
: 1763          %o'052525', %o'125252', %o'052525', %o'125252',
: 1764          %o'052525', %o'125252'),
: 1765 PAT12 : vector [21] initial (20,        ! PATTERN 12
: 1766          rep 3 of (%o'026455'), rep 3 of (%o'151322'),
: 1767          rep 2 of (%o'026455'), rep 2 of (%o'151322'),
: 1768          rep 2 of (%o'026455'),
: 1769          %o'151322', %o'026455', %o'151322', %o'026455',
: 1770          %o'151322', %o'026455', %o'151322', %o'026455'),
: 1771 PAT13 : vector [2] initial (1,          ! PATTERN 13
: 1772          %o'066666'),
: 1773 PAT14 : vector [17] initial (16,        ! PATTERN 14
: 1774          %o'000001', %o'000002', %o'000004', %o'000010',
: 1775          %o'000020', %o'000040', %o'000100', %o'000200',
: 1776          %o'000400', %o'001000', %o'002000', %o'004000',
: 1777          %o'010000', %o'020000', %o'040000', %o'100000'),
: 1778 PAT15 : vector [17] initial (16,        ! PATTERN 15
: 1779          %o'177776', %o'177775', %o'177773', %o'177767',
: 1780          %o'177757', %o'177737', %o'177677', %o'177577',
: 1781          %o'177377', %o'176777', %o'175777', %o'173777',
: 1782          %o'167777', %o'157777', %o'137777', %o'077777'),
: 1783 PAT16 : vector [17] initial (16,        ! PATTERN 16
: 1784          rep 3 of (%o'133331'), rep 3 of (%o'155554'),
: 1785          rep 2 of (%o'133331'), rep 2 of (%o'155554'),
: 1786          %o'133331', %o'155554', %o'133331', %o'155554',
: 1787          %o'133331', %o'155554'),
: 1788 PAT17 : vector [22] initial (21,        ! PATTERN 17
: 1789          %o'000000', rep 2 of (%o'106466'),
: 1790          rep 3 of (%o'071311'), rep 4 of (%o'106466'),
: 1791          rep 5 of (%o'071311'), rep 6 of (%o'106466')),
: 1792 PAT18 : vector [22] initial (21,        ! PATTERN 18
: 1793          %o'106466', %o'000000', %o'071311',
: 1794          rep 3 of (%o'106466'), rep 4 of (%o'071311'),
: 1795          rep 5 of (%o'106466'), rep 6 of (%o'071311')),
: 1796 PAT19 : vector [22] initial (21,        ! PATTERN 19
: 1797          %o'000000', rep 2 of (%o'134631'),
: 1798          rep 3 of (%o'043146'), rep 4 of (%o'134631'),
: 1799          rep 5 of (%o'043146'), rep 6 of (%o'134631')),
: 1800 PAT20 : vector [22] initial (21,        ! PATTERN 20
: 1801          %o'134631', %o'000000', %o'043146',
: 1802          rep 3 of (%o'134631'), rep 4 of (%o'043146'),
: 1803          rep 5 of (%o'134631'), rep 6 of (%o'043146')),
: 1804 PAT21 : vector [2] initial (1,          ! PATTERN 21
: 1805          %o'000000'),
: 1806 DPA_TBL : vector [DP_CNT] initial        ! (LBN)
: 1807          (RDM_CNT, PAT02, PAT03, PAT04, PAT05,
: 1808          PAT06, PAT07, PAT08, PAT09, PAT10, PAT11,
:          ! DATA PATTERN ADDRESS TABLE

```



```

: 1809          PAT12, PAT13, PAT14, PAT15, PAT16, PAT17,
: 1810          PAT18, PAT19, PAT20, PAT21),
: 1811
: 1812          BST_CNT : word initial (0),          ! CURRENT SEQUENTIAL BLOCK COUNT
: 1813          BST_DEV : word initial (0),          ! CURRENT SEQUENTIAL BLOCK DEVICE
: 1814          CURRENT_VECTOR : word,              ! CURRENT DEVICE'S VECTOR ADDRESS
: 1815          BRLEVEL : word,                     ! CURRENT DEVICE'S BR LEVEL
: 1816          DUOFF : word,                         ! STORAGE FOR DUP OFFSET INTO CST TABLE
: 1817          COMPARE_DATA : byte;                 ! FLAGGED CLEARED TO BYPASS HOST COMPARES
: 1818
: 1819 external routine
: 1820          NEX_TRAP : L$ISR novalue,
: 1821          SET_CPAR : novalue,
: 1822          SET_UPAR : novalue,
: 1823          OUT_IODQ,
: 1824          IN_IODQ : novalue,
: 1825          GET_PKT,
: 1826          PUT_PKT : novalue,
: 1827          GET_RETPKT,
: 1828          PUT_RETPKT : novalue,
: 1829          GET_IO_BUFF : novalue,
: 1830          PUT_IO_BUFF : novalue,
: 1831          PUTA_BUFF : novalue,
: 1832          SEND,
: 1833          WAIT : novalue,
: 1834          DROP_CTLR : novalue,
: 1835          DRV_CTLERR : novalue,
: 1836          MODULAS,
: 1837          EMSCHD : novalue,
: 1838          EMS_22 : novalue,
: 1839          EMS_EL : novalue,
: 1840          EMS_CMP : novalue,
: 1841          EMS_10 : novalue,
: 1842          EMS_12 : novalue,
: 1843          EMS_13 : novalue,
: 1844          EMS_14 : novalue,
: 1845          EMS_18 : novalue,
: 1846          EMS_21 : novalue,
: 1847          EMS_30 : novalue;

```

ZRQAM3
V01.2RD/RX EXERCISER
TEST SECTION14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 B11es-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)SEQ 0263
Page 8

```

: 1848 #sbttl 'TEST SECTION'
: 1849
: 1850 !+
: 1851 !   THIS SECTION CONTAINS THE TOP-LEVEL TEST CODE FOR THE RDRX EXERCISER.
: 1852 !   THE EXERCISER CONSISTS OF ONE TEST WHICH IS SUBDIVIDED INTO A NUMBER OF
: 1853 !   SUBTESTS. ALL SUBTESTS ARE DECLARED WITHIN THIS BLOCK.
: 1854 !-
: 1855
: 1856 BGNTST;
: 1857
: 1858 EOP_FLAG = TRUE;
: 1859 COMPARE_DATA = TRUE;
: 1860 DUP_FLAGS = .DUP_FLAGS AND (NOT SWP_DINT);
: 1861 INIT_TEST ();
: 1862
: 1863 if BIT_TST (SWP_FLAGS, SWF_CWC)
: 1864 then SWP_FLAGS = .SWP_FLAGS and (NOT SWF_HWC);
: 1865
: 1866 incr CTLR from 0 to (MAX_CTLR - 1) do
: 1867
: 1868     if (.CST [.CTLR, STATE] eq1 ONLINE) and
: 1869         (.DCT [.CTLR, STAT] eq1 ONLINE) and
: 1870         (.CST [.CTLR, U_CNT] gequ 0)
: 1871     then
: 1872
: 1873         incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + 4) by UNIT_SIZE do
: 1874
: 1875             if .CST [.CTLR, .OFFSET, D_STAT] eq1 ONLINE ! IF AT LEAST ONE UNIT ALIVE
: 1876             then
: 1877                 begin
: 1878                     EOP_FLAG = FALSE;
: 1879                     exitloop;
: 1880                 end;
: 1881
: 1882 if not .EOP_FLAG
: 1883 then
: 1884     MULTI_DRIVE ();
: 1885
: 1886 DELAY (10);
: 1887 ENDTST;

```

```

.TITLE ZRQAM3 RD/RX EXERCISER
.IDENT /V01.2/
.ENABL AMA

```

000000
000000

```

.PSECT $GGG$, RO
COMM.AREA:
.BLKW 24
DPST: .BLKW 2
MAX.LBN: .BLKW 4
STORAGE: .BLKW 4

```

000050
000054
000064

ZRQAM3
V01.2

RD/RX EXERCISER
TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

000074 000000V
 000076
 000100
 000102
 000104
 000106
 000110
 000116 000001
 000120 000000
 000122 000001
 000124 177777
 000126 000001
 000130 105613
 000132 000001
 000134 031463
 000136 000001
 000140 030221
 000142 000020
 000144 000001
 000146 000003
 000150 000007
 000152 000017
 000154 000037
 000156 000077
 000160 000177
 000162 000377
 000164 000777
 000166 001777
 000170 003777
 000172 007777
 000174 017777
 000176 037777
 000200 077777
 000202 177777
 000204 000020
 000206 177776
 000210 177774
 000212 177770
 000214 177760
 000216 177740
 000220 177700
 000222 177600
 000224 177400
 000226 177000
 000230 176000
 000232 174000
 000234 170000
 000236 160000
 000240 140000
 000242 100000
 000244 000000

INT.ADDR:
 .WORD AZINT0
 ICTLR: .BLKW 1
 MX1: .BLKW 1
 MX2: .BLKW 1
 MAD1: .BLKW 1
 MAD2: .BLKW 1
 LAST.PKT:
 .BLKW 3
 PAT02: .WORD 1
 .WORD 0
 PAT03: .WORD 1
 .WORD -1
 PAT04: .WORD 1
 .WORD -72165
 PAT05: .WORD 1
 .WORD 31463
 PAT06: .WORD 1
 .WORD 30221
 PAT07: .WORD 20
 .WORD 1
 .WORD 3
 .WORD 7
 .WORD 17
 .WORD 37
 .WORD 77
 .WORD 177
 .WORD 377
 .WORD 777
 .WORD 1777
 .WORD 3777
 .WORD 7777
 .WORD 17777
 .WORD 37777
 .WORD 77777
 .WORD -1
 PAT08: .WORD 20
 .WORD -2
 .WORD -4
 .WORD -10
 .WORD -20
 .WORD -40
 .WORD -100
 .WORD -200
 .WORD -400
 .WORD -1000
 .WORD -2000
 .WORD -4000
 .WORD -10000
 .WORD -20000
 .WORD -40000
 .WORD -100000
 .WORD 0

ZRQAM3
V01.2

RD/RX EXERCISER
TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 B11ss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

000246	000020	PAT09:	.WORD	20
000250	000000		.WORD	0
000252	000000		.WORD	0
000254	000000		.WORD	0
000256	177777		.WORD	-1
000260	177777		.WORD	-1
000262	177777		.WORD	-1
000264	000000		.WORD	0
000266	000000		.WORD	0
000270	177777		.WORD	-1
000272	177777		.WORD	-1
000274	000000		.WORD	0
000276	177777		.WORD	-1
000300	000000		.WORD	0
000302	177777		.WORD	-1
000304	000000		.WORD	0
000306	177777		.WORD	-1
000310	000001	PAT10:	.WORD	1
000312	133331		.WORD	-44447
000314	000020	PAT11:	.WORD	20
000316	052525		.WORD	52525
000320	052525		.WORD	52525
000322	052525		.WORD	52525
000324	125252		.WORD	-52526
000326	125252		.WORD	-52526
000330	125252		.WORD	-52526
000332	052525		.WORD	52525
000334	052525		.WORD	52525
000336	125252		.WORD	-52526
000340	125252		.WORD	-52526
000342	052525		.WORD	52525
000344	125252		.WORD	-52526
000346	052525		.WORD	52525
000350	125252		.WORD	-52526
000352	052525		.WORD	52525
000354	125252		.WORD	-52526
000356	000024	PAT12:	.WORD	24
000360	026455		.WORD	26455
000362	026455		.WORD	26455
000364	026455		.WORD	26455
000366	151322		.WORD	-26456
000370	151322		.WORD	-26456
000372	151322		.WORD	-26456
000374	026455		.WORD	26455
000376	026455		.WORD	26455
000400	151322		.WORD	-26456
000402	151322		.WORD	-26456
000404	026455		.WORD	26455
000406	026455		.WORD	26455
000410	151322		.WORD	-26456
000412	026455		.WORD	26455
000414	151322		.WORD	-26456
000416	026455		.WORD	26455

H5

ZRQAM3
V01.2

RD/RX EXERCISER
TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 B11es-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

SEQ 0266
Page 11

000420	151322		.WORD	-26456
000422	026455		.WORD	26455
000424	151322		.WORD	-26456
000426	026455		.WORD	26455
000430	000001	PAT13:	.WORD	1
000432	066666		.WORD	66666
000434	000020	PAT14:	.WORD	20
000436	000001		.WORD	1
000440	000002		.WORD	2
000442	000004		.WORD	4
000444	000010		.WORD	10
000446	000020		.WORD	20
000450	000040		.WORD	40
000452	000100		.WORD	100
000454	000200		.WORD	200
000456	000400		.WORD	400
000460	001000		.WORD	1000
000462	002000		.WORD	2000
000464	004000		.WORD	4000
000466	010000		.WORD	10000
000470	020000		.WORD	20000
000472	040000		.WORD	40000
000474	100000		.WORD	-100000
000476	000020	PAT15:	.WORD	20
000500	177776		.WORD	-2
000502	177775		.WORD	-3
000504	177773		.WORD	-5
000506	177767		.WORD	-11
000510	177757		.WORD	-21
000512	177737		.WORD	-41
000514	177677		.WORD	-101
000516	177577		.WORD	-201
000520	177377		.WORD	-401
000522	176777		.WORD	-1001
000524	175777		.WORD	-2001
000526	173777		.WORD	-4001
000530	167777		.WORD	-10001
000532	157777		.WORD	-20001
000534	137777		.WORD	-40001
000536	077777		.WORD	77777
000540	000020	PAT16:	.WORD	20
000542	133331		.WORD	-44447
000544	133331		.WORD	-44447
000546	133331		.WORD	-44447
000550	155554		.WORD	-22224
000552	155554		.WORD	-22224
000554	155554		.WORD	-22224
000556	133331		.WORD	-44447
000560	133331		.WORD	-44447
000562	155554		.WORD	-22224
000564	155554		.WORD	-22224
000566	133331		.WORD	-44447
000570	155554		.WORD	-22224

ZRQAM3
V01.2

RD/RX EXERCISER
TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

000572	133331		.WORD	-44447
000574	155554		.WORD	-22224
000576	133331		.WORD	-44447
000600	155554		.WORD	-22224
000602	000025	PAT17:	.WORD	25
000604	000000		.WORD	0
000606	106466		.WORD	-71312
000610	106466		.WORD	-71312
000612	071311		.WORD	71311
000614	071311		.WORD	71311
000616	071311		.WORD	71311
000620	106466		.WORD	-71312
000622	106466		.WORD	-71312
000624	106466		.WORD	-71312
000626	106466		.WORD	-71312
000630	071311		.WORD	71311
000632	071311		.WORD	71311
000634	071311		.WORD	71311
000636	071311		.WORD	71311
000640	071311		.WORD	71311
000642	106466		.WORD	-71312
000644	106466		.WORD	-71312
000646	106466		.WORD	-71312
000650	106466		.WORD	-71312
000652	106466		.WORD	-71312
000654	106466		.WORD	-71312
000656	000025	PAT18:	.WORD	25
000660	106466		.WORD	-71312
000662	000000		.WORD	0
000664	071311		.WORD	71311
000666	106466		.WORD	-71312
000670	106466		.WORD	-71312
000672	106466		.WORD	-71312
000674	071311		.WORD	71311
000676	071311		.WORD	71311
000700	071311		.WORD	71311
000702	071311		.WORD	71311
000704	106466		.WORD	-71312
000706	106466		.WORD	-71312
000710	106466		.WORD	-71312
000712	106466		.WORD	-71312
000714	106466		.WORD	-71312
000716	071311		.WORD	71311
000720	071311		.WORD	71311
000722	071311		.WORD	71311
000724	071311		.WORD	71311
000726	071311		.WORD	71311
000730	071311		.WORD	71311
000732	000025	PAT19:	.WORD	25
000734	000000		.WORD	0
000736	134631		.WORD	-43147
000740	134631		.WORD	-43147
000742	043146		.WORD	43146

ZRQAM3
V01.2

RD/RX EXERCISER
TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

000744	043146		.WORD	43146
000746	043146		.WORD	43146
000750	134631		.WORD	-43147
000752	134631		.WORD	-43147
000754	134631		.WORD	-43147
000756	134631		.WORD	-43147
000760	043146		.WORD	43146
000762	043146		.WORD	43146
000764	043146		.WORD	43146
000766	043146		.WORD	43146
000770	043146		.WORD	43146
000772	134631		.WORD	-43147
000774	134631		.WORD	-43147
000776	134631		.WORD	-43147
001000	134631		.WORD	-43147
001002	134631		.WORD	-43147
001004	134631		.WORD	-43147
001006	000025	PAT20:	.WORD	25
001010	134631		.WORD	-43147
001012	000000		.WORD	0
001014	043146		.WORD	43146
001016	134631		.WORD	-43147
001020	134631		.WORD	-43147
001022	134631		.WORD	-43147
001024	043146		.WORD	43146
001026	043146		.WORD	43146
001030	043146		.WORD	43146
001032	043146		.WORD	43146
001034	134631		.WORD	-43147
001036	134631		.WORD	-43147
001040	134631		.WORD	-43147
001042	134631		.WORD	-43147
001044	134631		.WORD	-43147
001046	043146		.WORD	43146
001050	043146		.WORD	43146
001052	043146		.WORD	43146
001054	043146		.WORD	43146
001056	043146		.WORD	43146
001060	043146		.WORD	43146
001062	000001	PAT21:	.WORD	1
001064	000000		.WORD	0
001066	000000G	DPA.TBL:	.WORD	RDM.CNT
001070	000116'		.WORD	PAT02
001072	000122'		.WORD	PAT03
001074	000126'		.WORD	PAT04
001076	000132'		.WORD	PAT05
001100	000136'		.WORD	PAT06
001102	000142'		.WORD	PAT07
001104	000204'		.WORD	PAT08
001106	000246'		.WORD	PAT09
001110	000310'		.WORD	PAT10
001112	000314'		.WORD	PAT11
001114	000356'		.WORD	PAT12

ZRQAM3
V01.2RD/RX EXERCISER
TEST SECTION14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)SEQ 0269
Page 14001116 000430'
001120 000434'
001122 000476'
001124 000540'
001126 000602'
001130 000656'
001132 000732'
001134 001006'
001136 001062'
001140 000000
001142 000000
001144001146
001150
001152.WORD PAT13
.WORD PAT14
.WORD PAT15
.WORD PAT16
.WORD PAT17
.WORD PAT18
.WORD PAT19
.WORD PAT20
.WORD PAT21
BST.CNT:.WORD 0
BST.DEV:.WORD 0
CURRENT.VECTOR:
.BLKW 1
BRLEVEL:.BLKW 1
DUOFF:.BLKW 1
COMPARE.DATA:
.BLKB 1.GLOBL CST, CST.ADDR, DCT, DCT.ADDR, RDRX.ADDR
.GLOBL IRDRX.ADDR, ICOM.ADDR, ICST.ADDR
.GLOBL IDCT.ADDR, DUPPKT, BST, TALLY
.GLOBL T.ADDR, C.ERR.TBL, MSCP.PKT, IPKT.ADDR
.GLOBL PKT.USE, RETPKT, RP.USE, RP.INDX
.GLOBL RP.ADDR, RDM.CNT, RANDOM, TRK.SGN
.GLOBL ELOG.PKT, BUFF.ADDR, BUFF.OWN
.GLOBL IODQ, IODQ.IN, IODQ.OUT, ENTRY.REASON
.GLOBL EOP.FLAG, DUP.FLAGS, CCTRL, CDISK
.GLOBL CUOFF, CTRL.CNT, DUR, QIO, FREE.MEM.ADDR
.GLOBL BYTS.PER.QIO, ST.CODE, SB.CODE
.GLOBL STEP, OF.RC, SA.REG, CMD.TIME
.GLOBL NEX, CRN.LOW, CRN.HIGH, P.INDEX
.GLOBL S.DUPPKT, S.PATTERN, CREDIT.BAL
.GLOBL INIT.OCCURED, NXT.PKT.2USE, DBM12
.GLOBL DBM18, DBM19, DBM20, DBM21, DBM23
.GLOBL DBM25, DBM26, DBM29, DBM108, DBM109
.GLOBL DBM110, DBM112, EH.0, EH.1, EH.2
.GLOBL EH.3, EH.4, EH.5, EH.6, EH.7, EH.8
.GLOBL EH.9, EH.10, EH.12, EH.13, MSG.02
.GLOBL MSG.03, EGS.02, EGD.10, EGD.11
.GLOBL EGD.12, EGD.13, EGD.14, EGD.15
.GLOBL EGD.16, EGD.17, EGD.18, EGD.19
.GLOBL EGD.20, EGD.21, EGD.22, EGD.23
.GLOBL EGD.24, EGH.30, EX.GDS, EX.ESP
.GLOBL EX.ELP, EX.RCD, EX.SDD, EX.ABP
.GLOBL EX.WRD, CRLF, SWP.ERROR, SWP.XFER
.GLOBL SWP.FLAGS, DUPROUND, SWP.RAT, SWP.DPAT
.GLOBL SWP.UCNT, SWP.UDPAT, L#LUN, L#UNIT
.GLOBL NEX.TRAP, SET.CPAR, SET.UPAR, OUT.IODQ
.GLOBL IN.IODQ, GET.PKT, PUT.PKT, GET.RETPKT
.GLOBL PUT.RETPKT, GET.IO.BUFF, PUT.IO.BUFF
.GLOBL PUTA.BUFF, SEND, WAIT, DROP.CTLR
.GLOBL DRV.CTLERR, MODULAS, EMSCMD, EMS.22

L5

ZRQAM3
V01.2

RD/RX EXERCISER
TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

.GLOBL EMS.EL, EMS.CMP, EMS.10, EMS.12
.GLOBL EMS.13, EMS.14, EMS.18, EMS.21
.GLOBL EMS.30, L#DLY

100000	BIT15--	-100000
040000	BIT14--	40000
020000	BIT13--	20000
010000	BIT12--	10000
004000	BIT11--	4000
002000	BIT10--	2000
001000	BIT09--	1000
000400	BIT08--	400
000200	BIT07--	200
000100	BIT06--	100
000040	BIT05--	40
000020	BIT04--	20
000010	BIT03--	10
000004	BIT02--	4
000002	BIT01--	2
000001	BIT00--	1
001000	BIT9--	1000
000400	BIT8--	400
000200	BIT7--	200
000100	BIT6--	100
000040	BIT5--	40
000020	BIT4--	20
000010	BIT3--	10
000004	BIT2--	4
000002	BIT1--	2
000001	BIT0--	1
000040	EF.START--	40
000037	EF.RESTART--	37
000036	EF.CONTINUE--	36
000035	EF.NEW--	35
000034	EF.PWR--	34
000340	PRI07--	340
000300	PRI06--	300
000240	PRI05--	240
000200	PRI04--	200
000140	PRI03--	140
000100	PRI02--	100
000040	PRI01--	40
000000	PRI00--	0
000004	EVL--	4
000010	LOT--	10
000020	ADR--	20
000040	IDU--	40
000100	ISR--	100
000200	UAM--	200
000400	BOE--	400
001000	PNT--	1000
002000	PRI--	2000

M5

ZRQAM3
V01.2

RD/RX EXERCISER
TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

SEQ 0271
Page 16

004000
010000
020000
040000
100000

IXE== 4000
IBE== 10000
IER== 20000
LOE== 40000
HOE== -100000

		.SBTTL	\$T1 TEST SECTION		
		.PSECT	\$CODE\$, RO		
000000					
000000	004137	000000G	\$T1: JSR R1,\$SAVE3	:	1847
000004	005746		TST -(SP)	:	
000006	112737	000001 000000G	MOVB #1,EOP.FLAG	:	1858
000014	112737	000001 001152'	MOVB #1,COMPARE.DATA	:	1859
000022	042737	000002 000000G	BIC #2,DUP.FLAGS	:	1860
000030	004737	000000V	JSR PC,INIT.TEST	:	1861
000034	032737	000020 000000G	BIT #20,SWP.FLAGS	:	1863
000042	001403		BEQ 1#	:	
000044	042737	000040 000000G	BIC #40,SWP.FLAGS	:	1864
000052	005003		1#: CLR R3	: CTLR	1866
000054	010346		2#: MOV R3,-(SP)	: CTLR,*	1868
000056	012746	000076	MOV #76,-(SP)		
000062	004737	000000G	JSR PC,BL#MUL		
000066	010001		MOV R0,R1		
000070	022626		CMP (SP)+,(SP)+		
000072	005761	000002G	TST CST+2(R1)		
000076	100040		BPL 6#		
000100	010346		MOV R3,-(SP)	: CTLR,*	1869
000102	012746	000022	MOV #22,-(SP)		
000106	004737	000000G	JSR PC,BL#MUL		
000112	022626		CMP (SP)+,(SP)+		
000114	005760	000000G	TST DCT(R0)		
000120	100027		BPL 6#		
000122	010346		MOV R3,-(SP)	: CTLR,*	1875
000124	012746	000037	MOV #37,-(SP)		
000130	004737	000000G	JSR PC,BL#MUL		
000134	012702	000003	MOV #3,R2	: *,OFFSET	1873
000140	010001		3#: MOV R0,R1	:	1875
000142	060201		ADD R2,R1	: OFFSET,*	
000144	006301		ASL R1		
000146	032761	020000 000000G	BIT #20000,CST(R1)		
000154	001403		BEQ 4#		
000156	105037	000000G	CLRB EOP.FLAG	:	1878
000162	000405		BR 5#	:	1877
000164	062702	000007	4#: ADD #7,R2	: *,OFFSET	1873
000170	020227	000031	CMP R2,#31	: OFFSET,*	
000174	003761		BLE 3#		
000176	022626		5#: CMP (SP)+,(SP)+		
000200	005203		6#: INC R3	: CTLR	1866
000202	000243		.WORD CLV!CLC		
000204	003723		BLE 2#		
000206	132737	000001 000000G	BITB #1,EOP.FLAG	:	1882

N5

ZRQAM3 RD/RX EXERCISER
V01.2 TEST SECTION

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0272
Page 17
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (2)

000214	001002		BNE	7\$			
000216	004737	000000V	JSR	PC,MULTI.DRIVE	:		1884
000222	012701	000012	MOV	#12,R1	:	*,\$\$TMP2	1886
000226	001410		BEQ	11\$			
000230	013700	000000G	MOV	L\$DLY,R0	:	*,\$\$TMP1	
000234	001403		BEQ	10\$			
000236	005016		CLR	(SP)	:	\$\$TMP	
000240	005300		DEC	R0	:	\$\$TMP1	
000242	001375		BNE	9\$			
000244	005301		DEC	R1	:	\$\$TMP2	
000246	000767		BR	8\$			
000250	005726		TST	(SP)+	:		
000252	000207		RTS	PC	:		1847

: Routine Size: 86 words, Routine Base: \$CODE\$ + 0000
: Maximum stack depth per invocation: 8 words

000000	004737	000000'		.SBTTL	T1 TEST SECTION		
000000			T1::				
000004	104466		1\$:	JSR	PC,\$T1	:	1886
000006	006000			TRAP	66		
000010	103773			ROR	R0		
000012	000207			BLO	1\$		
				RTS	PC		

: Routine Size: 6 words, Routine Base: \$CODE\$ + 0254
: Maximum stack depth per invocation: 2 words

```

: 1888 #sbttl 'INITIALIZATION TEST ROUTINES'
: 1889
: 1890 routine INIT_TEST : novalue =
: 1891
: 1892 !!
: 1893 !! THE INITIALIZATION TEST IS DESIGNED TO VERIFY THE EXISTENCE OF THE
: 1894 !! DEVICES AS CONFIGURED BY THE OPERATOR DURING THE HW DIALOG, AND TO
: 1895 !! BRING EACH DEVICE ONLINE IN PREPARATION FOR EITHER THE MULTI-DRIVE TEST
: 1896 !! OR THE DM EXERCISER.
: 1897 !!
: 1898 !! BASICALLY, THE DEVICES ARE BROUGHT ONLINE VIA "DRIVER_INIT", WHICH IS
: 1899 !! INVOKED IMMEDIATELY. ANY DEVICES WHICH FAIL DURING THIS PHASE WILL BE
: 1900 !! MARKED OFFLINE IN THEIR DCT AND CST. FOR THOSE DEVICES WHICH SURVIVE
: 1901 !! THE INITIALIZATION, THIS ROUTINE WILL ATTEMPT 1 OR 2 ACCESS COMMANDS TO
: 1902 !! EACH DISK VIA ROUTINE "ACCESS". THE INITIALIZATION TEST IS DEEMED A
: 1903 !! SUCCESS IF A BLOCK ON THE INNER TRACK OF EACH DISK CAN BE ACCESSED.
: 1904 !!-
: 1905
: 1906 begin
: 1907 DRIVER_INIT (); ! INIT DRIVER DATA AND DEVICES
: 1908
: 1909 incr CTLR from 0 to (MAX_CTLR - 1) do ! FOR EACH CONTROLLER
: 1910 begin
: 1911 SET_CPAR (.CTLR); ! SET UP COMMONLY-USED CONTROLLER-RELATED DATA ITEMS
: 1912
: 1913 if .CST_ADDR [STATE] eq1 ONLINE ! IF CONTROLLER IS STILL ALIVE
: 1914 then
: 1915
: 1916 incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do ! FOR EACH DISK
: 1917
: 1918 if (.CST_ADDR [.OFFSET, D_PRES] eq1 PRESENT) and
: 1919 (.CST_ADDR [.OFFSET, D_STAT] eq1 ONLINE) and
: 1920 (not .CST_ADDR [.OFFSET, D_FATAL])
: 1921 then
: 1922 begin
: 1923 SET_UPAR (.OFFSET); ! SET UP UNIT-RELATED DATA ITEMS
: 1924 IF SWP_DINT NEQ (.DUP_FLAGS AND SWP_DINT) ! IF DUP CAUSED INIT THEN SKIP THIS SECTION
: 1925 THEN ACCESS (); ! TRY ACCESS TO INNER TRACK
: 1926 end; ! IF UNIT IS PRESENT AND ONLINE
: 1927
: 1928 end; ! CONTROLLER LOOP
: 1929
: 1930 end; ! ROUTINE INIT_TEST

```

000000	004137	000000G	.SBTTL	INIT.TEST	INITIALIZATION TEST ROUTINES	
			INIT.TEST:			
000004	004737	000000V	JSR	R1, \$SAVE2		1890
000010	005002		JSR	PC, DRIVER_INIT		1907
000012	010246		CLR	R2	: CTLR	1909
000014	004737	000000G	14: MOV	R2, -(SP)	: CTLR, *	1911
000020	013700	000000G	JSR	PC, SET_CPAR		
			MOV	CST.ADDR, R0		1913

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555		
V01.2	INITIALIZATION TEST ROUTINES	14-Dec-1983 16:12:00	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (3)		
000024	005760	000002	TST	2(R0)	
000030	100035		BPL	4\$	
000032	012701	000003	MOV	#3,R1	; *,OFFSET
000036	010100		MOV	R1,R0	; OFFSET,*
000040	006300		ASL	R0	
000042	063700	000000G	ADD	CST.ADDR,R0	
000046	032710	040000	BIT	#40000,(R0)	
000052	001417		BEQ	3\$	
000054	032710	020000	BIT	#20000,(R0)	
000060	001414		BEQ	3\$; 1919
000062	032710	010000	BIT	#10000,(R0)	
000066	001011		BNE	3\$; 1920
000070	010116		MOV	R1,(SP)	; OFFSET,*
000072	004737	000000G	JSR	PC,SET.UPAR	
000076	032737	000002 000000G	BIT	#2,DUP.FLAGS	
000104	001002		BNE	3\$; 1924
000106	004737	000000V	JSR	PC,ACCESS	
000112	062701	000007	ADD	#7,R1	; *,OFFSET
000116	020127	000030	CMP	R1,#30	; OFFSET,*
000122	003745		BLE	2\$	
000124	005726		TST	(SP)+	; 1910
000126	005202		INC	R2	; CTLR
000130	000243		.WORD	CLV!CLC	; 1909
000132	003727		BLE	1\$	
000134	000207		RTS	PC	; 1890

; Routine Size: 47 words, Routine Base: \$CODE\$ + 0270
; Maximum stack depth per invocation: 5 words

```

: 1931 routine DRIVER_INIT : novalue =
: 1932
: 1933 !!+
: 1934 !! THIS ROUTINE IS EQUIVALENT IN FUNCTION TO THE INITIALIZATION ENTRY
: 1935 !! POINT OF A STANDARD DEVICE DRIVER. ITS RESPONSIBILITY IS TO INITIALIZE
: 1936 !! DRIVER DATA, AND TO BRING EACH RDRX CONTROLLER AND UNIT (DISK)
: 1937 !! ONLINE.
: 1938 !!-
: 1939
: 1940 begin
: 1941
: 1942 local
: 1943     PKT_ADDR;
: 1944
: 1945     PKT_ADDR = MSCP_PKT * 10;           ! ADDR (TEXT * 0) OF FIRST MSCP PACKET
: 1946     NXT_PKT_2USE = 0;                 ! NEXT PACKET TO ALLOCATE
: 1947
: 1948     incr COUNT from 0 to (PKT_CNT - 1) do ! FOR EACH MSCP PACKET
: 1949         begin
: 1950             PKT_USE [.COUNT] = -1;     ! MARK PACKET FREE
: 1951             MSCP_PKT [.COUNT, PKT_LO] = .PKT_ADDR; ! LOAD PKT ADDR INTO BUFFER DESCRIPTOR
: 1952             MSCP_PKT [.COUNT, PKT_HI] = 0;
: 1953             MSCP_PKT [.COUNT, CONNID] = CID_MSCP; ! SET CONNECTION ID TO MSCP ID
: 1954             PKT_ADDR = .PKT_ADDR + (PKT_LEN * 2); ! ADVANCE ADDR TO NEXT PACKET
: 1955         end;
: 1956
: 1957     incr CTRL from 0 to (MAX_CTRL - 1) do ! FOR EACH CONTROLLER
: 1958
: 1959         if .CST [.CTRL, IP_ADDR] neq 0 ! IF CONTROLLER IS PRESENT
: 1960         then
: 1961             begin
: 1962                 SET_CPAR (.CTRL); ! SET UP CURRENT CONTROLLER PARAMETERS
: 1963                 CURRENT_VECTOR = .CST_ADDR [VEC_ADDR]; ! SET CURRENT CONTROLLER'S VECTOR ADDRESS
: 1964                 BRLEVEL = .CST_ADDR [BR_LEV] + 5; ! SET CURRENT CONTROLLER'S BR LEVEL
: 1965                 CTRL_INIT (); ! INIT DEVICE AND CTRL DATA
: 1966
: 1967                 if .DCT_ADDR [STAT] eq 1 ONLINE ! IF CONTROLLER IS STILL ALIVE
: 1968                 then
: 1969
: 1970                     incr OFFSET from (0 * OF_UN) to (3 * UNIT_SIZE * OF_UN) by UNIT_SIZE do ! FOR EACH UNIT (DISK)
: 1971
: 1972                         if (.CST_ADDR [.OFFSET, D_PRES] eq 1 PRESENT) and ! IF UNIT EXISTS
: 1973                             (not .CST_ADDR [.OFFSET, D_FATAL])
: 1974                         then
: 1975                             begin
: 1976                                 SET_UPAR (.OFFSET); ! SET UP UNIT-RELATED DATA ITEMS
: 1977                                 UNIT_INIT (); ! BRING UNIT ONLINE
: 1978                                 end; ! IF UNIT EXISTS
: 1979
: 1980                     end; ! IF CONTROLLER IS PRESENT
: 1981
: 1982 end; ! ROUTINE DRIVER_INIT

```

Address	Label	OpCode	Comment	Instruction	Register/Value	Symbol	Line No.
000000	004137	000000G		.SBTTL DRIVER.INIT			
000004	012702	000012G		JSR	R1,#SAVE2		1931
000010	105037	000000G		MOV	#MSCP.PKT+12,R2	; *.PKT.ADDR	1945
000014	005001			CLRB	NXT.PKT.2USE		1946
000016	112761	000377 000000G	1#:	CLR	R1	; COUNT	1948
000024	010146			MOVB	#377,PKT.USE(R1)	; *,*(COUNT)	1950
000026	012746	000104		MOV	R1,-(SP)	; COUNT,*	1951
000032	004737	000000G		MOV	#104,-(SP)		
000036	010260	000000G		JSR	PC,BL#MUL		
000042	005060	000002G		MOV	R2,MSCP.PKT(R0)	; PKT.ADDR,*	
000046	105060	000011G		CLR	MSCP.PKT+2(R0)		1952
000052	062702	000104		CLRB	MSCP.PKT+11(R0)		1953
000056	022626			ADD	#104,R2	; *.PKT.ADDR	1954
000060	005201			CMP	(SP)+,(SP)+		1949
000062	020127	000013		INC	R1	; COUNT	1948
000066	003753			CMP	R1,#13	; COUNT,*	
000070	005002			BLE	1#		
000072	010246		2#:	CLR	R2	; CTLR	1957
000074	012746	000076		MOV	R2,-(SP)	; CTLR,*	1959
000100	004737	000000G		MOV	#76,-(SP)		
000104	022626			JSR	PC,BL#MUL		
000106	005760	000000G		CMP	(SP)+,(SP)+		
000112	001460			TST	CST(R0)		
000114	010246			BEQ	6#		
000116	004737	000000G		MOV	R2,-(SP)	; CTLR,*	1962
000122	013700	000000G		JSR	PC,SET.CPAR		
000126	016037	000002 001144'		MOV	CST.ADDR,R0		1963
000134	042737	177000 001144'		MOV	2(R0),CURRENT.VECTOR		
000142	005016			BIC	#177000,CURRENT.VECTOR		
000144	116016	000004		CLR	(SP)		1964
000150	012746	000005		MOVB	4(R0),(SP)		
000154	004737	000000G		MOV	#5,-(SP)		
000160	010037	001146'		JSR	PC,BL#SHF		
000164	004737	000000V		MOV	R0,BRLEVEL		
000170	005777	000000G		JSR	PC,CTLR.INIT		1965
000174	100026			TST	#DCT.ADDR		1967
000176	012701	000003		BPL	5#		
000202	010100		3#:	MOV	#3,R1	; *.OFFSET	1970
000204	006300			MOV	R1,R0	; OFFSET,*	1972
000206	063700	000000G		ASL	R0		
000212	032710	040000		ADD	CST.ADDR,R0		
000216	001410			BIT	#40000,(R0)		
000220	032710	010000		BEQ	4#		
000224	001005			BIT	#10000,(R0)		1973
000226	010116			BNE	4#		
000230	004737	000000G		MOV	R1,(SP)	; OFFSET,*	1976
000234	004737	000000V		JSR	PC,SET.UPAR		
000240	062701	000007	4#:	JSR	PC,UNIT.INIT		1977
000244	020127	000030		ADD	#7,R1	; *.OFFSET	1970
000250	003754			CMP	R1,#30	; OFFSET,*	
				BLE	3#		

F6

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 B11ss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (4)

SEQ 0277
Page 22

000252	022626	5#:	CMP	(SP)+,(SP)+	:	1961
000254	005202	6#:	INC	R2	: CTLR	1957
000256	000243		.WORD	CLV!CLC		
000260	003704		BLE	2#		
000262	000207		RTS	PC	:	1931

: Routine Size: 90 words, Routine Base: \$CODE\$ + 0426
: Maximum stack depth per invocation: 6 words


```

:      2036      WRT_RDRX (RCSA, RC_ALL, SA_GO);      ! SET "GO" BIT (START CTLR POLLING)
:      2037
:      2038      if SET_CTLR_CHAR ( ) eq1 SUCCESS      ! SET CONTROLLER CHARACTERISTICS
:      2039      then
:      2040      begin
:      2041      DCT_ADDR [STAT] = ONLINE;      ! MARK CONTROLLER ONLINE IN "DRIVER"
:      2042      CST_ADDR [STATE] = ONLINE;      ! MARK CONTROLLER ONLINE IN "PROGRAM"
:      2043      end;
:      2044      end
:      2045
:      2046      else      ! HARD INIT FAILED
:      2047      begin
:      2048      DROP_CTLR (.CCTLR, DU_INIT);      ! DROP ALL CONTROLLER'S UNITS
:      2049      end;
:      2050
:      2051      end;      ! ROUTINE CTLR_INIT

```

		.SBTTL CTLR.INIT INITIALIZATION TEST ROUTINES		
000000	010146		CTLR.INIT:	
			MOV R1, -(SP)	:
000002	004737	000000V	JSR PC,INI.CTLR.DAT	:
000006	012746	000340	MOV #340, -(SP)	:
000012	013700	000000G	MOV CCTLR, R0	
000016	006300		ASL R0	
000020	016046	000074'	MOV INT.ADDR(R0), -(SP)	
000024	013746	001144'	MOV CURRENT.VECTOR, -(SP)	
000030	012746	000003	MOV #3, -(SP)	
000034	104437		TRAP 37	
000036	052777	040000 000000G	BIS #40000, SDCT.ADDR	:
000044	013700	000000G	MOV CST.ADDR, R0	:
000050	016001	000006	MOV 6(R0), R1	
000054	000301		SWAB R1	
000056	042701	177760	BIC #177760, R1	
000062	010137	000000G	MOV R1, L#LUN	
000066	032737	000002 000000G	BIT #2, DUP.FLAGS	:
000074	001025		BNE 2#	:
000076	004737	000000V	JSR PC, REG.EXIST	:
000102	005700		TST R0	
000104	001410		BEQ 1#	:
000106	032737	000002 000000G	BIT #2, DUP.FLAGS	:
000114	001015		BNE 2#	:
000116	004737	000000V	JSR PC, VEC.BR.TEST	:
000122	005700		TST R0	
000124	001011		BNE 2#	
000126	013716	000000G	MOV CCTLR, (SP)	:
000132	012746	000002	MOV #2, -(SP)	:
000136	004737	000000G	JSR PC, DROP.CTLR	
000142	062706	000012	ADD #12, SP	:
000146	000450		BR 5#	:
000150	004737	000000V	JSR PC, HARD.INIT	:
000154	110001		MOV R0, R1	:
000156	042777	040000 000000G	BIC #40000, SDCT.ADDR	:
				*,RESULT
				2030

1983
2006
2007

2008
2009

2011
2013
2016
2020
2022
2025
2022
2024
2029
2030

I6

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (5)

SEQ 0280
Page 25

000164	120127	000001		CMPB	R1,#1	; RESULT,*	2032
000170	001026			BNE	3#		
000172	004737	000000V		JSR	PC,INI.RRING		2035
000176	012701	000001		MOV	#1,R1	; *,RC.REG	2036
000202	013700	000000G		MOV	RDRX.ADDR,R0		
000206	010160	000002		MOV	R1,2(R0)	; RC.REG,*	
000212	004737	000000V		JSR	PC,SET.CTLR.CHAR		2038
000216	020027	000001		CMP	R0,#1		
000222	001020			BNE	4#		
000224	052777	100000	000000G	BIS	#100000,#DCT.ADDR		2041
000232	013700	000000G		MOV	CST.ADDR,R0		2042
000236	052760	100000	000002	BIS	#100000,2(R0)		
000244	000407			BR	4#		2032
000246	013716	000000G	3#:	MOV	CCTLR,(SP)		2048
000252	012746	000002		MOV	#2,-(SP)		
000256	004737	000000G		JSR	PC,DROP.CTLR		
000262	005726			TST	(SP)+		2047
000264	062706	000010	4#:	ADD	#10,SP		2001
000270	012601		5#:	MOV	(SP)+,R1		1983
000272	000207			RTS	PC		

; Routine Size: 94 words, Routine Base: \$CODE\$ + 0712
; Maximum stack depth per invocation: 7 words

```

: 2052 routine INI_CTRL_DAT : novalue =
: 2053
: 2054 !+
: 2055 ! THIS ROUTINE IS RESPONSIBLE FOR INITIALIZING ALL CONTROLLER-RELATED
: 2056 ! DATA IN THE "DRIVER" PORTION OF THE EXERCISER. THIS INCLUDES THE
: 2057 ! CONTROLLER'S DCT AND OUTSTANDING COMMAND LIST.
: 2058 !
: 2059 ! IMPLICIT INPUTS:
: 2060 !     CCTLR - CURRENT CONTROLLER NUMBER
: 2061 !     DCT_ADDR - ADDRESS OF CURENT CONTROLLER'S DCT
: 2062 !-
: 2063
: 2064 begin
: 2065 DCT_ADDR [WORD0] = 0;
: 2066 DCT_ADDR [RR_BEG] = COMM_AREA + 8 + (.CCTLR * COMM_LEN * 2);
: 2067 DCT_ADDR [RR_END] = .DCT_ADDR [RR_BEG] + ((RRING_LEN - 1) * 4);
: 2068 DCT_ADDR [CR_BEG] = .DCT_ADDR [RR_END] + 4;
: 2069 DCT_ADDR [CR_END] = .DCT_ADDR [CR_BEG] + ((CRING_LEN - 1) * 4);
: 2070 DCT_ADDR [RR_POLL] = .DCT_ADDR [RR_BEG];
: 2071 DCT_ADDR [CR_POLL] = DCT_ADDR [CR_NEXT] = .DCT_ADDR [CR_BEG];
: 2072 end;

```

000000	004137	000000G	.SBTTL INI.CTRL.DAT INITIALIZATION TEST ROUTINES	
			INI.CTRL.DAT:	
000004	013701	000000G	JSR R1,\$SAVE2	2052
000010	005011		MOV DCT_ADDR,R1	2065
000012	012702	000004	CLR (R1)	
000016	060102		MOV #4,R2	2066
000020	013746	000000G	ADD R1,R2	
000024	012746	000050	MOV CCTLR,-(SP)	
000030	004737	000000G	MOV #50,-(SP)	
000034	062700	000010'	JSR PC,BL\$MUL	
000040	010012		ADD #COMM_AREA+10,R0	
000042	010061	000006	MOV R0,(R2)	
000046	062761	000014 000006	MOV R0,6(R1)	2067
000054	012700	000010	ADD #14,6(R1)	
000060	060100		MOV #10,R0	2068
000062	016110	000006	ADD R1,R0	
000066	062710	000004	MOV 6(R1),(R0)	
000072	011061	000012	ADD #4,(R0)	
000076	062761	000014 000012	MOV (R0),12(R1)	2069
000104	011261	000014	ADD #14,12(R1)	
000110	011061	000020	MOV (R2),14(R1)	2070
000114	011061	000016	MOV (R0),20(R1)	2071
000120	022626		MOV (R0),16(R1)	
000122	000207		CMP (SP)+,(SP)+	2064
			RTS PC	2052

; Routine Size: 42 words, Routine Base: \$CODE\$ + 1206
; Maximum stack depth per invocation: 6 words

```

: 2073 routine REG_EXIST =
: 2074
: 2075 !+
: 2076 ! THIS IS THE REGISTER EXISTENCE (OR "PROBE") TEST DESIGNED TO VERIFY
: 2077 ! THE PRESENCE OF AN RDRX DEVICE. THIS OBJECTIVE IS ACCOMPLISHED BY
: 2078 ! SETTING UP THE NON-EXISTENT MEMORY (NEX) TRAP VECTOR (LOCATION 4) AND
: 2079 ! ATTEMPTING TO READ WHAT IS ASSUMED TO BE THE DEVICE'S SA AND IP
: 2080 ! REGISTERS. IF THE NEX TRAP HANDLER IS INVOKED DUE TO AN ABSENT DEVICE,
: 2081 ! THEN THE GLOBAL DATUM "NEX" WILL BE SET TO "TRUE". THIS DATUM
: 2082 ! DETERMINES THE SUCCESS / FAILURE VALUE OF THIS ROUTINE.
: 2083 !-
: 2084
: 2085 begin
: 2086
: 2087 local
: 2088     TEMP : word,           ! TEMP FOR READING SA AND IP
: 2089     DUMMY : word;        ! AS THE NAME IMPLIES
: 2090
: 2091 if .ENTRY_REASON eq1 NEW_PASS
: 2092 then
: 2093     return SUCCESS;      ! SKIP TEST FOR NEXT PASS
: 2094
: 2095 OF_RC = 2;             ! SET UP TO READ SA FIRST
: 2096
: 2097 do
: 2098     begin
: 2099     NEX = FALSE;        ! SET TO "TRAP NOT RECEIVED"
: 2100     SETVEC (4, NEX_TRAP, PRI07); ! SET LOCATION 4 TRAP VECTOR ADDRESS
: 2101     TEMP = .(.RDRX_ADDR + .OF_RC); ! READ REGISTER (THEN TRAP OR CONTINUE)
: 2102     DUMMY = 0;        ! DUMMY INSTRUCTION TO COVER TRAP RETURN BUG
: 2103                     ! (TRAP RETURNS TO NEXT INSTRUCTION)
: 2104     CLRVEC (4);       ! CLEAR LOCATION 4 TRAP VECTOR ADDRESS
: 2105
: 2106     if .NEX           ! IF NEX TRAP OCCURRED
: 2107     then
: 2108     begin
: 2109     C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 2110     ERRDF (10, EGD_10, EMS_10); ! REGISTER EXISTENCE TEST FAILED
: 2111     SETPRI (PRI00);          ! LOWER PRIORITY
: 2112     return FAILURE;
: 2113     end
: 2114     else
: 2115     OF_RC = .OF_RC - 2;      ! SET UP FOR IP REG OR QUIT
: 2116
: 2117     end
: 2118 until .OF_RC lss 0;
: 2119
: 2120 return SUCCESS;
: 2121 end;

```

000000 004137 000000G

```

          SBTTL REG.EXIST INITIALIZATION TEST ROUTINES
REG.EXIST:

```

L6

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (7)

SEQ 0283
Page 28

000004	123727	000000G	000005		JSR	R1,\$SAVE2	:		2073
000012	001461				CMPB	ENTRY.REASON,#5	:		2091
000014	012737	000002	000000G		BEQ	3#	:		2093
000022	005037	000000G			MOV	#2,OF.RC	:		2095
000026	012746	000340		1#:	CLR	NEX	:		2099
000032	012746	000000G			MOV	#340,-(SP)	:		2100
000036	012746	000000G			MOV	#NEX.TRAP,-(SP)	:		
000042	012746	000004			MOV	#4,-(SP)	:		
000046	104437	000003			MOV	#3,-(SP)	:		
000050	013700	000000G			TRAP	37	:		
000054	063700	000000G			MOV	RDRX.ADDR,RO	:		2101
000060	011001				ADD	OF.RC,RO	:		
000062	005002				MOV	(RO),R1	:	*.TEMP	
000064	012700	000004			CLR	R2	:	DUMMY	2102
000070	104436				MOV	#4,RO	:		2104
000072	032737	000001	000000G		TRAP	36	:		
000100	001416				BIT	#1,NEX	:		2106
000102	013700	000000G			BEQ	2#	:		
000106	006300				MOV	CCTLR,RO	:		2109
000110	105260	000000G			ASL	RO	:		
000114	104455				INCB	C.ERR.TBL(RO)	:		
000116	000012				TRAP	55	:		2110
000120	000000G				.WORD	12	:		
000122	000000G				.WORD	EGD.10	:		
000124	005000				.WORD	EMS.10	:		
000126	104441				CLR	RO	:		2111
000130	062706	000010			TRAP	41	:		
000134	000413				ADD	#10,SP	:		2106
000136	162737	000002	000000G		BR	4#	:		2108
000144	062706	000010		2#:	SUB	#2,OF.RC	:		2115
000150	005737	000000G			ADD	#10,SP	:		2098
000154	002322				TST	OF.RC	:		2118
000156	012700	000001		3#:	BGE	1#	:		
000162	000207				MOV	#1,RO	:		2085
000164	005000			4#:	RTS	PC	:		
000166	000207				CLR	RO	:		2073
					RTS	PC	:		

; Routine Size: 60 words, Routine Base: \$CODE\$ + 1332
; Maximum stack depth per invocation: 9 words

```

: 2122 routine VEC_BR_TEST =
: 2123
: 2124 !!
: 2125 !! THIS ROUTINE ATTEMPTS TO VERIFY (A) THAT THE RDRX VECTOR ADDRESS GIVEN
: 2126 !! BY THE USER DURING THE HW DIALOG IS VALID, AND (B) THAT THE
: 2127 !! USER-SPECIFIED BUS REQUEST LEVEL FOR THE DEVICE IS CORRECT. THE FIRST
: 2128 !! OBJECTIVE IS ACCOMPLISHED BY SETTING THE CPU PRIORITY TO 0 AND FORCING
: 2129 !! AN RDRX INTERRUPT. IF THE USER SPECIFIED AND INCORRECT VECTOR ADDRESS,
: 2130 !! THEN THE RESULT MAY BE UNPREDICTABLE. FOR THIS REASON, THE MESSAGE
: 2131 !! "FUNCTIONAL TEST STARTED" IS PRINTED BEFORE THE TEST, AND
: 2132 !! "EXERCISER STARTED" IS PRINTED AT ITS SUCCESSFUL CONCLUSION. IF
: 2133 !! EITHER "FUNCTIONAL TEST ..." OR "EXERCISER ..." DOES NOT APPEAR, THEN
: 2134 !! PROGRAM CONTROL IS ASSUMED LOST AND A FATAL TRAP IS LIKELY TO OCCUR. AT
: 2135 !! THIS POINT, THE EXERCISER MUST BE STARTED AGAIN.
: 2136 !!
: 2137 !! IF THIS TEST SUCCEEDS, THEN THE BR LEVEL TEST IS RUN BY SETTING THE
: 2138 !! PROCESSOR PRIORITY TO THE ASSUMED INTERRUPT PRIORITY GIVEN BY THE
: 2139 !! USER. A FORCED INTERRUPT SHOULD NOT OCCUR. THEN, BY LOWERING THE
: 2140 !! PRIORITY BY ONE, THE DELAYED INTERRUPT SHOULD OCCUR.
: 2141 !!-
: 2142
: 2143 begin
: 2144
: 2145 if .ENTRY_REASON eq1 NEW_PASS
: 2146 then
: 2147     begin
: 2148         SETPRI (PRI00);           ! LOWER PRIORITY
: 2149         return SUCCESS;         ! SKIP TEST IF NEXT PASS
: 2150     end;
: 2151
: 2152 PRINTF (MSG_02);               ! "FUNCTIONAL TEST STARTED"
: 2153
: 2154 if INT_GEN () eq1 FALSE       ! FORCE AN INTERRUPT
: 2155 then
: 2156     begin                       ! IF INTERRUPT DID NOT OCCUR
: 2157         C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 2158         ERRDF (11, EGD_11, 0);  ! VECTOR TEST FAILED
: 2159         return FAILURE;
: 2160     end
: 2161 else
: 2162     begin                       ! INTERRUPT DID OCCUR
: 2163         PRINTF (MSG_03);        ! "EXERCISER STARTED"
: 2164         SETPRI (.BRLEVEL);     ! SET PRIORITY TO ASSUMED BR LEVEL
: 2165
: 2166         if INT_GEN () eq1 FALSE ! FORCE AN INTERRUPT (SHOULD NOT OCCUR)
: 2167         then
: 2168             begin               ! IF INTERRUPT DID NOT OCCUR
: 2169                 SETPRI (.BRLEVEL - #0'40'); ! LOWER PRIORITY BY 1
: 2170                 DELAY (1);      ! WAIT
: 2171
: 2172                 if .DCT_ADDR [SA_SAVE] neq 0 ! IF INTERRUPT DID OCCUR (SA_SAVE WOULD BE NON-ZERO)
: 2173                 then
: 2174                     begin

```

```

:      2175          SETPRI (PRI00);          ! RESTORE PROCESSOR PRIORITY TO 0
:      2176          return SUCCESS;         ! ONLY SUCCESSFUL EXIT POINT
:      2177          end;
:      2178
:      2179          end;
:      2180
:      2181          end;
:      2182
:      2183          SETPRI (PRI00);          ! COME HERE ONLY FOR BR TEST FAILURE
:      2184          C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
:      2185          ERRDF (12, EGD_12, EMS_12);
:      2186          return FAILURE;
:      2187          end;
    
```

```

000000 010146          .SBTTL VEC.BR.TEST INITIALIZATION TEST ROUTINES
VEC.BR.TEST:
000002 005746          MOV      R1, -(SP)          ;          2122
000004 123727 000000G 000005          TST      -(SP)          ;
000012 001003          CMPB    ENTRY.REASON, #5          ;          2145
000014 005000          BNE     1#          ;
000016 104441          CLR     R0          ;          2148
000020 000473          TRAP   41          ;
000022 012746 000000G          BR      7#          ;          2147
000026 012746 000001          1#:    MOV     #MSG.02, -(SP)          ;          2152
000032 010600          MOV     #1, -(SP)          ;
000034 104417          MOV     SP, R0          ; SP,*
000036 004737 000000V          TRAP   17          ;
000042 005700          JSR    PC, INT.GEN          ;          2154
000044 001012          TST     R0          ;
000046 013700 000000G          BNE     2#          ;
000052 006300          MOV     CCTLR, R0          ;          2157
000054 105260 000000G          ASL     R0          ;
000060 104455          INCB   C.ERR.TBL(R0)          ;
000062 000013          TRAP   55          ;          2158
000064 000000G          .WORD  13          ;
000066 000000          .WORD  EGD.11          ;
000070 000466          .WORD  0          ;
000072 012716 000000G          BR      9#          ;          2154
000076 012746 000001          2#:    MOV     #MSG.03, (SP)          ;          2163
000102 010600          MOV     #1, -(SP)          ;
000104 104417          MOV     SP, R0          ; SP,*
000106 013700 001146'          TRAP   17          ;
000112 104441          MOV     BRLEVEL, R0          ;          2164
000114 004737 000000V          TRAP   41          ;
000120 005700          JSR    PC, INT.GEN          ;          2166
000122 001035          TST     R0          ;
000124 013700 001146'          BNE     8#          ;
000130 162700 000040          MOV     BRLEVEL, R0          ;          2169
000134 104441          SUB     #40, R0          ;
000136 012701 000001          TRAP   41          ;
000142 001411          3#:    MOV     #1, R1          ; *, ##TMP2          2170
          BEQ     6#          ;
    
```


ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 B11-16 V3-555	SEQ 0286
V01.2	INITIALIZATION TEST ROUTINES	14-Dec-1983 16:12:00	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (8)	Page 31

000144	013700	000000G		MOV	L\$DLY,RO			
000150	001404			BEQ	5\$:	*	\$\$TMP1
000152	005066	000006	4\$:	CLR	6(SP)	:		\$\$TMP
000156	005300			DEC	RO	:		\$\$TMP1
000160	001374			BNE	4\$			
000162	005301		5\$:	DEC	R1	:		\$\$TMP2
000164	000766			BR	3\$			
000166	013700	000000G	6\$:	MOV	DCT.ADDR,RO	:		2172
000172	005760	000002		TST	2(RO)			
000176	001407			BEQ	8\$			
000200	005000			CLR	RO	:		2175
000202	104441			TRAP	41			
000204	062706	000006		ADD	#6,SP	:		2172
000210	012700	000001	7\$:	MOV	#1,RO	:		2174
000214	000416			BR	10\$			
000216	005726		8\$:	TST	(SP)+	:		2162
000220	005000			CLR	RO	:		2183
000222	104441			TRAP	41			
000224	013700	000000G		MOV	CCTI.R,RO	:		2184
000230	006300			ASL	RO			
000232	105260	000000G		INCB	C.ERR.TBL(RO)			
000236	104455			TRAP	55	:		2185
000240	000014			.WORD	14			
000242	000000G			.WORD	EGD.12			
000244	000000G			.WORD	EMS.12			
000246	022626		9\$:	CMP	(SP)+,(SP)+	:		2122
000250	005000			CLR	RO			
000252	005726		10\$:	TST	(SP)+			
000254	012601			MOV	(SP)+,R1			
000256	000207			RTS	PC			

: Routine Size: 88 words, Routine Base: \$CODE\$ + 1522
: Maximum stack depth per invocation: 7 words

```

: 2188 routine INT_GEN =
: 2189
: 2190 !!
: 2191 !! THIS ROUTINE BEGINS AN RDRX INITIALIZATION SEQUENCE, BUT ONLY
: 2192 !! COMPLETES THROUGH THE STEP 1 WRITE. ITS PURPOSE IS TO CREATE AN RDRX
: 2193 !! INTERRUPT (AT THE COMPLETEION OF STEP 1) IN ORDER TO HELP VERIFY THE
: 2194 !! THE USER-SPECIFIED VECTOR ADDRESS AND BUS REQUEST INTERRUPT LEVEL.
: 2195 !! A VALUE OF "TRUE" IS RETURNED TO THE CALLER IF AN INTERRUPT OCCURS,
: 2196 !! AND "FALSE" OTHERWISE. THE INTERRUPT IS VERIFIED BY A NON-ZERO VALUE
: 2197 !! IN THE "SA SAVE" WORD IN THE DEVICE'S DCT.
: 2198 !!-
: 2199
: 2200 begin
: 2201
: 2202 local
: 2203     SA : word;                ! STORAGE FOR STEP 1 READ AND WRITE
: 2204
: 2205     DCT_ADDR [SA_SAVE] = 0;    ! ZERO OUT SA SAVE WORD IN DCT
: 2206     WRT_RDRX (RCIP, RC_ALL, ALL_ONES); ! WRITE IP TO START INIT SEQUENCE
: 2207     DELAY (10);                ! WAIT
: 2208     SA = .RDRX_ADDR [RCSA, RC_ALL]; ! STEP 1 READ
: 2209     SA = (WR_RING + 8) or (.CURRENT_VECTOR + -2) or SA_INT; ! STEP 1 WRITE VALUE
: 2210     WRT_RDRX (RCSA, RC_ALL, .SA); ! STEP 1 WRITE
: 2211
: 2212     incr COUNT from 1 to 1600 do
: 2213         begin
: 2214             DELAY (5);          ! TOTAL DELAY COUNT OF 8,000
: 2215             BREAK;
: 2216
: 2217             if .DCT_ADDR [SA_SAVE] neq 0 ! IF SA WAS CHANGED
: 2218                 then
: 2219                     return TRUE; ! INTERRUPT OCCURED
: 2220             end;
: 2221
: 2222     return FALSE; ! IF INTERRUPT DID NOT OCCUR
: 2223 end;

```

Address	Offset	Label	Instruction	Comment	Line No.
000000	004137	000000G	INT.GEN: JSR	R1, \$SAVE2	2188
000004	024646		CMP	-(SP), -(SP)	
000006	013700	000000G	MOV	DCT_ADDR, R0	2205
000012	005060	000002	CLR	2(R0)	
000016	012700	177777	MOV	#-1, R0	2206
000022	010077	000000G	MOV	R0, \$RDRX_ADDR	
000026	012701	000012	MOV	#12, R1	2207
000032	001411		10: BEQ	40	
000034	013700	000000G	MOV	L\$DLY, R0	
000040	001404		BEQ	30	
000042	005066	000002	20: CLR	2(SP)	
000046	005300		DEC	R0	
000050	001374		BNE	20	
000052	005301		30: DEC	R1	

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.8L2;161 (9)

000054	000766			BR	1\$		
000056	013700	000000G	4\$:	MOV	RDRX.ADDR,R0	:	2208
000062	016016	000002		MOV	2(R0),(SP)	: *,RC.REG	
000066	013701	001144'		MOV	CURRENT.VECTOR,R1	:	2209
000072	006201			ASR	R1		
000074	006201			ASR	R1		
000076	010102			MOV	R1,R2	: *,SA	
000100	052702	111200		BIS	#111200,R2	: *,SA	
000104	010201			MOV	R2,R1	: SA,RC.REG	2210
000106	010160	000002		MOV	R1,2(R0)	: RC.REG,*	
000112	012702	003100		MOV	#3100,R2	: *,COUNT	2212
000116	012701	000005	5\$:	MOV	#5,R1	: *,\$\$TMP2	2214
000122	001411		6\$:	BEQ	9\$		
000124	013700	000000G		MOV	L\$DLY,R0	: *,\$\$TMP1	
000130	001404			BEQ	8\$		
000132	005066	000002	7\$:	CLR	2(SP)	: \$\$TMP	
000136	005300			DEC	R0	: \$\$TMP1	
000140	001374			BNE	7\$		
000142	005301		8\$:	DEC	R1	: \$\$TMP2	
000144	000766			BR	6\$		
000146	104422		9\$:	TRAP	22		
000150	013700	000000G		MOV	DCT.ADDR,R0	:	2217
000154	005760	000002		TST	2(R0)		
000160	001403			BEQ	10\$		
000162	012700	000001		MOV	#1,R0	:	2219
000166	000403			BR	11\$		
000170	005302		10\$:	DEC	R2	: COUNT	2212
000172	001351			BNE	5\$		
000174	005000			CLR	R0	:	2200
000176	022626		11\$:	CMP	(SP)+,(SP)+	:	2188
000200	000207			RTS	PC		

: Routine Size: 65 words, Routine Base: \$CODE\$ + 2002
: Maximum stack depth per invocation: 7 words

```

: 2224 routine HARD_INIT =
: 2225
: 2226 !*
: 2227 ! THIS ROUTINE PERFORMS THE FOUR READ / WRITE STEPS REQUIRED TO
: 2228 ! INITIALIZE AN RDRX DEVICE. IF NO READ ERRORS ARE DETECTED IN ANY OF
: 2229 ! THE FOUR STEPS, THEN A SUCCESS VALUE IS RETURNED TO THE CALLER.
: 2230 ! OTHERWISE, ADDITIONAL ATTEMPTS MAY BE MADE TO INITIALIZE THE DEVICE.
: 2231 ! IF ALL ATTEMPTS FAIL, A FAILURE INDICATION IS RETURNED.
: 2232 !-
: 2233
: 2234 begin
: 2235
: 2236 local
: 2237     IE_VEC : word;           ! IE-BIT-AND-VECTOR-ADDRESS/4 BYTE
: 2238                               ! (USED IN STEP 1 WRITE AND STEP 3 READ)
: 2239
: 2240 IE_VEC = .CURRENT_VECTOR + -2; ! GET VECTOR ADDR/4 (IE = 0)
: 2241
: 2242 incr ATTEMPTS from 1 to INI_ATT do
: 2243     begin
: 2244
: 2245     label
: 2246         STEP_1_READ,
: 2247         STEP_2_READ,
: 2248         STEP_3_READ,
: 2249         STEP_4_READ;
: 2250
: 2251     WRT_RDRX (RCIP, RC_ALL, ALL_ONES); ! WRITE IP TO START INIT SEQUENCE
: 2252     !!
: 2253     STEP 1 READ
: 2254     !!
: 2255     STEP = 1;
: 2256     STEP_1_READ:
: 2257         begin
: 2258
: 2259         incr COUNT from 1 to 100 do
: 2260             begin
: 2261                 DELAY (5);           ! TOTAL DELAY COUNT OF 500 FOR STEP 1
: 2262                 BREAK;
: 2263                 SA_REG = .RDRX_ADDR [RCSA, RC_ALL]; ! READ SA
: 2264
: 2265                 if (.SA_REG and S1_MASK) eq1 SA_S1 ! IF STEP 1 READ IS O.K.
: 2266                 then
: 2267                     leave STEP_1_READ;
: 2268
: 2269                 end;
: 2270
: 2271             exitloop;
: 2272             end;
: 2273
: 2274     !!
: 2275     STEP 1 WRITE
: 2276     !!

```

```

: 2277      SA_REG = (WR_RING + 8) or .IE_VEC;          ! STEP 1 WRITE VALUE
: 2278      WRT_RDRX (RCSA, RC_ALL, .SA_REG);          ! STEP 1 WRITE
: 2279      :
: 2280      :
: 2281      :
: 2282      :
: 2283      :
: 2284      :
: 2285      :
: 2286      :
: 2287      :
: 2288      :
: 2289      :
: 2290      :
: 2291      :
: 2292      :
: 2293      :
: 2294      :
: 2295      :
: 2296      :
: 2297      :
: 2298      :
: 2299      :
: 2300      :
: 2301      :
: 2302      :
: 2303      :
: 2304      :
: 2305      :
: 2306      :
: 2307      :
: 2308      :
: 2309      :
: 2310      :
: 2311      :
: 2312      :
: 2313      :
: 2314      :
: 2315      :
: 2316      :
: 2317      :
: 2318      :
: 2319      :
: 2320      :
: 2321      :
: 2322      :
: 2323      :
: 2324      :
: 2325      :
: 2326      :
: 2327      :
: 2328      :
: 2329      :

```

```

      SA_REG = (WR_RING + 8) or .IE_VEC;
      WRT_RDRX (RCSA, RC_ALL, .SA_REG);

STEP 2 READ

STEP = .STEP + 1;
STEP_2_READ:
  begin
    incr COUNT from 1 to 2000 do
      begin
        DELAY (5);
        BREAK;
        SA_REG = .RDRX_ADDR [RCSA, RC_ALL];
        if (.SA_REG and S2_MASK) eq1 (SA_S2 or WR_RING) ! IF STEP 2 READ IS O.K.
        then
          leave STEP_2_READ;
        end;
      end;
    exitloop;
  end;

STEP 2 WRITE
WRT_RDRX (RCSA, RC_ALL, .DCT_ADDR [RR_BEG]); ! RINGBASE-LO, PI = 0

STEP 3 READ

STEP = .STEP + 1;
STEP_3_READ:
  begin
    incr COUNT from 1 to 2000 do
      begin
        DELAY (5);
        BREAK;
        SA_REG = .RDRX_ADDR [RCSA, RC_ALL];
        if (.SA_REG and S3_MASK) eq1 (SA_S3 or .IE_VEC) ! IF STEP 3 READ IS O.K.
        then
          leave STEP_3_READ;
        end;
      end;
    exitloop;
  end;

STEP 3 WRITE

```


000060	001404				BEQ	4#			
000062	005066	000010		3#:	CLR	10(SP)	:	##TMP	
000066	005300				DEC	R0	:	##TMP1	
000070	001374				BNE	3#			
000072	005301			4#:	DEC	R1	:	##TMP2	
000074	000766				BR	2#			
000076	104422			5#:	TRAP	22			
000100	013700	000000G			MOV	RDRX.ADDR,R0	:		2263
000104	016066	000002	000006		MOV	2(R0),6(SP)	:	*,RC.REG	
000112	016637	000006	000000G		MOV	6(SP),SA.REG	:	RC.REG,*	
000120	016600	000006			MOV	6(SP),R0	:	SA.REG,*	2265
000124	042700	001777			BIC	#1777,R0			
000130	020027	004000			CMP	R0,#4000			
000134	001403				BEQ	6#	:		2267
000136	005302				DEC	R2	:	COUNT	2259
000140	001342				BNE	1#			
000142	000576				BR	24#	:		2243
000144	010437	000000G		6#:	MOV	R4,SA.REG	:	IE.VEC,*	2277
000150	052737	111000	000000G		BIS	#111000,SA.REG			
000156	013701	000000G			MOV	SA.REG,R1	:	*,RC.REG	2278
000162	013700	000000G			MOV	RDRX.ADDR,R0			
000166	010160	000002			MOV	R1,2(R0)	:	RC.REG,*	
000172	005237	000000G			INC	STEP	:		2282
000176	012702	003720			MOV	#3720,R2	:	*,COUNT	2286
000202	012701	000005		7#:	MOV	#5,R1	:	*,##TMP2	2288
000206	001411			8#:	BEQ	11#			
000210	013700	000000G			MOV	L#DLY,R0	:	*,##TMP1	
000214	001404				BEQ	10#			
000216	005066	000010		9#:	CLR	10(SP)	:	##TMP	
000222	005300				DEC	R0	:	##TMP1	
000224	001374				BNE	9#			
000226	005301			10#:	DEC	R1	:	##TMP2	
000230	000766				BR	8#			
000232	104422			11#:	TRAP	22			
000234	013700	000000G			MOV	RDRX.ADDR,R0	:		2290
000240	016066	000002	000004		MOV	2(R0),4(SP)	:	*,RC.REG	
000246	016637	000004	000000G		MOV	4(SP),SA.REG	:	RC.REG,*	
000254	016600	000004			MOV	4(SP),R0	:	SA.REG,*	2292
000260	042700	003400			BIC	#3400,R0			
000264	020027	010222			CMP	R0,#10222			
000270	001403				BEQ	12#	:		2294
000272	005302				DEC	R2	:	COUNT	2286
000274	001342				BNE	7#			
000276	000534				BR	26#	:		2243
000300	013700	000000G		12#:	MOV	DCT.ADDR,R0	:		2304
000304	016001	000004			MOV	4(R0),R1	:	*,RC.REG	
000310	013700	000000G			MOV	RDRX.ADDR,R0			
000314	010160	000002			MOV	R1,2(R0)	:	RC.REG,*	
000320	005237	000000G			INC	STEP	:		2308
000324	010403				MOV	R4,R3	:	IE.VEC,*	2318
000326	052703	020000			BIS	#20000,R3			
000332	012702	003720			MOV	#3720,R2	:	*,COUNT	2312
000336	012701	000005		13#:	MOV	#5,R1	:	*,##TMP2	2314

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (10)

000342	001411		14:	BEQ	17:			
000344	013700	000000G		MOV	L#DLY,RO		: *,##TMP1	
000350	001404			BEQ	16:			
000352	005066	000010	15:	CLR	10(SP)		: ##TMP	
000356	005300			DEC	R0		: ##TMP1	
000360	001374			BNE	15:			
000362	005301		16:	DEC	R1		: ##TMP2	
000364	000766			BR	14:			
000366	104422		17:	TRAP	22			
000370	013700	000000G		MOV	RDRX.ADDR,RO			2316
000374	016066	000002 000002		MOV	2(RO),2(SP)		: *,RC.REG	
000402	016637	000002 000000G		MOV	2(SP),SA.REG		: RC.REG,*	
000410	016600	000002		MOV	2(SP),R0		: SA.REG,*	2318
000414	042700	003400		BIC	#3400,R0			
000420	020003			CMP	R0,R3			
000422	001403			BEQ	18:			2320
000424	005302			DEC	R2		: COUNT	2312
000426	001343			BNE	13:			
000430	000457			BR	26:			2243
000432	013700	000000G	18:	MOV	RDRX.ADDR,RO			2330
000436	005060	000002		CLR	2(RO)			
000442	005237	000000G		INC	STEP			2334
000446	012703	003720		MOV	#3720,R3		: *.COUNT	2338
000452	012701	000005	19:	MOV	#5,R1		: *,##TMP2	2340
000456	001411		20:	BEQ	23:			
000460	013700	000000G		MOV	L#DLY,RO		: *,##TMP1	
000464	001404			BEQ	22:			
000466	005066	000010	21:	CLR	10(SP)		: ##TMP	
000472	005300			DEC	R0		: ##TMP1	
000474	001374			BNE	21:			
000476	005301		22:	DEC	R1		: ##TMP2	
000500	000766			BR	20:			
000502	104422		23:	TRAP	22			
000504	013700	000000G		MOV	RDRX.ADDR,RO			2342
000510	016016	000002		MOV	2(RO),(SP)		: *,RC.REG	
000514	011637	000000G		MOV	(SP),SA.REG		: RC.REG,*	
000520	011600			MOV	(SP),R0		: SA.REG,*	2344
000522	042700	003777		BIC	#3777,R0			
000526	020027	040000		CMP	R0,#40000			
000532	001403			BEQ	25:			2346
000534	005303			DEC	R3		: COUNT	2338
000536	001345			BNE	19:			
000540	000413		24:	BR	26:			2243
000542	012737	000001 000000G	25:	MOV	#1,CREDIT.BAL			2355
000550	005001			CLR	R1		: RC.REG	2356
000552	013700	000000G		MOV	RDRX.ADDR,RO			
000556	005060	000002		CLR	2(RO)			
000562	012700	000001		MOV	#1,R0			2243
000566	000414			BR	27:			
000570	005037	000000G	26:	CLR	CREDIT.BAL			2361
000574	013700	000000G		MOV	CCTRL,RO			2362
000600	006300			ASL	R0			
000602	105260	000000G		INCB	C.ERR.TBL(RO)			

J7

ZRQAM3 RD/RX EXERCISER
V01.2 INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0294
Page 39
VAX-11 B111-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (10)

000606	104455			TRAP	55				
000610	000015			.WORD	15				2363
000612	000000G			.WORD	EGD.13				
000614	000000G			.WORD	EMS.13				
000616	005000			CLR	R0				2234
000620	062706	000012	274:	ADD	#12.SP				2224
000624	000207			RTS	PC				

: Routine Size: 203 words, Routine Base: \$CODE\$ + 2204
: Maximum stack depth per invocation: 13 words

```

: 2366 routine INI_RRING : novalue =
: 2367
: 2368 !+
: 2369 ! THIS ROUTINE IS RESPONSIBLE FOR ALLOCATING ENOUGH MSCP PACKETS TO
: 2370 ! FILL AN RDRX RESPONSE RING. THE BUFFER DESCRIPTOR OF EACH PACKET
: 2371 ! (LOCATED IN FRONT OF THE PACKET ITSELF) IS LOADED INTO SUCCESSIVE
: 2372 ! RRING SLOTS. NOTE THAT THE BUFFER DESCRIPTORS HAVE BEEN INITIALIZED
: 2373 ! WITH THE FLAG AND OWNERSHIP BITS SET TO "1", MAKING EACH SLOT
: 2374 ! CONTROLLER-OWNED.
: 2375
: 2376 ! IMPLICIT INPUTS:
: 2377 !     CCTLR - CURRENT CONTROLLER NUMBER
: 2378 !     DCT_ADDR - ADDRESS OF CURRENT CONTROLLER'S DCT
: 2379 !-
: 2380
: 2381 begin
: 2382
: 2383 local
: 2384     index : word,
: 2385     RRING_ADDR;
: 2386
: 2387 RRING_ADDR = .DCT_ADDR [RR_BEG];           ! FIRST RESPONSE RING SLOT
: 2388
: 2389 incr COUNT from 1 to RRING_LEN do
: 2390     begin
: 2391         index = GET_PKT (.CCTLR);           ! GET AN MSCP PACKET
: 2392         .RRING_ADDR = .MSCP_PKT [.index, PKT_LO]; ! LOAD LO-ORDER BUFF DESC INTO SLOT
: 2393         .RRING_ADDR = .RRING_ADDR + 2;     ! ADVANCE TO SECOND WORD
: 2394         .RRING_ADDR = .MSCP_PKT [.index, PKT_HI]; ! LOAD HI-ORDER BUFF DESC INTO SLOT
: 2395         PKT_USE [.index] = .CCTLR;         ! PACKET IN USE
: 2396         .RRING_ADDR = .RRING_ADDR or ED_OWN or ED_FLAG; ! GIVE OWNERSHIP TO CONTRLLER
: 2397         .RRING_ADDR = .RRING_ADDR + 2;     ! ADVANCE TO NEXT SLOT
: 2398     end;
: 2399
: 2400 end;

```

			.SBTTL	INI.RRING INITIALIZATION TEST ROUTINES	
000000	004137	000000G	INI.RRING:		
			JSR	R1, \$SAVE4	2366
000004	013700	000000G	MOV	DCT_ADDR, R0	2387
000010	016004	0000004	MOV	4(R0), R4	*, RRING_ADDR
000014	013703	000000G	MOV	CCTLR, R3	2391
000020	012702	0000004	MOV	#4, R2	*, COUNT
000024	010346		MOV	R3, -(SP)	2389
000026	004737	000000G	JSR	PC, GET.PKT	2391
000032	010001		MOV	R0, R1	*, INDEX
000034	010116		MOV	R1, (SP)	INDEX, *
000036	012746	000104	MOV	#104, -(SP)	2392
000042	004737	000000G	JSR	PC, BL \$MUL	
000046	016024	000000G	MOV	MSCP.PKT(R0), (R4)+	*, RRING_ADDR
000052	016014	000002G	MOV	MSCP.PKT+2(R0), (R4)	*, RRING_ADDR
000056	013703	000000G	MOV	CCTLR, R3	2394
					2395

L7

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2:161 (11)

SEQ 0296
Page 41

000062	110361	000000G	MOVB	R3,PKT.USE(R1)	:	*,*(INDEX)	
000066	052724	140000	BIS	#140000,(R4)+	:	*,RRING.ADDR	2396
000072	022626		CMP	(SP)+,(SP)+	:		2390
000074	005302		DEC	R2	:	COUNT	2389
000076	001352		BNE	1#	:		
000100	000207		RTS	PC	:		2366

: Routine Size: 33 words, Routine Base: \$CODE\$ + 3032
: Maximum stack depth per invocation: 8 words

```

: 2401 routine SET_CTLR_CHAR =
: 2402
: 2403 !+
: 2404 ! THIS ROUTINE IS CALLED BY CTLR_INIT AFTER THE RDRX HAS BEEN HARD-
: 2405 ! INITIALIZED. ITS PURPOSE IS TO FORMAT AND SEND THE "SET CONTROLLER
: 2406 ! CHARACTERISTICS" COMMAND, AND TO VALIDATE THE RESPONSE (END MESSAGE).
: 2407 !
: 2408 ! IMPLICIT INPUTS:
: 2409 !     CCTLR - CURRENT CONTROLLER NUMBER
: 2410 !-
: 2411
: 2412 begin
: 2413
: 2414 !+
: 2415 ! MISCELLANEOUS INITIALIZATON
: 2416 !-
: 2417
: 2418
: 2419 QIO [.CCTLR] = 0;           ! INITIALIZE NO. OF OUTSTANDING QIOS
: 2420 CST [.CCTLR, U_CNT] = 0;   ! CLEAR UNITS IN CST TABLE
: 2421 incr COUNT from 0 to (RP_CNT - 1) do ! INITIALIZE RETURN PACKET POOL
: 2422     RP_USE [.COUNT] = -1;
: 2423
: 2424 IODQ_IN = IODQ_OUT = 0;   ! INITIALIZE I/O DONE QUEUE POINTERS
: 2425
: 2426
: 2427 P_INDEX = GET_PKT (.CCTLR); ! GET AN MSCP PACKET
: 2428 MSCP_PKT [.P_INDEX, MSGLEN] = SZ_SCC; ! PACKET SIZE
: 2429 MSCP_PKT [.P_INDEX, OPCODE] = OP_SCC; ! OPCODE = SET CTLR CHAR
: 2430 MSCP_PKT [.P_INDEX, C_FLAGS] = CF_MASK; ! CONTROLLER FLAGS
: 2431 MSCP_PKT [.P_INDEX, CMD_TYPE] = IMM_CMD; ! IMMEDIATE COMMAND
: 2432
: 2433 if SEND (.P_INDEX) eq 1 FAILURE ! ATTEMPT SEND
: 2434 then
: 2435     begin ! IF SEND WAS UNSUCCESSFUL
: 2436         C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 2437         ERRDF (20, EGD_20, 0); ! FATAL ERROR
: 2438         PUT_PKT (.P_INDEX); ! RETURN PACKET TO POOL
: 2439         DROP_CTLR (.CCTLR, DU_CFATAL); ! DROP CONTROLLER
: 2440         return FAILURE;
: 2441     end
: 2442 else
: 2443     begin ! IF SEND WAS SUCCESSFUL
: 2444
: 2445     do
: 2446         begin
: 2447             WAIT (); ! WAIT FOR RETPKT RESPONSE
: 2448             RP_INDX = OUT_IODQ (); ! GET INDEX OF RETPKT
: 2449             RP_ADDR = RETPKT + (.RP_INDX * RP_LEN * 2); ! CALCULATE RETPKT ADDRESS
: 2450
: 2451             if .RP_ADDR [MESTYP] neq MT_SEQ ! RETURN ALL RETPKTS NOT SENT BY CONTROLLER
: 2452             then
: 2453                 PUT_RETPKT (.RP_INDX);

```

```

: 2454
: 2455
: 2456     end
: 2457     until (.RP_ADDR [CONID] eql CID_DRIVER) or
: 2458         ((.RP_ADDR [MESTYP] eql MT_SEQ) and
: 2459          ((.RP_ADDR [ENDCOD] and OP_END) eql OP_END));
: 2460
: 2461     if .RP_ADDR [CONID] eql CID_DRIVER      ! IF RETPKT IS FROM "DRIVER"
: 2462     then
: 2463         begin
: 2464             PRINTF (DBM23);                ! "ERROR IN SET_CTLR_CHAR"
: 2465             PUT_RETPKT (.RP_INDX);        ! RELEASE RETURN PACKET
: 2466             DR_ERR ();                    ! DROP CONTROLLER
: 2467             return FAILURE;
: 2468         end
: 2469     else
: 2470         begin                                ! ELSE - RETPKT IS FROM DISK MSCP
: 2471             if (.RP_ADDR [ENDCOD] neq (OP_SCC or OP_END)) or      ! IF WRONG ENDCODE
: 2472                 ((.RP_ADDR [C_FLGS] and CF_MASK) neq CF_MASK)    ! OR FLAGS IN ERROR
: 2473             then
: 2474                 begin
: 2475                     C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 2476                     ERRDF (21, EGD_21, EMS_21);                ! FATAL ERROR
: 2477                     DROP_CTLR (.CCTLR, DU_CFATAL);            ! DROP CONTROLLER
: 2478                     PUT_RETPKT (.RP_INDX);                    ! RELEASE RETURN PACKET
: 2479                     return FAILURE;
: 2480                 end
: 2481             else
: 2482                 begin                                ! RETPKT HAS CORRECT ENDCODE
: 2483                     CMD_TIME = .RP_ADDR [C_TIME] + 2;
: 2484
: 2485                     if (.SWP_FLAGS and SWF_TRC) eql SWF_TRC
: 2486                     then
: 2487                         PRINTF (DBM25, .RP_ADDR [C_TIME]);
: 2488
: 2489                     end;                                ! RETPKT HAS CORRECT ENDCODE
: 2490
: 2491                 end;                                ! IF RETPKT WAS SENT BY DISK MSCP
: 2492
: 2493             PUT_RETPKT (.RP_INDX);
: 2494             return SUCCESS;                            ! IF SEND WAS SUCCESSFUL
: 2495         end;
: 2496     end;
: 2497 end;                                                ! ROUTINE SET-CTLR_CHAR

```

000000	010146		.SBTTL	SET.CTLR.CHAR INITIALIZATION TEST ROUTINES	
			SET.CTLR.CHAR:		
000002	013701	000000G	MOV	R1, -(SP)	2401
000006	105061	000000G	MOV	CCTLR, R1	2419
000012	010146		CLRB	QIO(R1)	
000014	012746	000076	MOV	R1, -(SP)	2420
			MOV	#76, -(SP)	

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	SEQ 0299
V01.2	INITIALIZATION TEST ROUTINES	14-Dec-1983 16:12:00	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (12)	Page 44
000020	004737 000000G		JSR PC,BL\$MUL	
000024	105060 000005G		CLRB CST*5(R0)	
000030	005000		CLR R0	: COUNT
000032	112760 000377 000000G	1#:	MOVB #377,RP.USE(R0)	: *,*(COUNT)
000040	005200		INC R0	: COUNT
000042	020027 000003		CMP R0,#3	: COUNT,*
000046	003771		BLE 1#	
000050	005037 000000G		CLR IODQ.OUT	
000054	005037 000000G		CLR IODQ.IN	: 2424
000060	010116		MOV R1,(SP)	
000062	004737 000000G		JSR PC,GET.PKT	: 2427
000066	010037 000000G		MOV RO,P.INDEX	
000072	010016		MOV RO,(SP)	: P.INDEX,*
000074	012746 000104		MOV #104,-(SP)	: 2428
000100	004737 000000G		JSR PC,BL\$MUL	
000104	012760 000040 000006G		MOV #40,MSCP.PKT*6(R0)	
000112	112760 000004 000022G		MOVB #4,MSCP.PKT*22(R0)	: 2429
000120	012760 000120 000030G		MOV #120,MSCP.PKT*30(R0)	: 2430
000126	005060 000004G		CLR MSCP.PKT*4(R0)	: 2431
000132	013716 000000G		MOV P.INDEX,(SP)	: 2433
000136	004737 000000G		JSR PC,SEND	
000142	005700		TST R0	
000144	001026		BNE 2#	
000146	013700 000000G		MOV CCTLR,R0	: 2436
000152	006300		ASL R0	
000154	105260 000000G		INCB C.ERR.TBL(R0)	
000160	104455		TRAP 55	: 2437
000162	000024		.WORD 24	
000164	000000G		.WORD EGD.20	
000166	000000		.WORD 0	
000170	013716 000000G		MOV P.INDEX,(SP)	: 2438
000174	004737 000000G		JSR PC,PUT.PKT	
000200	013716 000000G		MOV CCTLR,(SP)	: 2439
000204	012746 000006		MOV #6,-(SP)	
000210	004737 000000G		JSR PC,DROP.CTLR	
000214	005726		TST (SP)*	: 2435
000216	005000		CLR R0	: 2412
000220	000560		BR 11#	
000222	004737 000000G	2#:	JSR PC,WAIT	: 2447
000226	004737 000000G		JSR PC,OUT.IODQ	: 2448
000232	010037 000000G		MOV RO,RP,INDX	
000236	010016		MOV RO,(SP)	: RP,INDX,*
000240	012746 000060		MOV #60,-(SP)	: 2449
000244	004737 000000G		JSR PC,BL\$MUL	
000250	062700 000000G		ADD #RETPKT,R0	
000254	010037 000000G		MOV RO,RP,ADDR	
000260	132760 000360 000002		BITB #360,2(R0)	: 2451
000266	001404		BEQ 3#	
000270	013716 000000G		MOV RP,INDX,(SP)	: 2453
000274	004737 000000G		JSR PC,PUT.RETPKT	
000300	005726	3#:	TST (SP)*	: 2446
000302	013701 000000G		MOV RP,ADDR,R1	: 2456
000306	005000		CLR R0	

ZRQAM3 V01.2	RD/RX EXERCISER INITIALIZATION TEST ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Bliss-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (12)	SEQ 0300 Page 45
000310	126127 000003 000003	CMPB	3(R1),#3	
000316	001002	BNE	4#	
000320	005200	INC	RO	
000322	000407	BR	5#	
000324	132761 000360 000002	BITB	#360,2(R1)	2457
000332	001333	BNE	2#	
000334	105761 000014	TSTB	14(R1)	2458
000340	100330	BPL	2#	
000342	006000	ROR	RO	2460
000344	103015	BCC	6#	
000346	012716 000000G	MOV	#DBM23,(SP)	2463
000352	012746 000001	MOV	#1,-(SP)	
000356	010600	MOV	SP,RO	: SP,*
000360	104417	TRAP	17	
000362	013716 000000G	MOV	RP,INDX,(SP)	2464
000366	004737 000000G	JSR	PC,PUT.RETPKT	
000372	004737 000000V	JSR	PC,DR.ERR	2465
000376	000436	BR	8#	2460
000400	126127 000014 000204	CMPB	14(R1),#204	2471
000406	001007	BNE	7#	
000410	016100 000022	MOV	22(R1),RO	2472
000414	042700 177357	BIC	#177657,RO	
000420	020027 000120	CMP	RO,#120	
000424	001426	BEQ	9#	
000426	013700 000000G	MOV	CCTRL,RO	2475
000432	006300	ASL	RO	
000434	105260 000000G	INCB	C.ERR.TBL(RO)	
000440	104455	TRAP	55	2476
000442	000025	.WORD	25	
000444	000000G	.WORD	EGD.21	
000446	000000G	.WORD	EMS.21	
000450	013716 000000G	MOV	CCTRL,(SP)	2477
000454	012746 000006	MOV	#6,-(SP)	
000460	004737 000000G	JSR	PC,DROP.CTLR	
000464	013716 000000G	MOV	RP,INDX,(SP)	2478
000470	004737 000000G	JSR	PC,PUT.RETPKT	
000474	062706 000010	ADD	#10,SP	2471
000500	000433	BR	12#	2474
000502	016137 000024 000000G	MOV	24(R1),CMD.TIME	2483
000510	006337 000000G	ASL	CMD.TIME	
000514	032737 000001 000000G	BIT	#1,SMP.FLAGS	2485
000522	001411	BEQ	10#	
000524	016116 000024	MOV	24(R1),(SP)	2487
000530	012746 000000G	MOV	#DBM25,-(SP)	
000534	012746 000002	MOV	#2,-(SP)	
000540	010600	MOV	SP,RO	: SP,*
000542	104417	TRAP	17	
000544	022626	CMP	(SP),,(SP),	
000546	013716 000000G	MOV	RP,INDX,(SP)	2493
000552	004737 000000G	JSR	PC,PUT.RETPKT	
000556	012700 000001	MOV	#1,RO	2412
000562	062706 000006	ADD	#6,SP	2401
000566	000401	BR	13#	2412

D8

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0301
Page 46
VAX-11 B11ss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (12)

000570	005000	12#:	CLR	RO	
000572	012601	13#:	MOV	(SP)+,R1	
000574	000207		RTS	PC	

2401

: Routine Size: 191 words, Routine Base: \$CODE\$ + 3134
: Maximum stack depth per invocation: 8 words


```

: 2498 routine UNIT_INIT : novalue =
: 2499
: 2500 !!
: 2501 !! THIS ROUTINE IS CALLED FROM DRIVER_INIT FOR EACH CONFIGURED UNIT
: 2502 !! (DISK) WHICH IS ATTACHED TO A CONTROLLER THAT SURVIVED
: 2503 !! INITIALIZATION. ITS PURPOSE IS TO FORMAT AND SEND AN "ONLINE"
: 2504 !! MESSAGE, AND TO VERIFY THE RESPONSE.
: 2505 !!
: 2506 !! IMPLICIT INPUTS:
: 2507 !!     CCTLR - CURRENT CONTROLLER NUMBER
: 2508 !!     CDISK - CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 2509 !!     L#LUN - CURRENT (DRS) UNIT NUMBER
: 2510 !!     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 2511 !!-
: 2512
: 2513 begin
: 2514 local
: 2515     MAX0_LBNS : WORD UNSIGNED,      ! UNIT'S MAXIMUM LO WORD LBN
: 2516     MAX1_LBNS : WORD UNSIGNED;     ! UNIT'S MAXIMUM HI WORD LBN
: 2517
: 2518     P_INDEX = GET_PKT (.CCTLR);      ! GET AN MSCP PACKET
: 2519     MSCP_PKT [.P_INDEX, MSGLEN] = SZ_ONL; ! PACKET SIZE
: 2520     MSCP_PKT [.P_INDEX, DK_NUM] = .CDISK; ! SET DISK ADDRESS (RD/RX DISK NUMBER)
: 2521     MSCP_PKT [.P_INDEX, OPCODE] = OP_ONL; ! OPCODE FOR "ONLINE"
: 2522     MSCP_PKT [.P_INDEX, DDPAR] = BIT00; ! SHOW ALL ECC ERRORS IN ERROR LOG MESSAGES
: 2523     MSCP_PKT [.P_INDEX, CMD_TYPE] = SEQ_CMD; ! SEQUENTIAL COMMAND
: 2524
: 2525     if SEND (.P_INDEX) eq1 FAILURE    ! ATTEMPT TO SEND; IF CTLR IS OFFLINE
: 2526     then
: 2527         begin
: 2528             T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 2529             CST_ADDR [.CUOFF, D_FATAL] = TRUE; ! FATAL ERROR
: 2530             ERRDF (22, EGD_22, 0);
: 2531             DUR [.L#LUN] = DU_ONLINE; ! SETUP REASON TO DROP UNIT
: 2532             DODU (.L#LUN); ! DROP UNIT
: 2533             PUT_PKT (.P_INDEX); ! RETURN PACKET TO POOL
: 2534         end
: 2535     else
: 2536         begin ! OTHERWISE (SEND WAS SUCCESSFUL)
: 2537
: 2538         do
: 2539             begin
: 2540                 WAIT (); ! WAIT FOR RETPKT RESPONSE
: 2541                 RP_INDX = OUT_IODQ (); ! GET INDEX OF RETPKT
: 2542                 RP_ADDR = RETPKT + (.RP_INDX * RP_LEN * 2); ! CALCULATE RETPKT ADDRESS
: 2543
: 2544                 if .RP_ADDR [MESTYP] neq MT_SEQ ! RETURN ALL RETPKTS NOT SENT BY CONTROLLER
: 2545                 then
: 2546                     PUT_RETPKT (.RP_INDX);
: 2547
: 2548                 end
: 2549             until (.RP_ADDR [CONID] eq1 CID_DRIVER) or
: 2550                 ((.RP_ADDR [MESTYP] eq1 MT_SEQ) and

```

```

:      2551      ((.RP_ADDR [ENDCOD] and OP_END) eq1 OP_END));
:      2552
:      2553      if .RP_ADDR [CONID] eq1 CID_DRIVER      ! IF RETPKT IS FROM "DRIVER"
:      2554      then
:      2555      begin
:      2556      PRINTF (DBM26);      ! "ERROR IN UNIT_INIT"
:      2557      DR_ERR ();      ! DROP CONTROLLER
:      2558      end
:      2559      else
:      2560
:      2561      if .RP_ADDR [ENDCOD] neq (OP_ONL or OP_END) ! IF RETPKT IS FROM DISK MSCP
:      2562      then
:      2563      begin
:      2564      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
:      2565      CST_ADDR [.CUOFF, D_FATAL] = TRUE;
:      2566      ERRDF (23, EGD_23, EMS_21);      ! FATAL ERROR
:      2567      DUR [.L#LUN] = DU_ONLINE;      ! SETUP REASON TO DROP UNIT
:      2568      DODU (.L#LUN);      ! DROP UNIT
:      2569      end
:      2570      else
:      2571      begin      ! RETPKT HAS GOOD ENDCODE
:      2572      ST_CODE = .RP_ADDR [STSCOD];      ! GET STATUS CODE
:      2573      SB_CODE = .RP_ADDR [SUBCOD];      ! GET SUB-CODE
:      2574
:      2575      if .ST_CODE neq ST_SUC      ! IF STATUS CODE IS NOT SUCCESSFUL
:      2576      then
:      2577      begin
:      2578      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
:      2579      CST_ADDR [.CUOFF, D_FATAL] = TRUE;
:      2580      ERRDF (15, EGD_15, EMS_30);      ! ONLINE FAILED
:      2581      DUR [.L#LUN] = DU_ONLINE;      ! SET UP REASON FOR DROPPING UNIT
:      2582      DODU (.L#LUN);      ! DROP UNIT
:      2583      end
:      2584      else
:      2585      begin      ! SUCCESSFUL OPERATION
:      2586
:      2587      MAX0_LBNS = .RP_ADDR [SIZE0];      ! LOAD LOWER WORD OF UNIT SIZE
:      2588      MAX1_LBNS = .RP_ADDR [SIZE1];      ! LOAD UPPER WORD OF UNIT SIZE
:      2589
:      2590      if (.MAX0_LBNS eq1 0)      ! THIS SUBTRACTS ONE FROM THE TOTAL
:      2591      then      ! BECAUSE EVERYTHING STARTS AT 0
:      2592      begin      ! THROUGH (MAXIMUM - 1)
:      2593      MAX0_LBNS = #0'177777';
:      2594      MAX1_LBNS = .MAX1_LBNS - 1;
:      2595      end
:      2596      else
:      2597      MAX0_LBNS = .MAX0_LBNS - 1;
:      2598
:      2599      if (.CST_ADDR [.CUOFF + 2, D_BEG1] gtru .MAX1_LBNS) or      ! THIS SECTION CHECKS TO SEE
:      2600
:      2601      ((.CST_ADDR [.CUOFF + 2, D_BEG1] eq1u .MAX1_LBNS) and      ! IN SOFTWARE QUESTIONS WERE
:      2602      (.CST_ADDR [.CUOFF + 1, D_BEG0] gtru (.MAX0_LBNS - 1)))      ! DEVICE SPECIFIED
:      2603      ! note 1 less then max. or diagnosti
:      2604
: IF LBNS LISTED
:      2600
:      2601      TO LARGE FOR
:      2602
:      2603
: c will error

```

```

: 2604           then
: 2605           begin
: 2606             CST_ADDR [.CUOFF + 2, D_BEG1] = 0;
: 2607             CST_ADDR [.CUOFF + 1, D_BEG0] = 0;
: 2608           end;
: 2609
: 2610           if
: 2611             (.CST_ADDR [.CUOFF + 4, D_END1] gtru .MAX1_LBNS) or
: 2612             ((.CST_ADDR [.CUOFF + 4, D_END1] eqlu .MAX1_LBNS) and
: 2613              (.CST_ADDR [.CUOFF + 3, D_END0] gtru .MAX0_LBNS))
: 2614           then
: 2615             begin
: 2616               CST_ADDR [.CUOFF + 4, D_END1] = .MAX1_LBNS;
: 2617               CST_ADDR [.CUOFF + 3, D_END0] = .MAX0_LBNS;
: 2618             end;
: 2619
: 2620
: 2621
: 2622
: 2623           if (((.ENTRY_REASON eql RESTART) or
: 2624              (.ENTRY_REASON eql START)) and
: 2625              (.CRN_LOW leq 8) and
: 2626              (.CRN_HIGH eql 0))
: 2627           then
: 2628             THEN
: 2629             begin
: 2630               BST [.L$LUN, LO_WRD] = .CST_ADDR [.CUOFF + 1, D_BEG0];
: 2631               BST [.L$LUN, HI_WRD] = .CST_ADDR [.CUOFF + 2, D_BEG1];
: 2632               TRK_SGN [.L$LUN] = 1;
: 2633             end;
: 2634
: 2635           select neu .RP_ADDR [R_MODEL] of
: 2636             set
: 2637               [#0'6'] : CST_ADDR [.CUOFF, D_TYPE] = RD_51;
: 2638               [#0'7'] : CST_ADDR [.CUOFF, D_TYPE] = RX_50;
: 2639               [#0'10'] : CST_ADDR [.CUOFF, D_TYPE] = RD_52;
: 2640             [otherwise] : BEGIN
: 2641               ERRDF (25 ,EGD_24 ,EMS_30);
: 2642             END;
: 2643           tes;
: 2644
: 2645           if ((.RP_ADDR [U_FLGS] and UF_WPH) eql UF_WPH) and
: 2646              (.CST_ADDR [.CUOFF, D_PROT] eql UNPROTECTED)
: 2647           then
: 2648             begin
: 2649               T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 2650               CST_ADDR [.CUOFF, D_FATAL] = TRUE;
: 2651               ERRDF (16, EGD_16, EMS_30);
: 2652               DUR [.L$LUN] = DU_PROTECT;
: 2653             end;
: 2654
: 2655
: 2656

```

! operator error

! change beginning lbn to 0

! and ending lbn to max_lbn

! if restart or

! if continue

! and

! first initialization

! initialize block numbers

! LOAD sequential LBN table

!

! POSITIVE TRACKING DIRECTIO

! THIS SECTION LOADS TYPE INTO CST TABLE

! MODEL BYTE TELLS WHAT TYPE OF UNIT IN

! IDENTIFICATION BLOCK

! RD 51

! RX 50

! RD 52

! ERROR UNKNOWN DEVICE

! STATUS CODE IS O.K.

! WRITE-PROTECT CONFLICT

! SET REASON TO DROP UNIT

```

:      2657          DODU (.L#LUN);          ! DROP UNIT
:      2658          end
:      2659          else
:      2660          begin
:      2661          CST_ADDR [.CUOFF, D_STAT] = ONLINE;      ! WRITE PROTECT SWITCH IS O.K.
:      2662          CST [.CCTLR, U_CNT] = .CST [.CCTLR, U_CNT] + 1; ! SET ONLINE FLAG
:      2663          end;
:      2664          end;
:      2665          end;          ! IF RETPKT HAS CORRECT ENDCODE
:      2666
:      2667          PUT_RETPKT (.RP_INDX);
:      2668          end;          ! IF SEND WAS SUCCESSFUL
:      2669
:      2670          end;          ! ROUTINE UNIT-INIT

```

```

000000 004137 000000G          .SBTTL UNIT.INIT INITIALIZATION TEST ROUTINES
                                UNIT.INIT:
000004 024646          JSR      R1, #SAVES          ;          2498
000006 013746 000000G          CMP      -(SP), -(SP)          ;
000012 004737 000000G          MOV      CCTLR, -(SP)          ;          2518
000016 010037 000000G          JSR      PC, GET.PKT
000022 010016          MOV      R0, P.INDEX          ; P.INDEX, *          2519
000024 012746 000104          MOV      #104, -(SP)
000030 004737 000000G          JSR      PC, BL#MUL
000034 012760 000044 000006G          MOV      #44, MSCP.PKT+6(R0)
000042 013760 000000G 000016G          MOV      CDISK, MSCP.PKT+16(R0)          ;          2520
000050 112760 000011 000022G          MOVB     #11, MSCP.PKT+22(R0)          ;          2521
000056 012760 000001 000046G          MOV      #1, MSCP.PKT+46(R0)          ;          2522
000064 012760 000001 000004G          MOV      #1, MSCP.PKT+4(R0)          ;          2523
000072 013716 000000G          MOV      P.INDEX, (SP)          ;          2525
000076 004737 000000G          JSR      PC, SEND
000102 005700          TST      R0
000104 001033          BNE     1#
000106 013700 000000G          MOV      T.ADDR, R0          ;          2528
000112 105260 000063          INCB     63(R0)
000116 013700 000000G          MOV      CUOFF, R0          ;          2529
000122 006300          ASL      R0
000124 063700 000000G          ADD      CST.ADDR, R0
000130 052710 010000          BIS     #10000, (R0)
000134 104455          TRAP    55          ;          2530
000136 000026          .WORD   26
000140 000000G          .WORD   EGD.22
000142 000000          .WORD   0
000144 013700 000000G          MOV      L#LUN, R0          ;          2531
000150 112760 000007 000000G          MOVB     #7, DUR(R0)
000156 104451          TRAP    51          ;          2532
000160 013716 000000G          MOV      P.INDEX, (SP)          ;          2533
000164 004737 000000G          JSR      PC, PUT.PKT
000170 000137 005264'          JMP      24#          ;          2525
000174 004737 000000G          JSR      PC, WAIT          ;          2540
000200 004737 000000G          JSR      PC, OUT.IODQ          ;          2541

```

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0306
Page 51
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (13)

000204	010037	000000G		MOV	R0,RP.INDX		
000210	010016			MOV	R0,(SP)	; RP.INDX,*	2542
000212	012746	000060		MOV	#60,-(SP)		
000216	004737	000000G		JSR	PC,BL#MUL		
000222	062700	000000G		ADD	#RETPKT,R0		
000226	010037	000000G		MOV	R0,RP.ADDR		
000232	132760	000360	000002	BITB	#360,2(R0)		2544
000240	001404			BEQ	2#		
000242	013716	000000G		MOV	RP.INDX,(SP)		2546
000246	004737	000000G		JSR	PC,PUT.RETPKT		
000252	005726		2#:	TST	(SP)+		2539
000254	013701	000000G		MOV	RP.ADDR,R1		2549
000260	005000			CLR	R0		
000262	126127	000003	000003	CMPB	3(R1),#3		
000270	001002			BNE	3#		
000272	005200			INC	R0		
000274	000407			BR	4#		
000276	132761	000360	000002	3#:	BITB	#360,2(R1)	2550
000304	001333			BNE	1#		
000306	105761	000014		TSTB	14(R1)		2551
000312	100330			BPL	1#		
000314	006000		4#:	ROR	R0		2553
000316	103012			BCC	5#		
000320	012716	000000G		MOV	#DBM26,(SP)		2556
000324	012746	000001		MOV	#1,-(SP)		
000330	010600			MOV	SP,R0	; SP,*	
000332	104417			TRAP	17		
000334	004737	000000V		JSR	PC,DR.ERR		2557
000340	000137	005252'		JMP	22#		2555
000344	013705	000000G		5#:	MOV	CST.ADDR,R5	2565
000350	013766	000000G	000006	MOV	CUOFF,6(SP)		
000356	006366	000006		ASL	6(SP)		
000362	060566	000006		ADD	R5,6(SP)		
000366	126127	000014	000211	CMPB	14(R1),#211		2561
000374	001422			BEQ	6#		
000376	013700	000000G		MOV	T.ADDR,R0		2564
000402	105260	000062		INCB	62(R0)		
000406	052776	010000	000006	BIS	#10000,#6(SP)		2565
000414	104455			TRAP	55		2566
000416	000027			.WORD	27		
000420	000000G			.WORD	EGD.23		
000422	000000G			.WORD	EMS.21		
000424	013700	000000G		MOV	L#LUN,R0		2567
000430	112760	000007	000000G	MOV	#7,DUR(R0)		
000436	104451			TRAP	51		2568
000440	000445			BR	7#		
000442	116137	000016	000000G	6#:	MOVB	16(R1),ST.CODE	2561
000450	042737	177740	000000G	BIC	#177740,ST.CODE		2572
000456	016100	000016		MOV	16(R1),R0		
000462	006200			ASR	R0		2573
000464	006200			ASR	R0		
000466	006200			ASR	R0		
000470	006200			ASR	R0		

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0307
Page 52
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (13)

000472	006200		ASR	R0		
000474	042700	174000	BIC	#174000,R0		
000500	010037	000000G	MOV	R0,SB.CODE		
000504	005737	000000G	TST	ST.CODE	:	2575
000510	001423		BEQ	8#	:	
000512	013700	000000G	MOV	T.ADDR,R0	:	2578
000516	105260	000062	INCB	62(R0)	:	
000522	052776	010000 000006	BIS	#10000,86(SP)	:	2579
000530	104455		TRAP	55	:	2580
000532	000017		.WORD	17		
000534	000000G		.WORD	EGD.15		
000536	000000G		.WORD	EMS.30		
000540	013700	000000G	MOV	L#LUN,R0	:	2581
000544	112760	000007 000000G	MOVB	#7,DUR(R0)		
000552	104451		TRAP	51	:	2582
000554	000137	005254'	JMP	23#	:	2575
000560	016103	000044	MOV	44(R1),R3	:	*.MAX0.LBNS
000564	016104	000046	MOV	46(R1),R4	:	*.MAX1.LBNS
000570	005703		TST	R3	:	MAX0.LBNS
000572	001004		BNE	9#		
000574	012703	177777	MOV	#-1,R3	:	*.MAX0.LBNS
000600	005304		DEC	R4	:	MAX1.LBNS
000602	000401		BR	10#	:	2590
000604	005303		DEC	R3	:	MAX0.LBNS
000606	013700	000000G	MOV	CUOFF,R0	:	2597
000612	006300		ASL	R0	:	2599
000614	060500		ADD	R5,R0		
000616	011766	000004	MOV	(PC),4(SP)		
000622	060066	000004	ADD	R0,4(SP)		
000626	027604	000004	CMP	84(SP),R4	:	*.MAX1.LBNS
000632	101012		BHI	11#		
000634	001021		BNE	12#	:	2601
000636	013700	000000G	MOV	CUOFF,R0	:	2602
000642	006300		ASL	R0		
000644	060500		ADD	R5,R0		
000646	010302		MOV	R3,R2	:	MAX0.LBNS,*
000650	005302		DEC	R2		
000652	026002	000002	CMP	2(R0),R2		
000656	101410		BLOS	12#		
000660	005076	000004	CLR	84(SP)	:	2606
000664	013700	000000G	MOV	CUOFF,R0	:	2607
000670	006300		ASL	R0		
000672	060500		ADD	R5,R0		
000674	005060	000002	CLR	2(R0)		
000700	013702	000000G	MOV	CUOFF,R2	:	2611
000704	006302		ASL	R2		
000706	060502		ADD	R5,R2		
000710	026204	000010	CMP	10(R2),R4	:	*.MAX1.LBNS
000714	101010		BHI	13#		
000716	001017		BNE	14#	:	2613
000720	013700	000000G	MOV	CUOFF,R0	:	2614
000724	006300		ASL	R0		
000726	060500		ADD	R5,R0		

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	
V01.2	INITIALIZATION TEST ROUTINES	14-Dec-1983 16:12:00	DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (13)	
000730	026003	000006		
000734	101410			
000736	010462	000010	13#:	
000742	013700	000000G		
000746	006300			
000750	060500			
000752	010360	000006		
000756	123727	000000G 000002	14#:	
000764	001404			
000766	123727	000000G 000001		
000774	001031			
000776	023727	000000G 000010	15#:	
001004	003025			
001006	005737	000000G		
001012	001022			
001014	013704	000000G		
001020	010403			
001022	006303			
001024	006303			
001026	013700	000000G		
001032	006300			
001034	060500			
001036	016063	000002 000000G		
001044	017663	000004 000002G		
001052	112764	000001 000000G		
001060	005003		16#:	
001062	156103	000032		
001066	020327	000006		
001072	001011			
001074	013700	000000G		
001100	006300			
001102	060500			
001104	142710	000034		
001110	152710	000004		
001114	000432			
001116	020327	000007	17#:	
001122	001007			
001124	013700	000000G		
001130	006300			
001132	060500			
001134	142710	000034		
001140	000420			
001142	020327	000010	18#:	
001146	001011			
001150	013700	000000G		
001154	006300			
001156	060500			
001160	142710	000034		
001164	152710	000010		
001170	000404			
001172	104455		19#:	
001174	000031			
001176	000000G			

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0309
Page 54
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (13)

001200	000000G				.WORD	EMS.30		
001202	032761	020000	000022	20#:	BIT	#20000,22(R1)	:	2649
001210	001430				BEQ	21#	:	
001212	013700	000000G			MOV	CUOFF,R0	:	2650
001216	006300				ASL	R0	:	
001220	060500				ADD	R5,R0	:	
001222	005710				TST	(R0)	:	
001224	100022				BPL	21#	:	
001226	013700	000000G			MOV	T.ADDR,R0	:	2653
001232	105260	000062			INCB	62(R0)	:	
001236	052776	010000	000006		BIS	#10000,#6(SP)	:	2654
001244	104455				TRAP	55	:	2655
001246	000020				.WORD	20	:	
001250	000000G				.WORD	EGD.16	:	
001252	000000G				.WORD	EMS.30	:	
001254	013700	000000G			MOV	L#LUN,R0	:	2656
001260	112760	000011	000000G		MOVB	#11,DUR(R0)	:	
001266	104451				TRAP	51	:	2657
001270	000414				BR	23#	:	2649
001272	052776	020000	000006	21#:	BIS	#20000,#6(SP)	:	2661
001300	013716	000000G			MOV	CCTLR,(SP)	:	2662
001304	012746	000076			MOV	#76,-(SP)	:	
001310	004737	000000G			JSR	PC,BL#MUL	:	
001314	105260	000005G			INCB	CST+5(R0)	:	
001320	005726			22#:	TST	(SP)+	:	2660
001322	013716	000000G		23#:	MOV	RP.INDX,(SP)	:	2667
001326	004737	000000G			JSR	PC,PUT.RETPKT	:	
001332	062706	000010		24#:	ADD	#10,SP	:	2498
001336	000207				RTS	PC	:	

: Routine Size: 368 words, Routine Base: #CODE# + 3732
: Maximum stack depth per invocation: 13 words


```

: 2671 routine DR_ERR : novalue =
: 2672
: 2673 !+
: 2674 ! THIS ROUTINE IS DESIGNED TO PROCESS RETURN PACKETS THAT ORIGINATE AT
: 2675 ! THE "DRIVER" RATHER THAN THE DEVICE. DRIVER-ORIGINATED PACKETS INDICATE
: 2676 ! EITHER A FATAL DEVICE ERROR OR A COMMAND TIMEOUT. SINCE THIS ROUTINE IS
: 2677 ! ONLY CALLED DURING THE INITIALIZATION TEST, IT TREATS A COMMAND TIMEOUT
: 2678 ! AS AN INITIALIZATION ERROR.
: 2679 !
: 2680 ! IMPLICIT INPUTS:
: 2681 ! RP_ADDR - ADDRESS OF A RETPKT THAT ORIGINATED AT THE "DRIVER"
: 2682 ! (I.E., CONNECTION ID = CID_DRIVER)
: 2683 !-
: 2684
: 2685 begin
: 2686
: 2687 local
: 2688 REASON : word initial (DU_TIME);           ! ASSUME COMMAND TIMEOUT
: 2689
: 2690 if .RP_ADDR [MESTYP] eq1 MT_FATAL         ! IF FATAL DEVICE ERROR
: 2691 then
: 2692 REASON = DU_DFATAL;                       ! CHANGE REASON TO FATAL ERROR
: 2693
: 2694 DROP_CTLR (.CCTLR, .REASON);            ! DROP ALL UNITS ON CONTROLLER
: 2695 end;

```

Address	Hex	Dec	Code	Comment	Line No
000000	010146		.SBTTL	DR.ERR INITIALIZATION TEST ROUTINES	2671
000002	012701	000012	DR.ERR: MOV	R1, -(SP)	2685
000006	013700	000000G	MOV	#12, R1	2690
000012	116000	000002	MOV	RP.ADDR, R0	
000016	042700	177417	MOVB	2(R0), R0	
000022	020027	000060	BIC	#177417, R0	
000026	001002		CMP	R0, #60	
000030	012701	000005	BNE	1#	
000034	013746	000000G	MOV	#5, R1	2692
000040	010146		1#: MOV	CCTLR, -(SP)	2694
000042	004737	000000G	MOV	R1, -(SP)	
000046	022626		JSR	PC, DROP_CTLR	
000050	012601		CMP	(SP)+, (SP)+	2685
000052	000207		MOV	(SP)+, R1	2671
			RTS	PC	

; Routine Size: 22 words, Routine Base: \$CODE\$ + 5272
; Maximum stack depth per invocation: 4 words

```

: 2696 routine ACCESS : novalue =
: 2697
: 2698 !!+
: 2699 ! THIS ROUTINE IS CALLED BY INIT_TEST TO VERIFY THAT THE CURRENT DISK
: 2700 ! CAN BE ACCESSED. THIS OBJECTIVE IS ACCOMPLISHED BY FORMATTING AND
: 2701 ! SENDING ONE OR TWO MSCP ACCESS COMMANDS TO THE DISK, AND CHECKING
: 2702 ! THE STATUS FIELD OF THE RESPONSE MESSAGE(S).
: 2703 !
: 2704 ! IMPLICIT INPUTS:
: 2705 !     CCTLN - CURRENT CONTROLLER NUMBER
: 2706 !     CDISK - CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 2707 !     L$LUN - CURRENT (DRS) UNIT NUMBER
: 2708 !-
: 2709
: 2710 begin
: 2711
: 2712 local
: 2713     RESULT : word initial (FAILURE),           ! GUILTY UNTIL PROVEN INNOCENT
: 2714     LBN : word,
: 2715     PASS : word initial (1);                 ! LOOP PASS COUNT
: 2716
: 2717     ST_CODE = SB_CODE = 0;                   ! STATUS CODE AND SUB-CODE
: 2718     LBN = (((.MAX_LBN [.L$LUN] + 1) + -1) and so'77777') - 1;
: 2719                                           ! START WITH LAST LBN ON TOP SURFACE: [(X+1)/2] -1
: 2720
: 2721 do
: 2722     begin                                     ! LOOP STARTS HERE
: 2723     P_INDEX = GET_PKT (.CCTLN);              ! GET AN MSCP PACKET
: 2724     MSCP_PKT [.P_INDEX, DK_NUM] = .CDISK;   ! SET DISK ADDR (RD/RX DISK NUMBER)
: 2725     MSCP_PKT [.P_INDEX, OP_CODE] = OP_ACC; ! ACCESS OP CODE
: 2726     MSCP_PKT [.P_INDEX, BC_LO] = 512;      ! BYTE COUNT (1 BLOCK)
: 2727     MSCP_PKT [.P_INDEX, LBN_L] = .LBN;     ! LOGICAL BLOCK NUMBER
: 2728     MSCP_PKT [.P_INDEX, CMD_TYPE] = NON_SEQ_CMD; ! NON-SEQUENTIAL COMMAND
: 2729
: 2730     if SEND (.P_INDEX) eq 1 FAILURE         ! ATTEMPT TO SEND; IF CTLR NOT ONLINE
: 2731     then
: 2732         begin
: 2733             PUT_PKT (.P_INDEX);             ! RETURN PACKET TO POOL
: 2734             PASS = 2;                       ! NO MORE TRIES
: 2735         end
: 2736     else
: 2737         begin                                 ! IF SEND WAS SUCCESSFUL
: 2738
: 2739         do
: 2740             begin
: 2741                 WAIT ();                     ! WAIT FOR RESPONSE
: 2742                 RP_INDEX = OUT_IODQ ();     ! GET RETPKT (RESPONSE) INDEX
: 2743                 RP_ADDR = RETPKT + (.RP_INDEX * RP_LEN * 2); ! CALCULATE RETPKT ADDRESS
: 2744
: 2745                 if .RP_ADDR [MESTYP] neq MT_SEQ ! RETURN ALL RETPKTS NOT SENT BY CONTROLLER
: 2746                 then
: 2747                     PUT_RETPKT (.RP_INDEX);
: 2748

```

```

:      2749      end
:      2750      until (.RP_ADDR [CONID] eql CID_DRIVER) or
:      2751          ((.RP_ADDR [MESTYP] eql MT_SEQ) and
:      2752              ((.RP_ADDR [ENDCOD] and OP_END) eql OP_END));
:      2753
:      2754      if .RP_ADDR [CONID] eql CID_DRIVER ! IF RETPKT CAME FROM "DRIVER"
:      2755      then
:      2756          PASS = 2 ! NO MORE TRIES
:      2757      else
:      2758
:      2759          if .RP_ADDR [ENDCOD] neq (OP_ACC or OP_END)
:      2760          then
:      2761              begin
:      2762                  PRINTF (DBM29); ! "RETPKT HAS BAD ENDCODE"
:      2763                  EMSCMD ();
:      2764              end
:      2765          else
:      2766              begin ! RETPKT HAS CORRECT ENDCODE
:      2767                  ST_CODE = .RP_ADDR [STSCOD]; ! GET STATUS CODE FROM PACKET
:      2768                  SB_CODE = .RP_ADDR [SUBCOD]; ! GET SUB-CODE FROM PACKET
:      2769
:      2770                  if .ST_CODE eql ST_SUC ! IF STATUS CODE INDICATES SUCCESS
:      2771                  then
:      2772                      begin
:      2773                          RESULT = SUCCESS;
:      2774                          PASS = 2; ! NO NEED TO TRY AGAIN
:      2775                      end;
:      2776
:      2777                  end; ! IF RETPKT HAS CORRECT ENDCODE
:      2778
:      2779                  PUT_RETPKT (.RP_INDX);
:      2780                  end; ! IF SEND WAS SUCCESSFUL
:      2781
:      2782                  LBN = .LBN + 1; ! ADVANCE TO FIRST LBN OF BOTTOM SURFACE
:      2783                  PASS = .PASS + 1; ! SECOND PASS
:      2784                  end ! END OF PASS LOOP
:      2785      until .PASS gequ 3;
:      2786
:      2787      if .RESULT eql FAILURE
:      2788      then
:      2789          begin
:      2790              T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
:      2791              CST_ADDR [.CUOFF, D_FATAL] = TRUE; ! FATAL ERROR
:      2792              ERRDF (17, EGD_17, EMS_30); ! ACCESS FAILED
:      2793              DUR [.L#LUN] = DU_ACCESS; ! SET REASON TO DROP UNIT
:      2794              DODU (.L#LUN); ! DROP UNIT
:      2795              end; ! IF ACCESS FAILED
:      2796
:      2797      end; ! ROUTINE ACCESS

```

ZRQAM3
V01.2

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0313
Page 58
VAX-11 Blues-16 V3-555
DISK:USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (15)

000004	005003			CLR	R3		; RESULT	2710
000006	012702	000001		MOV	#1,R2		; *,PASS	
000012	005037	000000G		CLR	SB.CODE			2717
000016	005037	000000G		CLR	ST.CODE			
000022	013700	000000G		MOV	L#LUN,R0			2718
000026	006300			ASL	R0			
000030	016000	000054'		MOV	MAX.LBN(R0),R0			
000034	060200			ADD	R2,R0			
000036	006200			ASR	R0			
000040	010004			MOV	R0,R4		; *,LBN	
000042	042704	100000		BIC	#100000,R4		; *,LBN	
000046	005304			DEC	R4		; LBN	
000050	013746	000000G		MOV	CCTLR,-(SP)			2723
000054	004737	000000G	1#:	JSR	PC,GET.PKT			
000060	010037	000000G		MOV	R0,P.INDEX			
000064	010016			MOV	R0,(SP)		; P.INDEX,*	2724
000066	012746	000104		MOV	#104,-(SP)			
000072	004737	000000G		JSR	PC,BL#MUL			
000076	013760	000000G	000016G	MOV	CDISK,MSCP.PKT+16(R0)			
000104	112760	000020	000022G	MOVB	#20,MSCP.PKT+22(R0)			2725
000112	012760	001000	000026G	MOV	#1000,MSCP.PKT+26(R0)			2726
000120	010460	000046G		MOV	R4,MSCP.PKT+46(R0)		; LBN,*	2727
000124	012760	000002	000004G	MOV	#2,MSCP.PKT+4(R0)			2728
000132	013716	000000G		MOV	P.INDEX,(SP)			2730
000136	004737	000000G		JSR	PC,SEND			
000142	005700			TST	R0			
000144	001007			BNE	2#			
000146	013716	000000G		MOV	P.INDEX,(SP)			2733
000152	004737	000000G		JSR	PC,PUT.PKT			
000156	012702	000002		MOV	#2,R2		; *,PASS	2734
000162	000524			BR	9#			2730
000164	004737	000000G		JSR	PC,WAIT			2741
000170	004737	000000G	2#:	JSR	PC,OUT.IODQ			2742
000174	010037	000000G		MOV	R0,RP.INDX			
000200	010016			MOV	R0,(SP)		; RP.INDX,*	2743
000202	012746	000060		MOV	#60,-(SP)			
000206	004737	000000G		JSR	PC,BL#MUL			
000212	062700	000000G		ADD	#RETPKT,R0			
000216	010037	000000G		MOV	R0,RP.ADDR			
000222	132760	000360	000002	BITB	#360,2(R0)			2745
000230	001404			BEQ	3#			
000232	013716	000000G		MOV	RP.INDX,(SP)			2747
000236	004737	000000G		JSR	PC,PUT.RETPKT			
000242	005726			TST	(SP)+			2740
000244	013701	000000G	3#:	MOV	RP.ADDR,R1			2750
000250	005000			CLR	R0			
000252	126127	000003	000003	CHPB	3(R1),#3			
000260	001002			BNE	4#			
000262	005200			INC	R0			
000264	000407			BR	5#			
000266	132761	000360	000002	BITB	#360,2(R1)			2751
000274	001333			BNE	2#			
000276	105761	000014	4#:	TSTB	14(R1)			2752

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	SEQ 0314
V01.2	INITIALIZATION TEST ROUTINES	14-Dec-1983 16:12:00	DISK#USE#2:[DIETZ.RDRX]ZRQACO.BL2;161 (15)	Page 59
000302	100330			
000304	006000	5#:		2754
000306	103444			2756
000310	126127	000014	000220	2759
000316	001412			
000320	012716	000000G		
000324	012746	000001		2762
000330	010600			
000332	104417			
000334	004737	000000G		2763
000340	005726			2761
000342	000430			2759
000344	116137	000016	000000G	2767
000352	042737	177740	000000G	
000360	016100	000016		2768
000364	006200			
000366	006200			
000370	006200			
000372	006200			
000374	006200			
000376	042700	174000		
000402	010037	000000G		
000406	005737	000000G		
000412	001004			2770
000414	012703	000001		
000420	012702	000002	7#:	2773
000424	013716	000000G	8#:	2774
000430	004737	000000G		2779
000434	005204		9#:	
000436	005202			2782
000440	022626			2783
000442	020227	000003		2722
000446	103600			2785
000450	005703			
000452	001025			2787
000454	013700	000000G		
000460	105260	000062		2790
000464	013700	000000G		
000470	006300			2791
000472	063700	000000G		
000476	052710	010000		
000502	104455			
000504	000021			2792
000506	000000G			
000510	000000G			
000512	013700	000000G		
000516	112760	000010	000000G	2793
000524	104451			2794
000526	000207		10#:	2696

; Routine Size: 172 words, Routine Base: \$CODE\$ + 5346
 ; Maximum stack depth per invocation: 10 words

```

: 2798 #abttl 'MULTI-DRIVE TEST ROUTINES'
: 2799
: 2800 routine MULTI_DRIVE : novalue =
: 2801
: 2802 !*
: 2803 ! THIS SUBTEST IS THE MOST SIGNIFICANT PART OF THE ENTIRE PROGRAM. THE
: 2804 ! MULTI-DRIVE TEST IS A HOST-CONTROLLED EXERCISER DESIGNED TO GIVE THE
: 2805 ! USER AN INDICATION OF HOW ONE OR SEVERAL RDRX DRIVES WOULD PERFORM IN
: 2806 ! AN OPERATING SYSTEM ENVIRONMENT.
: 2807 !
: 2808 ! THIS ROUTINE ACTS AS AN "EXECUTIVE" TO THE WHOLE PROCESS. AFTER
: 2809 ! INVOKING MD_INIT TO INITIALIZE MULTI-DRIVE TEST DATA, THIS ROUTINE
: 2810 ! ENTERS A LOOP WHICH ISSUES QIOS TO ALL ACTIVE CONTROLLERS AND PROCESSES
: 2811 ! ANY RESPONSES. IN ADDITION, ALL OUTSTANDING COMMANDS ARE TIMED IN
: 2812 ! DRV_TIMCHK WHICH IS INVOKED EVERY SECOND. NORMAL TERMINATION OF THIS
: 2813 ! LOOP OCCURS WHEN QIOS ARE NO LONGER BEING ISSUED, AND ALL OUTSTANDING
: 2814 ! QIOS HAVE COMPLETED.
: 2815 !-
: 2816
: 2817 begin
: 2818
: 2819 MD_INIT (); ! INIT MULTI-DRIVE TEST DATA
: 2820 INIT_OCCURED = TRUE; ! INIT SEQUENCE DONE
: 2821 do
: 2822 begin ! START OF EXECUTIVE LOOP
: 2823
: 2824 incr CTLR from 0 to (MAX_CTLR - 1) do ! FOR EACH CONTROLLER
: 2825 begin ! start of controller loop
: 2826 SET_CPAR (.CTLR); ! SET UP CURRENT CONTROLLER PARAMETERS
: 2827
: 2828 SETPRI (PRI07); ! NO INTERRUPTS WHEN EXAMINING SA
: 2829 ICTLR = .CCTLR; ! FAKE INTERRUPTING CONTROLLER'S NUMBER
: 2830 ICST_ADDR = .CST_ADDR; ! FAKE INTERRUPTING CONTROLLER'S CST ADDRESS
: 2831 IDCT_ADDR = .DCT_ADDR; ! FAKE INTERRUPTING CONTROLLER'S DCT ADDRESS
: 2832 IRDRX_ADDR = .ICST_ADDR [IP_ADDR]; ! FAKE INTERRUPTING CONTROLLER'S ADDRESS
: 2833 IDCT_ADDR [SA_SAVE] = .IRDRX_ADDR [RCSA, RC_ALL]; ! CONTENTS OF THE SA REGISTER
: 2834
: 2835
: 2836 if BIT_TST (IDCT_ADDR [SA_SAVE], SA_ERR) ! IF SA SHOWS AN ERROR
: 2837 then
: 2838 begin
: 2839 FATAL_ERROR (); ! DECLARE FATAL ERROR
: 2840 SETPRI (PRI00); ! LOWER PRIORITY
: 2841 exitloop; ! QUIT
: 2842 end
: 2843
: 2844 else SETPRI (PRI00); ! IF NO ERROR, CONTINUE
: 2845
: 2846 if QIO_OK () ! IF O.K. TO ISSUE QIO(S) TO THIS CONTROLLER
: 2847 then
: 2848 begin ! THEN
: 2849 QIO_GEN (); ! GENERATE 1 OR 2 QIOS
: 2850

```

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK#USER2:(DIETZ.RDRX)ZRQACO.BL2;161 (16)

```

: 2851      if (.MX1 geq 0) and          ! IF SUCCESS ON FIRST QIO
: 2852      (not .EOP_FLAG)
: 2853      then
: 2854
: 2855      if SEND (.MX1) eq1 SUCCESS    ! ATTEMPT TO SEND IT. IF SUCCESS
: 2856      then
: 2857      QIO [.CTLR] = .QIO [.CTLR] + 1 ! INCR OUTSTANDING QIO COUNT
: 2858      else
: 2859      PUT_PKT (.MX1);              ! RETURN PACKET TO POOL
: 2860
: 2861      if (.MX2 geq 0) and          ! IF SUCCESS ON SECOND QIO
: 2862      (not .EOP_FLAG)
: 2863      then
: 2864
: 2865      if SEND (.MX2) eq1 SUCCESS    ! ATTEMPT TO SEND IT. IF SUCCESS
: 2866      then
: 2867      QIO [.CTLR] = .QIO [.CTLR] + 1 ! INCR OUTSTANDING QIO COUNT
: 2868      else
: 2869      PUT_PKT (.MX2);              ! RETURN PACKET TO POOL
: 2870
: 2871      end;                          ! O.K. TO ISSUE QIO(S)
: 2872
: 2873      end;                          ! end of CONTROLLER LOOP
: 2874      !all controllers should have 1 or 2 cmds in packets
: 2875
: 2876      PROC_RETPKT ();               ! PROCESS ANY RETURN PACKETS
: 2877
: 2878      end                             ! EXECUTIVE PROCESSING LOOP
: 2879
: 2880      until (not QIO_OUT ()) or
: 2881      (.EOP_FLAG);
: 2882      end;                          ! EXERCISER

```

000000	004137	000000G	.SBTTL	MULTI.DRIVE MULTI-DRIVE TEST ROUTINES	
			MULTI.DRIVE:		
000004	005746		JSR	R1, #SAVE2	2800
000006	004737	000000V	TST	-(SP)	
000012	112737	000001 000000G	JSR	PC, MD, INIT	2819
000020	005002		MOVB	#1, INIT.OCCURED	2820
000022	010246		18: CLR	R2	2824
000024	004737	000000G	28: MOV	R2, -(SP)	2826
000030	012700	000340	JSR	PC, SET.CPAR	
000034	104441		MOV	#340, R0	2828
000036	013737	000000G 000076'	TRAP	41	
000044	013737	000000G 000000G	MOV	CCTLR, ICTLR	2829
000052	013737	000000G 000000G	MOV	CST.ADDR, ICST.ADDR	2830
000060	017737	000000G 000000G	MOV	DCT.ADDR, IDCT.ADDR	2831
000066	013701	000000G	MOV	#ICST.ADDR, IRDRX.ADDR	2832
000072	013700	000000G	MOV	IDCT.ADDR, R1	2833
000076	016066	000002 000002	MOV	IRDRX.ADDR, R0	
000104	016661	000002 000002	MOV	2(R0), 2(SP)	: *.RC.REG
			MOV	2(SP), 2(R1)	: RC.REG.*

ZRQAM3
V01.2RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (16)SEQ 0317
Page 62

000112	016601	000002		MOV	2(SP),R1	:	2836
000116	042701	077777		BIC	#77777,R1	:	
000122	020127	100000		CMP	R1,#-100000	:	
000126	001006			BNE	3#	:	
000130	004737	000000V		JSR	PC,FATAL.ERROR	:	2839
000134	005000			CLR	R0	:	2840
000136	104441			TRAP	41	:	
000140	005726			TST	(SP)+	:	2838
000142	000464			BR	8#	:	
000144	005000		3#:	CLR	R0	:	2844
000146	104441			TRAP	41	:	
000150	004737	000000V		JSR	PC,QIO.OK	:	2846
000154	006000			ROR	R0	:	
000156	103052			BCC	7#	:	
000160	004737	000000V		JSR	PC,QIO.GEN	:	2849
000164	013700	000100'		MOV	MX1,R0	:	2851
000170	002421			BLT	5#	:	
000172	132737	000001 000000G		BITB	#1,EOP.FLAG	:	2852
000200	001015			BNE	5#	:	
000202	010016			MOV	R0,(SP)	:	2855
000204	004737	000000G		JSR	PC,SEND	:	
000210	020027	000001		CMP	R0,#1	:	
000214	001003			BNE	4#	:	
000216	105262	000000G		INCB	QIO(R2)	: *(CTLR)	2857
000222	000404			BR	5#	:	2855
000224	013716	000100'	4#:	MOV	MX1,(SP)	:	2859
000230	004737	000000G		JSR	PC,PUT.PKT	:	
000234	013700	000102'	5#:	MOV	MX2,R0	:	2861
000240	002421			BLT	7#	:	
000242	132737	000001 000000G		BITB	#1,EOP.FLAG	:	2862
000250	001015			BNE	7#	:	
000252	010016			MOV	R0,(SP)	:	2865
000254	004737	000000G		JSR	PC,SEND	:	
000260	020027	000001		CMP	R0,#1	:	
000264	001003			BNE	6#	:	
000266	105262	000000G		INCB	QIO(R2)	: *(CTLR)	2867
000272	000404			BR	7#	:	2865
000274	013716	000102'	6#:	MOV	MX2,(SP)	:	2869
000300	004737	000000G		JSR	PC,PUT.PKT	:	
000304	005726		7#:	TST	(SP)+	:	2825
000306	005202			INC	R2	: CTLR	2824
000310	000243			.WORD	CLV!CLC	:	
000312	003643			BLE	2#	:	
000314	004737	000000V	8#:	JSR	PC,PROC.RETPKT	:	2876
000320	004737	000000V		JSR	PC,QIO.OUT	:	2880
000324	006000			ROR	R0	:	
000326	103004			BCC	9#	:	
000330	132737	000001 000000G		BITB	#1,EOP.FLAG	:	2881
000336	001630			BEQ	1#	:	
000340	005726		9#:	TST	(SP)+	:	2800
000342	000207			RTS	PC	:	

H9

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 B111-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (16)

SEQ 0318

Page 63

: Routine Size: 114 words, Routine Base: \$CODE\$ + 6076
: Maximum stack depth per invocation: 7 words

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0319
Page 64
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (17)

```

: 2883 routine MD_INIT : novalue =
: 2884
: 2885 !+
: 2886 ! THIS ROUTINE IS CALLED BY ROUTINE MULTI-DRIVE TO INITIALIZE DATA ITEMS
: 2887 ! USED BY THE MULTI-DRIVE TEST.
: 2888 !-
: 2889
: 2890 begin
: 2891
: 2892 if NOT .INIT_OCCURED           ! IF THIS IS A START
: 2893 then                           ! PARTITION FREE MEMORY INTO I/O BUFFERS
: 2894     INIT_IO_BUFF ();
: 2895
: 2896 if (.ENTRY_REASON neq CONT) and ! IF START, RESTART, OR PWR FAIL
: 2897     (.ENTRY_REASON neq NEW_PASS)
: 2898 then
: 2899
: 2900     incr CTLR from 0 to (MAX_CTLR - 1) do
: 2901         begin
: 2902             SET_CPAR (.CTLR);
: 2903
: 2904             incr DISK from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 2905                 begin
: 2906                     SET_UPAR (.DISK);
: 2907                     DPST [.L#LUN] = DP_CNT;           ! INITIALIZE DATA PATTERN SEQUENCE TABLE
: 2908                 end;
: 2909
: 2910             end;
: 2911
: 2912     incr COUNT from 0 to (QIO_PER_CTLR * MAX_CTLR - 1) do ! INITIALIZE I/O BUFFER ALLOCATION
: 2913         BUFF_OWN [.COUNT] = -1;                       ! TABLE
: 2914
: 2915 end;                                                     ! ROUTINE MD_INIT

```

			.SBTTL	MD.INIT MULTI-DRIVE TEST ROUTINES	
000000	004137	000000G	MD.INIT:JSR	R1,#SAVE2	2883
000004	132737	000001 000000G	BITB	#1,INIT.OCCURED	2892
000012	001002		BNE	1#	
000014	004737	000000V	JSR	PC,INIT.IO.BUFF	2894
000020	123727	000000G 000003	1#:CMPB	ENTRY.REASON,#3	2896
000026	001433		BEQ	4#	
000030	123727	000000G 000005	2#:CMPB	ENTRY.REASON,#5	2897
000036	001427		BEQ	4#	
000040	005002		CLR	R2	: CTLR 2900
000042	010246		2#:MOV	R2,-(SP)	: CTLR,* 2902
000044	004737	000000G	JSR	PC,SET.CPAR	
000050	012701	000003	3#:MOV	#3,R1	: *.DISK 2904
000054	010116		MOV	R1,(SP)	: DISK,* 2906
000056	004737	000000G	JSR	PC,SET.UPAR	
000062	013700	000000G	MOV	L#LUN,R0	
000066	112760	000025 000050'	MOV	#25,DPST(R0)	2907
000074	062701	000007	ADD	#7,R1	: *.DISK 2904

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0320
Page 65
VAX-11 B11es-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (17)

000100	020127	000030			CMP	R1,#30				
000104	003763				BLE	3#			; DISK,*	
000106	005726				TST	(SP)+				
000110	005202				INC	R2				2901
000112	000243				.WORD	CLV:CLC			; CTLR	2900
000114	003752				BLE	2#				
000116	005000			4#:	CLR	RO			; COUNT	2912
000120	112760	000377	000000G	5#:	MOVB	#377,BUFF.OWN(RO)			; *,*(COUNT)	2913
000126	005200				INC	RO			; CCUNT	2912
000130	020027	000007			CMP	RO,#7			; COUNT,*	
000134	003771				BLE	5#				
000136	000207				RTS	PC				2883

; Routine Size: 48 words, Routine Base: \$CODE\$ + 6442
; Maximum stack depth per invocation: 5 words

		.SBTTL	INIT.IO.BUFF MULTI-DRIVE TEST ROUTINES		
000000	004137	000000G	INIT.IO.BUFF:		
			JSR	R1,\$SAVE3	
000004	013701	000000G	MOV	FREE.MEM.ADDR,R1	2916
000010	011100		MOV	(R1),R0	2941
000012	006300		ASL	R0	
000014	060100		ADD	R1,R0	2936
000016	062700	000002	ADD	#2,R0	2941
000022	062701	000003	ADD	#3,R1	2942
000026	010137	000000G	MOV	R1,BUFF.ADDR	
000032	042737	000001 000000G	BIC	#1,BUFF.ADDR	
000040	032737	000037 000000G	1#: BIT	#37,BUFF.ADDR	2944
000046	001404		BEQ	2#	
000050	062737	000002 000000G	ADD	#2,BUFF.ADDR	2945
000056	000770		BR	1#	2944
000060	010046		2#: MOV	R0,-(SP)	2947
000062	163716	000000G	SUB	BUFF.ADDR,(SP)	
000066	012746	000010	MOV	#10,-(SP)	
000072	004737	000000G	JSR	PC,BL#DIV	
000076	010037	000000G	MOV	R0,BYTS.PER.QIO	
000102	042737	000037 000000G	BIC	#37,BYTS.PER.QIO	
000110	023727	000000G 002000	CMP	BYTS.PER.QIO,#2000	2950
000116	101403		BLOS	3#	
000120	012737	002000 000000G	MOV	#2000,BYTS.PER.QIO	2952
000126	023727	000000G 000040	3#: CMP	BYTS.PER.QIO,#40	2954
000134	103005		BHIS	4#	
000136	104454		TRAP	54	2957
000140	000002		.WORD	2	
000142	000000G		.WORD	EGS.02	
000144	000000		.WORD	0	
000146	104444		TRAP	44	
000150	012702	000001	4#: MOV	#1,R2	2961
000154	010201		5#: MOV	R2,R1	2965
000156	006301		ASL	R1	
000160	010200		MOV	R2,R0	INDEX,*
000162	006300		ASL	R0	
000164	016003	177776G	MOV	BUFF.ADDR-2(R0),R3	
000170	063703	000000G	ADD	BYTS.PER.QIO,R3	
000174	010361	000000G	MOV	R3,BUFF.ADDR(R1)	
000200	005202		INC	R2	INDEX
000202	020227	000007	CMP	R2,#7	INDEX,*
000206	003762		BLE	5#	
000210	022626		CMP	(SP)+,(SP)+	2936
000212	000207		RTS	PC	2916

; Routine Size: 70 words, Routine Base: #CODE# + 6602
; Maximum stack depth per invocation: 8 words

```

: 2968 routine QIO_OK =
: 2969
: 2970 !+
: 2971 ! THIS ROUTINE IS CALLED BY THE MULTI_DRIVE "EXECUTIVE" IN ORDER TO
: 2972 ! DETERMINE WHETHER OR NOT A QIO REQUEST (OR QIO PAIR) SHOULD BE
: 2973 ! GENERATED TO THE CURRENT CONTROLLER. A VALUE OF "TRUE" IS RETURNED IF
: 2974 ! THE CONTROLLER MEETS 3 REQUIREMENTS:
: 2975 !
: 2976 !     A. THE CONTROLLER IS ONLINE;
: 2977 !     B. THE NUMBER OF OUTSTANDING QIOS IS AT LEAST 2 LESS THAN THE
: 2978 !         MAXIMUM ALLOWED FOR ANY ONE CONTROLLER;
: 2979 !     C. THERE IS AT LEAST ONE DISK ONLINE TO THE CONTROLLER.
: 2980 !
: 2981 ! IF ANY OF THESE TEST FAIL, THEN A VALUE OF "FALSE" IS RETURNED.
: 2982 !
: 2983 ! IMPLICIT INPUTS:
: 2984 !     CCTLR - CURRENT CONTROLLER NUMBER
: 2985 !     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 2986 !-
: 2987
: 2988 if (.CST_ADDR [STATE] eq1 ONLINE) and           ! IF CONTROLLER IS ONLINE
: 2989 (not .EOP_FLAG) and
: 2990 ((.QIO [.CCTLR] + 2) lequ QIO_PER_CTLR) and      ! IF OUTSTANDING QIO COUNT IS O.K.
: 2991 (.CST_ADDR [U_CNT] neq 0)                       ! IF THERE IS VALID UNIT
: 2992 then
: 2993     return TRUE                                  ! "TRUE" EXIT POINT
: 2994 else
: 2995     return FALSE;                               ! "FALSE" EXIT POINT

```

000000	013700	000000G	QIO.OK: .SBTTL	QIO.OK MULTI-DRIVE TEST ROUTINES	
000004	005760	000002	MOV	CST.ADDR,RO	2988
000010	100027		TST	2(RO)	
000012	132737	000001 000000G	BPL	1#	
000020	001023		BITB	#1,EOP.FLAG	2989
000022	013700	000000G	BNE	1#	
000026	116000	000000G	MOV	CCTLR,RO	2990
000032	042700	177400	MOVB	QIO(RO),RO	
000036	062700	000002	BIC	#177400,RO	
000042	020027	000010	ADD	#2,RO	
000046	101010		CMP	RO,#10	
000050	013700	000000G	BHI	1#	
000054	105760	000005	MOV	CST.ADDR,RO	2991
000060	001403		TSTB	5(RO)	
000062	012700	000001	BEQ	1#	
000066	000207		MOV	#1,RO	2968
000070	005000		RTS	PC	
000072	000207		1#:	CLR	RO
				RTS	PC

```

; Routine Size: 30 words,      Routine Base: $CODE$ + 7016
; Maximum stack depth per invocation: 0 words

```

```

: 2996 routine QIO_OUT =
: 2997
: 2998 !+
: 2999 ! THIS ROUTINE IS CALLED BY THE MULTI_DRIVE EXECUTIVE FOR DETERMINING THE
: 3000 ! END OF THE MULTI-DRIVE TEST. ITS PURPOSE IS TO EXAMINE THE QIO VECTOR
: 3001 ! FOR ANY OUTSTANDING QIOS ON ANY CONTROLLER. A VALUE OF "TRUE" IS
: 3002 ! RETURNED IF THERE IS AT LEAST ONE QIO OUTSTANDING ON ANY CONTROLLER.
: 3003 ! OTHERWISE, "FALSE" IS RETURNED INDICATING NO OUTSTANDING QIOS.
: 3004 !-
: 3005
: 3006 begin
: 3007
: 3008 incr CTLR from 0 to (MAX_CTLR - 1) do
: 3009 begin
: 3010 SET_CPAR (.CTLR);           ! SET UP CURRENT CONTROLLER PARAMETERS
: 3011
: 3012 if .CST_ADDR [STATE] eq1 ONLINE   ! IF CONTROLLER IS ONLINE
: 3013 then
: 3014     return TRUE;
: 3015
: 3016 end;
: 3017
: 3018 return FALSE;             ! EXIT - NO CONTROLLERS ONLINE
: 3019 end;

```

Address	Hex	Label	Code	Comment	Line
000000	010146		.SBTTL	QIO.OUT MULTI-DRIVE TEST ROUTINES	
000002	005001	QIO.OUT:	MOV	R1, -(SP)	2996
000004	010146		CLR	R1	3008
000006	004737	000000G	1\$: MOV	R1, -(SP)	3010
000012	013700	000000G	JSR	PC, SET.CPAR	
000016	005760	000002	MOV	CST.ADDR, R0	3012
000022	100004		TST	2(R0)	
000024	005726		BPL	2\$	
000026	012700	000001	TST	(SP)+	3014
000032	000405		MOV	#1, R0	
000034	005726		BR	3\$	
000036	005201		2\$: TST	(SP)+	3009
000040	000243		INC	R1	3008
			.WORD	CLV!CLC	
000042	003760		BLE	1\$	
000044	005000		CLR	R0	3006
000046	012601		3\$: MOV	(SP)+, R1	
000050	000207		RTS	PC	2996

; Routine Size: 21 words, Routine Base: \$CODE\$ + 7112
; Maximum stack depth per invocation: 3 words

```

: 3020 routine QIO_GEN : novalue =
: 3021
: 3022 :+
: 3023 : THIS ROUTINE IS CALLED BY THE MULTI_DRIVE EXECUTIVE FOR AN ONLINE
: 3024 : CONTROLLER ELIGIBLE TO RECEIVE I/O TRANSFER REQUESTS. IT IS
: 3025 : RESPONSIBLE FOR SECURING ONE OR TWO MSCP PACKETS AND LOADING THEM
: 3026 : WITH VARIOUS PARAMETERS COMPRISING THE I/O REQUEST. THE I/O REQUEST
: 3027 : GENERATED HERE IS DESTINED TO A PARTICULAR UNIT SELECTED AT RANDOM FROM
: 3028 : THOSE CONFIGURED UNDER THE CURRENT CONTROLLER.
: 3029
: 3030 : EACH FIELD OF THE PACKET(S) IS LOADED WITHIN INDIVIDUAL ROUTINES
: 3031 : (QIO_FUNC, QIO_LBN, QIO_SIZE, ETC.). MOST OF THE VALUES SELECTED FOR
: 3032 : EACH FIELD ARE BASED ON A SET OF RANDOM NUMBER GENERATED AT THE START.
: 3033
: 3034 : UNDER NORMAL CIRCUMSTANCES, ONLY ONE I/O REQUEST IS GENERATED. HOWEVER,
: 3035 : IF THIS I/O REQUEST IS A "WRITE", AND IF THE OPERATOR SELECTED THE
: 3036 : OPTION FOR MOST WRITE-COMPARES, THEN A SECOND "READ" REQUEST WILL BE
: 3037 : GENERATED WITH THE SAME LBN AND BYTE COUNT.
: 3038
: 3039 : AFTER THE PACKET(S) HAVE BEEN LOADED, THIS ROUTINE REGAINS CONTROL
: 3040 : AND ATTEMPTS TO GET ONE OR TWO I/O BUFFERS FOR THE ACTUAL DATA
: 3041 : TRANSFERS. THE SUCCESS / FAIL STATUS OF THIS ENTIRE OPERATION IS
: 3042 : PASSED BACK TO THE CALLER THROUGH THE GLOBALS "MX1" AND "MX2"; THEY
: 3043 : CONTAIN VALID MSCP PACKET INDECES, OR -1.
: 3044
: 3045 : Note that the DUP Exerciser is located inside the QIO_FUNC routine.
: 3046 : Every so often the Dup exerciser will run and return the MSCP Exerciser
: 3047 : to it's normal state.
: 3048
: 3049
: 3050 : IMPLICIT INPUTS:
: 3051 :     CCTLR - CURRENT CONTROLLER NUMBER
: 3052 : -
: 3053
: 3054 begin
: 3055     MX2 = -1;                                ! ASSUME FAILURE IN SECURING 2ND PACKET
: 3056
: 3057     if (MX1 = GET_PKT (.CCTLR)) les 0        ! TRY TO GET 1ST PACKET. IF FAILURE
: 3058     then
: 3059         return;                               ! NO POINT IN CONTINUING
: 3060
: 3061     if (MX2 = GET_PKT (.CCTLR)) les 0        ! TRY TO GET 2ND PACKET. IF FAILURE
: 3062     then
: 3063         begin
: 3064             PUT_PKT (.MX1);                    ! RETURN 1ST PACKET TO POOL
: 3065             MX1 = -1;                          ! INDICATE FAILURE
: 3066             return;                            ! DONE
: 3067         end;
: 3068
: 3069     MAD1 = MSCP_PKT * (.MX1 * PKT_LEN * 2);  ! CALCULATE STARTING ADDRESSES
: 3070     MAD2 = MSCP_PKT * (.MX2 * PKT_LEN * 2);  ! OF BOTH PACKETS
: 3071     GET_RANDOM ();                             ! GENERATE A SET OF RANDOM NUMBERS
: 3072     QIO_UNIT ();                              ! LOAD RANDOM UNIT NUMBER INTO PACKETS

```


ZRGAM3
V01.2RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 B11es-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRGACO.BL2;161 (21)

```

:      3073
:      3074      if .EOP_FLAG
:      3075      then
:      3076          return;
:      3077
:      3078      QIO_FUNC ();
ND MSCP OPCODE0
:      3079
EXERCISER !†
:      3080
EXERCISER
:      3081
:      3082      if (.MX1 lss 0) OR (.EOP_FLAG)
:      3083      then return;
:      3084
:      3085      QIO_LBN ();
:      3086      QIO_SIZE ();
:      3087      GET_IO_BUFF (MAD1 [BUF_0]);
:      3088
:      3089      if .MX2 geq 0
:      3090      then
:      3091          begin
:      3092              GET_IO_BUFF (MAD2 [BUF_0]);
:      3093
:      3094              if .MAD2 [BUF_0] eqa 0
:      3095              then
:      3096                  begin
:      3097
:      3098                      if .MAD1 [BUF_0] neq 0
:      3099                      then
:      3100                          begin
:      3101                              PUT_IO_BUFF (MAD1 [BUF_0]);
:      3102                              MAD1 [BUF_0] = 0;
:      3103                              end;
:      3104
:      3105                              PUT_PKT (.MX2);
:      3106                              MX2 = -1;
:      3107                              end;
:      3108
:      3109                      end;
:      3110
:      3111              if .MAD1 [BUF_0] eqa 0
:      3112              then
:      3113                  begin
:      3114                      PUT_PKT (.MX1);
:      3115                      MX1 = -1;
:      3116                      end
:      3117              else
:      3118
:      3119                  if .MAD1 [OPCODE] eq 1 OP_WRT
:      3120                  then
:      3121                      FILL_BUFF ();
:      3122
:      3123          end;
:
: RETURN IF NO UNIT ONLINE
:
: LOAD RANDOM IMSCP FUNCTION CODE (OPCODE)0 OR IDUP EXERCISER TEST A
: THIS IS THE POINT WHERE THE DUP EXERCISER WILL CUT IN TO THE MSCP
: START WRITTING AND READING DBN'S ONCE FINISHED IT WILL RETURN THE
: TO ITS NORMAL MSCP MODE....
: IF IT WAS IN DUP TEST AND FAILED TO GET A ENVELOPE RETURN
: NO POINT IN CONTINUING
:
: LOAD LBN (RANDOM OR SEQUENTIAL)
: LOAD RANDOM BYTE COUNT
: TRY TO GET AN I/O BUFFER
:
: IF TWO QIOS ARE TO BE ISSUED
:
: TRY TO GET 2ND I/O BUFFER
:
: IF 2ND BUFFER ALLOCATION FAILED
:
: IF 1ST I/O BUFFER WAS ALLOCATED
:
: RETURN 1ST I/O BUFFER TO POOL
: MARK IT AS FAILED
:
: RETURN 2ND PACKET TO POOL
: INDICATE FAILURE
: IF 2ND I/O BUFFER ALLOCATION FAILED
:
: IF TWO QIOS ARE TO BE ISSUED
:
: IF 1ST I/O BUFFER ALLOCATION FAILED
:
: RETURN 1ST PACKET TO POOL
: INDICATE FAILURE
:
: OTHERWISE, IF 1ST OPCODE IS A WRITE (ALL IS O.K.)
: FILL 1ST I/O BUFFER WITH APPROPRIATE DATA PATTERN
:
: ROUTINE QIO_GEN

```

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (21)

SEQ 0327
Page 72

Address	OpCode	Operand1	Operand2	Label	Instruction	Comment	Line No.
000000	012737	177777	000102'		.SBTTL	QIO.GEN MULTI-DRIVE TEST ROUTINES	
000006	013746	000000G		QIO.GEN:	MOV #1, MX2		3055
000012	004737	000000G			MOV CCTL, -(SP)		3057
000016	010037	000100'			JSR PC, GET.PKT		
000022	005726				MOV R0, MX1		
000024	005700				TST (SP),		
000026	002572				TST R0	; MX1	
000030	013746	000000G			BLT 6		3059
000034	004737	000000G			MOV CCTL, -(SP)		3061
000040	010037	000102'			JSR PC, GET.PKT		
000044	005726				MOV R0, MX2		
000046	005700				TST (SP),		
000050	002011				TST R0	; MX2	
000052	013746	000100'			BGE 1		
000056	004737	000000G			MOV MX1, -(SP)		3064
000062	012737	177777	000100'		JSR PC, PUT.PKT		
000070	005726				MOV #1, MX1		3065
000072	000207				TST (SP),		3061
000074	013746	000100'			RTS PC		3063
000100	012746	000104		10:	MOV MX1, -(SP)		3069
000104	004737	000000G			MOV #104, -(SP)		
000110	062700	000000G			JSR PC, BL#MUL		
000114	010037	000104'			ADD #MSCP, PKT, R0		
000120	013716	000102'			MOV R0, MAD1		
000124	012746	000104			MOV MX2, (SP)		3070
000130	004737	000000G			MOV #104, -(SP)		
000134	062700	000000G			JSR PC, BL#MUL		
000140	010037	000106'			ADD #MSCP, PKT, R0		
000144	004737	000000V			MOV R0, MAD2		
000150	004737	000000V			JSR PC, GET.RANDOM		3071
000154	132737	000001	000000G		JSR PC, QIO.UNIT		3072
000162	001112				BITB #1, EOP.FLAG		3074
000164	004737	000000V			BNE 5		3020
000170	005737	000100'			JSR PC, QIO.FUNC		3078
000174	002505				TST MX1		3082
000176	132737	000001	000000G		BLT 5		
000204	001101				BITB #1, EOP.FLAG		
000206	004737	000000V			BNE 5		3020
000212	004737	000000V			JSR PC, QIO.LBN		3085
000216	013716	000104'			JSR PC, QIO.SIZE		3086
000222	062716	000032			MOV MAD1, (SP)		3087
000226	004737	000000G			ADD #32, (SP)		
000232	005737	000102'			JSR PC, GET.IO.BUFF		
000236	002437				TST MX2		3089
000240	013716	000106'			BLT 3		
000244	062716	000032			MOV MAD2, (SP)		3092
000250	004737	000000G			ADD #32, (SP)		
000254	013700	000106'			JSR PC, GET.IO.BUFF		
000260	005760	000032			MOV MAD2, R0		3094
000264	001024				TST 32(R0)		
000266	013700	000104'			BNE 3		
000272	062700	000032			MOV MAD1, R0		3098
					ADD #32, R0		

E10

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	SEQ 0328
V01.2	MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:00	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (21)	Page 73
000276	005710		TST (R0)	
000300	001407		BEQ 2#	
000302	010016		MOV R0,(SP)	
000304	004737	000000G	JSR PC,PUT.IO.BUFF	3101
000310	013700	000104'	MOV MAD1,R0	
000314	005060	000032	CLR 32(R0)	3102
000320	013716	000102'	MOV MX2,(SP)	
000324	004737	000000G	JSR PC,PUT.PKT	3105
000330	012737	177777 000102'	MOV #-1,MX2	
000336	013700	000104'	MOV MAD1,R0	3106
000342	005760	000032	TST 32(R0)	3111
000346	001010		BNE 4#	
000350	013716	000100'	MOV MX1,(SP)	
000354	004737	000000G	JSR PC,PUT.PKT	3114
000360	012737	177777 000100'	MOV #-1,MX1	
000366	000410		BR 5#	3115
000370	013700	000104'	MOV MAD1,R0	3111
000374	126027	000022 000042	CMPB 22(R0),#42	3119
000402	001002		BNE 5#	
000404	004737	000000V	JSR PC,FILL.BUFF	
000410	062706	000006	ADD #6,SP	3121
000414	000207		RTS PC	3054 3020

; Routine Size: 135 words, Routine Base: \$CODE\$ + 7164
 ; Maximum stack depth per invocation: 4 words

```

: 3124 routine GET_RANDOM : novalue =
: 3125
: 3126 !+
: 3127 ! THIS ROUTINE IS CALLED BY QIO_GEN TO GENERATE A SET OF RANDOM NUMBERS,
: 3128 ! AND TO STORE THEM INTO THE RANDOM NUMBER TABLE (RANDOM). THE RANDOM
: 3129 ! NUMBERS ARE USED TO SELECT I/O REQUEST PARAMETERS FOR THE CURRENT QIO
: 3130 ! OR QIO PAIR. IN ADDITION, IF DATA PATTERN #1 IS BEING USED, THESE
: 3131 ! RANDOM NUMBERS WILL BE USED IN THE WRITE OPERATION.
: 3132 !-
: 3133
: 3134 begin
: 3135
: 3136 own
: 3137 SEED : word initial (173),
: 3138 NEXT_RANDOM : word initial (245);
: 3139
: 3140 incr COUNT from 0 to (RDM_LEN - 1) do
: 3141 begin
: 3142 SEED = (.SEED + .NEXT_RANDOM + 1) * 4;
: 3143 NEXT_RANDOM = (.NEXT_RANDOM / 4) * .SEED;
: 3144 RANDOM [.COUNT] = .NEXT_RANDOM;
: 3145 end;
: 3146
: 3147 end;

```

```

001153 .PSECT $GGG$, R0
001154 000255 SEED: .WORD 255
001156 000365 NEXT_RANDOM:
        .WORD 365

```

```

007602 .SBTTL GET_RANDOM MULTI-DRIVE TEST ROUTINES
        .PSECT $CODE$, R0

```

```

000000 004137 000000G GET_RANDOM:
000004 013702 001154' JSR R1,$SAVE3 ; 3124
000010 013703 001156' MOV SEED,R2 ; 3142
000014 005001 MOV NEXT_RANDOM,R3
000016 010300 10: CLR R1 ; COUNT 3140
000020 060200 MOV R3,R0 ; 3142
000022 006300 ADD R2,R0
000024 006300 ASL R0 ; 3141
000026 010037 001154' MOV R0,SEED ; 3142
000032 062737 000004 001154' ADD #4,SEED
000040 010346 MOV R3,-(SP) ; 3143
000042 012746 000004 MOV #4,-(SP)
000046 004737 000000G JSR PC,BL$DIV
000052 013702 001154' MOV SEED,R2
000056 060200 ADD R2,R0

```

G10

ZRQAM3 RD/RX EXERCISER
V01.2 MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (22)

SEQ 0330
Page 75

000060	010037	001156'	MOV	R0,NEXT.RANDNUM		
000064	010003		MOV	R0,R3	; NEXT.RANDNUM,*	3144
000066	010361	000000G	MOV	R3,RANDOM(R1)	; *,*(COUNT)	
000072	022626		CMP	(SP)+,(SP)+	;	3141
000074	062701	000002	ADD	#2,R1	; *,COUNT	3140
000100	020127	000036	CMP	R1,#36	; COUNT,*	
000104	003744		BLE	1#		
000106	000207		RTS	PC	;	3124

; Routine Size: 36 words, Routine Base: \$CODE\$ + 7602
; Maximum stack depth per invocation: 7 words

ZRQAM3
V01.2RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2:161 (23) Page 76

```

: 3148 routine QIO_UNIT : novalue =
: 3149
: 3150 !+
: 3151 ! THIS ROUTINE IS CALLED BY QIO_GEN TO SELECT ONE UNIT
: 3152 ! CONFIGURED UNDER THE CURRENT CONTROLLER (CCTL) TO BE USED FOR THE
: 3153 ! CURRENT QIO OR QIO PAIR. THE UNIT SELECTED IS BASED ON THE NUMBER OF
: 3154 ! UNITS ELIGIBLE TO RECEIVE AN I/O REQUEST (FROM 1 TO 4). IF A "YES"
: 3155 ! ANSWER IS GIVEN FOR THE RANDOM UNIT MODE THE UNITS WILL BE PICKED AT
: 3156 ! RANDOM. IF NOT THEN THE UNITS WILL BE PICKED IN A SEQUENTIAL ORDER.
: 3157 !
: 3158 ! IMPLICIT INPUTS:
: 3159 !     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 3160 !
: 3161 ! IMPLICIT OUTPUTS:
: 3162 !     THE RD/RX DISK NUMBER (DISK ADDRESS) IS LOADED INTO THE
: 3163 !     APPROPRIATE FIELD OF BOTH MSCP PACKETS.
: 3164 !-
: 3165
: 3166 begin
: 3167     own
: 3168         RAT_COUNT : word initial (0);
: 3169     local
: 3170         x          : word,
: 3171         MOD_COUNT  : byte,
: 3172         TBL_COUNT  : byte,
: 3173         SELECT_RD  : byte;
: 3174
: 3175 MOD_COUNT = 0;                                ! find out if there is a unit online
: 3176
: 3177 incr OFFSET from (0 + OF_UN) to (3 + UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3178
: 3179     if (.CST_ADDR [.OFFSET, D_PRES] eq1 PRESENT) and
: 3180     (.CST_ADDR [.OFFSET, D_STAT] eq1 ONLINE) and
: 3181     (not .CST_ADDR [.OFFSET, D_FATAL])
: 3182     then
: 3183         begin
: 3184             MOD_COUNT = .MOD_COUNT + 1;
: 3185         end;
: 3186
: 3187 if .MOD_COUNT eq1 0                                ! IF no units on line then END OF PASS
: 3188 then
: 3189     begin
: 3190         EOP_FLAG = TRUE;
: 3191         return;
: 3192     end;
: 3193
: 3194
: 3195 if ((.SMP_FLAGS and SWF_UNT) neq SWF_UNT)          !check if sequential mode 0 = SEQUENTIAL, 1 = RANDOM
: 3196 then                                              ! SEQUENTIAL UNIT MODE
: 3197     begin
: 3198         do
: 3199             if (.BST_DEV eq1 0)
: 3200             then

```

ZRQAM3
V01.2RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK#USER2:(DIETZ.RDRX)ZRQACO.BL2;161 (23)SEQ 0332
Page 77

```

:      3201          BST_DEV = MAX_DRIVES - 1          ! Make it one less than max drive # or 3
:      3202          else
:      3203          BST_DEV = .BST_DEV - 1          ! else just subtract 1 from current device
:      3204
:      3205          until (.CST_ADDR [.BST_DEV * UNIT_SIZE + OF_UN, D_PRES] eq1 PRESENT) and ! do until find a unit online
:      3206          (.CST_ADDR [.BST_DEV * UNIT_SIZE + OF_UN, D_STAT] eq1 ONLINE) and
:      3207          (not .CST_ADDR [.BST_DEV * UNIT_SIZE + OF_UN, D_FATAL]);
:      3208          SET_UPAR (.BST_DEV * UNIT_SIZE + OF_UN);          ! set up characteristics
:      3209          MAD1 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];          ! drop in command pa
cket
:      3210          MAD2 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];
:      3211          return;
:      3212          end
:      3213          else
:      3214          begin
:      3215          !!
:      3216          !! RANDOM SELECTION OF DRIVES
:      3217          !!
:      3218          !! this part selects the device by the ratio
:      3219          !!
:      3220          rat_count = .rat_count + 1;          ! increment counter from 0 to 100
:      3221          if (.rat_count geq 100) then rat_count = 0;          ! in case counter gets to large reinit to 0
:      3222          if (.rat_count les .swp_rat)
:      3223          then
:      3224          SELECT_RD = true          ! if counter greater than swap ratio then do
:      3225          else          ! and reinitate the counter
:      3226          SELECT_RD = false;          ! if counter less than ratio do a rd-51
:      3227
:      3228          if (100 eq1 .swp_rat)          ! if ratio equal 100 do rd
:      3229          then
:      3230          SELECT_RD = true;
:      3231
:      3232          if (0 eq1 .swp_rat)          ! if ratio equals 0 do rx
:      3233          then
:      3234          SELECT_RD = FALSE;
:      3235          !!
:      3236          !! IF RD51e SELECTED
:      3237          !!
:      3238          !! COUNT NUMBER OF RDs AVAILABLE
:      3239          !!
:      3240          if .SELECT_RD
:      3241          then
:      3242          begin
:      3243          MOD_COUNT = 0;          ! COUNT THE NUMBER OF RDs IN THE SYSTEM
:      3244          incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
:      3245
:      3246          if (.CST_ADDR [.OFFSET, D_PRES] eq1 PRESENT) and
:      3247          (.CST_ADDR [.OFFSET, D_STAT] eq1 ONLINE) and
:      3248          (.CST_ADDR [.OFFSET, D_TYPE] neq RX_50) and
:      3249          (not .CST_ADDR [.OFFSET, D_FATAL])
:      3250          then
:      3251          begin
:      3252          STORAGE [.MOD_COUNT] = .OFFSET;
:      3253          MOD_COUNT = .MOD_COUNT + 1;

```

```

:
: 3254           end;
: 3255
: 3256 : SELECT ONE OF THE RD51s
: 3257 :
: 3258   if .MOD_COUNT neq 0           ! IF AT LEAST ONE RD51 PRESENT
: 3259   then
: 3260       begin
: 3261         X = ((.RANDOM [.TBL_COUNT] and #0'77777') mod .MOD_COUNT);
: 3262         SET_UPAR (.STORAGE [.X]);
: 3263         MAD1 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];
: 3264         MAD2 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];
: 3265         return;
: 3266       end;
: 3267   end
: 3268   else
: 3269 :
: 3270 : IF NO RD51 SELECTED, SELECT AN RX50
: 3271 :
: 3272 : COUNT THE NUMBER OF RX50s
: 3273 :
: 3274   begin
: 3275     MOD_COUNT = 0;           ! COUNT THE NUMBER OF RXs IN THE SYSTEM
: 3276     incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3277
: 3278       if (.CST_ADDR [.OFFSET, D_PRES] eq1 PRESENT) and
: 3279         (.CST_ADDR [.OFFSET, D_STAT] eq1 ONLINE) and
: 3280         (.CST_ADDR [.OFFSET, D_TYPE] eq1 RX_50) and
: 3281         (not .CST_ADDR [.OFFSET, D_FATAL])
: 3282       then
: 3283         begin
: 3284           STORAGE [.MOD_COUNT] = .OFFSET;
: 3285           MOD_COUNT = .MOD_COUNT + 1;
: 3286         end;
: 3287
: 3288 :
: 3289 : AND CHOOSE ONE OF THE RXs
: 3290 :
: 3291   if .MOD_COUNT neq 0           ! IF AT LEAST ONE RD51 PRESENT
: 3292   then
: 3293       begin
: 3294         X = ((.RANDOM [.TBL_COUNT] and #0'77777') mod .MOD_COUNT);
: 3295         SET_UPAR (.STORAGE [.X]);
: 3296         MAD1 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];
: 3297         MAD2 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];
: 3298         return;
: 3299       end;
: 3300   end;
: 3301
: 3302 :
: 3303 :
: 3304 : IF NO UNIT SELECTED SO FAR BY ABOVE METHOD, SELECT ANY ONE AT RANDOM
: 3305 :
: 3306 : COUNT ALL UNITS AVAILABLE

```



```

: 3307 !
: 3308 MOD_COUNT = 0; ! COUNT THE NUMBER OF ONLINE UNITS IN THE SYSTEM
: 3309 incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3310
: 3311 if (.CST_ADDR [.OFFSET, D_PRES] eq1 PRESENT) and
: 3312 (.CST_ADDR [.OFFSET, D_STAT] eq1 ONLINE) and
: 3313 (not .CST_ADDR [.OFFSET, D_FATAL])
: 3314 then
: 3315 begin
: 3316 STORAGE [.MOD_COUNT] = .OFFSET;
: 3317 MOD_COUNT = .MOD_COUNT + 1;
: 3318 end;
: 3319
: 3320 !
: 3321 ! AND CHOOSE ONE OF THEM
: 3322 !
: 3323 if .MOD_COUNT neq 0 ! IF AT LEAST ONE RD51 PRESENT
: 3324 then
: 3325 begin
: 3326 X = ((.RANDOM [.TBL_COUNT] and #0'77777') mod .MOD_COUNT);
: 3327 SET_UPAR (.STORAGE [.X]);
: 3328 MAD1 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];
: 3329 MAD2 [DK_NUM] = .CST_ADDR [.CUOFF, D_DISK_NUM];
: 3330 return;
: 3331 end;
: 3332 end;
: 3333 EOP_FLAG = TRUE; ! JUST IN CASE NO UNITS ONLINE
: 3334 end; ! ROUTINE QIO_UNIT

```

```

001160 .PSECT $GGG$, R0
001160 000000 RAT.COUNT:
      .WORD 0

007712 .SBTTL QIO.UNIT MULTI-DRIVE TEST ROUTINES
      .PSECT $CODE$, R0

000000 004137 000000G QIO.UNIT:
000004 105002 JSR R1, $SAVE4 ; 3148
000006 013703 CLR R2 ; MOD.COUNT 3175
000012 012700 MOV CST.ADDR, R3 ; 3179
000016 010001 MOV #6, R0 ; #.OFFSET 3177
000020 060301 10: MOV R0, R1 ; OFFSET, # 3179
000022 032711 ADD R3, R1
000026 001407 BIT #40000, (R1)
000030 032711 BEQ 2# ;
000034 001404 BIT #20000, (R1) ; 3180
000036 032711 BEQ 2# ;
000042 001001 BIT #10000, (R1) ; 3181
000044 105202 BNE 2# ;
      INCB R2 ; MOD.COUNT 3184

```

L10

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 B11e-16 V3-555	SEQ 0335
V01.2	MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:00	DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (23)	Page 80
000046	062700	000016	24:	ADD #16,R0 ; *,OFFSET 3177
000052	020027	000060		CMP R0,#60 ; OFFSET,*
000056	003757			BLE 1#
000060	105702			TSTB R2 ; MOD.COUNT 3187
000062	001002			BNE 3# ; 3190
000064	000137	010664'		JMP 23#
000070	032737	000002	000000G	34: BIT #2,SWP.FLAGS ; 3195
000076	001072			BNE 7#
000100	013701	001142'		MOV BST.DEV,R1 ; 3199
000104	005701			44: TST R1 ;
000106	001004			BNE 5#
000110	012737	000003	001142'	MOV #3,BST.DEV ; 3201
000116	000402			BR 6# ; 3199
000120	005337	001142'		54: DEC BST.DEV ; 3203
000124	013701	001142'		64: MOV BST.DEV,R1 ; 3205
000130	010146			MOV R1,-(SP)
000132	012746	000007		MOV #7,-(SP)
000136	004737	000000G		JSR PC,BL#MUL
000142	010004			MOV R0,R4
000144	022626			CMP (SP)+,(SP)+
000146	006300			ASL R0
000150	060300			ADD R3,R0
000152	062700	000006		ADD #6,R0
000156	032710	040000		BIT #40000,(R0)
000162	001750			BEQ 4#
000164	032710	020000		BIT #20000,(R0) ; 3206
000170	001745			BEQ 4#
000172	032710	010000		BIT #10000,(R0) ; 3207
000176	001342			BNE 4#
000200	010446			MOV R4,-(SP) ; 3208
000202	062716	000003		ADD #3,(SP)
000206	004737	000000G		JSR PC,SET.UPAR
000212	013700	000104'		MOV MAD1,R0 ; 3209
000216	013701	000000G		MOV CUOFF,R1
000222	006301			ASL R1
000224	063701	000000G		ADD CST.ADDR,R1
000230	111160	000016		MOVB (R1),16(R0)
000234	042760	177774	000016	BIC #177774,16(R0)
000242	013700	000106'		MOV MAD2,R0 ; 3210
000246	111160	000016		MOVB (R1),16(R0)
000252	042760	177774	000016	BIC #177774,16(R0)
000260	005726			TST (SP)+ ; 3195
000262	000207			RTS PC ; 3197
000264	005237	001160'		74: INC RAT.COUNT ; 3220
000270	023727	001160'	000144	CMP RAT.COUNT,#144 ; 3221
000276	002402			BLT 8#
000300	005037	001160'		CLR RAT.COUNT
000304	013701	000000G		84: MOV SWP.RAT,R1 ; 3222
000310	023701	001160'		CMP RAT.COUNT,R1
000314	002003			BGE 9#
000316	112700	000001		MOVB #1,R0 ; *,SELECT.RD 3224
000322	000401			BR 10# ; 3222
000324	105000			94: CLRB R0 ; SELECT.RD 3226

M10

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0336
Page 81
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (23)

000326	020127	000144	10#:	CMP	R1,#144	:	3228
000332	001002			BNE	11#		
000334	112700	000001		MOVB	#1,R0	;*,SELECT.RD	3230
000340	005701		11#:	TST	R1	:	3232
000342	001001			BNE	12#		
000344	105000			CLRB	R0	;SELECT.RD	3234
000346	105002		12#:	CLRB	R2	;MOD.COUNT	3243
000350	006000			ROR	R0	;SELECT.RD	3240
000352	103037			BCC	16#		
000354	012701	000003		MOV	#3,R1	;*,OFFSET	3244
000360	010100		13#:	MOV	R1,R0	;OFFSET,*	3246
000362	006300			ASL	R0		
000364	060300			ADD	R3,R0		
000366	032710	040000		BIT	#40000,(R0)		
000372	001417			BEQ	14#		
000374	032710	020000		BIT	#20000,(R0)	:	3247
000400	001414			BEQ	14#		
000402	132710	000034		BITB	#34,(R0)	:	3248
000406	001411			BEQ	14#		
000410	032710	010000		BIT	#10000,(R0)	:	3249
000414	001006			BNE	14#		
000416	005000			CLR	R0	:	3252
000420	150200			BISB	R2,R0	;MOD.COUNT,*	
000422	006300			ASL	R0		
000424	010160	000064'		MOV	R1,STORAGE(R0)	;OFFSET,*	
000430	105202			INCB	R2	;MOD.COUNT	3253
000432	062701	000007	14#:	ADD	#7,R1	;*,OFFSET	3244
000436	020127	000030		CMP	R1,#30	;OFFSET,*	
000442	003746			BLE	13#		
000444	105702		15#:	TSTB	R2	;MOD.COUNT	3258
000446	001436			BEQ	19#		
000450	000472			BR	22#	:	3261
000452	012701	000003	16#:	MOV	#3,R1	;*,OFFSET	3276
000456	010100		17#:	MOV	R1,R0	;OFFSET,*	3278
000460	006300			ASL	R0		
000462	060300			ADD	R3,R0		
000464	032710	040000		BIT	#40000,(R0)		
000470	001417			BEQ	18#		
000472	032710	020000		BIT	#20000,(R0)	:	3279
000476	001414			BEQ	18#		
000500	132710	000034		BITB	#34,(R0)	:	3280
000504	001011			BNE	18#		
000506	032710	010000		BIT	#10000,(R0)	:	3281
000512	001006			BNE	18#		
000514	005000			CLR	R0	:	3284
000516	150200			BISB	R2,R0	;MOD.COUNT,*	
000520	006300			ASL	R0		
000522	010160	000064'		MOV	R1,STORAGE(R0)	;OFFSET,*	
000526	105202			INCB	R2	;MOD.COUNT	3285
000530	062701	000007	18#:	ADD	#7,R1	;*,OFFSET	3276
000534	020127	000030		CMP	R1,#30	;OFFSET,*	
000540	003746			BLE	17#		
000542	000740			BR	15#	:	3291

N10

ZRQAM3 V01.2	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Bliss-16 V3-555 DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (23)	SEQ 0337 Page 82
000544 105002		19#:	CLRB R2	; MOD.COUNT 3308
000546 012701 000003			MOV #3,R1	; *,OFFSET 3309
000552 010100		20#:	MOV R1,R0	; OFFSET,* 3311
000554 006300			ASL R0	
000556 063700 000000G			ADD CST.ADDR,R0	
000562 032710 040000			BIT #40000,(R0)	
000566 001414			BEQ 21#	
000570 032710 020000			BIT #20000,(R0)	; 3312
000574 001411			BEQ 21#	
000576 032710 010000			BIT #10000,(R0)	; 3313
000602 001006			BNE 21#	
000604 005000			CLR R0	; 3316
000606 150200			BISB R2,R0	; MOD.COUNT,*
000610 006300			ASL R0	
000612 010160 000064'			MOV R1,STORAGE(R0)	; OFFSET,*
000616 105202			INCB R2	; MOD.COUNT 3317
000620 062701 000007		21#:	ADD #7,R1	; *,OFFSET 3309
000624 020127 000030			CMP R1,#30	; OFFSET,*
000630 003750			BLE 20#	
000632 105702			TSTB R2	; MOD.COUNT 3323
000634 001446			BEQ 23#	
000636 005000		22#:	CLR R0	; 3326
000640 150400			BISB R4,R0	; TBL.COUNT,*
000642 006300			ASL R0	
000644 016046 000000G			MOV RANDOM(R0),-(SP)	
000650 042716 100000			BIC #100000,(SP)	
000654 005046			CLR -(SP)	
000656 110216			MOVB R2,(SP)	; MOD.COUNT,*
000660 004737 000000G			JSR PC,BL#MOD	
000664 010003			MOV R0,R3	; *,X
000666 006300			ASL R0	; 3327
000670 016016 000064'			MOV STORAGE(R0),(SP)	
000674 004737 000000G			JSR PC,SET.UPAR	
000700 013700 000104'			MOV MAD1,R0	; 3328
000704 013701 000000G			MOV CUOFF,R1	
000710 006301			ASL R1	
000712 063701 000000G			ADD CST.ADDR,R1	
000716 111160 000016			MOVB (R1),16(R0)	
000722 042760 177774 000016			BIC #177774,16(R0)	
000730 013700 000106'			MOV MAD2,R0	; 3329
000734 111160 000016			MOVB (R1),16(R0)	
000740 042760 177774 000016			BIC #177774,16(R0)	
000746 022626			CMP (SP)+,(SP)+	; 3323
000750 000207			RTS PC	; 3325
000752 112737 000001 000000G		23#:	MOVB #1,EOP.FLAG	; 3333
000760 000207			RTS PC	; 3148

; Routine Size: 249 words, Routine Base: #CODE# + 7712
; Maximum stack depth per invocation: 8 words

```

: 3335 routine QIO_FUNC : novalue =
: 3336
: 3337 !*
: 3338 ! THIS ROUTINE IS CALLED BY QIO_GEN TO SELECT THE I/O FUNCTION (OPCODE)
: 3339 ! TO BE USED FOR THE CURRENT QIO OR QIO PAIR. THE FUNCTION IS DETERMINED
: 3340 ! BY THE FOLLWING ALGORITHM:
: 3341 !
: 3342 !     IF THE CHOSEN UNIT IS PROTECTED
: 3343 !     THEN
: 3344 !         FUNCTION = READ
: 3345 !     ELSE (UNPROTECTED)
: 3346 !         FUNCTION (WRITE OR READ) IS BASED ON A RANDOM
: 3347 !         NUMBER
: 3348 !
: 3349 ! IN ADDITION, IF THE OPERATOR SELECTED THE OPTION OF PERFORMING WRITE-
: 3350 ! COMPARES AT THE HOST, AND IF A "WRITE" FUNCTION WAS CHOSEN ABOVE FOR
: 3351 ! THE FIRST QIO, THEN A "READ" OPCODE IS LOADED INTO THE SECOND MSCP
: 3352 ! ENVELOPE. OTHERWISE, THE SECOND MSCP ENVELOPE IS RETURNED TO THE POOL.
: 3353 !
: 3354 ! THIS ROUTINE ALSO DECIDES WHEATHER IT IS TIME TO RUN THE DUP EXERCISER.
: 3355 ! THE EXERCISER WRITES 25 LBNS FOR EVERY 1 DBN. FOR INITIALIZATION
: 3356 ! REASONS THE MULTIPLE OF 25 LBNS CAN BE READ FOR 1 * MULTIPLE OF DBNS.
: 3357 ! DUP EXERCISER WRITES X BLOCKS PER PASS THRU THE DUP ROUTINE. THEREFORE
: 3358 ! THE RATIO OF 1 TO 25 MUST BE MULTIPLIED BY X FOR A RATIO OF 1 TO
: 3359 ! 25 * X. SO EVERY 25*X LBN'S READ OFF A WINCHESTER UNIT X DBN'S WILL
: 3360 ! BE READ IF ASKED BY THE USER. X is represented by the VARIABLE "DUPROUND".
: 3361 ! DUPROUND MAY BE CHANGED IN THE SOFTWARE QUESTION GIVEN AT THE START OF
: 3362 ! RUNNING THE EXERCISER PROGARM. THE DUP EXERCISER THEN REINITIALIZES
: 3363 ! THE CONTROLLER AND CONTINUES AS IF THE REGULAR MSCP EXERCISER WAS
: 3364 ! NEVER INTERRUPTED. NOTE THAT THE REINITIALIZATION PROCESSES TAKES
: 3365 ! A COUPLE OF SECONDS WHICH ONCE ADDED UP AFTER A COUPLE MILLION READS
: 3366 ! CAN PROVE TO BE QUITE TIMELY. THEREFORE I SUGGEST THAT IF A LARGE AMOUNT OF
: 3367 ! I/O TRANSFERS ARE TO BE DONE THE "DUPROUND" VARIABLE BE RAISED TO SPEED UP
: 3368 ! UP THE PROCESS.
: 3369 !
: 3370 ! IMPLICIT INPUTS:
: 3371 !     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 3372 !     CUOFF - CURRENT UNIT CST OFFSET
: 3373 !
: 3374 ! IMPLICIT OUTPUTS:
: 3375 !     THE OPCODE FIELD OF ONE OR BOTH MSCP ENVELOPES IS LOADED.
: 3376 !-
: 3377
: 3378 begin
: 3379
: 3380 local
: 3381     FUNC ; word; ! OPCODE (READ OR WRITE)
: 3382 !PRINTX (DER9);
: 3383     DUOFF = .CUOFF; !TEMPERARILY STORE VALUE INCASE ANOTHER COMMANDS LEFT IN QUEUE
: 3384     IF ((.CST_ADDR [.DUOFF + 6, D_COUNT] LEQ 0) AND ! IF MSCP FUNC CNT EQUAL TO 0
: 3385         (.CST_ADDR [.DUOFF, D_TYPE] NEQ RX_50) AND !IF WINCHESTER DISK
: 3386         (.CST_ADDR [.DUOFF + 5, NODUPMEDIA] NEQ 1)) !IF NODUPMEDIA FLAG BIT IS CLEAR
: 3387     THEN

```

```

: 3388 BEGIN
: 3389 PUT_PKT (.MX2); ! RETURN 2nd ENVELOPE TO POOL
: 3390 MX2 = -1; ! INDICATE FAILURE
: 3391 DUP (); ! DO DUP TEST
: 3392 !PRINTX (DBM111); ! PRINT MSCP EXERCISER REINIT
: 3393 CST_ADDR [.DUOFF + 6, D_COUNT] = (25 * .dupround); ! REINITIALIZE MSCP FUNC COUNTER
: 3394
: 3395 ! ***** THIS SECTION REINITIALIZES 2 ENVELOPES SO THE MSCP EXERCISER CAN
: 3396 ! ***** PROCEED AS BEFORE THE DUP EXERCISER STARTED *****
: 3397
: 3398 DUP_FLAGS = .DUP_FLAGS OR SWP_DINT; ! SET DUP INIT FLAG
: 3399 INIT_TEST (); ! THIS REINITIALIZES THE CONTROLLER FOR MSCP MODE
: 3400 DUP_FLAGS = .DUP_FLAGS AND (NOT SWP_DINT); ! CLEAR DUP INIT FLAG
: 3401
: 3402 MX2 = -1; ! ASSUME FAILURE IN SECURING 2ND ENVELOPE
: 3403 IF (MX1 = GET_PKT (.CCTLR)) LSS 0 ! TRY TO GET 1ST ENVELOPE.
: 3404 OR (.EOP_FLAG) ! IF FAILURE
: 3405 THEN RETURN; ! NO POINT IN CONTINUING
: 3406 IF (MX2 = GET_PKT (.CCTLR)) LSS 0 ! TRY TO GET 2ND ENVELOPE.
: 3407 OR (.EOP_FLAG) ! IF FAILURE
: 3408 THEN BEGIN
: 3409 PUT_PKT (.MX1); ! RETURN 1ST ENVELOPE TO POOL
: 3410 MX1 = -1; ! INDICATE FAILURE
: 3411 RETURN; ! DONE
: 3412 END;
: 3413
: 3414 MAD1 = MSCP_PKT + (.MX1 * PKT_LEN * 2); ! CALCULATE STARTING ADDRESSES
: 3415 MAD2 = MSCP_PKT + (.MX2 * PKT_LEN * 2); ! OF BOTH ENVELOPES
: 3416 GET_RANDOM (); ! GENERATE A SET OF RANDOM NUMBERS
: 3417 QIO_UNIT (); ! LOAD RANDOM UNIT NUMBER INTO ENVELOPES
: 3418 END;
: 3419
: 3420 !*****
: 3421 ! START OF ROUTINE MSCP
: 3422 !*****
: 3423
: 3424 CST_ADDR [.CUOFF + 6, D_COUNT] = .CST_ADDR [.CUOFF + 6, D_COUNT] - 1; ! DECREMENT MSCP FUNC COUNTER
: 3425
: 3426 MAD2 [OPCODE] = 0; ! ASSUME 2ND PACKET NOT NEEDED
: 3427
: 3428 if .CST_ADDR [.CUOFF, D_PROT] eq1 PROTECTED ! IF UNIT IS PROTECTED
: 3429 then
: 3430 FUNC = OP_RD ! SET FUNCTION TO READ
: 3431 else
: 3432
: 3433 if (.RANDOM [1] and 1) ! USE 2ND RANDOM NUMBER TO SELECT
: 3434 then
: 3435 FUNC = OP_RD ! READ
: 3436 else
: 3437 FUNC = OP_WRT; ! WRITE
: 3438
: 3439 if (MAD1 [OPCODE] = .FUNC) eq1 OP_WRT ! LOAD CHOSEN OPCODE. IF WRITE
: 3440 then

```

```

: 3441      begin
: 3442      MAD1 [CMD_TYPE] = NON_SEQ_CMD;      ! NON-SEQUENTIAL COMMAND
: 3443
: 3444      if BIT_TST (SWP_FLAGS, SWF_CWC)      ! IF CONTROLLER DOES WRITE-COMPARES
: 3445      then
: 3446          MAD1 [MODIFY] = MD_CMP;          ! ADD COMPARE MODIFIER
: 3447
: 3448      if BIT_TST (SWP_FLAGS, SWF_HWC)      ! IF HOST DOES WRITE-COMPARES
: 3449      then
: 3450          begin
: 3451              MAD2 [OPCODE] = OP_RD;        ! SET READ OPCODE INTO 2ND MSCP PACKET
: 3452              MAD2 [CMD_TYPE] = NON_SEQ_CMD; ! NON-SEQUENTIAL COMMAND
: 3453          end;
: 3454
: 3455      end
: 3456      else
: 3457          begin
: 3458              MAD1 [CMD_TYPE] = NON_SEQ_CMD; ! NON-SEQUENTIAL COMMAND
: 3459
: 3460              if BIT_TST (SWP_FLAGS, SWF_CRC) ! IF CONTROLLER DOES READ-COMPARES - FUNCTION IS READ
: 3461              then
: 3462                  MAD1 [MODIFY] = MD_CMP;    ! ADD COMPARE MODIFIER
: 3463
: 3464              end;
: 3465
: 3466          if .MAD2 [OPCODE] eq 0            ! IF NO OPCODE IN 2ND PACKET
: 3467          then
: 3468              begin
: 3469                  PUT_PKT (.MX2);           ! RETURN 2ND PACKET TO POOL
: 3470                  MX2 = -1;               ! MARK IT UNUSED
: 3471              end;
: 3472
: 3473      end;                                ! ROUTINE QIO_FUNC

```

Address	Offset	Label	Instruction	Comment	Address
000000	004137	000000G	.SBTTL	QIO.FUNC MULTI-DRIVE TEST ROUTINES	
			QIO.FUNC:		
000004	013737	000000G 001150'	JSR	R1, #SAVE3	3335
000012	013702	000000G	MOV	CUOFF, DUOFF	3383
000016	013701	001150'	MOV	CST, ADDR, R2	3384
000022	010100		MOV	DUOFF, R1	
000024	006300		MOV	R1, R0	
000026	060200		ASL	R0	
000030	005760	000014	ADD	R2, R0	
000034	003146		TST	14(R0)	
000036	010100		BGT	4#	
000040	006300		MOV	R1, R0	3385
000042	060200		ASL	R0	
000044	132710	000034	ADD	R2, R0	
000050	001540		BITB	#34, (R0)	
000052	010100		BEQ	4#	
000054	006300		MOV	R1, R0	3386
			ASL	R0	

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0341
Page 86
VAX-11 B11es-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (24)

000056	060200		ADD	R2,R0		
000060	005760	000012	TST	12(R0)		
000064	100532		BMI	4#		
000066	013746	000102'	MOV	MX2,-(SP)	:	
000072	004737	000000G	JSR	PC,PUT.PKT	:	3389
000076	012737	177777 000102'	MOV	#-1,MX2	:	
000104	004737	000000V	JSR	PC,DUP	:	3390
000110	013701	001150'	MOV	DUOFF,R1	:	3391
000114	006301		ASL	R1	:	3393
000116	063701	000000G	ADD	CST.ADDR,R1		
000122	013716	000000G	MOV	DUPROUND,(SP)		
000126	012746	000031	MOV	#31,-(SP)		
000132	004737	000000G	JSR	PC,BL#MUL		
000136	010061	000014	MOV	RO,14(R1)		
000142	052737	000002 000000G	BIS	#2,DUP.FLAGS	:	3398
000150	004737	000270'	JSR	PC,INIT.TEST	:	3399
000154	042737	000002 000000G	BIC	#2,DUP.FLAGS	:	3400
000162	012737	177777 000102'	MOV	#-1,MX2	:	3402
000170	013716	000000G	MOV	CCTRL,(SP)	:	3403
000174	004737	000000G	JSR	PC,GET.PKT		
000200	010037	000100'	MOV	RO,MX1		
000204	002426		BLT	2#		
000206	132737	000001 000000G	BITB	#1,EOP.FLAG	:	3404
000214	001022		BNE	2#	:	3335
000216	013716	000000G	MOV	CCTRL,(SP)	:	3406
000222	004737	000000G	JSR	PC,GET.PKT		
000226	010037	000102'	MOV	RO,MX2		
000232	002404		BLT	1#		
000234	132737	000001 000000G	BITB	#1,EOP.FLAG	:	3407
000242	001411		BEQ	3#		
000244	013716	000100'	MOV	MX1,(SP)	:	
000250	004737	000000G	JSR	PC,PUT.PKT	:	3409
000254	012737	177777 000100'	MOV	#-1,MX1	:	
000262	022626		CMP	(SP)+,(SP)+	:	3410
000264	000207		RTS	PC	:	3408
000266	013716	000100'	MOV	MX1,(SP)	:	3414
000272	012746	000104	MOV	#104,-(SP)		
000276	004737	000000G	JSR	PC,BL#MUL		
000302	062700	000000G	ADD	#MSCP.PKT,RO		
000306	010037	000104'	MOV	RO,MAD1		
000312	013716	000102'	MOV	MX2,(SP)	:	
000316	012746	000104	MOV	#104,-(SP)		3415
000322	004737	000000G	JSR	PC,BL#MUL		
000326	062700	000000G	ADD	#MSCP.PKT,RO		
000332	010037	000106'	MOV	RO,MAD2		
000336	004737	007602'	JSR	PC,GET.RANDOM	:	3416
000342	004737	007712'	JSR	PC,QIO.UNIT	:	3417
000346	062706	000010	ADD	#10,SP	:	3388
000352	013700	000000G	MOV	CUOFF,RO	:	3424
000356	006300		ASL	RO		
000360	063700	000000G	ADD	CST.ADDR,RO		
000364	005360	000014	DEC	14(R0)		
000370	013700	000106'	MOV	MAD2,RO	:	3426

F11

ZRGAM3 V01.2	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Bliss-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRGACO.BL2;161 (24)	SEQ 0342 Page 87
000374	012703	000022	MOV #22,R3	
000400	060003		ADD R0,R3	
000402	105013		CLRB (R3)	
000404	013700	000000G	MOV CUOFF,R0	
000410	006300		ASL R0	3428
000412	063700	000000G	ADD CST,ADDR,R0	
000416	032710	100000	BIT #100000,(R0)	
000422	001404		BEQ 5#	3430
000424	032737	000001 000002G	BIT #1,RANDOM+2	3433
000432	001403		BEQ 6#	
000434	012701	000041	5#: MOV #41,R1	3435
000440	000402		BR 7#	3433
000442	012701	000042	6#: MOV #42,R1	3437
000446	013700	000104'	7#: MOV MAD1,R0	3439
000452	013702	000000G	MOV SWP,FLAGS,R2	3444
000456	110160	000022	MOVB R1,22(R0)	3439
000462	020127	000042	CMP R1,#42	
000466	001024		BNE 9#	
000470	012760	000002 000004	MOV #2,4(R0)	3442
000476	032702	000020	BIT #20,R2	3444
000502	001403		BEQ 8#	
000504	012760	040000 000024	MOV #40000,24(R0)	3446
000512	032702	000040	8#: BIT #40,R2	3448
000516	001421		BEQ 10#	
000520	112713	000041	MOVB #41,(R3)	3451
000524	013701	000106'	MOV MAD2,R1	3452
000530	012761	000002 000004	MOV #2,4(R1)	
000536	000411		BR 10#	3439
000540	012760	000002 000004	9#: MOV #2,4(R0)	3458
000546	032702	000004	BIT #4,R2	3460
000552	001403		BEQ 10#	
000554	012760	040000 000024	MOV #40000,24(R0)	3462
000562	105713		10#: TSTB (R3)	3466
000564	001010		BNE 11#	
000566	013746	000102'	MOV MX2,-(SP)	3469
000572	004737	000000G	JSR PC,PUT.PKT	
000576	012737	177777 000102'	MOV #-1,MX2	3470
000604	005726		TST (SP)+	3468
000606	000207		11#: RTS PC	3335

; Routine Size: 196 words, Routine Base: \$CODE\$ + 10674
 ; Maximum stack depth per invocation: 9 words

```

: 3474 !†
: 3475
: 3476 ROUTINE DUP : NOVALUE =
: 3477 !*
: 3478 ! THIS ROUTINE IS CALLED BY QIO_FUNC AFTER 25 * "DUPROUND" RD/WTS .
: 3479 ! THIS EXERCISER WAS PLACED IN THE MIDDLE OF THE MSCP EXERCISER SO A
: 3480 ! COMMON INITIALIZATION AND OTHER ROUTINES COULD BE USED.
: 3481 ! THE DUP EXERCISER WILL RUN A READ ONLY OR A WRITE/READ COMPARE
: 3482 ! TO THE DIAGNOSTIC BLOCKS OR DBN'S. IT WILL RECORD
: 3483 ! THE STATICS IN THE TALLY TABLES.
: 3484 !
: 3485 ! THE PROGRAM USES CONTROLLER LOCAL PROGRAMS TO WRITE AND READ THE DBN'S.
: 3486 ! WHEN WRITTING TO THE DBN'S A ONE WORD PATTERN WILL BE SELECTED
: 3487 ! AND COPY TO A 256 WORD BLOCK. THE ROUTINE WILL WRITE TO "DUPROUND" AMOUNT
: 3488 ! OF DBN SEQUENTIAL BLOCKS. IF A THERE ARE CONTROLLER LOCAL PROGRAMS AND
: 3489 ! IF THE USER SO DESIRES A WRITE AND READ OF EACH BLOCK AND A COMPARISON
: 3490 ! TO THE DATA PATTERN WILL BE GIVEN TO TEST FOR FAULTY DBN'S. IF THE USER
: 3491 ! DOES NOT WANT TO WRITE TO THE DBN'S ONLY A READ WILL BE GIVEN AND NO
: 3492 ! COMPARISON WILL BE DONE. THE BAD BLOCKS FOUND IN THE COMPARISON TEST
: 3493 ! WILL NOT BE LISTED IN THE RCT TABLES.
: 3494 !
: 3495 ! AFTER THE EXERCISER HAS EXAMINE "DUPROUND" AMOUNT OF BLOCKS IT WILL
: 3496 ! REINITIATE THE ENVELOPES SO THAT THE MSCP EXERCISER MAY CONTINUE AS
: 3497 ! BEFORE.
: 3498 !
: 3499 ! IMPLICIT INPUTS:
: 3500 ! CCTLN - CURRENT CONTROLLER NUMBER
: 3501 ! CST_ADDRS - CONTAINS THE CURRENT CONTROLLER STATUS TABLE
: 3502 ! CUOFF - CURRENT OFFSET IN CST TABLE FOR PARTICULAR DRIVE
: 3503 !
: 3504 ! IMPLICIT OUTPUTS:
: 3505 ! S_PATTERN - PATTERN BEING WRITTEN TO AND READ FROM DBN'S
: 3506 !-
: 3507 BEGIN
: 3508 OWN
: 3509 TEMP : WORD;
: 3510
: 3511 !PRINTX (DBM110);
: 3512 !PRINTX (DER10);
: 3513
: 3514 until (.CRN_LOW eqv .RP_ADDR [CRF_LO]) or ! TO ENSURE THAT ALL RETURN MESSAGES HAVE BEEN PROCESSED
: 3515 (.EOP_FLAG eqv true) do ! Make sure all MSCP commands are completed
: 3516 begin
: 3517 BREAK; ! BREAK FOR ACT
: 3518 PROC_RETPKT(); ! PROCESS RETURN PACKET TO SEE IF OK FOR DUP
: 3519 RP_INDX = .RP_INDX + 1; ! INCREMENT RP_INDX
: 3520 if .RP_INDX geq RP_CNT then (RP_INDX = 0); ! MAKE SURE THE COUNTER DOES NOT GET TO BIG
: 3521 RP_ADDR = RETPKT + (.RP_INDX * RP_LEN + 2); ! CALCULATE RETPKT ADDRESS
: 3522 end;
: 3523
: 3524
: 3525 S_PATTERN = .RANDOM [1]; !OTHER UNIT VARIABLES
: 3526

```

```

ZRGAM3      RD/RX EXERCISER      14-Dec-1983 16:12:07      VAX-11 B1100-16 V3-555      SEQ 0344
V01.2      MULTI-DRIVE TEST ROUTINES      14-Dec-1983 16:12:00      DISK#USER2:[DIETZ.RDRX]ZRGACO.BL2;161 (25)      Page 89

:          3527 IF (.CST_ADDR [.DUOFF + 5, D_DBN] + .dupround) GEQ 144 ! TEST TO SEE IF NEXT DBN'S TO LARGE
:          3528 THEN (CST_ADDR [.DUOFF + 5, D_DBN] = 0); ! CIRCLE AROUND IF DBN TO LARGE
:          3529
:          3530 DUPIDLE (); ! DO A GET DUST STATUS TO FIND IF LOCAL DUP MEDIA
:          3531 IF .CST_ADDR [.DUOFF + 5, NODUPMEDIA] EQL 1 THEN RETURN; ! IF DUP LOCAL MEDIA NOT THERE THEN RETURN
:          3532
:          3533 TEMP = .CST_ADDR [.DUOFF + 5, D_DBN];
:          3534 INCR DBNCNT FROM (.TEMP + 1) TO (.TEMP + .dupround) DO ! INCREMENT FROM RELATIVE DBN TO DBN + dupro
und
:          3535 BEGIN
:          3536 IF .CST_ADDR [.DUOFF + 5, DUPWRITE] ! IF WRITE FLAG SET IN CST TABLE THEN WRITE
DBN'S
:          3537 THEN
:          3538 BEGIN
:          3539 DUPIDLE (); ! MAKE SURE THE CONTROLLER IS IN AN IDLE STA
TE
:          3540 DUPWRTDBN (); ! CALL ROUTINE TO HANDLE WRITTING ROUTINES
:          3541 END;
:          3542
:          3543 DUPIDLE (); ! MAKE SURE CONTROLLER IN IDLE STATE
:          3544 DUPREDDBN (); ! CALL ROUTINE TO HANDLE READING DBN'S
:          3545
:          3546 CST_ADDR [.DUOFF + 5, D_DBN] = .CST_ADDR [.DUOFF + 5, D_DBN] + 1; ! INCREMENT RELATIVE DBN COUNTER
:          3547
:          3548 IF .CST_ADDR [.DUOFF + 5, DUPERROR] EQL 1 ! ERROR IN DUP REINITIALIZE
:          3549 THEN RETURN; ! AND RETURN
:          3550 END;
:          3551 END;
    
```

```

001162      .PSECT  $GGG$,  R0
001162      TEMP:  .BLKW  1

011504      .SBTTL  DUP MULTI-DRIVE TEST ROUTINES
           .PSECT  $CODE$,  R0

000000 004137 000000G      DUP:  JSR      R1, $SAVE3      ;          3476
000004 013700 000000G      1$:  MOV      RP, ADDR, R0      ;          3514
000010 023760 000000G 000004      CMP      CRN, LOW, 4(R0)
000016 001433      BEQ      3$
000020 123727 000000G 000001      CMPB    EOP, FLAG, #1      ;          3515
000026 001427      BEQ      3$
000030 104422      TRAP    22      ;          3516
000032 004737 000000V      JSR      PC, PROC, RETPKT      ;          3518
000036 005237 000000G      INC      RP, INDX      ;          3519
000042 023727 000000G 000004      CMP      RP, INDX, #4      ;          3520
000050 002402      BLT      2$
000052 005037 000000G      CLR      RP, INDX
000056 013746 000000G      2$:  MOV      RP, INDX, -(SP)      ;          3521
000062 012746 0000060      MOV      #60, -(SP)
000066 004737 000000G      JSR      PC, BL $MUL
000072 062700 000000G      ADD     $RETPKT, R0
000076 010037 000000G      MOV     R0, RP, ADDR
000102 022626      CMP     (SP)+, (SP)+      ;          3516
    
```

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Blues-16 V3-555	SEQ 0345	
V01.2	MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:00	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (25)	Page 90	
000104	000737			3514	
000106	013737	000002G 000000G	34:	MOV RANDOM+2,S.PATTERN	3525
000114	013700	001150'		MOV DUOFF,RO	3527
000120	006300			ASL RO	
000122	063700	000000G		ADD CST.ADDR,RO	
000126	005001			CLR R1	
000130	156001	000012		BISB 12(RO),R1	
000134	063701	000000G		ADD DUPROUND,R1	
000140	020127	000220		CMP R1,#220	
000144	002402			BLT 44	
000146	105060	000012		CLRB 12(RO)	3528
000152	004737	000000V	44:	JSR PC,DUPIDLE	3530
000156	013700	001150'		MOV DUOFF,RO	3531
000162	006300			ASL RO	
000164	063700	000000G		ADD CST.ADDR,RO	
000170	005760	000012		TST 12(RO)	
000174	100456			BMI 84	
000176	116037	000012 001162'		MOVB 12(RO),TEMP	3533
000204	105037	001163'		CLRB TEMP+1	
000210	013702	001162'		MOV TEMP,R2	3534
000214	063702	000000G		ADD DUPROUND,R2	
000220	013700	001150'		MOV DUOFF,RO	3536
000224	006300			ASL RO	
000226	063700	000000G		ADD CST.ADDR,RO	
000232	010003			MOV RO,R3	
000234	062703	000012		ADD #12,R3	
000240	013701	001162'		MOV TEMP,R1	3534
000244	000427			BR 74 ; *,DBNCNT	
000246	032713	010000	54:	BIT #10000,(R3)	3536
000252	001404			BEQ 64	
000254	004737	000000V		JSR PC,DUPIDLE	3539
000260	004737	000000V		JSR PC,DUPWRTDBN	3540
000264	004737	000000V	64:	JSR PC,DUPIDLE	3543
000270	004737	000000V		JSR PC,DUPREDDBN	3544
000274	013700	001150'		MOV DUOFF,RO	3546
000300	006300			ASL RO	
000302	063700	000000G		ADD CST.ADDR,RO	
000306	010003			MOV RO,R3	
000310	062703	000012		ADD #12,R3	
000314	105213			INCB (R3)	
000316	032713	040000		BIT #40000,(R3)	3548
000322	001003			BNE 84	3549
000324	005201		74:	INC R1 ; DBNCNT	3534
000326	020102			CMP R1,R2 ; DBNCNT,*	
000330	003746			BLE 54	
000332	000207		84:	RTS PC	3476

; Routine Size: 110 words, Routine Base: #CODE# + 11504
 ; Maximum stack depth per invocation: 7 words

; 3552

```

: 3553 ROUTINE DUPWRTOBN : NOVALUE =
: 3554
: 3555 !*
: 3556 ! THIS ROUTINE IS CALLED BY DUP ROUTINE TO USE THE CONTROLLER LOCAL PROGRAM
: 3557 ! "WRTOBN". TO USE THE PROGRAM THE OPTIONAL DUP SUB-PROTOCOL IS USED TO
: 3558 ! COMMUNICATE WITH THE CONTROLLER. THE PROGRAM WRITES TO A DIAGNOSTIC BLOCK (DBN)
: 3559 ! THE WORD IN "S_PATTERN" IS WRITTEN TO THE 256 WORDS IN THE DBN. IF AN ERROR OCCURS
: 3560 ! WHILE RUNNING THE CONTROLLER LOCAL PROGRAM THE ERROR IS USUALLY REPORTED IN THE
: 3561 ! DUP BUFFER. (EX. ILLEGAL UNIT NUMBER, ILLEGAL BLK #, DEVICE ERROR, ZERO LENGHT MSG)
: 3562 !
: 3563 !     IMPLICIT INPUTS:
: 3564 !     CST_ADDR  - CONTAINS THE CURRENT CONTROLLER STATUS TABLE
: 3565 !     DUOFF    - CURRENT OFFSET IN CST TABLE FOR PARTICULAR DRIVE
: 3566 !     S_PATTERN - CONTAINS PATTERN WORD!-
: 3567 BEGIN
: 3568 !PRINTX (DER11);
: 3569 T_ADDR [T_DBN_WT] = .T_ADDR [T_DBN_WT] + 1;      ! INCREMENT # OF WRITES GIVEN
: 3570
: 3571 MSCP_PKT [.MX1, MSGLEN] = SZ_ELP;                ! PACKET SIZE                                EXECUTE LOCAL PROGRAM WRT DB
:
: 3572 MSCP_PKT [.MX1, OPCODE] = OP_ELP;                ! OPCODE = EXECUTE LOCAL PROGRAM
: 3573 MSCP_PKT [.MX1, L1] = %ascii'W';                 ! FILL IN PROGRAM NAME WITH ASCII LETTERS
: 3574 MSCP_PKT [.MX1, L2] = %ascii'R';
: 3575 MSCP_PKT [.MX1, L3] = %ascii'T';
: 3576 MSCP_PKT [.MX1, L4] = %ascii'D';
: 3577 MSCP_PKT [.MX1, L5] = %ascii'B';
: 3578 MSCP_PKT [.MX1, L6] = %ascii'N';
: 3579 MSCP_PKT [.MX1, MODIFY] = 1;                    ! STANDALONE MODIFIER
: 3580 MSCP_PKT [.MX1, MSGTYP] = IMM_CMD;              ! CALL IT IMMEDIATE
: 3581 DUPCOMMAND ();                                  ! SENDS AND RECEIVES THE COMMAND
: 3582
: 3583 IF .CST_ADDR [.DUOFF + 5, DUPERROR] EQL 1        !status error
: 3584 THEN RETURN;                                    ! AND RETURN
: 3585
: 3586 DO (MX1 = GET_PKT (.CCTLR))
: 3587 UNTIL (.MX1 GEQ 0);                             ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3588
: 3589 MSCP_PKT [.MX1, MSGLEN] = SZ_REC;                ! PACKET SIZE                                RECIEVE DATA
: 3590 MSCP_PKT [.MX1, OPCODE] = OP_RCD;              ! OPCODE = RECEIVE DATA
: 3591 MSCP_PKT [.MX1, BC_LO] = 80;                    ! BYTE COUNT TO BE TRANSFERED EQUALS 2 ****see pg 26 of DUP s
:
: pec
: 3592 MSCP_PKT [.MX1, BUF_0] = DUPPKT;               ! LOAD DESCRIBTOR BUFFER
: 3593 MSCP_PKT [.MX1, MODIFY] = 0;
: 3594 MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD;              ! CALL IT sequential
: 3595 DUPCOMMAND ();                                  ! SENDS AND RECEIVES THE COMMAND
: 3596
: 3597 IF (.CST_ADDR [.DUOFF + 5, DUPERROR] EQL 1) OR  !status error
: 3598 (.DUPPKT [DUPTYPE] NEQU 1) OR                  !dup type error
: 3599 (.DUPPKT [DUPMSG] NEQU 6)
: 3600 THEN
: 3601     (HARD_ERROR ());
: 3602     CST_ADDR [.DUOFF + 5, DUPERROR] = 1;        ! SET FLAG
: 3603     RETURN;);                                    ! NO POINT IN CONTINUING
: 3604
: 3605 DO (MX1 = GET_PKT (.CCTLR))

```

```

:      3606      UNTIL (.MX1 GEQ 0);      ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
:      3607
:      3608      MSCP_PKT [.MX1, MSGLEN] = SZ_SEN;      ! PACKET SIZE      SEND DATA
:      3609      MSCP_PKT [.MX1, OPCODE] = OP_SDD;      ! OPCODE = SEND DATA
:      3610      MSCP_PKT [.MX1, BC_LO] = 6;      ! BYTE COUNT TO BE TRANSFERED EQUALS 6
:      3611      MSCP_PKT [.MX1, BUF_0] = DUPPKT;      ! LOAD DESCRIPTOR BUFFER
:      3612      DUPPKT [DUPBF0] = .CST_ADDR [.DUOFF, D_DISK_NUM]; !LOAD UNIT NUMBER (RDRX)
:      3613      DUPPKT [DUPBF1] = .CST_ADDR [.DUOFF + 5, D_DBN]; ! LOAD DBN NUMBER
:      3614      DUPPKT [DUPBF2] = .S_PATTERN;      ! LOAD PATTERN
:      3615      MSCP_PKT [.MX1, MODIFY] = 0;
:      3616      MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD;      ! CALL IT sequential
:      3617      DUPCOMMAND ();      ! SENDS AND RECEIVES THE COMMAND
:      3618
:      3619      IF .CST_ADDR [.DUOFF + 5, DUPERROR] EQL 1      ! status error
:      3620      THEN RETURN;
:      3621
:      3622      DO (MX1 = GET_PKT (.CCTLR))
:      3623      UNTIL (.MX1 GEQ 0);      ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
:      3624
:      3625      MSCP_PKT [.MX1, MSGLEN] = SZ_REC;      ! PACKET SIZE      RECEIVE DATA
:      3626      MSCP_PKT [.MX1, OPCODE] = OP_RCD;      ! OPCODE = RECEIVE DATA
:      3627      MSCP_PKT [.MX1, BC_LO] = 4;      ! BYTE COUNT TO BE TRANSFERED EQUALS 4
:      3628      MSCP_PKT [.MX1, BUF_0] = DUPPKT;      ! LOAD DESCRIPTOR BUFFER
:      3629      MSCP_PKT [.MX1, MODIFY] = 0;
:      3630      MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD;      ! CALL IT sequential
:      3631      DUPCOMMAND ();      ! SENDS AND RECEIVES THE COMMAND
:      3632
:      3633      IF (.CST_ADDR [.DUOFF + 5, DUPERROR] EQL 1) or      !status error
:      3634      (.DUPPKT [DUPTYPE] NEQU 3) or      !dup type error
:      3635      (.DUPPKT [DUPMSG] NEQU 3) or
:      3636      (.DUPPKT [DUPBF1] NEQU 0)      !non successful write code
:      3637      THEN
:      3638      (HARD_ERROR ();
:      3639      CST_ADDR [.DUOFF + 5, DUPERROR] = 1;      ! SET FLAG
:      3640      RETURN;);      ! NO POINT IN CONTINUING
:      3641
:      3642      DO (MX1 = GET_PKT (.CCTLR))
:      3643      UNTIL (.MX1 GEQ 0);      ! TRY TO GET AN ENVELOPE.
:      3644
:      3645      T_ADDR [T_BLK_WT] = .T_ADDR [T_BLK_WT] + 1;      !INCREMENT COUNTER IF A SUCCESS
:      3646
:      3647      END;

```

000000	010146		.SBTTL	DUPWRTDBN MULTI-DRIVE TEST ROUTINES	
			DUPWRTDBN;		
000002	013700	000000G	MOV	R1, -(SP)	3553
000006	005260	000052	MOV	T_ADDR, R0	3569
			INC	52(R0)	
000012	013746	000100'	MOV	MX1, -(SP)	3571
000016	012746	000104	MOV	#104, -(SP)	
000022	004737	000000G	JSR	PC, BL#MUL	
000026	012760	000022 000006G	MOV	#22, MSCP.PKT+6(R0)	

000034	112760	000003	000022G		MOVB	#3,MSCP.PKT+22(RO)	:	3572
000042	112760	000127	000026G		MOVB	#127,MSCP.PKT+26(RO)	:	3573
000050	112760	000122	000027G		MOVB	#122,MSCP.PKT+27(RO)	:	3574
000056	112760	000124	000030G		MOVB	#124,MSCP.PKT+30(RO)	:	3575
000064	112760	000104	000031G		MOVB	#104,MSCP.PKT+31(RO)	:	3576
000072	112760	000102	000032G		MOVB	#102,MSCP.PKT+32(RO)	:	3577
000100	112760	000116	000033G		MOVB	#116,MSCP.PKT+33(RO)	:	3578
000106	012760	000001	000024G		MOV	#1,MSCP.PKT+24(RO)	:	3579
000114	142760	000360	000010G		BICB	#360,MSCP.PKT+10(RO)	:	3580
000122	004737	000000V			JSR	PC,DUPCOMMAND	:	3581
000126	013700	001150'			MOV	DUOFF,RO	:	3583
000132	006300				ASL	RO	:	
000134	063700	000000G			ADD	CST.ADDR,RO	:	
000140	032760	040000	000012		BIT	#40000,12(RO)	:	
000146	001402				BEQ	1#	:	
000150	022626				CMP	(SP)+,(SP)+	:	3553
000152	000505				BR	3#	:	3584
000154	013716	000000G		1#:	MOV	CCTLR,(SP)	:	3586
000160	004737	000000G			JSR	PC.GET.PKT	:	
000164	010037	000100'			MOV	RO,MX1	:	
000170	002771				BLT	1#	:	3587
000172	010016				MOV	RO,(SP)	:	3589
000174	012746	000104			MOV	#104,-(SP)	:	
000200	004737	000000G			JSR	PC,BL#MUL	:	
000204	012760	000034	000006G		MOV	#34,MSCP.PKT+6(RO)	:	
000212	112760	000005	000022G		MOVB	#5,MSCP.PKT+22(RO)	:	3590
000220	012760	000120	000026G		MOV	#120,MSCP.PKT+26(RO)	:	3591
000226	012760	000000G	000032G		MOV	#DUPPKT,MSCP.PKT+32(RO)	:	3592
000234	005060	000024G			CLR	MSCP.PKT+24(RO)	:	3593
000240	142760	000360	000010G		BICB	#360,MSCP.PKT+10(RO)	:	3594
000246	152760	000020	000010G		BISB	#20,MSCP.PKT+10(RO)	:	
000254	004737	000000V			JSR	PC,DUPCOMMAND	:	3595
000260	013700	001150'			MOV	DUOFF,RO	:	3597
000264	006300				ASL	RO	:	
000266	063700	000000G			ADD	CST.ADDR,RO	:	
000272	032760	040000	000012		BIT	#40000,12(RO)	:	
000300	001016				BNE	2#	:	
000302	013700	000000G			MOV	DUPPKT,RO	:	3598
000306	042700	007777			BIC	#7777,RO	:	
000312	020027	010000			CMP	RO,#10000	:	
000316	001007				BNE	2#	:	
000320	013700	000000G			MOV	DUPPKT,RO	:	3599
000324	042700	170000			BIC	#170000,RO	:	
000330	020027	000006			CMP	RO,#6	:	
000334	001415				BEQ	4#	:	
000336	004737	000000V		2#:	JSR	PC,HARD.ERROR	:	3601
000342	013700	001150'			MOV	DUOFF,RO	:	3602
000346	006300				ASL	RO	:	
000350	063700	000000G			ADD	CST.ADDR,RO	:	
000354	052760	040000	000012		BIS	#40000,12(RO)	:	
000362	062706	000006			ADD	#6,SP	:	3597
000366	000504			3#:	BR	5#	:	3601
000370	013716	000000G		4#:	MOV	CCTLR,(SP)	:	3605

M11

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0349
Page 94
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (26)

000374	004737	000000G			JSR	PC,GET.PKT		
000400	010037	000100'			MOV	R0,MX1		
000404	002771				BLT	4#	:	3606
000406	010016				MOV	R0,(SP)	:	3608
000410	012746	000104			MOV	#104,-(SP)		
000414	004737	000000G			JSR	PC,BL#MUL		
000420	012760	000034	000006G		MOV	#34,MSCP.PKT+6(R0)		
000426	112760	000004	000022G		MOVB	#4,MSCP.PKT+22(R0)	:	3609
000434	012760	000006	000026G		MOV	#6,MSCP.PKT+26(R0)	:	3610
000442	012760	000000G	000032G		MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3611
000450	013701	001150'			MOV	DUOFF,R1	:	3612
000454	006301				ASL	R1		
000456	063701	000000G			ADD	CST.ADDR,R1		
000462	111137	000000G			MOVB	(R1),DUPPKT		
000466	042737	177774	000000G		BIC	#177774,DUPPKT		
000474	013701	001150'			MOV	DUOFF,R1	:	3613
000500	006301				ASL	R1		
000502	063701	000000G			ADD	CST.ADDR,R1		
000506	116137	000012	000002G		MOVB	12(R1),DUPPKT+2		
000514	105037	000003G			CLRB	DUPPKT+3		
000520	013737	000000G	000004G		MOV	S.PATTERN,DUPPKT+4	:	3614
000526	005060	000024G			CLR	MSCP.PKT+24(R0)	:	3615
000532	142760	000360	000010G		BICB	#360,MSCP.PKT+10(R0)	:	3616
000540	152760	000020	000010G		BISB	#20,MSCP.PKT+10(R0)		
000546	004737	000000V			JSR	PC,DUPCOMMAND	:	3617
000552	013700	001150'			MOV	DUOFF,R0	:	3619
000556	006300				ASL	R0		
000560	063700	000000G			ADD	CST.ADDR,R0		
000564	032760	040000	000012		BIT	#40000,12(R0)		
000572	001403				BEQ	6#		
000574	062706	000010			ADD	#10,SP	:	3553
000600	000524			5#:	BR	10#	:	3620
000602	013716	000000G		6#:	MOV	CCTLR,(SP)	:	3622
000606	004737	000000G			JSR	PC,GET.PKT		
000612	010037	000100'			MOV	R0,MX1		
000616	002771				BLT	6#	:	3623
000620	010016				MOV	R0,(SP)	:	3625
000622	012746	000104			MOV	#104,-(SP)		
000626	004737	000000G			JSR	PC,BL#MUL		
000632	012760	000034	000006G		MOV	#34,MSCP.PKT+6(R0)		
000640	112760	000005	000022G		MOVB	#5,MSCP.PKT+22(R0)	:	3626
000646	012760	000004	000026G		MOV	#4,MSCP.PKT+26(R0)	:	3627
000654	012760	000000G	000032G		MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3628
000662	005060	000024G			CLR	MSCP.PKT+24(R0)	:	3629
000666	142760	000360	000010G		BICB	#360,MSCP.PKT+10(R0)	:	3630
000674	152760	000020	000010G		BISB	#20,MSCP.PKT+10(R0)		
000702	004737	000000V			JSR	PC,DUPCOMMAND	:	3631
000706	013700	001150'			MOV	DUOFF,R0	:	3633
000712	006300				ASL	R0		
000714	063700	000000G			ADD	CST.ADDR,R0		
000720	032760	040000	000012		BIT	#40000,12(R0)		
000726	001021				BNE	7#		
000730	013700	000000G			MOV	DUPPKT,R0	:	3634

N11

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0350
Page 95
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (26)

000734	042700	007777		BIC	#7777,R0		
000740	020027	030000		CMP	R0,#30000		
000744	001012			BNE	7#		
000746	013700	000000G		MOV	DUPPKT,R0	;	3635
000752	042700	170000		BIC	#170000,R0		
000756	020027	000003		CMP	R0,#3		
000762	001003			BNE	7#		
000764	005737	000002G		TST	DUPPKT+2	;	3636
000770	001413			BEQ	8#		
000772	004737	000000V	7#:	JSR	PC,HARD.ERROR	;	3638
000776	013700	001150'		MOV	DUOFF,R0	;	3639
001002	006300			ASL	R0		
001004	063700	000000G		ADD	CST.ADDR,R0		
001010	052760	040000 000012		BIS	#40000,12(R0)		
001016	000413			BR	9#	;	3633
001020	013716	000000G	8#:	MOV	CCTLR,(SP)	;	3642
001024	004737	000000G		JSR	PC,GET.PKT		
001030	010037	000100'		MOV	R0,MX1		
001034	002771			BLT	8#	;	3643
001036	013700	000000G		MOV	T.ADDR,R0	;	3645
001042	005260	000050		INC	50(R0)		
001046	062706	000012	9#:	ADD	#12,SP	;	3567
001052	012601		10#:	MOV	(SP)+,R1	;	3553
001054	000207			RTS	PC	;	

; Routine Size: 279 words, Routine Base: \$CODE\$ + 12040
; Maximum stack depth per invocation: 7 words


```

: 3701 DO (MX1 = GET_PKT (.CCTLR))
: 3702 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3703
: 3704 MSCP_PKT [.MX1, MSGLEN] = SZ_SEN; ! PACKET SIZE SEND DATA
: 3705 MSCP_PKT [.MX1, OPCODE] = OP_SDD; ! OPCODE = SEND DATA
: 3706 MSCP_PKT [.MX1, BC_LO] = 4; ! BYTE COUNT TO BE TRANSFERED EQUALS 4
: 3707 MSCP_PKT [.MX1, BUF_0] = DUPPKT; ! LOAD DESCRIPTOR BUFFER
: 3708 DUPPKT [DUPBF0] = .CST_ADDR [.DUOFF, D_DISK_NUM]; ! LOAD UNIT NUMBER (RDRX)
: 3709 DUPPKT [DUPBF1] = .CST_ADDR [.DUOFF + 5, D_DBN]; ! LOAD DBN NUMBER
: 3710 MSCP_PKT [.MX1, MODIFY] = 0;
: 3711 MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD; ! CALL IT sequential
: 3712 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 3713
: 3714 IF .CST_ADDR [.DUOFF + 5, DUPERROR] EQL 1 !status error
: 3715 THEN RETURN;
: 3716
: 3717 DO (MX1 = GET_PKT (.CCTLR))
: 3718 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3719
: 3720 MSCP_PKT [.MX1, MSGLEN] = SZ_REC; ! PACKET SIZE RECEIVE DATA
: 3721 MSCP_PKT [.MX1, OPCODE] = OP_RCD; ! OPCODE = GET DUST STATUS
: 3722 MSCP_PKT [.MX1, BC_LO] = 514; ! BYTE COUNT TO BE TRANSFERED EQUALS 512
: 3723 MSCP_PKT [.MX1, BUF_0] = DUPPKT; ! LOAD DESCRIPTOR BUFFER
: 3724 MSCP_PKT [.MX1, MODIFY] = 0;
: 3725 MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD; ! CALL IT sequential
: 3726 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 3727
: 3728 IF (.CST_ADDR [.DUOFF + 5, DUPERROR] EQL 1) OR !status error
: 3729 (.DUPPKT [DUPTYPE] NEQU 6) OR !dup type error
: 3730 (.DUPPKT [DUPMSG] NEQU 2)
: 3731 THEN
: 3732 (HARD_ERROR ());
: 3733 CST_ADDR [.DUOFF + 5, DUPERROR] = 1; ! SET FLAG
: 3734 RETURN;); ! NO POINT IN CONTINUING
: 3735
: 3736 DO (MX1 = GET_PKT (.CCTLR))
: 3737 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3738
: 3739 T_ADDR [T_BLK_RD] = .T_ADDR [T_BLK_RD] + 1; !IF DUP NO ERROR THEN INCREMENT COUNTER
: 3740
: 3741 END;

```

		.SBTTL DUPREDDBN MULTI-DRIVE TEST ROUTINES		
000000 010146		DUPREDDBN:		
000002	013700	000000G	MOV R1, -(SP)	3648
000006	005260	000056	MOV T_ADDR, R0	3665
000012	013746	000100'	INC 56(R0)	
000016	012746	000104	MOV MX1, -(SP)	3667
000022	004737	000000G	MOV #104, -(SP)	
000026	012760	000022 000006G	JSR PC, BL#MUL	
000034	112760	000003 000022G	MOV #22, MSCP.PKT+6(R0)	
			MOVB #3, MSCP.PKT+22(R0)	3668

ZRQAM3 V01.2	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES		14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Bliss-16 V3-555 DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (27)	SEQ 0353 Page 98		
000042	112760	000122	000026G	MOVB	#122,MSCP.PKT+26(RO)	:	3669
000050	112760	000105	000027G	MOVB	#105,MSCP.PKT+27(RO)	:	3670
000056	112760	000104	000030G	MOVB	#104,MSCP.PKT+30(RO)	:	3671
000064	112760	000104	000031G	MOVB	#104,MSCP.PKT+31(RO)	:	3672
000072	112760	000102	000032G	MOVB	#102,MSCP.PKT+32(RO)	:	3673
000100	112760	000116	000033G	MOVB	#116,MSCP.PKT+33(RO)	:	3674
000106	012760	000001	000024G	MOV	#1,MSCP.PKT+24(RO)	:	3675
000114	142760	000360	000010G	BICB	#360,MSCP.PKT+10(RO)	:	3676
000122	004737	000000V		JSR	PC,DUPCOMMAND	:	3677
000126	013700	001150'		MOV	DUOFF,RO	:	3679
000132	006300			ASL	RO	:	
000134	063700	000000G		ADD	CST.ADDR,RO	:	
000140	032760	040000	000012	BIT	#40000,12(RO)	:	
000146	001402			BEQ	1#	:	
000150	022626			CHP	(SP)+,(SP)+	:	3648
000152	000505			BR	3#	:	3680
000154	013716	000000G	1#:	MOV	CCTLR,(SP)	:	3682
000160	004737	000000G		JSR	PC,GET.PKT	:	
000164	010037	000100'		MOV	RO,MX1	:	
000170	002771			BLT	1#	:	3683
000172	010016			MOV	RO,(SP)	:	3685
000174	012746	000104		MOV	#104,-(SP)	:	
000200	004737	000000G		JSR	PC,BL#MUL	:	
000204	012760	000034	000006G	MOV	#34,MSCP.PKT+6(RO)	:	
000212	112760	000005	000022G	MOVB	#5,MSCP.PKT+22(RO)	:	3686
000220	012760	000120	000026G	MOV	#120,MSCP.PKT+26(RO)	:	3687
000226	012760	000000G	000032G	MOV	#DUPPKT,MSCP.PKT+32(RO)	:	3688
000234	005060	000024G		CLR	MSCP.PKT+24(RO)	:	3689
000240	142760	000360	000010G	BICB	#360,MSCP.PKT+10(RO)	:	3690
000246	152760	000020	000010G	BISB	#20,MSCP.PKT+10(RO)	:	
000254	004737	000000V		JSR	PC,DUPCOMMAND	:	3691
000260	013700	001150'		MOV	DUOFF,RO	:	3693
000264	006300			ASL	RO	:	
000266	063700	000000G		ADD	CST.ADDR,RO	:	
000272	032760	040000	000012	BIT	#40000,12(RO)	:	
000300	001016			BNE	2#	:	
000302	013700	000000G		MOV	DUPPKT,RO	:	3694
000306	042700	007777		BIC	#7777,RO	:	
000312	020027	010000		CHP	RO,#10000	:	
000316	001007			BNE	2#	:	
000320	013700	000000G		MOV	DUPPKT,RO	:	3695
000324	042700	170000		BIC	#170000,RO	:	
000330	020027	000005		CHP	RO,#5	:	
000334	001415			BEQ	4#	:	
000336	004737	000000V	2#:	JSR	PC,HARD.ERROR	:	3697
000342	013700	001150'		MOV	DUOFF,RO	:	3698
000346	006300			ASL	RO	:	
000350	063700	000000G		ADD	CST.ADDR,RO	:	
000354	052760	040000	000012	BIS	#40000,12(RO)	:	
000362	062706	000006		ADD	#6,SP	:	3693
000366	000501		3#:	BR	5#	:	3697
000370	013716	000000G	4#:	MOV	CCTLR,(SP)	:	3701
000374	004737	000000G		JSR	PC,GET.PKT	:	

000400	010037	000100'	MOV	R0,MX1		
000404	002771		BLT	4#	:	3702
000406	010016		MOV	R0,(SP)	:	3704
000410	012746	000104	MOV	#104,-(SP)		
000414	004737	000000G	JSR	PC,BL#MUL		
000420	012760	000034 000006G	MOV	#34,MSCP.PKT+6(R0)		
000426	112760	000004 000022G	MOVB	#4,MSCP.PKT+22(R0)	:	3705
000434	012760	000004 000026G	MOV	#4,MSCP.PKT+26(R0)	:	3706
000442	012760	000000G 000032G	MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3707
000450	013701	001150'	MOV	DUOFF,R1	:	3708
000454	006301		ASL	R1		
000456	063701	000000G	ADD	CST.ADDR,R1		
000462	111137	000000G	MOVB	(R1),DUPPKT		
000466	042737	177774 000000G	BIC	#177774,DUPPKT		
000474	013701	001150'	MOV	DUOFF,R1	:	3709
000500	006301		ASL	R1		
000502	063701	000000G	ADD	CST.ADDR,R1		
000506	116137	000012 000002G	MOVB	12(R1),DUPPKT+2		
000514	105037	000003G	CLRB	DUPPKT+3		
000520	005060	000024G	CLR	MSCP.PKT+24(R0)	:	3710
000524	142760	000360 000010G	BICB	#360,MSCP.PKT+10(R0)	:	3711
000532	152760	000020 000010G	BISB	#20,MSCP.PKT+10(R0)		
000540	004737	000000V	JSR	PC,DUPCOMMAND	:	3712
000544	013700	001150'	MOV	DUOFF,R0	:	3714
000550	006300		ASL	R0		
000552	063700	000000G	ADD	CST.ADDR,R0		
000556	032760	040000 000012	BIT	#40000,12(R0)		
000564	001403		BEQ	6#		
000566	062706	000010	ADD	#10,SP	:	3648
000572	000521		BR	10#	:	3715
000574	013716	000000G	MOV	CCTLR,(SP)	:	3717
000600	004737	000000G	JSR	PC.GET.PKT		
000604	010037	000100'	MOV	R0,MX1		
000610	002771		BLT	6#	:	3718
000612	010016		MOV	R0,(SP)	:	3720
000614	012746	000104	MOV	#104,-(SP)		
000620	004737	000000G	JSR	PC,BL#MUL		
000624	012760	000034 000006G	MOV	#34,MSCP.PKT+6(R0)		
000632	112760	000005 000022G	MOVB	#5,MSCP.PKT+22(R0)	:	3721
000640	012760	001002 000026G	MOV	#1002,MSCP.PKT+26(R0)	:	3722
000646	012760	000000G 000032G	MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3723
000654	005060	000024G	CLR	MSCP.PKT+24(R0)	:	3724
000660	142760	000360 000010G	BICB	#360,MSCP.PKT+10(R0)	:	3725
000666	152760	000020 000010G	BISB	#20,MSCP.PKT+10(R0)		
000674	004737	000000V	JSR	PC,DUPCOMMAND	:	3726
000700	013700	001150'	MOV	DUOFF,R0	:	3728
000704	006300		ASL	R0		
000706	063700	000000G	ADD	CST.ADDR,R0		
000712	032760	040000 000012	BIT	#40000,12(R0)		
000720	001016		BNE	7#		
000722	013700	000000G	MOV	DUPPKT,R0	:	3729
000726	042700	007777	BIC	#7777,R0		
000732	020027	060000	CMP	R0,#60000		

F12

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	SEQ 0355
V01.2	MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:00	DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (27)	Page 100
000736	001007			
000740	013700	000000G	BNE 7#	
000744	042700	170000	MOV DUPPKT,RO	3730
000750	020027	000002	BIC #170000,RO	
000754	001413		CMP RO,#2	
000756	004737	000000V	BEQ 8#	
000762	013700	001150'	7#: JSR PC,HARD.ERROR	3732
000766	006300		MOV DUOFF,RO	3733
000770	063700	000000G	ASL RO	
000774	052760	040000 000012	ADD CST.ADDR,RO	
001002	000413		BIS #40000,12(RO)	
001004	013716	000000G	BR 9#	3728
001010	004737	000000G	8#: MOV CCTLR,(SP)	3736
001014	010037	000100'	JSR PC,GET.PKT	
001020	002771		MOV RO,MX1	
001022	013700	000000G	BLT 8#	3737
001026	005260	000054	MOV T.ADDR,RO	3739
001032	062706	000012	INC 54(RO)	
001036	012601		9#: ADD #12,SP	3663
001040	000207		10#: MOV (SP)+,R1	3648
			RTS PC	

; Routine Size: 273 words, Routine Base: #CODE# + 13116
; Maximum stack depth per invocation: 7 words

; 3742

```

: 3743
: 3744 ROUTINE DUPCOMMAND : NOVALUE =
: 3745
: 3746 !+
: 3747 ! THIS ROUTINE IS CALLED BY DUP TO PROCESS COMMANDS.
: 3748 ! THE COMMAND ENVELOPES ARE FILLED IN DUP ROUTINES IN THE "MX1" INDEX.
: 3749 ! WITH THE INDEX THIS ROUTINE SENDS THE COMMAND, WAITS FOR A
: 3750 ! RESPONSES AND THEN PROCESSES THE RETURN PACKET.
: 3751 !-
: 3752 BEGIN
: 3753 !PRINTX (DER13);
: 3754
: 3755 MSCP_PKT [.MX1, CREDITS] = 0;           ! DUP DOES NOT USE THE CREDIT SYSTEM
: 3756 MSCP_PKT [.MX1, CONNID] = CID_DUP;    ! MAKE PACKAGE EQUAL A DUP COMMAND
: 3757 MSCP_PKT [.MX1, DK_NUM] = 0;         ! DISK NUMBER (NOT APPLICABLE)
: 3758
: 3759 IF SEND (.MX1) EQLU FAILURE           ! ATTEMPT SEND; IF CTRL IS OFFLINE
: 3760 THEN
: 3761     BEGIN
: 3762     PUT_PKT (.MX1);
: 3763     MX1 = -1;
: 3764     CST_ADDR [.DUOFF + 5, DUPERROR] = 1; ! RETURN ENVELOPE TO POOL
: 3765     PRINTF (DBM112);                   ! "DUP: PKT NOT AVAILABLE"
: 3766     END
: 3767
: 3768 ELSE
: 3769     do
: 3770         begin
: 3771         BREAK;                            ! BREAK FOR ACT
: 3772         PROC_RETPKT ();                  ! PROCESS RETURN PACKET TO SEE IF OK FOR DUP
: 3773         end
: 3774     until (.CRN_LOW eqLU .RP_ADDR [CRF_LO]) or ! TO ENSURE THAT ALL RETURN MESSAGES HAVE BEEN PROCESSED
: 3775     (.EOP_FLAG eq1 true);                ! or end of pass caused by error
: 3776 END;

```

			.SBTTL	DUPCOMMAND MULTI-DRIVE TEST ROUTINES	
000000	013746	000100'	DUPCOMMAND:		
			MOV	MX1, -(SP)	3755
000004	012746	000104	MOV	#104, -(SP)	
000010	004737	000000G	JSR	PC, BL#MUL	
000014	142760	000017 000010G	BICB	#17, MSCP.PKT+10(RO)	
000022	112760	000002 000011G	MOVB	#2, MSCP.PKT+11(RO)	3756
000030	005060	000016G	CLR	MSCP.PKT+16(RO)	3757
000034	013716	000100'	MOV	MX1, (SP)	3759
000040	004737	000000G	JSR	PC, SEND	
000044	005700		TST	RO	
000046	001027		BNE	1#	
000050	013716	000100'	MOV	MX1, (SP)	3762
000054	004737	000000G	JSR	PC, PUT.PKT	
000060	012737	177777 000100'	MOV	#-1, MX1	3763
000066	013700	001150'	MOV	DUOFF, RO	3764
000072	006300		ASL	RO	

H12

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (28)

SEQ 0357
Page 102

000074	063700	000000G		ADD	CST.ADDR,RO		
000100	052760	040000	000012	BIS	#40000,12(RO)		
000106	012716	000000G		MOV	#DBM112,(SP)	:	3765
000112	012746	000001		MOV	#1,-(SP)		
000116	010600			MOV	SP,RO	: SP,+	
000120	104417			TRAP	17		
000122	005726			TST	(SP)+	:	3761
000124	000415			BR	2#	:	3759
000126	104422		1#:	TRAP	22	:	3770
000130	004737	000000V		JSR	PC,PROC.RETPKT	:	3772
000134	013700	000000G		MOV	RP.ADDR,RO	:	3774
000140	023760	000000G	000004	CMP	CRN.LOW,4(RO)		
000146	001404			BEQ	2#		
000150	123727	000000G	000001	CMPB	EOP.FLAG,#1	:	3775
000156	001363			BNE	1#		
000160	022626		2#:	CMP	(SP)+,(SP)+	:	3752
000162	000207			RTS	PC	:	3744

: Routine Size: 58 words, Routine Base: \$CODE\$ + 14160
: Maximum stack depth per invocation: 5 words

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 B11e-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (29)

SEQ 0358
Page 103

```

: 3777
: 3778 ROUTINE DUPIDLE : NOVALUE =
: 3779 !+
: 3780 ! THIS ROUTINE IS CALLED BY DUP ROUTINE TO INSURE THAT THE CONTROLLER
: 3781 ! IS NOT IN A ACTIVE STATE. IF CALLED AND THE CONTROLLER IS IN AN ACTIVE
: 3782 ! STATE THE CONTROLLER WILL GIVE AN ABORT COMMAND WHICH SHOULD KILL THE
: 3783 ! CURRENT JOB OR LOCAL PROGRAM.
: 3784 !-
: 3785 BEGIN
: 3786 CST_ADDR [.DUOFF + 5, DUPERROR] = 0;          !CLEAR DUP ERROR FLAG;
: 3787
: 3788 MSCP_PKT [.MX1, MSGLEN] = SZ_GDS;           ! PACKET SIZE                GET DUST STATUS
: 3789 MSCP_PKT [.MX1, OPCODE] = OP_GDS;         ! OPCODE = GET DUST STATUS
: 3790 MSCP_PKT [.MX1, MODIFY] = 0;
: 3791 MSCP_PKT [.MX1, MSGTYP] = IMM_CMD;         ! CALL IT IMMEDIATE
: 3792 DUPCOMMAND ();                             ! SENDS AND RECEIVES THE COMMAND
: 3793                                             ! GDS ONLY RETURNS SUCCESS or it don't return
: 3794
: 3795 DO (MX1 = GET_PKT (.CCTLR))
: 3796 UNTIL (.MX1 GEQ 0);                          ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRGRAM ERROR
: 3797
: 3798 if .CST_ADDR [.DUOFF + 5, D_ACTIVE] neq IDLE ! if not in idle state then abort the program
: 3799 then
: 3800     begin
: 3801         MSCP_PKT [.MX1, MSGLEN] = SZ_ABT;     ! PACKET SIZE                ABORT CMD
: 3802         MSCP_PKT [.MX1, OPCODE] = OP_ABT;     ! OPCODE = ABORT PROGRAM
: 3803         MSCP_PKT [.MX1, MODIFY] = 0;
: 3804         MSCP_PKT [.MX1, MSGTYP] = IMM_CMD;     ! CALL IT IMMEDIATE
: 3805         DUPCOMMAND ();                         ! SENDS AND RECEIVES THE COMMAND
: 3806                                             !ONLY ERROR IS already in idle state
: 3807     DO (MX1 = GET_PKT (.CCTLR))
: 3808     UNTIL (.MX1 GEQ 0);                          ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRGRAM ERROR
: 3809     end;
: 3810 end;

```

Address	Hex	Dec	Label	Instruction	Comment	Line
000000	010146					
000002	013700	001150'	DUPIDLE:	MOV R1, -(SP)		3778
000006	006300			MOV DUOFF, R0		3786
000010	063700	000000G		ASL R0		
000014	042760	040000 000012		ADD CST_ADDR, R0		
000022	013746	000100'		BIC #40000, 12(R0)		
000026	012746	000104		MOV MX1, -(SP)		3788
000032	004737	000000G		MOV #104, -(SP)		
000036	012760	000014 000006G		JSR PC, BL#MUL		
000044	112760	000001 000022G		MOV #14, MSCP_PKT+6(R0)		
000052	005060	000024G		MOVB #1, MSCP_PKT+22(R0)		3789
000056	142760	000360 000010G		CLR MSCP_PKT+24(R0)		3790
000064	004737	014160'		BICB #360, MSCP_PKT+10(R0)		3791
000070	013716	000000G		JSR PC, DUPCOMMAND		3792
000074	004737	000000G	1#:	MOV CCTLR, (SP)		3795
000100	010037	000100'		JSR PC, GET_PKT		
				MOV R0, MX1		

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0359
Page 104
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (29)

000104	010001			MOV	R0,R1		; MX1,*	3796
000106	002770			BLT	1#			
000110	013700	001150'		MOV	DUOFF,R0			3798
000114	006300			ASL	R0			
000116	063700	000000G		ADD	CST.ADDR,R0			
000122	032760	020000	000012	BIT	#20000,12(R0)			
000130	001432			BEQ	3#			
000132	010116			MOV	R1,(SP)			3801
000134	012746	000104		MOV	#104,-(SP)			
000140	004737	000000G		JSR	PC,BL#MUL			
000144	012760	000014	000006G	MOV	#14,MSCP.PKT+6(R0)			
000152	112760	000006	000022G	MOVB	#6,MSCP.PKT+22(R0)			3802
000160	005060	000024G		CLR	MSCP.PKT+24(R0)			3803
000164	142760	000360	000010G	BICB	#360,MSCP.PKT+10(R0)			3804
000172	004737	014160'		JSR	PC,DUPCOMMAND			3805
000176	013716	000000G	2#:	MOV	CCTLR,(SP)			3807
000202	004737	000000G		JSR	PC,GET.PKT			
000206	010037	000100'		MOV	R0,MX1			
000212	002771			BLT	2#			3808
000214	005726			TST	(SP)+			3800
000216	022626		3#:	CMP	(SP)+,(SP)+			3785
000220	012601			MOV	(SP)+,R1			3778
000222	000207			RTS	PC			

: Routine Size: 74 words, Routine Base: \$CODE\$ + 14344
: Maximum stack depth per invocation: 5 words

```

: 3811 routine QIO_LBN : novalue =
: 3812
: 3813 !+
: 3814 ! THIS ROUTINE IS CALLED BY QIO_GEN TO SELECT THE LOGICAL BLOCK NUMBER TO
: 3815 ! BE USED FOR THE CURRENT QIO OR QIO PAIR.
: 3816 !
: 3817 ! IF THE OPERATOR CHOSE THE RANDOM BLOCK MODE OPTION, THEN THE LBN IS
: 3818 ! RANDOMLY CHOSEN WITHIN THE SPECIFIED LIMITS FOR THE LBN.
: 3819 ! OTHERWISE, THE NEXT SEQUENTIAL LBN IS DERIVED FROM THE BLOCK SEQUENCE
: 3820 ! TABLE (BST).
: 3821 !
: 3822 ! IMPLICIT INPUTS:
: 3823 !     L#LUN - CURRENT (DIAGNOSTIC SUPERVIOR) UNIT NUMBER
: 3824 !
: 3825 ! IMPLICIT OUTPUTS:
: 3826 !     THE LBN IS LOADED INTO ONE OR BOTH MSCP PACKETS.
: 3827 !-
: 3828
: 3829 begin
: 3830
: 3831 local
: 3832     RD_DISK : byte;      ! FLAG TO INDICATE WINCHESTER DISK SELECTED
: 3833
: 3834
: 3835 if .CST_ADDR [.CUOFF, D_TYPE] eql RX_50
: 3836 then
: 3837     RD_DISK = FALSE
: 3838 else
: 3839     RD_DISK = TRUE;
: 3840
: 3841 MAD1 [LBN_L] = .BST [.L#LUN, LO_WRD];      ! LOAD LBN INTO 1ST PACKET
: 3842 MAD1 [LBN_H] = .BST [.L#LUN, HI_WRD];      ! LOAD LBN INTO 1ST PACKET
: 3843
: 3844 if .MX2 geq 0      ! IF 2 QIOS
: 3845 then
: 3846     begin
: 3847         MAD2 [LBN_L] = .BST [.L#LUN, LO_WRD];      ! LOAD LBN INTO 2ND PACKET
: 3848         MAD2 [LBN_H] = .BST [.L#LUN, HI_WRD];      ! LOAD LBN INTO 2ND PACKET
: 3849     end;
: 3850
: 3851 if BIT_TST (SWP_FLAGS, SWF_BLK)      ! IF RANDOM BLOCK MODE
: 3852 then
: 3853     begin
: 3854         if NOT ((.RD_DISK) and
: 3855             (((.RANDOM [0] and %o'077777') mod (99)) lequ 33))
: 3856             then
: 3857                 begin
: 3858                     if .CST_ADDR [.CUOFF + 2, D_BEG1] eqlu .CST_ADDR [.CUOFF + 4, D_END1]
: 3859                         then
: 3860                             BST [.L#LUN, LO_WRD] = .CST_ADDR [.CUOFF + 1, D_BEG0] +
: 3861                                 ! select low lbn from random
: 3862                                 MODULAS (.CST_ADDR [.CUOFF + 1, D_BEG0], .CST_ADDR [.CUOFF + 3, D_END0])
: 3863                             else
: 3864                                 ! if upper word of beg trk and end t

```

```

n winchesters
: 3856
: 3857
: THE SAME
: 3858
: 3859
g trk same
: 3860
number WINDOW
: 3861
: 3862
: 3863
rk different

```

ZRQAM3
V01.2RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQAC0.BL2;161 (30)SEQ 0361
Page 106

```

3864      begin
3865      BST [.L$LUN, HI_WRD] = .CST_ADDR [.CUOFF + 2, D_BEG1] +      ! select upper lbn from window
3866      MODULAS (.CST_ADDR [.CUOFF + 2, D_BEG1], .CST_ADDR [.CUOFF + 4, D_END1]);
3867
3868      if .BST [.L$LUN, HI_WRD] eq1 .CST_ADDR [.CUOFF + 4, D_END1]      ! IF UPPER WORD EQUALS HI LIMIT BE S
3869      then .BST [.L$LUN, LO_WRD] = MODULAS (0, .CST_ADDR [.CUOFF + 3, D_END0]); ! WORD DOES NOT PASS HI LIMIT
3870
3871      if .BST [.L$LUN, HI_WRD] eq1 .CST_ADDR [.CUOFF + 2, D_BEG1]      ! if upper word equal lo limit make
3872      then BST [.L$LUN, LO_WRD] = %0'177777' - MODULAS (.CST_ADDR [.CUOFF + 1, D_BEG0], %0'177777');
3873      ! word is above lo limit
3874
3875      if .BST [.L$LUN, HI_WRD] gtr .CST_ADDR [.CUOFF + 2, D_BEG1] and
3876      .BST [.L$LUN, HI_WRD] les .CST_ADDR [.CUOFF + 4, D_END1]      ! if neither of the above then any n
3877      then BST [.L$LUN, LO_WRD] = .RANDOM [5];
3878      end;
3879      end;
3880
3881      else
3882      begin      ! ELSE - SEQUENTIAL LBN MODE
3883      if (.TRK_SGN [.L$LUN] geq 1)
3884      then      ! if positive track direction add one to multiword
3885      (if .BST [.L$LUN, LO_WRD] eqlu %0'177777'
3886      then
3887      begin
3888      BST [.L$LUN, LO_WRD] = 0;
3889      BST [.L$LUN, HI_WRD] = .BST [.L$LUN, HI_WRD] + 1;
3890      end
3891      else
3892      BST [.L$LUN, LO_WRD] = .BST [.L$LUN, LO_WRD] + 1)
3893      ! if negative track direction subtract one from multiword
3894      if .BST [.L$LUN, LO_WRD] eqlu %0'0'
3895      then
3896      begin
3897      BST [.L$LUN, LO_WRD] = %0'177777';
3898      BST [.L$LUN, HI_WRD] = .BST [.L$LUN, HI_WRD] - 1;
3899      end
3900      else
3901      BST [.L$LUN, LO_WRD] = .BST [.L$LUN, LO_WRD] - 1;
3902
3903
3904      if .BST [.L$LUN, LO_WRD] gequ (.CST_ADDR [.CUOFF + 3, D_END0]) and ! if hi limit then change direction
3905      .BST [.L$LUN, HI_WRD] gequ (.CST_ADDR [.CUOFF + 4, D_END1])
3906      then TRK_SGN [.L$LUN] = -1;
3907
3908      if .BST [.L$LUN, LO_WRD] lequ (.CST_ADDR [.CUOFF + 1, D_BEG0] + 1) and ! if low limit then change direction
3909      .BST [.L$LUN, HI_WRD] lequ (.CST_ADDR [.CUOFF + 2, D_BEG1])
3910      then TRK_SGN [.L$LUN] = 1;
3911      end;
3912      end;

```

! ROUTINE QIO_LBN

000000 004137 000000G

.SBTTL QIO.LBN MULTI-DRIVE TEST ROUTINES
QIO.LBN:JSR R1,\$SAVE5

3811

M12

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0362
Page 107
VAX-11 B11es-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (30)

000004	005746			TST	-(SP)		
000006	013703	000000G		MOV	CST.ADDR,R3	:	3835
000012	013704	000000G		MOV	CUOFF,R4		
000016	010400			MOV	R4,R0		
000020	006300			ASL	R0		
000022	060300			ADD	R3,R0		
000024	132710	000034		BITB	#34,(R0)		
000030	001002			BNE	1#		
000032	105005			CLRB	R5	: RD.DISK	3837
000034	000402			BR	2#	:	3835
000036	112705	000001	1#:	MOVB	#1,R5	: *,RD.DISK	3839
000042	013700	000104'	2#:	MOV	MAD1,R0	:	3841
000046	013702	000000G		MOV	L#LUN,R2		
000052	010201			MOV	R2,R1		
000054	006301			ASL	R1		
000056	006301			ASL	R1		
000060	012716	000000G		MOV	#BST,(SP)		
000064	060116			ADD	R1,(SP)		
000066	017660	000000	000046	MOV	80(SP),46(R0)		
000074	062701	000002G		ADD	#BST+2,R1	:	3842
000100	011160	000050		MOV	(R1),50(R0)		
000104	005737	000102'		TST	MX2	:	3844
000110	002407			BLT	3#		
000112	013700	000106'		MOV	MAD2,R0	:	3847
000116	017660	000000	000046	MOV	80(SP),46(R0)		
000124	011160	000050		MOV	(R1),50(R0)	:	3848
000130	032737	001000	000000G	3#:	BIT	#1000,SWP.FLAGS	3851
000136	001002			BNE	4#		
000140	000137	015344'		JMP	11#		
000144	006005		4#:	ROR	R5	: RD.DISK	3854
000146	103014			BCC	5#		
000150	013746	000000G		MOV	RANDOM,-(SP)	:	3855
000154	042716	100000		BIC	#100000,(SP)		
000160	012746	000143		MOV	#143,-(SP)		
000164	004737	000000G		JSR	PC,BL#MOD		
000170	022626			CMP	(SP)+,(SP)+		
000172	020027	000041		CMP	R0,#41		
000176	101565			BLOS	10#		
000200	010400		5#:	MOV	R4,R0	:	3858
000202	006300			ASL	R0		
000204	060300			ADD	R3,R0		
000206	016002	000004		MOV	4(R0),R2		
000212	010400			MOV	R4,R0		
000214	006300			ASL	R0		
000216	060300			ADD	R3,R0		
000220	020260	000010		CMP	R2,10(R0)		
000224	001021			BNE	6#		
000226	010405			MOV	R4,R5	:	3860
000230	006305			ASL	R5		
000232	060305			ADD	R3,R5		
000234	016546	000002		MOV	2(R5),-(SP)	:	3861
000240	010400			MOV	R4,R0		
000242	006300			ASL	R0		

N12

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0363
Page 108
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (30)

000244	060300				ADD	R3,R0			
000246	016046	000006			MOV	6(R0),-(SP)			
000252	004737	000000G			JSR	PC,MODULAS			
000256	066500	000002			ADD	2(R5),R0	:		3860
000262	010076	000004			MOV	R0,84(SP)			
000266	000530				BR	9#	:		
000270	010246			6#:	MOV	R2,-(SP)	:		3858
000272	016046	000010			MOV	10(R0),-(SP)	:		3866
000276	004737	000000G			JSR	PC,MODULAS			
000302	060200				ADD	R2,R0	:		3865
000304	010011				MOV	R0,(R1)			
000306	013702	000000G			MOV	L#LUN,R2	:		3868
000312	006302				ASL	R2			
000314	006302				ASL	R2			
000316	013700	000000G			MOV	CUOFF,R0			
000322	006300				ASL	R0			
000324	063700	000000G			ADD	CST.ADDR,R0			
000330	026260	000002G	000010		CMP	BST+2(R2),10(R0)			
000336	001015				BNE	7#			
000340	005016				CLR	(SP)	:		3869
000342	013700	000000G			MOV	CUOFF,R0			
000346	006300				ASL	R0			
000350	063700	000000G			ADD	CST.ADDR,R0			
000354	016046	000006			MOV	6(R0),-(SP)			
000360	004737	000000G			JSR	PC,MODULAS			
000364	010072	000000G			MOV	R0,8BST(R2)			
000370	005726				TST	(SP)+			
000372	013701	000000G		7#:	MOV	L#LUN,R1	:		3871
000376	006301				ASL	R1			
000400	006301				ASL	R1			
000402	013700	000000G			MOV	CUOFF,R0			
000406	006300				ASL	R0			
000410	063700	000000G			ADD	CST.ADDR,R0			
000414	026160	000002G	000004		CMP	BST+2(R1),4(R0)			
000422	001021				BNE	8#			
000424	013700	000000G			MOV	CUOFF,R0	:		3872
000430	006300				ASL	R0			
000432	063700	000000G			ADD	CST.ADDR,R0			
000436	016016	000002			MOV	2(R0),(SP)			
000442	012746	177777			MOV	#-1, -(SP)			
000446	004737	000000G			JSR	PC,MODULAS			
000452	012761	177777	000000G		MOV	#-1,BST(R1)			
000460	160061	000000G			SUB	R0,BST(R1)			
000464	005726				TST	(SP)+			
000466	013701	000000G		8#:	MOV	L#LUN,R1	:		3875
000472	006301				ASL	R1			
000474	006301				ASL	R1			
000476	013700	000000G			MOV	CUOFF,R0			
000502	006300				ASL	R0			
000504	063700	000000G			ADD	CST.ADDR,R0			
000510	026160	000002G	000004		CMP	BST+2(R1),4(R0)			
000516	003414				BLE	9#			
000520	013700	000000G			MOV	CUOFF,R0	:		3876

000524	006300		ASL	R0		
000526	063700	000000G	ADD	CST.ADDR,R0		
000532	026160	000002G 000010	CMP	BST+2(R1),10(R0)		
000540	002003		BGE	9#		
000542	013761	000012G 000000G	MOV	RANDOM+12,BST(R1)	:	
000550	022626		CMP	(SP)+,(SP)+	:	3877
000552	000471		BR	17#	:	3857
000554	062702	000000G	ADD	#TRK.SGN,R2	:	3851
000560	105712		TSTB	(R2)	:	3883
000562	003413		BLE	13#		
000564	027627	000000 177777	CMP	#0(SP),#-1	:	3885
000572	001004		BNE	12#		
000574	005076	000000	CLR	#0(SP)	:	3888
000600	005211		INC	(R1)	:	3889
000602	000415		BR	15#	:	3885
000604	005276	000000	INC	#0(SP)	:	3892
000610	000412		BR	15#	:	3883
000612	005776	000000	TST	#0(SP)	:	3894
000616	001005		BNE	14#		
000620	012776	177777 000000	MOV	#-1,#0(SP)	:	3897
000626	005311		DEC	(R1)	:	3898
000630	000402		BR	15#	:	3894
000632	005376	000000	DEC	#0(SP)	:	3901
000636	010400		MOV	R4,R0	:	3904
000640	006300		ASL	R0		
000642	060300		ADD	R3,R0		
000644	027660	000000 000006	CMP	#0(SP),6(R0)		
000652	103410		BLO	16#		
000654	010400		MOV	R4,R0	:	3905
000656	006300		ASL	R0		
000660	060300		ADD	R3,R0		
000662	021160	000010	CMP	(R1),10(R0)		
000666	103402		BLO	16#		
000670	112712	000377	MOVB	#377,(R2)	:	3906
000674	010400		MOV	R4,R0	:	3908
000676	006300		ASL	R0		
000700	060300		ADD	R3,R0		
000702	016000	000002	MOV	2(R0),R0		
000706	005200		INC	R0		
000710	027600	000000	CMP	#0(SP),R0		
000714	101010		BHI	17#		
000716	010400		MOV	R4,R0	:	3909
000720	006300		ASL	R0		
000722	060300		ADD	R3,R0		
000724	021160	000004	CMP	(R1),4(R0)		
000730	101002		BHI	17#		
000732	112712	000001	MOVB	#1,(R2)	:	3910
000736	005726		TST	(SP)+	:	3811
000740	000207		RTS	PC		

; Routine Size: 241 words, Routine Base: #CODE# + 14570
; Maximum stack depth per invocation: 11 words

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0365
Page 110
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (31)

```

3913 routine QIO_SIZE : novalue =
3914
3915 !*
3916 ! THIS ROUTINE IS CALLED BY QIO_GEN TO SELECT THE I/O TRANSFER BYTE COUNT
3917 ! TO BE USED FOR THE CURRENT QIO OR QIO PAIR. THE BYTE COUNT IS
3918 ! DETERMINED BY A RANDOM NUMBER, AND WILL ALWAYS FALL BETWEEN 1 AND THE
3919 ! I/O BUFFER SIZE (BYTS_PER_QIO). It is assumed that BYTS_PER_QIO will
3920 ! never be larger than one binary word or 65000 bytes.
3921 !
3922 ! IMPLICIT OUTPUTS:
3923 ! THE BYTE COUNT IS LOADED INTO ONE OR BOTH MSCP PACKETS.
3924 !-
3925
3926 begin
3927
3928 local
3929     SIZE : word,
3930     BLOCKS_LEFT : word;
3931
3932     SIZE = ((.RANDOM [4] and %o'077777') mod (.BYTS_PER_QIO + 1)) and %o'177760'; !GET BYTE COUNT FROM RANDOM NUMBER
3933
3934     if .SIZE eq 0
3935     then
3936         SIZE = 16;
3937
3938     if .CST_ADDR [.CUOFF + 4, D_END1] gtru .MAD1 [LBN_H]
3939     then BLOCKS_LEFT = %o'177777'
3940     else BLOCKS_LEFT = .CST_ADDR [.CUOFF + 3, D_END0] - .MAD1 [LBN_L] + 1;
3941
3942     if ((.SIZE + BYTES_PER_SECT - 1) / BYTES_PER_SECT) gtru .BLOCKS_LEFT
3943     then
3944         SIZE = .BLOCKS_LEFT * BYTES_PER_SECT;
3945
3946     MAD1 [BC_LO] = .SIZE;
3947
3948     if .MX2 geq 0
3949     then
3950         MAD2 [BC_LO] = .SIZE;
3951
3952 end;

```

000000	004137	000000G	.SBTTL	QIO.SIZE MULTI-DRIVE TEST ROUTINES	
			QIO.SIZE:		
000004	013746	000010G	JSR	R1, #SAVE3	3913
000010	042716	100000	MOV	RANDOM+10, -(SP)	3932
000014	013746	000000G	BIC	#100000, (SP)	
000020	005216		MOV	BYTS.PER.QIO, -(SP)	
000022	004737	000000G	INC	(SP)	
000026	010003		JSR	PC, BL#MOD	
000030	042703	000017	MOV	R0, R3	; *.SIZE
000034	001002		BIC	#17, R3	; *.SIZE
			BNE	18	3934

D13

ZRGAM3 V01.2	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 B11-16 V3-555 DISK\$USER2:([DIETZ.RDRX])ZRGACO.BL2;161 (31)	SEQ 0366 Page 111
000036	012703	000020	MOV #20,R3	
000042	013700	000000G	MOV CUOFF,R0	; *,SIZE 3936
000046	006300		ASL R0	; 3938
000050	063700	000000G	ADD CST.ADDR,R0	
000054	013701	000104'	MOV MAD1,R1	
000060	026061	000010 000050	CMP 10(R0),50(R1)	
000066	101403		BLOS 2#	
000070	012702	177777	MOV #-1,R2	; *,BLOCKS.LEFT 3939
000074	000413		BR 3#	; 3938
000076	013700	000000G	MOV CUOFF,R0	; 3940
000102	006300		ASL R0	
000104	063700	000000G	ADD CST.ADDR,R0	
000110	016000	000006	MOV 6(R0),R0	
000114	166100	000046	SUB 46(R1),R0	
000120	010002		MOV R0,R2	; *,BLOCKS.LEFT
000122	005202		INC R2	; BLOCKS.LEFT
000124	010316		MOV R3,(SP)	; SIZE,* 3942
000126	062716	000777	ADD #777,(SP)	
000132	012746	001000	MOV #1000,-(SP)	
000136	004737	000000G	JSR PC,BL\$DIV	
000142	005726		TST (SP)+	
000144	020002		CMP R0,R2	; *,BLOCKS.LEFT
000146	101405		BLOS 4#	
000150	010200		MOV R2,R0	; BLOCKS.LEFT,* 3944
000152	000300		SWAB R0	
000154	105000		CLRB R0	
000156	006300		ASL R0	
000160	010003		MOV R0,R3	; *,SIZE
000162	010361	000026	MOV R3,26(R1)	; SIZE,* 3946
000166	005737	000102'	TST MX2	; 3948
000172	002404		BLT 5#	
000174	013700	000106'	MOV MAD2,R0	; 3950
000200	010360	000026	MOV R3,26(R0)	; SIZE,*
000204	022626		CMP (SP)+,(SP)+	; 3926
000206	000207		RTS PC	; 3913

; Routine Size: 68 words, Routine Base: \$CODE\$ + 15532
; Maximum stack depth per invocation: 8 words

```

: 3953 routine FILL_BUFF : novalue =
: 3954
: 3955 !!
: 3956 !! THIS ROUTINE IS CALLED BY QIO_GEN TO LOAD THE I/O BUFFER DESCRIBED IN
: 3957 !! THE FIRST MSCP PACKET WITH THE APPROPRIATE DATA PATTERN.
: 3958 !!
: 3959 !! THE DATA PATTERN TO BE SELECTED IS BASED ON THE FOLLOWING ALGORITHM:
: 3960 !!
: 3961 !!     IF THE OPERATOR DEFINED A DATA PATTERN
: 3962 !!     THEN
: 3963 !!         SELECT IT
: 3964 !!     ELSE
: 3965 !!         GET DATA PATTERN NUMBER FROM SW P-TABLE
: 3966 !!         IF DATA PATTERN NUMBER = 0
: 3967 !!         THEN
: 3968 !!             GET DATA PATTERN NUMBER FROM THE UNIT'S ENTRY
: 3969 !!             IN THE DATA PATTERN SEQUENCE TABLE (DPST)
: 3970 !!
: 3971 !! NOTE THAT PATTERN # 1 CONSISTS OF RANDOM NUMBERS, AND PATTERNS # 17 -
: 3972 !! 21 USE THE ACTUAL LBN OF THE WRITE REQUEST.
: 3973 !!
: 3974 !! IMPLICIT INPUTS:
: 3975 !!     L&LUN - CURRENT (DRS) UNIT NUMBER
: 3976 !!-
: 3977
: 3978 begin
: 3979
: 3980 local
: 3981     DP_NUM : word,           ! DATA PATTERN NUMBER SELECTED
: 3982     DP_ADDR,                ! ADDR OF DATA PATTERN (LENGTH)
: 3983     IOB_ADDR,               ! I/O BUFFER ADDRESS (DESTINATION)
: 3984     SRC_ADDR,               ! WORKING SOURCE ADDRESS
: 3985     COUNT : word;          ! NO. OF WORDS IN DATA PATTERN
: 3986
: 3987 if BIT_TST (SMP_FLAGS, SWF_UDP) ! IF USER DEFINED A DATA PATTERN
: 3988 then
: 3989     DP_ADDR = SMP_UCNT       ! SELECT IT
: 3990 else
: 3991     begin
: 3992
: 3993     if .SMP_DPAT neq 0       ! IF USER SELECTED A PRE-DEFINED DATA PATTERN
: 3994     then
: 3995         DP_NUM = .SMP_DPAT  ! SELECT IT
: 3996     else
: 3997         begin
: 3998             DP_NUM = .DPST [.L&LUN]; ! GET PATTERN NUMBER FROM SEQUENCE TABLE
: 3999             DPST [.L&LUN] = .DPST [.L&LUN] + 1; ! ADVANCE TO NEXT PATTERN NUMBER
: 4000
: 4001             if .DPST [.L&LUN] gtru DP_CNT ! CHECK FOR HIGH LIMIT
: 4002             then
: 4003                 DPST [.L&LUN] = 1;
: 4004
: 4005         end;

```

```

: 4006
: 4007 DP_ADDR = .DPA_TBL [.DP_NUM - 1]; ! ADDRESS OF DATA PATTERN (COUNT)
: 4008
: 4009 if .DP_NUM gequ 17
: 4010 then
: 4011
: 4012 if .DP_NUM ! CHECK MACRO ! IF PATTERN 17, 19, OR 21
: 4013 then
: 4014 (.DP_ADDR + 2) = .MAD1 [LBN_L] ! LOAD LBN INTO FIRST WORD OF PATTERN
: 4015 else
: 4016 (.DP_ADDR + 4) = .MAD1 [LBN_L]; ! LOAD LBN INTO SECOND WORD OF PATTERN
: 4017
: 4018 end;
: 4019
: 4020 IOB_ADDR = .MAD1 [BUF_0]; ! I/O BUFFER ADDRESS
: 4021 COUNT = ..DP_ADDR; ! NO. OF WORDS IN DATA PATTERN
: 4022 SRC_ADDR = .DP_ADDR + 2; ! START OF THE ACTUAL DATA PATTERN
: 4023
: 4024 incr N from 1 to ((.MAD1 [BC_LO] + 1) / 2) do ! FOR EACH WORD IN THIS WRITE REQUEST
: 4025 begin
: 4026 .IOB_ADDR = ..SRC_ADDR; ! MOVE 1 WORD
: 4027 IOB_ADDR = .IOB_ADDR + 2; ! ADVANCE DESTINATION ADDRESS
: 4028 SRC_ADDR = .SRC_ADDR + 2; ! ADVANCE SOURCE ADDRESS
: 4029 COUNT = .COUNT - 1; ! DECREMENT COUNT
: 4030
: 4031 if .COUNT eql 0 ! IF END OF DATA PATTERN
: 4032 then
: 4033 begin
: 4034 COUNT = ..DP_ADDR; ! REPEAT DATA PATTERN
: 4035 SRC_ADDR = .DP_ADDR + 2;
: 4036 end;
: 4037
: 4038 end; ! WORD TRANSFER LOOP
: 4039
: 4040 end; ! ROUTINE FILL_BUFF

```

```

000000 004137 000000G .SBTTL FILL.BUFF MULTI-DRIVE TEST ROUTINES
: FILL.BUFF:
000004 005746 JSR R1, $SAVES ; 3953
000006 032737 TST -(SP) ;
000014 001403 000100 000000G BIT @100, SWP.FLAGS ; 3987
000016 012703 000000G BEQ 1# ;
000022 000443 MOV @SWP.UCNT, R3 ; *, DP.ADDR 3989
000024 013700 000000G BR 5# ; 3987
000030 001402 1#: MOV SWP.DPAT, R0 ; 3993
000032 010001 BEQ 2# ;
000034 000414 MOV R0, R1 ; *, DP.NUM 3995
000036 013700 000000G BR 3# ; 3993
000042 062700 000050' 2#: MOV L$LUN, R0 ; 3998
000046 005001 ADD @DPST, R0 ;
000050 151001 CLR R1 ; DP.NUM
: BISB (R0), R1 ; *, DP.NUM

```

ZRQAM3 V01.2	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Blues-16 V3-555 DISK#USER2:(DIETZ.RDRX)ZRQACO.BL2;161 (32)	SEQ 0369 Page 114
000052	105210		INCB (R0)	3999
000054	121027	000025	CMPB (R0),#25	4001
000060	101402		BLOS 3#	
000062	112710	000001	MOVB #1,(R0)	4003
000066	010100		MOV R1,R0	4007
000070	006300	3#:	ASL R0	
000072	016003	001064'	MOV DPA.TBL-2(R0),R3	
000076	020127	000021	CMP R1,#21	4009
000102	103413		BLO 5#	
000104	013700	000104'	MOV MAD1,R0	4014
000110	006001		ROR R1	4012
000112	103004		BCC 4#	
000114	016063	000046 000002	MOV 46(R0),2(R3)	4014
000122	000403		BR 5#	4012
000124	016063	000046 000004	MOV 46(R0),4(R3)	4016
000132	013700	000104'	MOV MAD1,R0	4020
000136	016005	000032	MOV 32(R0),R5	
000142	011302		MOV (R3),R2	4021
000144	012704	000002	MOV #2,R4	4022
000150	060304		ADD R3,R4	
000152	010416		MOV R4,(SP)	
000154	016046	000026	MOV 26(R0),-(SP)	4024
000160	005216		INC (SP)	
000162	012746	000002	MOV #2,-(SP)	
000166	004737	000000G	JSR PC,BL#DIV	
000172	005001		CLR R1	
000174	000412		BR 7#	
000176	017625	000004	MOV #4(SP),(R5)+	4026
000202	062766	000002 000004	ADD #2,4(SP)	4028
000210	005302		DEC R2	4029
000212	001003		BNE 7#	4031
000214	011302		MOV (R3),R2	4034
000216	010466	000004	MOV R4,4(SP)	4035
000222	005201		INC R1	4024
000224	020100	7#:	CMP R1,R0	
000226	003763		BLE 6#	
000230	062706	000006	ADD #6,SP	
000234	000207		RTS PC	3953

; Routine Size: 79 words, Routine Base: #CODE# + 15742
; Maximum stack depth per invocation: 10 words

```

: 4041 routine PROC_RETPKT : novalue =
: 4042
: 4043 !*
: 4044 !
: 4045 ! THIS ROUTINE IS CALLED FROM THE MULTI_DRIVE "EXECUTIVE" AND DUP_COMMAND TO CHECK FOR
: 4046 ! AND PROCESS ANY RETURN PACKETS THAT HAVE BEEN "SENT" BY THE "DRIVER"
: 4047 ! PORTION OF THE PROGRAM. THE I/O DONE QUEUE (IODQ) ACTS AS THE LINK
: 4048 ! BETWEEN THE TWO PROGRAM PARTS; IT HOLDS INDECES OF RETURN PACKETS WHICH
: 4049 ! REQUIRE PROCESSING.
: 4050 !
: 4051 ! UNDER THE MULTI-DRIVE TEST, RETURN PACKETS ORIGINATE FROM TWO SOURCES:
: 4052 ! 1. MSCP - THE MORE COMMON, DESCRIBING A COMPLETED I/O
: 4053 ! OPERATION.
: 4054 ! 2. DUP - THE LESS COMMON, DESCRIBING A PORTION OF I/O
: 4055 ! COMMUNICATIONS WITH THE CONTROLLER PROGRAM.
: 4056 ! 3. THE PROGRAM "DRIVER" - DESCRIBING A CONTROLLER ERROR OR
: 4057 ! COMMAND TIMEOUT.
: 4058 !-
: 4059 while .IODQ_IN neq .IODQ_OUT do ! DO UNTIL I/O DONE QUEUE IS EMPTY
: 4060 begin
: 4061 RP_INDX = OUT_IODQ (); ! GET INDEX OF NEXT RETPKT AND ADVANCE OUT POINTER
: 4062 RP_ADDR = RETPKT + (.RP_INDX * RP_LEN * 2); ! CALCULATE RETPKT ADDRESS
: 4063 if NOT (.RP_ADDR [CONID] eq1 CID_DUP) ! if not DUP then
: 4064 then (SET_CPAR (.RP_ADDR [CTLR])); ! SET UP CURRENT CONTROLLER PARAMETERS
: 4065
: 4066 selectneu .RP_ADDR [CONID] of ! CONNECTION ID INDICATES PACKET SOURCE
: 4067 set
: 4068
: 4069 [CID_MSCP] : IO_RETPKT (); ! DISK MSCP (I/O TRANSFER DONE)
: 4070 [CID_DUP] : DIO_RETPKT (); ! DUP (I/O TRANSFER DONE)
: 4071 [CID_DRIVER] : DR_RETPKT (); ! MESSAGE FROM "DRIVER"
: 4072
: 4073 [otherwise] : PRINTF (DBM12, .RP_ADDR [CONID]);!"CONN ID = XXXXX RECEIVED"
: 4074 tes;
: 4075
: 4076 end; ! UNTIL I/O DONE QUEUE IS EMPTY

```

Address	Offset	Hex	Hex	Label	Operation	Comment	Line
000000	010146			.SBTTL	PROC.RETPKT MULTI-DRIVE TEST ROUTINES		
000002	023737	000000G	000000G	PROC.RETPKT:	MOV R1, -(SP)		4041
000010	001467			1*:	CMP IODQ.IN, IODQ.OUT		4059
000012	004737	000000G			BEQ 7*		
000016	010037	000000G			JSR PC, OUT_IODQ		4061
000022	010046				MOV R0, RP_INDX		
000024	012746	000060			MOV R0, -(SP)	; RP_INDX, *	4062
000030	004737	000000G			MOV #60, -(SP)		
000034	062700	000000G			JSR PC, BL#MUL		
000040	010037	000000G			ADD #RETPKT, R0		
000044	126027	000003	000002		MOV R0, RP_ADDR		
000052	001406				CMPB 3(R0), #2		4063
000054	116016	000002			BEQ 2*		
					MOVB 2(R0), (SP)		4064

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0371
Page 116
VAX-11 B110-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.8L2;161 (33)

000060	042716	177760		BIC	#177760,(SP)		
000064	004737	000000G		JSR	PC,SET.CPAR		
000070	013700	000000G	2#:	MOV	RP.ADDR,R0	:	4066
000074	005001			CLR	R1		
000076	156001	000003		BISB	3(R0),R1		
000102	005701			TST	R1		
000104	001003			BNE	3#		
000106	004737	000000V		JSR	PC,IO.RETPKT	:	4069
000112	000424			BR	6#	:	4066
000114	020127	000002	3#:	CMP	R1,#2		
000120	001003			BNE	4#		
000122	004737	000000V		JSR	PC,DIO.RETPKT	:	4070
000126	000416			BR	6#	:	4066
000130	020127	000003	4#:	CMP	R1,#3		
000134	001003			BNE	5#		
000136	004737	000000V		JSR	PC,DR.RETPKT	:	4071
000142	000410			BR	6#	:	4066
000144	010116		5#:	MOV	R1,(SP)	:	4073
000146	012746	000000G		MOV	#DBM12,-(SP)		
000152	012746	000002		MOV	#2,-(SP)		
000156	010600			MOV	SP,R0	: SP,*	
000160	104417			TRAP	17		
000162	022626			CMP	(SP)+,(SP)+		
000164	022626		6#:	CMP	(SP)+,(SP)+	:	4060
000166	000705			BR	1#	:	4059
000170	012601		7#:	MOV	(SP)+,R1	:	4041
000172	000207			RTS	PC		

: Routine Size: 62 words, Routine Base: #CODE# + 16200
: Maximum stack depth per invocation: 7 words

```

: 4077 :+
: 4078 ROUTINE DIO_RETPKT : NOVALUE =
: 4079
: 4080 :+
: 4081 : THIS ROUTINE IS CALLED BY PROC_RETPKT TO HANDLE ALL DUP I/O TRANSFER
: 4082 : RETURN PACKETS. PROCESSING OF THESE PACKETS INCLUDES DECLARING ANY
: 4083 : HARD ERRORS THAT MAY HAVE OCCURRED, UPDATING THE STATISTICS.
: 4084 :
: 4085 : IMPLICIT INPUTS:
: 4086 : RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4087 : T_ADDR - ADDRESS OF THE CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 4088 : CST_ADDR - ADDRESS OF THE CURRENT CONTROLLER'S CST
: 4089 : DUOFF - CST OFFSET FOR THE CURRENT UNIT
: 4090 : L#LUN - CURRENT UNIT NUMBER
: 4091 : CCTLN - CURRENT CONTROLLER NUMBER
: 4092 :
: 4093 : IMPLICIT OUTPUTS
: 4094 : CST_ADDR [.DUOFF + 5, NODUPMEDIA] - IF THIS BIT SET NO DUP EXERCISER
: 4095 :
: 4096 :-
: 4097 BEGIN
: 4098
: 4099 LOCAL FLAG : BYTE INITIAL(BYTE(TRUE)),
: 4100 SUM2 : WORD,
: 4101 SUM : WORD; ! TOTAL NUMBER OF BYTES TRANSFERRED TO/FROM A UNIT
: 4102 !PRINTX (DER18);
: 4103
: 4104 IF .RP_ADDR [STATUS] NEQU ST_SUC ! IF STATUS CODE INDICATES ERROR
: 4105 THEN
: 4106 BEGIN
: 4107 CST_ADDR [.DUOFF + 5, DUPERROR] = 1; ! SET DUP ERROR FLAG
: 4108 HARD_ERROR ();
: 4109 IF .RP_ADDR [ENDCOD] EQLU (OP_ELP + OP_END) OR ! IF ENDCODE IS EXECUTE LOCAL PROGRAM
: 4110 .RP_ADDR [ENDCOD] EQLU (OP_GDS + OP_END) ! OR GET DUST STATUS
: 4111 THEN BEGIN
: 4112 CST_ADDR [.DUOFF + 5, NODUPMEDIA] = 1; ! TURN OFF DUP EXERCISR
: 4113 END;
: 4114 END
: 4115 ELSE ! ELSE - I/O WAS SUCCESSFUL
: 4116 BEGIN
: 4117
: 4118 IF .RP_ADDR [ENDCOD] EQLU (OP_GDS + OP_END) ! IF ENDCODE IS GET DUST STATUS
: 4119 THEN
: 4120 BEGIN
: 4121 IF .RP_ADDR [9,11,1,0] EQL 1
: 4122 THEN CST_ADDR [.DUOFF + 5, D_ACTIVE] = ACTIVE ! CONTROLLER IN AN ACTIVE STAE
: 4123 ELSE CST_ADDR [.DUOFF + 5, D_ACTIVE] = IDLE; ! CONTROLLER IN AN IDLE STATE
: 4124 IF .RP_ADDR [9,9,1,0] NEQ 1 THEN ! TEST TO SEE IF CONTROLLER LOCAL PROGRAMS(PG 18 OF DUP DOC)
: 4125 BEGIN
: 4126 HARD_ERROR ();
: 4127 CST_ADDR [.DUOFF + 5, NODUPMEDIA] = 1; ! TURN OFF DUP EXERCISR
: 4128 END;
: 4129 END;

```

```

:      4130
:      4131
:      4132 IF (.RP_ADDR [ENDCOD] EQL (OP_RCD + OP_END)) AND
:      4133 (.DUPPKT [DUPTYPE] EQL 6) AND
:      4134 (.DUPPKT [DUPMSG] EQL 2) AND           !IF IT IS A RECEIVE DBN COMMAND WITH TYPE 6 AND MESSAGE 2 THEN
:      4135 (.CST_ADDR [.DUOFF + 5, DUPWRITE] EQLU 1) ! IF WRITE FLAG SET IN CST TABLE THEN COMPARE BLOCKS
:      4136 THEN DUP_COMPARE ();
:      4137
:      4138 END;                                ! COMPARE THE FOLLOWING 512 BYTES
:      4139
:      4140 PUT_RETPKT (.RP_INDX);
:      4141 END;                                ! ROUTINE DIO_RETPKT

```

```

000000 010146          .SBTTL DIO.RETPKT MULTI-DRIVE TEST ROUTINES
DIO.RETPKT:
000002 112700 000001  MOV R1, -(SP) ; 4078
000006 013701 000000G  MOVB #1, R0 ; *,FLAG 4097
000012 005761 000016  MOV RP_ADDR, R1 ; 4104
000016 001435  BEQ 2#
000020 013700 001150'  MOV DUOFF, R0 ; 4107
000024 006300  ASL R0
000026 063700 000000G  ADD CST_ADDR, R0
000032 052760 040000 000012  BIS #40000, 12(R0)
000040 004737 000000V  JSR PC, HARD_ERROR ; 4108
000044 013700 000000G  MOV RP_ADDR, R0 ; 4109
000050 126027 000014 000203  CMPB 14(R0), #203
000056 001404  BEQ 1#
000060 126027 000014 000201  CMPB 14(R0), #201 ; 4110
000066 001112  BNE 6#
000070 013700 001150'  1#: MOV DUOFF, R0 ; 4112
000074 006300  ASL R0
000076 063700 000000G  ADD CST_ADDR, R0
000102 052760 100000 000012  BIS #100000, 12(R0)
000110 000501  BR 6# ; 4104
000112 126127 000014 000201  2#: CMPB 14(R1), #201 ; 4118
000120 001036  BNE 5#
000122 013700 001150'  MOV DUOFF, R0 ; 4122
000126 006300  ASL R0
000130 063700 000000G  ADD CST_ADDR, R0
000134 032761 004000 000022  BIT #4000, 22(R1) ; 4121
000142 001404  BEQ 3#
000144 052760 020000 000012  BIS #20000, 12(R0) ; 4122
000152 000403  BR 4# ; 4121
000154 042760 020000 000012  3#: BIC #20000, 12(R0) ; 4123
000162 032761 001000 000022  4#: BIT #1000, 22(R1) ; 4124
000170 001012  BNE 5#
000172 004737 000000V  JSR PC, HARD_ERROR ; 4126
000176 013700 001150'  MOV DUOFF, R0 ; 4127
000202 006300  ASL R0
000204 063700 000000G  ADD CST_ADDR, R0
000210 052760 100000 000012  BIS #100000, 12(R0)

```


L13

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0374
Page 119
VAX-11 B1es-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (34)

000216	013700	000000G	5#:	MOV	RP.ADDR,R0	:	4132
000222	126027	000014 000205		CMPB	14(R0),#205		
000230	001031			BNE	6#		
000232	013700	000000G		MOV	DUPPKT,R0	:	4133
000236	042700	007777		BIC	#7777,R0		
000242	020027	060000		CMP	R0,#60000		
000246	001022			BNE	6#		
000250	013700	000000G		MOV	DUPPKT,R0	:	4134
000254	042700	170000		BIC	#170000,R0		
000260	020027	000002		CMP	R0,#2		
000264	001013			BNE	6#		
000266	013700	001150'		MOV	DUOFF,R0	:	4135
000272	006300			ASL	R0		
000274	063700	000000G		ADD	CST.ADDR,R0		
000300	032760	010000 000012		BIT	#10000,12(R0)		
000306	001402			BEQ	6#		
000310	004737	000000V		JSR	PC,DUP.COMPARE	:	4136
000314	013746	000000G	6#:	MOV	RP.INDX,-(SP)	:	4140
000320	004737	000000G		JSR	PC,PUT.RETPKT		
000324	005726			TST	(SP)+	:	4097
000326	012601			MOV	(SP)+,R1	:	4078
000330	000207			RTS	PC		

: Routine Size: 109 words, Routine Base: \$CODE\$ + 16374
: Maximum stack depth per invocation: 3 words

: 4142

```

: 4143 ROUTINE DUP_COMPARE : NOVALUE =
: 4144
: 4145
: 4146 !+
: 4147 ! THIS ROUTINE IS CALLED BY DIO_RETPKT WHEN THE RECEIVE DATA COMMAND
: 4148 ! IS BEING PROCESSED. THIS COMMAND COMPARES THE WRITTEN BUFFER WITH
: 4149 ! THE PATERN WORD GIVEN IN SEND DATA COMMAND. FOR EVERY WORD COMPARED
: 4150 ! THE ROUTINE INCREMENTS THE TALLY TABLE. IF THE COMPARE SHOWS AN
: 4151 ! ERROR. THE DBN HARD ERROR COUNTER WILL BE INCREMENTED AND THE
: 4152 ! THE DBN NUMBER AND BYTE COUNT WILL BE PRINTED.
: 4153
: 4154 ! IMPLICIT INPUTS:
: 4155 ! S_PATTERN ! THE SAVED PATTERN WRITTEN TO THE DBN'S
: 4156 ! S_DUPPKT ! THE POINTER FOR DUP BUFFER
: 4157 ! T_ADDR ! THE ADDRESS OF THE TALLY TABLE FOR THIS UNIT
: 4158 ! CST_ADDR ! THE ADDRESS OF PRESENT CONTROLLER STATUS TABLE
: 4159 !-
: 4160 BEGIN
: 4161
: 4162 OWN
: 4163 COUNT : WORD;
: 4164
: 4165 !PRINTX (DER19);
: 4166 S_DUPPKT = 0;
: 4167 INCR COUNT FROM 1 TO 256 DO !INDEX PIONTER FOR DATA STORED IN MSCP ENV PACKET
: 4168 BEGIN
: 4169 S_DUPPKT = .S_DUPPKT + 2; ! INITIALLY THIS SKIPS THE FIRST WORD OF DUPPKT
: 4170 IF .(DUPPKT + .S_DUPPKT) NEQ .S_PATTERN THEN !IF THE CONTENTS OF DBN DOESN'T EQUAL PATTERN
: 4171 BEGIN
: 4172 CST_ADDR [.DUOFF + 5, DUPERROR] = 1; ! SET DUP ERROR FLAG
: 4173 ERRHRD (46, EH_10, EMS_22); !LIST ERROR
: 4174 EXITLOOP;
: 4175 END;
: 4176 END; !GO THROUGH ALL DBN WORDS
: 4177 END; !END ROUTINE DUP-COMPARE

```

```

001164 .PSECT $GGG$, RO
001164 COUNT: .BLKW 1

```

```

016726 .SBTTL DUP_COMPARE MULTI-DRIVE TEST ROUTINES
.PSECT $CODE$, RO

```

```

000000 010146 DUP_COMPARE:
000002 005037 000000G MOV R1, -(SP) ; 4143
000006 012701 000400 CLR S.DUPPKT ; 4166
000012 062737 000002 000000G MOV #400, R1 ; *,COUNT 4167
000020 013700 000000G 1$: ADD #2, S.DUPPKT ; 4169
000024 026037 000000G 000000G MOV S.DUPPKT, RO ; 4170
000032 001415 CMP DUPPKT(RO), S.PATTERN
BEQ 2$

```

N13

ZRQAM3	RD/RX EXERCISER		14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	SEQ 0376
V01.2	MULTI-DRIVE TEST ROUTINES		14-Dec-1983 16:12:00	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (35)	Page 121
000034	013700	001150'			4172
000040	006300		MOV	DUOFF,R0	
000042	063700	000000G	ASL	R0	
000046	052760	040000 000012	ADD	CST.ADDR,R0	
000054	104456		BIS	#40000,12(R0)	
000056	000056		TRAP	56	4173
000060	000000G		.WORD	56	
000062	000000G		.WORD	EH.10	
000064	000402		.WORD	EMS.22	
000066	005301	2\$:	BR	3\$	4171
000070	001350		DEC	R1	4167
000072	012601	3\$:	BNE	1\$	
000074	000207		MOV	(SP)+,R1	4143
			RTS	PC	

```

: Routine Size: 31 words,      Routine Base: $CODE$ + 16726
: Maximum stack depth per invocation: 3 words

```

```

:      4178
:      4179
:      4180

```

```

: 4181 routine IO_RETPKT : novalue =
: 4182
: 4183 !!
: 4184 !! THIS ROUTINE IS CALLED BY PROC_RETPKT TO HANDLE ALL I/O TRANSFER
: 4185 !! RETURN PACKETS. PROCESSING OF THESE PACKETS INCLUDES DECLARING ANY
: 4186 !! HARD ERRORS THAT MAY HAVE OCCURRED, UPDATING THE STATISTICS, AND
: 4187 !! PERFORMING MOST WRITE-COMPARES IF REQUIRED.
: 4188 !!
: 4189 !! IMPLICIT INPUTS:
: 4190 !! CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 4191 !! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4192 !! T_ADDR - ADDRESS OF CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 4193 !! CCTLN - CURRENT CONTROLLER NUMBER
: 4194 !! L$LUN - CURRENT UNIT NUMBER
: 4195 !!-
: 4196
: 4197 begin
: 4198
: 4199 local
: 4200 FLAG : byte initial (byte (TRUE));
: 4201
: 4202 FSET_UPAR (); ! FIND UNIT'S ENTRY IS CST AND SET UP UNIT-RELATED DATA
: 4203 ST_CODE = .RP_ADDR [STSCOD]; ! GET STATUS CODE FROM RETPKT
: 4204 SB_CODE = .RP_ADDR [SUBCOD]; ! GET SUB-CODE, IF ANY
: 4205
: 4206 if (.ST_CODE neq ST_SUC) ! IF STATUS CODE INDICATES ERROR
: 4207 then
: 4208 begin
: 4209 HARD_ERROR (); ! UPDATE ERROR COUNT
: 4210 COMPARE_DATA = FALSE; ! NO POINT IN DOING MOST COMPARES ON ERROR
: 4211
: 4212 if (.ST_CODE neq ST_OFL) and ! DROP UNIT IF ERROR COUNTS EXCEEDS LIMIT
: 4213 (.st_code neq ST_AVL) and
: 4214 (.T_ADDR [ERR_HARD] gequ .SWP_ERROR)
: 4215 then
: 4216 begin
: 4217 DUR [.L$LUN] = DU_HERR; ! LOAD REASON FOR DROPPING UNIT
: 4218 DODU (.L$LUN); ! DROP UNIT
: 4219 end;
: 4220
: 4221 end;
: 4222
: 4223 if .ST_CODE eq1 ST_SUC ! IF I/O WAS SUCCESSFUL
: 4224 then
: 4225 begin
: 4226 UPD_IO_TALLY (); ! UPDATE I/O TALLY (STATISTICS)
: 4227
: 4228 if .RP_ADDR [ENDCOD] eq1 (OP_WRT or OP_END)
: 4229 then COMPARE_DATA = TRUE; ! MOST COMPARES MAY BE ALLOWED IF NO FURTHER ERRORS
: 4230
: 4231 if (BIT_TST (SWP_FLAGS, SWF_HMC)) and ! IF HOST IS DOING WRITE-COMPARES
: 4232 (.COMPARE_DATA)
: 4233 then

```

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0378
Page 123
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (36)

```

:      4234          FLAG = HOST_WRT_CHK ( );
:      4235
:      4236          end;
:      4237
:      4238          if .FLAG
:      4239          then
:      4240              SWEEP ( );
:      4241
:      4242          QIO [.CCTLR] = .QIO [.CCTLR] - 1;
:      4243          end;

```

```

: SAVE I/O PACKET OR DO WRITE-CHECK
: IF FLAG IS STILL TRUE
: DEALLOCATE BUFFER(S) AND RETPKT(S)
: DECREMENT NO. OF OUTSTANDING QIOS
: ROUTINE IO_RETPKT

```

			.SBTTL	IO.RETPKT MULTI-DRIVE TEST ROUTINES	
000000	004137	000000G	IO.RETPKT:		
000004	112701	000001	JSR	R1,#SAVE2	4181
000010	004737	000000V	MOVB	#1,R1	4197
000014	013700	000000G	JSR	PC,FSET.UPAR	4202
000020	116037	000016 000000G	MOV	RP,ADDR,RO	4203
000026	042737	177740 000000G	MOVB	16(RO),ST.CODE	
000034	016002	000016	BIC	#177740,ST.CODE	
000040	006202		MOV	16(RO),R2	4204
000042	006202		ASR	R2	
000044	006202		ASR	R2	
000046	006202		ASR	R2	
000050	006202		ASR	R2	
000052	042702	174000	BIC	#174000,R2	
000056	010237	000000G	MOV	R2,SB.CODE	
000062	005737	000000G	TST	ST.CODE	4206
000066	001433		BEQ	2#	
000070	004737	000000V	JSR	PC,HARD.ERROR	4209
000074	105037	001152'	CLRB	COMPARE.DATA	4210
000100	023727	000000G 000003	CMF	ST.CODE,#3	4212
000106	001420		BEQ	1#	
000110	023727	000000G 000004	CMF	ST.CODE,#4	4213
000116	001414		BEQ	1#	
000120	013700	000000G	MOV	T.ADDR,RO	4214
000124	026037	000014 000000G	CMF	14(RO),SWP.ERROR	
000132	103406		BLO	1#	
000134	013700	000000G	MOV	L#LUN,RO	4217
000140	112760	000004 000000G	MOVB	#4,DUR(RO)	
000146	104451		TRAP	51	4218
000150	005737	000000G	1#:	TST	4223
000154	001026		BNE	4#	
000156	004737	000000V	2#:	JSR	4226
000162	013700	000000G	MOV	PC,UPD.IO.TALLY	
000166	126027	000014 000242	CMF	RP,ADDR,RO	4228
000174	001003		CMF	14(RO),#242	
000176	112737	000001 001152'	BNE	3#	
000204	032737	000040 000000G	3#:	MOVB	4229
000212	001407		BIT	#40,SWP.FLAGS	4231
000214	032737	000001 001152'	BEQ	4#	
000222	001403		BIT	#1,COMPARE.DATA	4232
			BEQ	4#	

D14

ZRQAM3	RD/RX EXERCISER		14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	SEQ 0379
V01.2	MULTI-DRIVE TEST ROUTINES		14-Dec-1983 16:12:00	DISK\$USER2:(DIETZ.RDRX)ZRQACO.BL2;161 (36)	Page 124
000224	004737	000000V			
000230	110001				4234
000232	006001		4\$:		
000234	103002				4238
000236	004737	000000V			
000242	013700	000000G	5\$:		4240
000246	105360	000000G			4242
000252	000207				4181

: Routine Size: 86 words, Routine Base: \$CODE\$ + 17024
: Maximum stack depth per invocation: 5 words

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (37)

SEQ 0380

Page 125

```

: 4244 routine FSET_UPAR : novalue =
: 4245
: 4246 !*
: 4247 ! THIS ROUTINE IS CALLED BY IO_RETPKT AND OTHERS TO SEARCH THE CURRENT
: 4248 ! CONTROLLER STATUS TABLE (CST) FOR THE DISK ADDRESS WHICH IS
: 4249 ! CONTAINED IN THE CURRENT RETURN PACKET. WHEN FOUND, THE OFFSET INTO THE
: 4250 ! CST IS USED AS INPUT TO SET_UPAR, WHICH SETS UP CURRENT UNIT-RELATED
: 4251 ! DATA PARAMETERS.
: 4252 !
: 4253 ! IMPLICIT INPUTS:
: 4254 ! RP_ADDR - ADDRESS OF CURRENT RETURN PACKET
: 4255 ! CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 4256 !-
: 4257
: 4258 begin
: 4259
: 4260 incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do ! FOR EACH UNIT IN CST
: 4261
: 4262 if .CST_ADDR [.OFFSET, D_DISK_NUM] eq1 .RP_ADDR [DISK]
: 4263 ! IF RETPKT UNIT NUMBER MATCHES CST ENTRY
: 4264 then
: 4265 begin
: 4266 SET_UPAR (.OFFSET); ! SET UP UNIT-RELATED DATA
: 4267 return; ! DONE
: 4268 end;
: 4269
: 4270 PRINTF (DBM19, .RP_ADDR [DISK], .CCTLR); ! "CAN'T FIND DISK XXX IN CST X"
: 4271 end; ! ROUTINE FSET_UPAR

```

000000	004137	000000G	.SBTTL FSET.UPAR MULTI-DRIVE TEST ROUTINES	
000004	012702	000003	FSET.UPAR:	
000010	010201		JSR R1, #SAVE4	4244
000012	006301		MOV #3, R2	4260
000014	063701	000000G	1#: MOV R2, R1	4262
000020	013700	000000G	ASL R1	
000024	016004	000010	AD() CST_ADDR, R1	
000030	111103		MOV RP_ADDR, R0	
000032	042703	177774	MOV 10(R0), R4	
000036	020304		MOVB (R1), R3	
000040	001005		BIC #177774, R3	
000042	010246		CMP R3, R4	
000044	004737	000000G	BNE 2#	
000050	005726		MOV R2, -(SP)	4266
000052	000207		JSR PC, SET_UPAR	
000054	062702	000007	TST (SP)+	4262
000060	020227	000030	RTS PC	4265
000064	003751		2#: ADD #7, R2	4260
000066	013746	000000G	CMP R2, #30	
000072	013700	000000G	BLE 1#	
000076	016046	000010	MOV CCTLR, -(SP)	4270
			MOV RP_ADDR, R0	
			MOV 10(R0), -(SP)	

F14

ZRQAM3 RD/RX EXERCISER
VOL 2 MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (37)

SEQ 0381
Page 126

000102	012746	000000G	MOV	#DBM19,-(SP)		
000106	012746	000003	MOV	#3,-(SP)		
000112	010600		MOV	SP,R0	; SP,*	
000114	104417		TRAP	17		
000116	062706	000010	ADD	#10,SP	;	4258
000122	000207		RTS	PC	;	4244

; Routine Size: 42 words, Routine Base: \$CODE\$ + 17300
; Maximum stack depth per invocation: 11 words


```

: 4272 routine HARD_ERROR : novalue =
: 4273
: 4274 !+
: 4275 ! THIS ROUTINE IS CALLED BY IO_RETPKT, DIO_RETPKT, AND OTHERS TO INCREMENT THE HARD
: 4276 ! ERROR STATISTIC FIELD FOR THE CURRENT UNIT. IF THE HARD ERROR COUNT
: 4277 ! HAS EXCEEDED THE OPERATOR-SPECIFIED LIMIT, THEN THE UNIT IS DROPPED
: 4278 ! FROM TESTING.
: 4279 !
: 4280 ! IMPLICIT INPUTS:
: 4281 !     L$LUN - CURRENT UNIT NUMBER
: 4282 !     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 4283 !     CUOFF - CST OFFSET FOR CURRENT UNIT
: 4284 !     T_ADDR - ADDRESS OF CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 4285 !-
: 4286
: 4287 begin
: 4288     T_ADDR [ERR_HARD] = .T_ADDR [ERR_HARD] + 1;           ! INCREMENT UNIT'S HARD ERROR COUNT
: 4289 if .RP_ADDR [CONID] EQL CID_MSCP
: 4290 THEN
: 4291     selectoneu .ST_CODE of
: 4292     set
: 4293
: 4294     [ST_SUC]:         if .SB_CODE neq 0                       ! SUCCESS WITH NON-ZERO SUB-CODE
: 4295                     then
: 4296                         begin
: 4297                             if .SB_CODE eql 4
: 4298                             then
: 4299                                 begin
: 4300                                     T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1
: 4301                                 end
: 4302                             else
: 4303                                 T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4304
: 4305                                 ERRHRD (44, EGH_30, EMS_30);
: 4306                                 end;
: 4307
: 4308     [ST_CMD]:         begin                                     ! INVALID COMMAND
: 4309                         T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4310                         ERRHRD (31, EGH_30, EMS_30);
: 4311                         end;
: 4312
: 4313     [ST_ABO]:         begin                                     ! COMMAND ABORTED
: 4314                         T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4315                         ERRHRD (32, EGH_30, EMS_30);
: 4316                         end;
: 4317
: 4318     [ST_OFL] :       begin                                     ! OFFLINE
: 4319                         T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4320                         ERRDF (18, EGD_18, EMS_18);
: 4321                         DUR [.L$LUN] = DU_DFATAL;           ! DEVICE FATAL ERROR
: 4322                         DODU (.L$LUN);                       ! DROP UNIT
: 4323                         end;
: 4324

```

```

: 4325
: 4326      [ST_AVL] :      begin
: 4327                          T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4328                          ERRDF (24, EGD_18, EMS_18);
: 4329                          DUR [.L#LUN] = DU_AV;                                ! DEVICE WENT AVAILABLE STATE
: 4330                          DODU (.L#LUN);                                        ! DROP UNIT
: 4331                          end;
: 4332
: 4333      [ST_MFE]:      begin                                                    ! MEDIA FORMAT ERROR
: 4334                          T_ADDR [ERR_HRD_SEK] = .T_ADDR [ERR_HRD_SEK] + 1;
: 4335                          ERRHRD (35, EGH_30, EMS_30);
: 4336                          end;
: 4337
: 4338      [ST_WPT]:      begin                                                    ! DEVICE WRITE PROTECTED
: 4339
: 4340                          if .SB_CODE eq1 128
: 4341                          then
: 4342                              T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1
: 4343                          else
: 4344                              T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4345
: 4346                          ERRHRD (36, EGH_30, EMS_30);
: 4347                          end;
: 4348
: 4349      [ST_CMP]:      begin                                                    ! COMPARE ERROR
: 4350                          T_ADDR [ERR_HRD_DAT] = .T_ADDR [ERR_HRD_DAT] + 1;
: 4351                          ERRHRD (37, EGH_30, EMS_30);
: 4352                          end;
: 4353
: 4354      [ST_DAT]:      begin                                                    ! DATA ERROR
: 4355
: 4356                          if .SB_CODE eq1 2
: 4357                          then
: 4358                              T_ADDR [ERR_HRD_SEK] = .T_ADDR [ERR_HRD_SEK] + 1
: 4359                          else
: 4360                              T_ADDR [ERR_HRD_DAT] = .T_ADDR [ERR_HRD_DAT] + 1;
: 4361
: 4362                          ERRHRD (38, EGH_30, EMS_30);
: 4363                          end;
: 4364
: 4365      [ST_HST]:      begin                                                    ! HOST ACCESS ERROR
: 4366                          T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4367                          ERRHRD (39, EGH_30, EMS_30);
: 4368                          end;
: 4369
: 4370      [ST_CNT]:      begin                                                    ! CONTROLLER ERROR
: 4371                          T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4372                          ERRHRD (40, EGH_30, EMS_30);
: 4373                          end;
: 4374
: 4375      [ST_DRV]:      begin                                                    ! DRIVE ERROR
: 4376
: 4377                          if .SB_CODE eq1 3

```

ZRQAM3
V01.2RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK\$USER2:(DIETZ.RDRX)ZRQACO.BL2;161 (38)SEQ 0384
Page 129

```

: 4378      then
: 4379      T_ADDR [ERR_HRD_SEK] = .T_ADDR [ERR_HRD_SEK] + 1
: 4380      else
: 4381      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4382
: 4383      ERRHRD (41, EGH_30, EMS_30);
: 4384      end;
: 4385
: 4386      [ST_DIA]:      begin                                ! MESSAGE FROM INTERNAL DIAGNOSTICS
: 4387      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4388      ERRHRD (43, EGH_30, EMS_30);
: 4389      end;
: 4390
: 4391      [otherwise]:  begin                                ! PRINT STATUS CODE IF NO MATCH
: 4392      C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 4393      ERRHRD (45, EGH_30, EMS_30);
: 4394      end;
: 4395
: 4396      tes;
: 4397      if .RP_ADDR [CONID] EQL CID_DUP
: 4398      THEN
: 4399
: 4400      selectoneu .RP_ADDR [STSCOD] of
: 4401      SET
: 4402      [%o'0']      : begin                                ! if status code succesful
: 4403      if .RP_ADDR [ENCODE] EQLU (OP_GDS + OP_END) and ! IF ENCODE IS GET DUST STATUS
: 4404      .RP_ADDR [9,9,1,0] NEQ 1                        ! TEST TO SEE IF CONTROLLER LOCAL PR
:
: 4405      then                                             ! (PG 18 OF DUP DOC)
: 4406      BEGIN
: 4407      ERRHRD (47, EH_12, EMS_30);      ! UNABLE TO LOAD LOCAL CONTROLLER DU
:
: 4408      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4409      END
: 4410      else
: 4411      begin
: 4412      if (.DUPPKT [DUPTYPE] eql 5)                    ! if fatal error
: 4413      then
: 4414      begin
: 4415      DUR [.L$LUN] = DU_DFATAL;
: 4416      DODU (.L$LUN);                                ! FATAL DEVICE ERROR DROP UNIT);
: 4417      end;                                           ! SET REASON FOR DROPPING UNIT
: 4418      selectoneu .DUPPKT [DUPMSG] of
: 4419      SET
: 4420      [%o'0'] : begin
: 4421      errhrd (99, eh_13, ems_30); ! succesfull message
: 4422      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4423      end;
: 4424      [%o'1'] : begin
: 4425      errhrd (48, eh_13, ems_30); ! illegal unit number
: 4426      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4427      end;
: 4428      [%o'2'] : begin
: 4429      errhrd (49, eh_13, ems_30); ! illegal relative or physical b
:
: 4430      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;

```

OGRAMS

P MEDIA

lock #

```

: 4431
: 4432
: 4433
: 4434
: 4435
: 4436
: 4437
: 4438
: 4439
: 4440
: 4441
: 4442
D) + 1;
: 4443
: 4444
: 4445
: 4446
: 4447
: 4448
: 4449
: 4450
: 4451
: 4452
: 4453
: 4454
: 4455
: 4456
: 4457
: 4458
: 4459
: 4460
: 4461
: 4462
: 4463
: 4464
: 4465
: 4466
: 4467
: 4468
: 4469
: 4470
: 4471
: 4472
: 4473
: 4474
: 4475
: 4476
: 4477
: 4478

end;

[no'3'] : begin
errhrd (50, eh_13, ems_30); ! device error
T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
end;

[no'4'] : begin
errhrd (51, eh_13, ems_30); ! zero lenght message
T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
end;

[OTHERWISE] : begin
ERRHRD (98, EH_8, EMS_30); ! DUP UNKNOWN STATUS CODE
C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HR
end;

tes;

end;

[no'1'] : begin
ERRHRD (52, EH_7, EMS_30); ! INVALID COMMAND
T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
end;

[no'2'] : begin
ERRHRD (53, EH_7, EMS_30); ! NO REGION AVAILABLE
T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
end;

[no'3'] : begin
ERRHRD (54, EH_7, EMS_30); ! NO REGION SUITABLE
T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
end;

[no'4'] : begin
ERRHRD (55, EH_7, EMS_30); ! PROGRAM NOT KNOWN
T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
end;

[no'5'] : begin
ERRHRD (56, EH_7, EMS_30); ! LOAD FAILURE
T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
end;

[no'6'] : begin
ERRHRD (57, EH_7, EMS_30); ! STANDALONE
T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
end;

[OTHERWISE] : begin
ERRHRD (58, EH_8, EMS_30); ! DUP UNKNOWN STATUS CODE
C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
end;

TES;

end;

! ROUTINE HARD_ERROR

```

000004	013701	000000G		MOV	T.ADDR,R1	:	4288
000010	005261	000014		INC	14(R1)	:	
000014	013700	000000G		MOV	RP.ADDR,R0	:	4289
000020	105760	000003		TSTB	3(R0)	:	
000024	001157			BNE	12#	:	
000026	013700	000000G		MOV	ST.CODE,R0	:	4291
000032	001022			BNE	3#	:	
000034	013702	000000G		MOV	SB.CODE,R2	:	4294
000040	001574			BEQ	16#	:	
000042	012703	000062		MOV	#62,R3	:	4301
000046	060103			ADD	R1,R3	:	
000050	020227	000004		CMP	R2,#4	:	4298
000054	001002			BNE	1#	:	
000056	105213			INCB	(R3)	:	4301
000060	000402			BR	2#	:	4298
000062	105263	000001	1#:	INCB	1(R3)	:	4304
000066	104456		2#:	TRAP	56	:	4306
000070	000054			.WORD	54	:	
000072	000000G			.WORD	EGH.30	:	
000074	000000G			.WORD	EMS.30	:	
000076	000567			BR	18#	:	4291
000100	020027	000001	3#:	CMP	R0,#1	:	
000104	001007			BNE	4#	:	
000106	105261	000063		INCB	63(R1)	:	4310
000112	104456			TRAP	56	:	4311
000114	000037			.WORD	37	:	
000116	000000G			.WORD	EGH.30	:	
000120	000000G			.WORD	EMS.30	:	
000122	000567			BR	20#	:	4291
000124	020027	000002	4#:	CMP	R0,#2	:	
000130	001007			BNE	5#	:	
000132	105261	000062		INCB	62(R1)	:	4315
000136	104456			TRAP	56	:	4316
000140	000040			.WORD	40	:	
000142	000000G			.WORD	EGH.30	:	
000144	000000G			.WORD	EMS.30	:	
000146	000576			BR	24#	:	4291
000150	020027	000003	5#:	CMP	R0,#3	:	
000154	001015			BNE	6#	:	
000156	105261	000062		INCB	62(R1)	:	4320
000162	104455			TRAP	55	:	4321
000164	000022			.WORD	22	:	
000166	000000G			.WORD	EGD.18	:	
000170	000000G			.WORD	EMS.18	:	
000172	013700	000000G		MOV	L#LUN,R0	:	4322
000176	112760	000005	000000G	MOVB	#5,DUR(R0)	:	
000204	104451			TRAP	51	:	4323
000206	000570			BR	26#	:	4291
000210	020027	000004	6#:	CMP	R0,#4	:	
000214	001015			BNE	7#	:	
000216	105261	000062		INCB	62(R1)	:	4327
000222	104455			TRAP	55	:	4328
000224	000030			.WORD	30	:	

ZRQAM3
V01.2RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00SEQ 0387
Page 132
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (38)

000226	000000G			.WORD	EGD.18		
000230	000000G			.WORD	EMS.18		
000232	013700	000000G		MOV	L#LUN,R0	:	4329
000236	112760	000013	000000G	MOVB	#13,DUR(R0)	:	
000244	104451			TRAP	51	:	4330
000246	000562			BR	28#	:	4291
000250	020027	000005	7#:	CMP	R0,#5	:	
000254	001007			BNE	8#	:	
000256	105261	000060		INCB	60(R1)	:	4334
000262	104456			TRAP	56	:	4335
000264	000043			.WORD	43		
000266	000000G			.WORD	EGH.30		
000270	000000G			.WORD	EMS.30		
000272	000550			BR	28#	:	4291
000274	020027	000006	8#:	CMP	R0,#6	:	
000300	001020			BNE	11#	:	
000302	012702	000062		MOV	#62,R2	:	4342
000306	060102			ADD	R1,R2	:	
000310	023727	000000G	000200	CMP	SB.CODE,#200	:	4340
000316	001003			BNE	9#	:	
000320	105262	000001		INCB	1(R2)	:	4342
000324	000401			BR	10#	:	4340
000326	105212		9#:	INCB	(R2)	:	4344
000330	104456		10#:	TRAP	56	:	4346
000332	000044			.WORD	44		
000334	000000G			.WORD	EGH.30		
000336	000000G			.WORD	EMS.30		
000340	000525			BR	28#	:	4291
000342	020027	000007	11#:	CMP	R0,#7	:	
000346	001007			BNE	13#	:	
000350	105261	000061		INCB	61(R1)	:	4350
000354	104456			TRAP	56	:	4351
000356	000045			.WORD	45		
000360	000000G			.WORD	EGH.30		
000362	000000G			.WORD	EMS.30		
000364	000513		12#:	BR	28#	:	4291
000366	020027	000010	13#:	CMP	R0,#10	:	
000372	001020			BNE	17#	:	
000374	012702	000060		MOV	#60,R2	:	4358
000400	060102			ADD	R1,R2	:	
000402	023727	000000G	000002	CMP	SB.CODE,#2	:	4356
000410	001002			BNE	14#	:	
000412	105212			INCB	(R2)	:	4358
000414	000402			BR	15#	:	4356
000416	105262	000001	14#:	INCB	1(R2)	:	4360
000422	104456		15#:	TRAP	56	:	4362
000424	000046			.WORD	46		
000426	000000G			.WORD	EGH.30		
000430	000000G			.WORD	EMS.30		
000432	000470		16#:	BR	28#	:	4291
000434	020027	000011	17#:	CMP	R0,#11	:	
000440	001007			BNE	19#	:	
000442	105261	000063		INCB	63(R1)	:	4366

M14

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (38)

SEQ 0388
Page 133

000446	104456			TRAP	56	:	4367
000450	000047			.WORD	47		
000452	000000G			.WORD	EGH.30		
000454	000000G			.WORD	EMS.30		
000456	000456		18#:	BR	28#	:	4291
000460	020027	000012	19#:	CMP	RO,#12		
000464	001007			BNE	21#		
000466	105261	000062		INCB	62(R1)	:	4371
000472	104456			TRAP	56	:	4372
000474	000050			.WORD	50		
000476	000000G			.WORD	EGH.30		
000500	000000G			.WORD	EMS.30		
000502	000444		20#:	BR	28#	:	4291
000504	020027	000013	21#:	CMP	RO,#13		
000510	001016			BNE	25#		
000512	023727	000000G 000003		CMP	SB.CODE,#3	:	4377
000520	001003			BNE	22#		
000522	105261	000060		INCB	60(R1)	:	4379
000526	000402			BR	23#	:	4377
000530	105261	000062	22#:	INCB	62(R1)	:	4381
000534	104456		23#:	TRAP	56	:	4383
000536	000051			.WORD	51		
000540	000000G			.WORD	EGH.30		
000542	000000G			.WORD	EMS.30		
000544	000423		24#:	BR	28#	:	4291
000546	020027	000037	25#:	CMP	RO,#37		
000552	001007			BNE	27#		
000554	105261	000062		INCB	62(R1)	:	4387
000560	104456			TRAP	56	:	4388
000562	000053			.WORD	53		
000564	000000G			.WORD	EGH.30		
000566	000000G			.WORD	EMS.30		
000570	000411		26#:	BR	28#	:	4291
000572	013700	000000G	27#:	MOV	CCTLR,RO	:	4392
000576	006300			ASL	RO		
000600	105260	000000G		INCB	C.ERR.TBL(RO)		
000604	104456			TRAP	56	:	4393
000606	000055			.WORD	55		
000610	000000G			.WORD	EGH.30		
000612	000000G			.WORD	EMS.30		
000614	013700	000000G	28#:	MOV	RP.ADDR,RO	:	4397
000620	126027	000003 000002		CMPB	3(RO),#2		
000626	001401			BEQ	29#		
000630	000207			RTS	PC		
000632	116001	000016	29#:	MOVB	16(RO),R1	:	4400
000636	042701	177740		BIC	#177740,R1		
000642	001111			BNE	37#		
000644	126027	000014 000201		CMPB	14(RO),#201	:	4403
000652	001011			BNE	30#		
000654	032760	001000 000022		BIT	#1000,22(RO)	:	4404
000662	001005			BNE	30#		
000664	104456			TRAP	56	:	4407
000666	000057			.WORD	57		

N14

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (38)

SEQ 0389
Page 134

000670	000000G		.WORD	EH.12		
000672	000000G		.WORD	EMS.30		
000674	000543		BR	42‡	:	4408
000676	013700	000000G	30‡: MOV	DUPPKT,R0	:	4412
000702	042700	007777	BIC	#7777,R0		
000706	020027	050000	CMP	R0,#50000		
000712	001006		BNE	31‡		
000714	013700	000000G	MOV	L‡LUN,R0	:	4415
000720	112760	000005 000000G	MOVB	#5,DUR(R0)		
000726	104451		TRAP	51	:	4416
000730	013700	000000G	31‡: MOV	DUPPKT,R0	:	4418
000734	042700	170000	BIC	#170000,R0		
000740	001005		BNE	32‡		
000742	104456		TRAP	56	:	4421
000744	000143		.WORD	143		
000746	000000G		.WORD	EH.13		
000750	000000G		.WORD	EMS.30		
000752	000530		BR	44‡	:	4422
000754	020027	000001	32‡: CMP	R0,#1	:	4418
000760	001005		BNE	33‡		
000762	104456		TRAP	56	:	4425
000764	000060		.WORD	60		
000766	000000G		.WORD	EH.13		
000770	000000G		.WORD	EMS.30		
000772	000520		BR	44‡	:	4426
000774	020027	000002	33‡: CMP	R0,#2	:	4418
001000	001005		BNE	34‡		
001002	104456		TRAP	56	:	4429
001004	000061		.WORD	61		
001006	000000G		.WORD	EH.13		
001010	000000G		.WORD	EMS.30		
001012	000510		BR	44‡	:	4430
001014	020027	000003	34‡: CMP	R0,#3	:	4418
001020	001005		BNE	35‡		
001022	104456		TRAP	56	:	4433
001024	000062		.WORD	62		
001026	000000G		.WORD	EH.13		
001030	000000G		.WORD	EMS.30		
001032	000464		BR	42‡	:	4434
001034	020027	000004	35‡: CMP	R0,#4	:	4418
001040	001005		BNE	36‡		
001042	104456		TRAP	56	:	4437
001044	000063		.WORD	63		
001046	000000G		.WORD	EH.13		
001050	000000G		.WORD	EMS.30		
001052	000470		BR	44‡	:	4438
001054	104456		36‡: TRAP	56	:	4441
001056	000142		.WORD	142		
001060	000000G		.WORD	EH.8		
001062	000000G		.WORD	EMS.30		
001064	000474		BR	46‡	:	4442
001066	020127	000001	37‡: CMP	R1,#1	:	4400
001072	001005		BNE	38‡		

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0390
Page 135
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (38)

001074	104456			TRAP	56		4449
001076	000064			.WORD	64	:	
001100	000000G			.WORD	EH.7		
001102	000000G			.WORD	EMS.30		
001104	000437			BR	42:	:	4450
001106	020127	000002	38:	CMP	R1,#2	:	4400
001112	001005			BNE	39:		
001114	104456			TRAP	56	:	4453
001116	000065			.WORD	65		
001120	000000G			.WORD	EH.7		
001122	000000G			.WORD	EMS.30		
001124	000427			BR	42:	:	4454
001126	020127	000003	39:	CMP	R1,#3	:	4400
001132	001005			BNE	40:		
001134	104456			TRAP	56	:	4457
001136	000066			.WORD	66		
001140	000000G			.WORD	EH.7		
001142	000000G			.WORD	EMS.30		
001144	000433			BR	44:	:	4458
001146	020127	000004	40:	CMP	R1,#4	:	4400
001152	001005			BNE	41:		
001154	104456			TRAP	56	:	4461
001156	000067			.WORD	67		
001160	000000G			.WORD	EH.7		
001162	000000G			.WORD	EMS.30		
001164	000423			BR	44:	:	4462
001166	020127	000005	41:	CMP	R1,#5	:	4400
001172	001011			BNE	43:		
001174	104456			TRAP	56	:	4465
001176	000070			.WORD	70		
001200	000000G			.WORD	EH.7		
001202	000000G			.WORD	EMS.30		
001204	013700	000000G	42:	MOV	T.ADDR,RO	:	4466
001210	105260	000062		INCB	62(RO)	:	
001214	000207			RTS	PC	:	4400
001216	020127	000006	43:	CMP	R1,#6	:	4400
001222	001011			BNE	45:		
001224	104456			TRAP	56	:	4469
001226	000071			.WORD	71		
001230	000000G			.WORD	EH.7		
001232	000000G			.WORD	EMS.30		
001234	013700	000000G	44:	MOV	T.ADDR,RO	:	4470
001240	105260	000063		INCB	63(RO)	:	
001244	000207			RTS	PC	:	4400
001246	104456		45:	TRAP	56	:	4473
001250	000072			.WORD	72		
001252	000000G			.WORD	EH.8		
001254	000000G			.WORD	EMS.30		
001256	013700	000000G	46:	MOV	CCTL,RO	:	4474
001262	006300			ASL	RO	:	
001264	105260	000000G		INCB	C.ERR.TBL(RO)	:	
001270	000207			RTS	PC	:	4272

C15

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 B11es-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (38)

SEQ 0391
Page 136

: Routine Size: 349 words, Routine Base: \$CODE\$ + 17424
: Maximum stack depth per invocation: 6 words

```

: 4479 routine UPD_IO_TALLY : novalue =
: 4480
: 4481 !!
: 4482 !! THIS ROUTINE IS CALLED FROM IO_RETPKT FOR ALL I/O TRANSFER RETURN
: 4483 !! PACKETS WITH "SUCCESS" STATUS CODES. ITS PURPOSE IS TO UPDATE ALL THE
: 4484 !! APPROPRIATE STATISTICAL FIELDS FOR THE CURRENT UNIT. A CHECK IS ALSO
: 4485 !! MADE ON THE TOTAL NUMBER OF BYTES TRANSFERRED THUS FAR; IF THE
: 4486 !! OPERATOR-SPECIFIED LIMIT HAS BEEN REACHED, THEN THE UNIT IS DROPPED.
: 4487 !!
: 4488 !! IMPLICIT INPUTS:
: 4489 !! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4490 !! T_ADDR - ADDRESS OF THE CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 4491 !! CST_ADDR - ADDRESS OF THE CURRENT CONTROLLER'S CST
: 4492 !! CUOFF - CST OFFSET FOR THE CURRENT UNIT
: 4493 !! L&LUN - CURRENT UNIT NUMBER
: 4494 !!-
:
: 4495
: 4496 begin
: 4497
: 4498 local
: 4499 THOUSANDS : word,
: 4500 MILLIONS : word;
: 4501
: 4502 if .RP_ADDR [ENDCOD] eq1 (OP_RD or OP_END)
: 4503 then
: 4504 begin
: 4505 T_ADDR [TOT_READS_LO] = .T_ADDR [TOT_READS_LO] + 1;
: 4506 T_ADDR [BYTES_READ_LO] = .T_ADDR [BYTES_READ_LO] + .RP_ADDR [BCNT_LO];
: 4507 T_ADDR [TOT_BYT_READ_LO] = .T_ADDR [TOT_BYT_READ_LO] + .RP_ADDR [BCNT_LO];
: 4508 OVF_CHK (T_ADDR [TOT_READS_LO]);
: 4509 OVF_CHK (T_ADDR [BYTES_READ_LO]);
: 4510 OVF_CHK (T_ADDR [TOT_BYT_READ_LO]);
: 4511 end
: 4512 else
: 4513
: 4514 if .RP_ADDR [ENDCOD] eq1 (OP_WRT or OP_END)
: 4515 then
: 4516 begin
: 4517 T_ADDR [TOT_WRITES_LO] = .T_ADDR [TOT_WRITES_LO] + 1;
: 4518 T_ADDR [BYTES_WRIT_LO] = .T_ADDR [BYTES_WRIT_LO] + .RP_ADDR [BCNT_LO];
: 4519 T_ADDR [TOT_BYT_WRT_LO] = .T_ADDR [TOT_BYT_WRT_LO] + .RP_ADDR [BCNT_LO];
: 4520 OVF_CHK (T_ADDR [TOT_WRITES_LO]);
: 4521 OVF_CHK (T_ADDR [BYTES_WRIT_LO]);
: 4522 OVF_CHK (T_ADDR [TOT_BYT_WRT_LO]);
: 4523 end;
: 4524
: 4525 if (.RP_ADDR [ENDCOD] eq1 (OP_RD or OP_END)) or
: 4526 (.RP_ADDR [ENDCOD] eq1 (OP_WRT or OP_END))
: 4527 then
: 4528 begin
: 4529 MILLIONS = .T_ADDR [MBYTES_READ] + .T_ADDR [MBYTES_WRT];
: 4530 THOUSANDS = .T_ADDR [BYTES_READ_HI] + .T_ADDR [BYTES_WRIT_HI];
: 4531

```

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (39)

SEQ 0393
Page 138

```

: 4532      if .THOUSANDS gequ 1000
: 4533      then
: 4534          begin
: 4535              MILLIONS = .MILLIONS + 1;           ! COUNT THE LOWER OVERFLOW TOO!
: 4536              THOUSANDS = .THOUSANDS - 1000;
: 4537          end;
: 4538
: 4539      !
: 4540      ! THIS ADDED BECAUSE IT WILL TAKE FOREVER TO TRANSFER ON THE ORDER OF A MEGABYTE TO A FLOPPY
: 4541      ! BUT IT IS A MUCH MORE REASONABLE MEASURE FOR THE RD51 WINCHESTER.  THE QUESTION NOW REFERS TO
: 4542      ! THE TOTAL DATA TRANSFER TO THE CONTROLLER AND THIS IS PRETTY CLOSE SINCE THE FLOPPIES GET
: 4543      ! ABOUT 1/1000 THE DATA THE HARD DISK(S) GET.
: 4544      !
: 4545
: 4546      if .SWP_XFER eq 0                               ! IF THERE IS A TRANSFER LIMIT
: 4547      then
: 4548          begin
: 4549              if .THOUSANDS gtru 100
: 4550              then
: 4551                  EOP_FLAG = TRUE;                   ! SET END-OF-PASS FLAG
: 4552              end
: 4553          else
: 4554              begin
: 4555                  if .MILLIONS gequ .SWP_XFER       ! IF TRANSFER LIMIT IS REACHED
: 4556                  then
: 4557                      EOP_FLAG = TRUE;               ! SET END-OF-PASS FLAG
: 4558                  end;
: 4559              end;
: 4560      end;                                           !end of read/write tallying
:                                                         ! ROUTINE UPD_IO_TALLY
    
```

				.SBTTL	UPD.IO.TALLY MULTI-DRIVE TEST ROUTINES		
000000	004137	000000G		UPD.IO.TALLY:			
000004	013701	000000G		JSR	R1, #SAVE2	4479	
000010	126127	000014	000241	MOV	RP, ADDR, R1	4502	
000016	001027			CMPB	14(R1), #241		
000020	013700	000000G		BNE	1#		
000024	005260	000016		MOV	T, ADDR, R0	4505	
000030	066110	000020		INC	16(R0)		
000034	066160	000020	000032	ADD	20(R1), (R0)	4506	
000042	012746	000016		ADD	20(R1), 32(R0)	4507	
000046	060016			MOV	#16, -(SP)	4508	
000050	004737	000000V		ADD	R0, (SP)		
000054	013716	000000G		JSR	PC, OVFLCHK		
000060	004737	000000V		MOV	T, ADDR, (SP)	4509	
000064	013716	000000G		JSR	PC, OVFLCHK		
000070	062716	000032		MOV	T, ADDR, (SP)	4510	
000074	000435			ADD	#32, (SP)		
000076	126127	000014	000242	BR	2#		
000104	001034			1#:	CMPB	14(R1), #242	4514
000106	013700	000000G		BNE	3#		
000112	005260	000024		MOV	T, ADDR, R0	4517	
				INC	24(R0)		

ZRQAM3 V01.2	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Bliss-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (39)	
000116	066160 000020 000006	ADD	20(R1),6(R0)	4518
000124	066160 000020 000040	ADD	20(R1),40(R0)	4519
000132	012746 000024	MOV	#24,-(SP)	4520
000136	060016	ADD	RO,(SP)	
000140	004737 000000V	JSR	PC,OVF.CHK	
000144	013716 000000G	MOV	T.ADDR,(SP)	4521
000150	062716 000006	ADD	#6,(SP)	
000154	004737 000000V	JSR	PC,OVF.CHK	
000160	013716 000000G	MOV	T.ADDR,(SP)	4522
000164	062716 000040	ADD	#40,(SP)	
000170	004737 000000V	JSR	PC,OVF.CHK	
000174	005726	TST	(SP),*	4516
000176	013700 000000G	MOV	RP.ADDR,RO	4525
000202	126027 000014 000241	CMPB	14(RO),#241	
000210	001404	BEQ	4#	
000212	126027 000014 000242	CMPB	14(RO),#242	4526
000220	001034	BNE	8#	
000222	013700 000000G	MOV	T.ADDR,RO	4529
000226	016002 000004	MOV	4(RO),R2	*,MILLIONS
000232	066002 000012	ADD	12(RO),R2	*,MILLIONS
000236	016001 000002	MOV	2(RO),R1	*,THOUSANDS
000242	066001 000010	ADD	10(RO),R1	*,THOUSANDS
000246	020127 001750	CMP	R1,#1750	THOUSANDS, *
000252	103403	BLO	5#	
000254	005202	INC	R2	MILLIONS
000256	162701 001750	SUB	#1750,R1	*,THOUSANDS
000262	013700 000000G	MOV	SWP.XFER,RO	
000266	001004	BNE	6#	
000270	020127 000144	CMP	R1,#144	THOUSANDS, *
000274	101406	BLOS	8#	
000276	000402	BR	7#	
000300	020200	CMP	R2,RO	MILLIONS, *
000302	103403	BLO	8#	
000304	112737 000001 000000G	MOVB	#1,EOP.FLAG	
000312	000207	RTS	PC	4557 4479

; Routine Size: 102 words, Routine Base: \$CODE\$ + 20716
; Maximum stack depth per invocation: 5 words

```

: 4561 routine OVF_CHK (ADDR) : novalue =
: 4562
: 4563 !!+
: 4564 !! THIS ROUTINE IS CALLED FROM UPD_IO_TALLY TO CHECK FOR OVERFLOW IN
: 4565 !! CERTAIN STATISTICAL FIELDS OF THE CURRENT UNIT. SPECIFICALLY, THE
: 4566 !! LOW-ORDER FIELD OF THE NUMBER OF BYTES READ OR WRITTEN IS CHECKED FOR
: 4567 !! EXCEEDING 1000. IF TRUE, THEN THE HIGH-ORDER COUNT IS INCREMENTED. IF
: 4568 !! THAT EXCEEDS 1000, THEN THE MEGABYTE COUNT IS INCREMENTED.
: 4569 !!
: 4570 !! INPUTS:
: 4571 !! ADDR - ADDRESS OF THE BYTES_READ_LO OR BYTES_WRIT_LO FIELD FOR
: 4572 !! THE CURRENT UNIT (SEE STATISTIC TABLE (TALLY) LAYOUT)
: 4573 !!-
: 4574
: 4575 begin
: 4576
: 4577 while ..ADDR gequ 1000 do ! IF LO-ORDER OVERFLOW
: 4578 begin
: 4579 .ADDR = ..ADDR - 1000; ! SUBTRACT 1000
: 4580 (.ADDR + 2) = .(.ADDR + 2) + 1; ! INCR HI-ORDER
: 4581 end;
: 4582
: 4583 if .(.ADDR + 2) gequ 1000 ! IF HI-ORDER OVERFLOW
: 4584 then
: 4585 begin
: 4586 (.ADDR + 2) = .(.ADDR + 2) - 1000; ! SUBTRACT 1000
: 4587 (.ADDR + 4) = .(.ADDR + 4) + 1; ! INCREMENT MBYTES
: 4588 end;
: 4589
: 4590 end; ! ROUTINE OVF_CHK
    
```

		.SBTTL	OVF.CHK MULTI-DRIVE TEST ROUTINES	
000000	010146		OVF.CHK:MOV R1, -(SP)	4561
000002	016600	000004	MOV 4(SP), R0	4577
000006	012701	000002	MOV #2, R1	4580
000012	060001		ADD R0, R1	
000014	021027	001750	1\$: CMP (R0), #1750	4577
000020	103404		BLO 2\$	
000022	162710	001750	SUB #1750, (R0)	4579
000026	005211		INC (R1)	4580
000030	000771		BR 1\$	4577
000032	021127	001750	2\$: CMP (R1), #1750	4583
000036	103404		BLO 3\$	
000040	162711	001750	SUB #1750, (R1)	4586
000044	005260	000004	INC 4(R0)	4587
000050	012601		3\$: MOV (SP)+, R1	4561
000052	000207		RTS PC	

; Routine Size: 22 words, Routine Base: \$CODE\$ + 21232
 ; Maximum stack depth per invocation: 2 words

```

: 4591 routine HOST_WRT_CHK =
: 4592
: 4593 !+
: 4594 ! THIS ROUTINE IS CALLED FROM IO_RETPKT FOR ALL I/O TRANSFER RETURN
: 4595 ! PACKETS WITH "SUCCESS" STATUS CODES, BUT ONLY IF THE HOST WRITE-COMPARE
: 4596 ! OPTION WAS SELECTED BY THE OPERATOR.
: 4597 !
: 4598 ! IF THE CURRENT RETPKT BEING PROCESSED IS A WRITE FUNCTION, THEN THE
: 4599 ! PACKET INDEX (RP_INDX) IS SAVED IN THE CONTROLLER'S RETURN PACKET SAVE
: 4600 ! AREA (RP_SAVE). OTHERWISE, THE PACKET IS A READ, SO ITS ASSOCIATED
: 4601 ! WRITE PACKET IS REMOVED FROM THE SAVE AREA, AND A BYTE-BY-BYTE
: 4602 ! COMPARISON IS PERFORMED ON THE TWO I/O BUFFERS. ANY DIFFERENCES
: 4603 ! ENCOUNTERED RESULTS IN THE DECLARATION OF A HARD ERROR.
: 4604 !
: 4605 ! IMPLICIT INPUTS:
: 4606 ! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4607 ! RP_INDX - INDEX OF THE CURRENT RETURN PACKET
: 4608 !-
: 4609
: 4610 begin
: 4611
: 4612 local
: 4613   BUFF1 : ref block [MAX_XFER * 2, byte], ! I/O BUFFER ADDRESS
: 4614   BUFF2 : ref block [MAX_XFER * 2, byte], ! I/O BUFFER ADDRESS
: 4615   BUFFW, ! I/O BUFFER ADDRESS
: 4616   COUNT : word, ! BYTE COUNT
: 4617   FLAG : byte initial (byte (TRUE)),
: 4618   index : signed word;
: 4619
: 4620 if .RP_ADDR [ENDCOD] eq1 (OP_WRT or OP_END) ! IF WRITE OPERATION
: 4621 then
: 4622   FLAG = FALSE ! DON'T CALL SWEEP FROM IO_RETPKT
: 4623 else
: 4624   if (.RP_ADDR [ENDCOD] eq1 (OP_RD or OP_END)) and
: 4625     ((index = RPS_REM ()) geq 0) ! IF ASSOCIATED WRITE PACKET IS FOUND ELSE ENCODE IS READ
: 4626   then
: 4627     begin
: 4628       BUFFW = RETPKT [.index, BUFF_0]; ! ADDR OF ADDR OF WRITE I/O BUFFER
: 4629       BUFF1 = ..BUFFW; ! ADDR OF WRITE I/O BUFFER
: 4630       BUFF2 = .RP_ADDR [BUFF_0]; ! ADDR OF READ I/O BUFFER
: 4631       COUNT = .RP_ADDR [BCNT_LO]; ! BYTE COUNT
: 4632
: 4633       incr I from 1 to .COUNT do ! FOR EACH BYTE IN BUFFERS
: 4634
: 4635         if .(.BUFF1)<0, 8, 0> eq1 .(.BUFF2)<0, 8, 0> ! IF BYTES COMPARE O.K.
: 4636         then
: 4637           begin
: 4638             BUFF1 = .BUFF1 + 1; ! ADVANCE WRITE BUFFER ADDR
: 4639             BUFF2 = .BUFF2 + 1; ! ADVANCE READ BUFFER ADDR
: 4640           end
: 4641         else
: 4642           begin ! ELSE - COMPARE ERROR
: 4643             T_ADDR [ERR_HARD] = .T_ADDR [ERR_HARD] + 1;

```

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0397
Page 142
VAX-11 Bliss-16 V3-555
DISK#USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (41)

```

:      4644      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
:      4645      ERRHRD (42, EGH_30, 0);      ! I/O REQUEST FAILED
:      4646      EMS_CMP (RETPKT + (.index * RP_LEN * 2));
:      4647
:      4648      if .T_ADDR [ERR_HARD] gequ .SWP_ERROR
:      4649      then
:      4650      begin
:      4651      DUR [.L$LUN] = DU_HERR; ! IF ERROR COUNT EXCEEDED
:      4652      DODU (.L$LUN);      ! DROP UNIT
:      4653      end;
:      4654
:      4655      exitloop;      ! NO NEED TO CONTINUE
:      4656      end;      ! IF COMPARE ERROR
:      4657
:      4658      end;      ! IF ASSOCIATED WRITE RETPKT WAS FOUND
:      4659
:      4660      return (.FLAG);
:      4661      end;      ! ROUTINE HOST_WRT_CHK
    
```

Address	Label	OpCode	Comment	Address
000000	004137	000000G	.SBTTL HOST.WRT.CHK: HOST.WRT.CHK MULTI-DRIVE TEST ROUTINES	
000004	005746		JSR R1, \$SAVES	4591
000006	112705	000001	TST -(SP)	
000012	013700	000000G	MOVB #1, R5 ; *,FLAG	4610
000016	126027	000014 000242	MOV RP.ADDR, R0 ;	4620
000024	001002		CMPB 14(R0), #242	
000026	105005		BNE 1#	
000030	000500		CLRB R5 ; FLAG	4622
000032	126027	000014 000241	BR 6# ;	4620
000040	001074		1# : CMPB 14(R0), #241 ;	4624
000042	004737	000000V	BNE 6# ;	
000046	005700		JSR PC, RPS.REM ;	4625
000050	002470		TST R0 ; INDEX	
000052	010046		BLT 6# ; INDEX, *	4628
000054	012746	000060	MOV R0, -(SP)	
000060	004737	000000G	MOV #60, -(SP)	
000064	010066	000004	JSR PC, BL \$MUL	
000070	062700	000024G	MOV R0, 4(SP)	
000074	011002		ADD #RETPKT+24, R0 ; *,BUFFW	
000076	013700	000000G	MOV (R0), R2 ; BUFFW, BUFF1	4629
000102	016003	000024	MOV RP.ADDR, R0 ;	4630
000106	016004	000020	MOV 24(R0), R3 ; *,BUFF2	
000112	005001		MOV 20(R0), R4 ; *,COUNT	4631
000114	000442		CLR R1 ; I	4633
000116	121213		BR 4# ;	
000120	001003		2# : CMPB (R2), (R3) ; BUFF1, BUFF2	4635
000122	005202		BNE 3# ;	
000124	005203		INC R2 ; BUFF1	4638
000126	000435		INC R3 ; BUFF2	4639
000130	013700	000000G	BR 4# ;	4635
000134	005260	000014	3# : MOV T.ADDR, R0 ;	4643
			INC 14(R0) ;	

ZRQAM3	RD/RX EXERCISER	14-Dec-1983 16:12:07	VAX-11 Bliss-16 V3-555	SEQ 0398
V01.2	MULTI-DRIVE TEST ROUTINES	14-Dec-1983 16:12:00	DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (41)	Page 143
000140	105260 000063	INCB	63(R0)	4644
000144	104456	TRAP	56	4645
000146	000052	.WORD	52	
000150	000000G	.WORD	EGH.30	
000152	000000	.WORD	0	
000154	016616 000004	MOV	4(SP),(SP)	4646
000160	062716 000000G	ADD	#RETPKT,(SP)	
000164	004737 000000G	JSR	PC,EMS.CMP	
000170	013700 000000G	MOV	T.ADDR,R0	4648
000174	026037 000014 000000G	CMP	14(R0),SWP.ERROR	
000202	103412	BLO	5#	
000204	013700 000000G	MOV	L#LUN,R0	4651
000210	112760 000004 000000G	MOVB	#4,DUR(R0)	
000216	104451	TRAP	51	4652
000220	000403	BR	5#	4642
000222	005201	INC	R1	4633
000224	020104	CMP	R1,R4	
000226	003733	BLE	2#	
000230	022626	CMP	(SP)+,(SP)+	4627
000232	005000	CLR	R0	4610
000234	150500	BISB	R5,R0	
000236	005726	TST	(SP)+	
000240	000207	RTS	PC	4591

: Routine Size: 81 words, Routine Base: #CODE# + 21306
 : Maximum stack depth per invocation: 11 words

```

: 4662 routine SWEEP : novalue =
: 4663
: 4664 !*
: 4665 ! THIS ROUTINE IS CALLED FROM IO_RETPKT AND OTHERS TO DEALLOCATE THE
: 4666 ! RESOURCES ASSOCIATED WITH THE CURRENT RETURN PACKET. THIS INCLUDES THE
: 4667 ! PACKET ITSELF AND THE I/O BUFFER. IN ADDITION, IF THE HOST IS
: 4668 ! PERFORMING WRITE-COMPARES, AND IF THE CURRENT RETURN PACKET IS A READ
: 4669 ! FUNCTION, THEN THE CURRENT CONTROLLER'S RP_SAVE AREA IS SEARCHED FOR
: 4670 ! THE ASSOCIATED WRITE RETPKT SO THAT ITS RESOURCES CAN ALSO BE
: 4671 ! DEALLOCATED.
: 4672 !
: 4673 ! IMPLICIT INPUTS:
: 4674 ! RP_ADDR - ADDRESS OF CURRENT RETURN PACKET
: 4675 ! RP_INDX - INDEX OF CURRENT RETURN PACKET
: 4676 !-
: 4677
: 4678 begin
: 4679
: 4680 local
: 4681 index : signed word;
: 4682
: 4683 if (.RP_ADDR [ENDCOD] and OP_MSK) eal OP_RD ! IF READ OPCODE OR ENDCODE
: 4684 then
: 4685
: 4686 if BIT_TST (SWP_FLAGS, SWF_HWC) ! IF HOST IS DOING WRITE-COMPARES
: 4687 then
: 4688
: 4689 if (index = RPS_REM ()) geq 0 ! IF ASSOCIATED WRITE RETPKT IS FOUND
: 4690 then
: 4691 begin
: 4692 PUT_IO_BUFF (RETPKT [.index, BUFF_0]); ! RETURN WRITE I/O BUFFER TO POOL
: 4693 PUT_RETPKT (.index); ! RETURN WRITE PACKET TO POOL
: 4694 end;
: 4695
: 4696 PUT_IO_BUFF (RP_ADDR [BUFF_0]); ! RETURN CURRENT I/O BUFFER TO POOL
: 4697 PUT_RETPKT (.RP_INDX); ! RETURN CURRENT RETPKT TO POOL
: 4698 end; ! ROUTINE SWEEP
    
```

			.SBTTL	SWEEP MULTI-DRIVE TEST ROUTINES	
000000	010146		SWEEP:	MOV R1, -(SP)	4662
000002	013700	000000G		MOV RP_ADDR, R0	4683
000006	116000	000014		MOVB 14(R0), R0	
000012	042700	177600		BIC #177600, R0	
000016	020027	000041		CMP R0, #41	
000022	001026			BNE 1\$	
000024	032737	000040 000000G		BIT #40, SWP_FLAGS	4686
000032	001422			BEQ 1\$	
000034	004737	000000V		JSR PC, RPS.REM	4689
000040	010001			MOV R0, R1	! *, INDEX
000042	002416			BLT 1\$	
000044	010146			MOV R1, -(SP)	! INDEX, *
000046	012746	000060		MOV #60, -(SP)	4692

L15

000052	004737	000000G		JSR	PC,BL\$MUL		
000056	062700	000024G		ADD	\$RETPKT+24,R0		
000062	010016			MOV	R0,(SP)		
000064	004737	000000G		JSR	PC,PUT.IO.BUFF		
000070	010116			MOV	R1,(SP)	; INDEX,*	4693
000072	004737	000000G		JSR	PC,PUT.RETPKT		
000076	022626			CMP	(SP)+,(SP)+		4691
000100	013746	000000G	1\$:	MOV	RP.ADDR,-(SP)		4696
000104	062716	000024		ADD	\$24,(SP)		
000110	004737	000000G		JSR	PC,PUT.IO.BUFF		
000114	013716	000000G		MOV	RP.INDX,(SP)		4697
000120	004737	000000G		JSR	PC,PUT.RETPKT		
000124	005726			TST	(SP)+		4678
000126	012601			MOV	(SP)+,R1		4662
000130	000207			RTS	PC		

: Routine Size: 45 words, Routine Base: \$CODE\$ + 21550
 : Maximum stack depth per invocation: 4 words

```

: 4699 routine RPS_REM =
: 4700
: 4701 !+
: 4702 ! THIS ROUTINE SEARCHES THE CURRENT CONTROLLER'S RP_SAVE AREA FOR A
: 4703 ! RETURN PACKET WHOSE COMMAND REFERENCE NUMBER (CRN) IS ONE LESS THAN THE
: 4704 ! CRN OF THE CURRENT RETURN PACKET (I.E., SEARCHING FOR THE SAVED WRITE
: 4705 ! OPERATION ASSOCIATED WITH THE CURRENT READ OPERATION). IF FOUND, THE
: 4706 ! RP_SAVE ENTRY IS CLEARED (TO -1) AND THE RETPKT INDEX OF THE WRITE
: 4707 ! OPERATION IS RETURNED TO THE CALLER.
: 4708 !
: 4709 ! IMPLICIT INPUTS:
: 4710 !     RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4711 !
: 4712 ! OUTPUTS:
: 4713 !     INDEX (VALUE OF THIS ROUTINE) - INDEX OF THE RETPKT CONTAINING
: 4714 !     A CRN WHICH IS ONE LESS THAN THE CURRENT
: 4715 !-
: 4716
: 4717 begin
: 4718
: 4719 local
: 4720     index : signed word initial (-1);                                ! ASSUME NOT FOUND
: 4721
: 4722 incr COUNT from 0 to RP_CNT - 1 do                                ! FOR EACH ENTRY IN RP_SAVE
: 4723
: 4724     if (.RP_USE [.COUNT] eq1 .CCTLR) and                            ! IF THIS IS A VALID RETPKT
: 4725         (.RETPKT [.COUNT, ENDCOD] eq1 (OP_WRT or OP_END))
: 4726     then
: 4727
: 4728         if ((.RETPKT [.COUNT, CRF_LO] eq1 (.RP_ADDR [CRF_LO] - 1)) and ! IF CORRECT CRN
: 4729             (.RETPKT [.COUNT, CRF_HI] eq1 .RP_ADDR [CRF_HI])) or
: 4730             ((.RETPKT [.COUNT, CRF_HI] eq1 (.RP_ADDR [CRF_HI] - 1)) and
: 4731              (.RETPKT [.COUNT, CRF_LO] eq1 #o'177777') and
: 4732              (.RP_ADDR [CRF_LO] eq1 0))
: 4733         then
: 4734             begin
: 4735                 index = .COUNT;                                ! INDEX TO BE RETURNED
: 4736                 exitloop;                                       ! DONE
: 4737             end;
: 4738
: 4739 return .index;
: 4740 end;                                                                ! ROUTINE RPS_REM

```

000000	004137	000000G	.SBTTL	RPS.REM MULTI-DRIVE TEST ROUTINES	
000004	012704	177777	RPS.REM: JSR	R1, #SAVE4	4699
000010	005003		MOV	#-1, R4	: *.INDEX
000012	116300	000000G	CLR	R3	: COUNT
000016	020037	000000G	1\$: MOV	RP.USE(R3), R0	: *(COUNT), *
000022	001053		CMP	R0, CCTLR	
000024	010346		BNE	4\$	
000026	012746	000060	MOV	R3, -(SP)	: COUNT, *
			MOV	#60, -(SP)	4725

N15

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0402
Page 147

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (43)

000032	004737	000000G		JSR	PC, BL \$MUL		
000036	022626			CMP	(SP)+, (SP)+		
000040	126027	000014G	000242	CMPB	RETPKT+14(R0), #242		
000046	001041			BNE	4\$		
000050	010346			MOV	R3, -(SP)	; COUNT, *	4728
000052	012746	000060		MOV	#60, -(SP)		
000056	004737	000000G		JSR	PC, BL \$MUL		
000062	022626			CMP	(SP)+, (SP)+		
000064	013701	000000G		MOV	RP.ADDR, R1		
000070	016102	000004		MOV	4(R1), R2		
000074	005302			DEC	R2		
000076	026002	000004G		CMP	RETPKT+4(R0), R2		
000102	001004			BNE	2\$		
000104	026061	000006G	000006	CMP	RETPKT+6(R0), 6(R1)	;	4729
000112	001415			BEQ	3\$		
000114	016102	000006	2\$:	MOV	6(R1), R2	;	4730
000120	005302			DEC	R2		
000122	026002	000006G		CMP	RETPKT+6(R0), R2		
000126	001011			BNE	4\$		
000130	026027	000004G	177777	CMP	RETPKT+4(R0), #-1	;	4731
000136	001005			BNE	4\$		
000140	005761	000004		TST	4(R1)	;	4732
000144	001002			BNE	4\$		
000146	010304		3\$:	MOV	R3, R4	; COUNT, INDEX	4735
000150	000404			BR	5\$;	4734
000152	005203		4\$:	INC	R3	; COUNT	4722
000154	020327	000003		CMP	R3, #3	; COUNT, *	
000160	003714			BLE	1\$		
000162	010400		5\$:	MOV	R4, R0	; INDEX, *	4717
000164	000207			RTS	PC	;	4699

; Routine Size: 59 words, Routine Base: \$CODE\$ + 21702
; Maximum stack depth per invocation: 8 words

ZRQAM3
V01.2

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0403
Page 148
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2:161 (44)

```

: 4741 routine DR_RETPKT : novalue =
: 4742
: 4743 !.
: 4744 ! THIS ROUTINE IS CALLED BY PROC_RETPKT FOR ALL PACKETS ORIGINATING AT
: 4745 ! THE "DRIVER" PORTION OF THE PROGRAM. THIS INCLUDES PACKETS DESCRIBING
: 4746 ! FATAL DEVICE ERRORS.
: 4747 !
: 4748 ! FOR FATAL DEVICE ERRORS, THIS ROUTINE RELEASES ALL RESOURCES HELD BY
: 4749 ! THE CONTROLLER. THE CONTROLLER IS MARKED OFFLINE IN ITS CST, AND ALL
: 4750 ! UNITS ATTACHED TO THE CONTROLLER ARE DROPPED.
: 4751 !
: 4752 ! IMPLICIT INPUTS:
: 4753 ! RP_INDX - INDEX OF THE CURRENT RETURN PACKET
: 4754 ! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4755 ! CST_ADDR - ADDRESS OF THE CURRENT CONTROLLER'S CST
: 4756 ! CCTLN - CURRENT CONTROLLER NUMBER
: 4757 !-
: 4758
: 4759 begin
: 4760
: 4761
: 4762 PUTA_BUFF (); : RELEASE ALL I/O BUFFERS HELD BY CONTROLLER
: 4763
: 4764 incr index from 0 to RP_CNT - 1 do : FOR EACH ENTRY IN CONTROLLER'S RP_SAVE
: 4765
: 4766 if .RP_USE [.index] eql .CCTLN : IF VALID RETPKT INDEX
: 4767 then
: 4768 PUT_RETPKT (.index); : RETURN RETPKT TO POOL
: 4769
: 4770 QIO [.CCTLN] = 0; : CLEAR NO. OF OUTSTANDING QIOS
: 4771 CST_ADDR [STATE] = OFFLINE; : MARK CST OFFLINE
: 4772 DROP_CTLN (.CCTLN, DU_CFATAL); : DROP CONTROLLER'S UNITS
: 4773 PUT_RETPKT (.RP_INDX); : PUT BACK RETPKT
: 4774 end; : ROUTINE DR_RETPKT

```

```

000000 010146 .SBTTL DR.RETPKT MULTI-DRIVE TEST ROUTINES
DR.RETPKT:
000002 004737 000000G MOV R1,-(SP) ; 4741
000006 005001 JSR PC,PUTA.BUFF ; 4762
000010 116100 000000G CLR R1 ; INDEX 4764
000014 020037 000000G 1$: MOVB RP.USE(R1),R0 ; *(INDEX),* 4766
000020 001004 CMP R0,CCTLN
000022 010146 BNE 2$
000024 004737 000000G MOV R1,-(SP) ; INDEX,* 4768
000030 005726 JSR PC,PUT.RETPKT
000032 005201 TST (SP)*
000034 020127 000003 2$: INC R1 ; INDEX 4764
000040 003763 CMP R1,#3 ; INDEX,*
000042 013701 000000G BLE 1$
000046 105061 000000G MOV CCTLN,R1 ; 4770
000052 013700 000000G CLRB QIO(R1)
MOV CST.ADDR,R0 ; 4771

```

C16

000056	042760	100000	000002	BIC	#100000,2(R0)		
000064	010146			MOV	R1,-(SP)	:	4772
000066	012746	000006		MOV	#6,-(SP)		
000072	004737	000000G		JSR	PC,DROP.CTLR		
000076	013716	000000G		MOV	RP,INDX,(SP)	:	4773
000102	004737	000000G		JSR	PC,PUT.RETPKT		
000106	022626			CMP	(SP),,(SP),	:	4759
000110	012601			MOV	(SP),,R1	:	4741
000112	000207			RTS	PC		

: Routine Size: 38 words, Routine Base: \$CODE\$ + 22070
: Maximum stack depth per invocation: 4 words

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0405
Page 150
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (45)

```

:      4775  #sbttl 'RDRX INTERRUPT SERVICE ROUTINES'
:      4776
:      4777  !!
:      4778  !
:      4779  !   THERE EXISTS AN RDRX INTERRUPT SERVICE ROUTINE FOR EACH DEVICE
:      4780  !   CONTROLLER. EACH SERVICE ROUTINE BEGINS BY SIMPLY SETTING THE
:      4781  !   APPROPRIATE CONTROLLER NUMBER INTO "ICTLR". ALL SERVICE ROUTINES THEN
:      4782  !   BRANCH TO A COMMON INTERRUPT PROCESSING ROUTINE.
:      4783  !-
:      4784  BGNSRV (AZINTO);
:      4785  ICTLR = 0;
:      4786  AZINT ();
:      4787  ENDSRV;

```

```

000000 010046          .SBTTL  AZINTO RDRX INTERRUPT SERVICE ROUTINES
000002 005037 000076'  AZINTO::MOV   RO,-(SP)      ;
000006 004737 000000V  CLR     ICTLR           ;
000012 012600          JSR     PC,AZINT          ;
000014 000002          MOV     (SP),RO          ;
                                RTI

```

```

: Routine Size: 7 words,      Routine Base: %CODE$ + 22204
: Maximum stack depth per invocation: 2 words

```



```

: 4788 routine AZINT : novalue =
: 4789
: 4790 !!+
: 4791 !! THIS IS THE COMMON INTERRUPT SERVICE ROUTINE FOR ALL RDRX CONTROLLERS.
: 4792 !! AFTER CALCULATING THE DCT ADDRESS FOR THE INTERRUPTING DEVICE, THIS
: 4793 !! ROUTINE WILL SAVE THE CURRENT CONTENTS OF THE SA REGISTER IN THE DCT.
: 4794 !! THEN, IF THE "IGNORE INTERRUPT" BIT IS SET, NO FURTHER ACTION IS TAKEN.
: 4795 !! OTHERWISE, THE SA VALUE IS CHECKED FOR A FATAL ERROR, AND THE COMMAND
: 4796 !! AND RESPONSE RINGS ARE POLLED.
: 4797 !!-
: 4798
: 4799 begin
: 4800 IDCT_ADDR = DCT + (.ICTLR * DCT_LEN * 2); ! GET DCT ADDRESS
: 4801 ICST_ADDR = CST + (.ICTLR * CST_LEN * 2); ! GET CST ADDRESS
: 4802 IRDRX_ADDR = .ICST_ADDR [IP_ADDR]; ! GET RDRX ADDRESS
: 4803 ICOM_ADDR = COMM_AREA + (.ICTLR * COMM_LEN * 2); ! GET COMM_AREA ADDR
: 4804 IDCT_ADDR [SA_SAVE] = .IRDRX_ADDR [RCSA, RC_ALL]; ! SAVE SA REGISTER
: 4805
: 4806 if .IDCT_ADDR [IG_INT] ! IGNORE INTERRUPT?
: 4807 then
: 4808 return; ! RETURN IF INTERRUPTS IGNORED
: 4809
: 4810 if BIT_TST (IDCT_ADDR [SA_SAVE], SA_ERR) ! IF FATAL ERROR
: 4811 then
: 4812 FATAL_ERROR ()
: 4813 else
: 4814 begin
: 4815 POLL_CRING (); ! POLL COMMAND RING
: 4816 POLL_RRING (); ! POLL RESPONSE RING
: 4817 end;
: 4818
: 4819 end;

```

000000	010146		AZINT:	MOV	R1, -(SP)		4788
000002	005746			TST	-(SP)		
000004	013701	000076'		MOV	ICTLR, R1		4800
000010	010146			MOV	R1, -(SP)		
000012	012746	000022		MOV	#22, -(SP)		
000016	004737	000000G		JSR	PC, BL#MUL		
000022	062700	000000G		ADD	#DCT, R0		
000026	010037	000000G		MOV	R0, IDCT.ADDR		
000032	010116			MOV	R1, (SP)		4801
000034	012746	000076		MOV	#76, -(SP)		
000040	004737	000000G		JSR	PC, BL#MUL		
000044	062700	000000G		ADD	#CST, R0		
000050	010037	000000G		MOV	R0, ICST.ADDR		
000054	011037	000000G		MOV	(R0), IRDRX.ADDR	; ICST.ADDR, *	4802
000060	010116			MOV	R1, (SP)		4803
000062	012746	000050		MOV	#50, -(SP)		
000066	004737	000000G		JSR	PC, BL#MUL		
000072	062700	000000'		ADD	#COMM.AREA, R0		

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0407
Page 152
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (46)

000076	010037	000000G		MOV	RO,ICOM.ADDR		
000102	013700	000000G		MOV	IDCT.ADDR,R0	:	
000106	013701	000000G		MOV	IRDRX.ADDR,R1	:	4804
000112	016166	000002	000010	MOV	2(R1),10(SP)	:	
000120	016660	000010	000002	MOV	10(SP),2(R0)	:	*,RC.REG
000126	032710	040000		BIT	#40000,(R0)	:	RC.REG,*
000132	001016			BNE	2#	:	*,IDCT.ADDR
000134	016600	000010		MOV	10(SP),R0	:	4806
000140	042700	077777		BIC	#77777,R0	:	4788
000144	020027	100000		CMP	R0,#-100000	:	4810
000150	001003			BNE	1#	:	
000152	004737	000000V		JSR	PC,FATAL.ERROR	:	
000156	000404			BR	2#	:	4812
000160	004737	000000V	1#:	JSR	PC,POLL.CRING	:	4810
000164	004737	000000V		JSR	PC,POLL.RRING	:	4815
000170	062706	000012	2#:	ADD	#12,SP	:	4816
000174	012601			MOV	(SP)+,R1	:	4788
000176	000207			RTS	PC	:	

: Routine Size: 64 words, Routine Base: \$CODE\$ + 2222
: Maximum stack depth per invocation: 7 words

```

: 4820 routine FATAL_ERROR : novalue =
: 4821
: 4822 !+
: 4823 ! THIS ROUTINE IS CALLED BY THE INTERRUPT SERVICE ROUTINE (AZINT) UPON
: 4824 ! DETECTING AN UNRECOVERABLE ERROR THROUGH THE DEVICE'S SA REGISTER.
: 4825 ! ITS PURPOSE IS TO CLEAN UP DEVICE DATA IN THE "DRIVER" PORTION OF
: 4826 ! THE EXERCISER, AND TO INFORM THE "PROGRAM" PORTION OF THE EVENT VIA
: 4827 ! RETURN PACKET.
: 4828
: 4829 ! IMPLICIT INPUTS:
: 4830 ! ICTLR - INTERRUPTING CONTROLLER NUMBER
: 4831 ! IDCT_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 4832 ! ICST_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S CST
: 4833 ! IRDRX_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S IP REGISTER
: 4834 !-
: 4835
: 4836 begin
: 4837
: 4838 local
: 4839     index : signed word,
: 4840     U_SAVE : word;
: 4841
: 4842     SA_REG = .IDCT_ADDR [SA_SAVE];
: 4843     U_SAVE = .L$LUN;           ! SAVE PRE-INTERRUPT CURRENT UNIT NUMBER
: 4844     C_ERR_TBL [.ICTLR, C_ERR_HRD] = .C_ERR_TBL [.ICTLR, C_ERR_HRD] + 1; ! SET CURRENT UNIT TO FIRST IN CONTROLLER
: 4845     L$LUN = .ICST_ADDR [OF_UN, D_UNIT]; ! SET CURRENT UNIT TO FIRST IN CONTROLLER
: 4846     ERRDF (14, EGD_14, EMS_14);      ! FATAL CONTROLLER ERROR
: 4847     L$LUN = .U_SAVE;               ! RESTORE PRE-INTERRUPT CURRENT UNIT
: 4848     DRV_CTLERR (.ICTLR);           ! CLEAN UP DRIVER DATA FOR CONTROLLER
: 4849
: 4850     if (index = GET_RETPKT (.ICTLR)) les 0 ! TRY TO GET A RETPKT; IF FAILURE
: 4851     then
: 4852         PRINTF (DBM18) ! "FATAL_ERROR: RETPKT NOT AVAILABLE"
: 4853     else
: 4854         begin ! IF RETPKT WAS ALLOCATED
: 4855             RETPKT [.index, CONID] = CID_DRIVER; ! SET CONNECTION ID TO "DRIVER"
: 4856             RETPKT [.index, MESTYP] = MT_FATAL; ! FATAL ERROR
: 4857             RETPKT [.index, CTLR] = .ICTLR; ! CONTROLLER NUMBER
: 4858             IN_IODQ (.index); ! LOAD RETPKT INDEX INTO IODQ
: 4859             end; ! IF RETPKT WAS ALLOCATED
: 4860
: 4861     end; ! ROUTINE FATAL_ERR

```

000000	004137	000000G	.SBTTL	FATAL.ERROR RDRX INTERRUPT SERVICE ROUTINES	
			FATAL.ERROR:		
000004	013700	000000G	JSR	R1, \$SAVE2	4820
000010	016037	000002 000000G	MOV	IDCT.ADDR, R0	4842
000016	013701	000000G	MOV	2(R0), SA.REG	
000022	013700	000076'	MOV	L\$LUN, R1	4843
000026	006300		MOV	ICTLR, R0	4844
000030	105260	000000G	ASL	R0	
			INCB	C.ERR.TBL(R0)	

H16

ZRQAM3 V01.2	RD/RX EXERCISER RDRX INTERRUPT SERVICE ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Bliss-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (47)	SEQ 0409 Page 154
000034	013700	000000G	MOV ICST.ADDR,R0	4845
000040	016002	000006	MOV 6(R0),R2	
000044	000302		SWAB R2	
000046	042702	177760	BIC #177760,R2	
000052	010237	000000G	MOV R2,L\$LUN	
000056	104455		TRAP 55	4846
000060	000016		.WORD 16	
000062	000000G		.WORD EGD.14	
000064	000000G		.WORD EMS.14	
000066	010137	000000G	MOV R1,L\$LUN	4847
000072	013746	000076'	MOV ICTLR,-(SP)	4848
000076	004737	000000G	JSR PC,DRV.CTLERR	
000102	013716	000076'	MOV ICTLR,(SP)	4850
000106	004737	000000G	JSR PC,GET.RETPKT	
000112	010001		MOV RO,R1	4852
000114	002007		BGE 1#	
000116	012716	000000G	MOV #DBM18,(SP)	4852
000122	012746	000001	MOV #1,-(SP)	
000126	010600		MOV SP,RO	4850
000130	104417		TRAP 17	
000132	000424		BR 2#	4855
000134	010116		MOV R1,(SP)	4855
000136	012746	000060	MOV #60,-(SP)	
000142	004737	000000G	JSR PC,BL\$MUL	
000146	062700	000002G	ADD #RETPKT+2,RO	
000152	112760	000003 000001	MOVB #3,1(RO)	
000160	013702	000076'	MOV ICTLR,R2	4857
000164	042702	177760	BIC #177760,R2	
000170	112710	000060	MOVB #60,(RO)	
000174	150210		BISB R2,(RO)	
000176	010116		MOV R1,(SP)	4858
000200	004737	000000G	JSR PC,IN.IODQ	
000204	022626		CMP (SP)+,(SP)+	4836
000206	000207		RTS PC	4820

; Routine Size: 68 words, Routine Base: \$CODE\$ + 22422
 ; Maximum stack depth per invocation: 7 words

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0410
Page 155
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (48)

```

: 4862 routine POLL_CRING : novalue =
: 4863
: 4864 !!
: 4865 !! THIS ROUTINE IS CALLED BY THE RDRX INTERRUPT SERVICE ROUTINE (AZINT)
: 4866 !! FOR EACH DEVICE INTERRUPT EXCEPT DURING INITIALIZATION OR FATAL ERROR.
: 4867 !! ITS PURPOSE IS TO SCAN THE DEVICE'S COMMAND RING AND CHECK FOR ANY
: 4868 !! COMMAND SLOTS THAT HAVE BEEN "TAKEN" BY THE CONTROLLER. SUCH SLOTS
: 4869 !! HAVE BEEN RETURNED TO THE HOST, INDICATED BY A ZERO OWNERSHIP BIT. FOR
: 4870 !! EACH SLOT THAT HAS BEEN RETURNED TO THE HOST, THE CRING COUNT IS
: 4871 !! DECREMENTED, AND THE CR_POLL ADDRESS IS ADVANCED TO THE NEXT SLOT IN
: 4872 !! THE COMMAND RING.
: 4873 !!
: 4874 !! IMPLICIT INPUTS:
: 4875 !! ICTLR - INTERRUPTING CONTROLLER NUMBER
: 4876 !! IDCT_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 4877 !! ICOM_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S COMM_AREA
: 4878 !!-
: 4879
: 4880 begin
: 4881
: 4882 while ((.IDCT_ADDR [CRING_CNT] gtru 0) and ! WHILE # OF COMMANDS IN CRING > 0 AND
: 4883 not (BIT_TST ((.IDCT_ADDR [CR_POLL] + 2), ED_OWN))) do ! CURRENT SLOT IS HOST-OWNED
: 4884 begin
: 4885 IDCT_ADDR [CRING_CNT] = .IDCT_ADDR [CRING_CNT] - 1; ! DECREMENT # CMDS IN CRING
: 4886 IDCT_ADDR [CR_POLL] = .IDCT_ADDR [CR_POLL] + 4; ! ADVANCE TO NEXT SLOT TO POLL
: 4887
: 4888 if .IDCT_ADDR [CR_POLL] gtra .IDCT_ADDR [CR_END] ! IF BEYOND END OF RING
: 4889 then
: 4890 IDCT_ADDR [CR_POLL] = .IDCT_ADDR [CR_BEG]; ! SET POINTER TO TOP OF CRING
: 4891
: 4892 end;
: 4893
: 4894 ICOM_ADDR [CMD_INT] = 0; ! CLEAR COMMAND INTERRUPT WORD IN RING HEADER
: 4895 end;

```

Address	Offset	Hex	Assembly	Comment	Line
000000	004137	000000G	.SBTTL POLL.CRING RDRX INTERRUPT SERVICE ROUTINES		
			POLL.CRING:		
000004	013701	000000G	JSR	R1, \$SAVE2	4862
000010	012702	000016	MOV	IDCT_ADDR, R1	4882
000014	060102		MOV	#16, R2	4886
000016	105711		ADD	R1, R2	
000020	001422		1\$: TSTB	(R1)	4882
000022	016100	000016	BEQ	2\$	
000026	016000	000002	MOV	16(R1), R0	4883
000032	042700	077777	MOV	2(R0), R0	
000036	020027	100000	BIC	#77777, R0	
000042	001411		CMF	R0, #-100000	
000044	105311		BEQ	2\$	
000046	062712	000004	DECB	(R1)	4885
000052	021261	000012	ADD	#4, (R2)	4886
000056	101757		CMF	(R2), 12(R1)	4888
			BLOS	1\$	

J16

ZRQAM3 V01.2	RD/RX EXERCISER RDRX INTERRUPT SERVICE ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 B111-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (48)	SEQ 0411 Page 156	
000060 016112 000010		MOV	10(R1),(R2)	:	4890
000064 000754		BR	1\$:	4882
000066 013700 000000G	2\$:	MOV	ICOM.ADDR,R0	:	4894
000072 005060 000004		CLR	4(R0)	:	
000076 000207		RTS	PC	:	4862

; Routine Size: 32 words, Routine Base: \$CODE\$ + 22632
; Maximum stack depth per invocation: 4 words

```

: 4896 routine POLL_RRING : novalue =
: 4897
: 4898 !+
: 4899 ! THIS ROUTINE IS CALLED BY THE RDRX INTERRUPT SERVICE ROUTINE (AZINT)
: 4900 ! FOR EACH DEVICE INTERRUPT EXCEPT DURING INITIALIZATION OR FATAL ERROR.
: 4901 ! ITS PURPOSE IS TO SCAN THE DEVICE'S RESPONSE RING AND CHECK FOR ANY
: 4902 ! SLOTS WHICH HAVE BEEN RETURNED TO THE HOST (OWNERSHIP BIT = 0). FOR
: 4903 ! EACH SUCH SLOT, THE ASSOCIATED MESSAGE IS PROCESSED BASED ON ITS
: 4904 ! CONNECTION ID (MSCP OR DUP). AFTER PROCESSING, THE MESSAGE PACKET
: 4905 ! IS RE-INITIALIZED AND RETURNED TO THE CONTROLLER (OWNERSHIP BIT SET
: 4906 ! TO 1).
: 4907
: 4908 ! IMPLICIT INPUTS:
: 4909 !     ICTLR - NUMBER OF INTERRUPTING CONTROLLER
: 4910 !     IDCT_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 4911 !-
: 4912
: 4913 begin
: 4914
: 4915 while not (BIT_TST ((.IDCT_ADDR [RR_POLL] + 2), ED_OWN)) do           ! WHILE 0 = 0
: 4916     begin
: 4917     IPKT_ADDR = ..IDCT_ADDR [RR_POLL] - 10; ! ADDRESS OF RESPONSE PACKET
: 4918
: 4919     if NOT (.IPKT_ADDR [CONNID] eal CID_DUP)
: 4920     then (CREDIT_BAL = .CREDIT_BAL + .IPKT_ADDR [CREDITS]);           ! It was notice that Dup was sending
: 4921                                             ! credits back which it should not
: 4922     selectoneu .IPKT_ADDR [CONNID] of
: 4923     set
: 4924
: 4925     [CID_MSCP] : MSCP_RSP ();
: 4926     [CID_DUP] : DUP_RSP ();
: 4927
: 4928     [otherwise] : PRINTF (DBM20, .IPKT_ADDR [CONNID], .IRDRX_ADDR);
: 4929                                             ! "BAD CONN ID = XXXXX RECEIVED FROM XXXXXX"
: 4930     tes;
: 4931
: 4932     IPKT_ADDR [MSGLEN] = MSG_LEN + 2; ! RE-INIT PKT FIELDS; MESSAGE LENGTH
: 4933     IDCT_ADDR [RR_POLL] = .IDCT_ADDR [RR_POLL] + 2; ! ADVANCE TO HI ORDER WORD OF RING SLOT
: 4934     .IDCT_ADDR [RR_POLL] = .IPKT_ADDR [PKT_HI]; ! RETURN SLOT TO CONTROLLER
: 4935     .IDCT_ADDR [RR_POLL] = ..IDCT_ADDR [RR_POLL] or ED_OWN or ED_FLAG; ! OWNERSHIP TOO
: 4936     IDCT_ADDR [RR_POLL] = .IDCT_ADDR [RR_POLL] + 2; ! ADVANCE TO NEXT RRING SLOT
: 4937
: 4938     if .IDCT_ADDR [RR_POLL] gtra .IDCT_ADDR [RR_END]; ! IF BEYOND END OF RING
: 4939     then
: 4940     IDCT_ADDR [RR_POLL] = .IDCT_ADDR [RR_BEG]; ! CYCLE TO TOP OF RING
: 4941
: 4942     end; ! WHILE LOOP
: 4943     ICOM_ADDR [RSP_INT] = 0; ! CLEAR RESPONSE INTERRUPT WORD IN RING HEADER
: 4944 end;

```

000000 004137 000000G

.SBTTL POLL_RRING RDRX INTERRUPT SERVICE ROUTINES
POLL_RRING:

L16

ZRQAM3 V01.2	RD/RX EXERCISER RDRX INTERRUPT SERVICE ROUTINES	14-Dec-1983 16:12:07 14-Dec-1983 16:12:00	VAX-11 Bliss-16 V3-555 DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (49)	SEQ 0413 Page 158
000004	013702	000000G	JSR R1,\$SAVE3	4896
000010	062702	000014	MOV IDCT.ADDR,R2	4915
000014	011200		ADD #14,R2	
000016	016000	000002	1\$: MOV (R2),R0	
000022	042700	077777	MOV 2(R0),R0	
000026	020027	100000	BIC #77777,R0	
000032	001506		CMP R0,#-100000	
000034	017237	000000	BEQ 6\$	
000042	162737	000012	MOV #0(R2),IPKT.ADDR	4917
000050	013700	000000G	SUB #12,IPKT.ADDR	
000054	005001		MOV IPKT.ADDR,R0	4919
000056	156001	000011	CLR R1	
000062	020127	000002	BISB 11(R0),R1	
000066	001410		CMP R1,#2	
000070	116003	000010	BEQ 2\$	
000074	042703	177760	MOVB 10(R0),R3	4920
000100	063703	000000G	BIC #177760,R3	
000104	010337	000000G	ADD CREDIT.BAL,R3	
000110	005701		MOV R3,CREDIT.BAL	
000112	001003		2\$: TST R1	4922
000114	004737	000000V	BNE 3\$	
000120	000421		JSR PC,MSCP.RSP	4925
000122	020127	000002	BR 5\$	4922
000126	001003		3\$: CMP R1,#2	
000130	004737	000000V	BNE 4\$	
000134	000413		JSR PC,DUP.RSP	4926
000136	013746	000000G	BR 5\$	4922
000142	010146		4\$: MOV IRDRX.ADDR,-(SP)	4928
000144	012746	000000G	MOV R1,-(SP)	
000150	012746	000003	MOV #DBM20,-(SP)	
000154	010600		MOV #3,-(SP)	
000156	104417		MOV SP,R0	: SP,*
000160	062706	000010	TRAP 17	
000164	013700	000000G	ADD #10,SP	
000170	012760	000074	5\$: MOV IPKT.ADDR,R0	4932
000176	013701	000000G	MOV #74,6(R0)	
000202	010102		MOV IDCT.ADDR,R1	4933
000204	062702	000014	MOV R1,R2	
000210	062712	000002	ADD #14,R2	
000214	016072	000002	ADD #2,(R2)	
000222	052772	140000	MOV 2(R0),#0(R2)	4934
000230	062712	000002	BIS #-40000,#0(R2)	4935
000234	021261	000006	ADD #2,(R2)	4936
000240	101665		CMP (R2),6(R1)	4938
000242	016112	000004	BLOS 1\$	
000246	000662		MOV 4(R1),(R2)	4940
000250	013700	000000G	BR 1\$	4915
000254	005060	000006	6\$: MOV ICOM.ADDR,R0	4943
000260	000207		CLR 6(R0)	
			RTS PC	4896

; Routine Size: 89 words, Routine Base: \$CODE\$ + 22732
; Maximum stack depth per invocation: 10 words


```

: 4945 !†
: 4946 ROUTINE DUP_RSP : NOVALUE =
: 4947
: 4948 !‡
: 4949     THIS ROUTINE IS CALLED BY POLL_RRING FOR EACH DUP RESPONSE
: 4950     ITS GENERAL PURPOSE IS TO ACT ON A DATAGRAM OR SEQUENTIAL MESSAGE.
: 4951     IF THE MESSAGE TYPE IS SEQUENTIAL, THE ROUTINE COPIES THE
: 4952     CONTENTS OF THE MESSAGE ENVELOPE INTO A RETURN PACKET SO THAT THE
: 4953     ENVELOPE CAN BE RETURNED TO THE CONTROLLER.
: 4954
: 4955     IMPLICIT INPUTS:
: 4956     ICTLR - INTERRUPTING CONTROLLER NUMBER
: 4957     IPKT_ADDR - ADDRESS OF MSCP ENVELOPE CONTAINING RESPONSE
: 4958     !-
: 4959     begin
: 4960
: 4961     local
: 4962     R_INDEX : signed word,
: 4963     SRC_ADDR,
: 4964     DST_ADDR,
: 4965     R_ADDR : ref block [RP_LEN, word] field (RP_FIELDS);
: 4966     !PRINTX (DER34);
: 4967
: 4968     incr COUNT from 0 to PKT_CNT - 1 do
: 4969
: 4970     if (.MSCP_PKT [.COUNT, CRN_LO] eql .IPKT_ADDR [CRN_LO]) and      ! IF THIS IS THE ASSOC CMD
: 4971     (.MSCP_PKT [.COUNT, CRN_HI] eql .IPKT_ADDR [CRN_HI]) and
: 4972     (.MSCP_PKT [.COUNT, PKT_LO] neq .IPKT_ADDR [PKT_LO]) and
: 4973     ((.MSCP_PKT [.COUNT, OPCODE] and OP_END) neq OP_END) and
: 4974     (.MSCP_PKT [.COUNT, CONNID] eql CID_DUP) and
: 4975     ((.IPKT_ADDR [OPCODE] and OP_END) eql OP_END)
: 4976     then
: 4977     begin
: 4978     P_INDEX = .COUNT;          ! SET PKT NUMBER
: 4979     exitloop;
: 4980     end;
: 4981
: 4982     if .P_INDEX les 0          ! IF COMMAND NOT FOUND
: 4983     then
: 4984     begin
: 4985     PRINTF (DBM108, .IPKT_ADDR [CRN_LO]); ! UNKNOWN COMMAND REF. NUMBER
: 4986     return;
: 4987     end;
: 4988
: 4989     if (R_INDEX = GET_RETPKT (.ICTLR)) les 0 ! IF RETPKT IS NOT AVAILABLE
: 4990     then
: 4991     PRINTF (DBM112)          ! "DUP-RSP: RETPKT NOT AVAILABLE"
: 4992     else
: 4993     begin
: 4994     SRC_ADDR = .IPKT_ADDR + 6; ! SET UP COPY (SKIP OVER PKT DESC)
: 4995     R_ADDR = DST_ADDR = RETPKT + (.R_INDEX * RP_LEN * 2); ! START OF ALLOCATED RETPKT
: 4996
: 4997     incr COUNT from 1 to RP_LEN do

```

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0415
Page 160
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (50)

```

:      4998          begin
:      4999          .DST_ADDR = ..SRC_ADDR;
:      5000          DST_ADDR = .DST_ADDR + 2;
:      5001          SRC_ADDR = .SRC_ADDR + 2;
:      5002          end;
:      5003
:      5004          IN_IODQ (.R_INDEX);
:      5005          end;
:      5006
:      5007
:      5008          if .P_INDEX geq 0
:      5009          then
:      5010              PUT_PKT (.P_INDEX);
:      5011
:      5012          end;

```

```

: COPY 1 WORD
: ADVANCE DESTINATION ADDR
: ADVANCE SOURCE ADDR
: COPY LOOP

: PUT RETPKT INDEX INTO IODQ
: IF RETPKT WAS ALLOCATED

: IF ASSOC CMD PKT WAS FOUND

: RETURN COMMAND PACKET TO POOL

: ROUTINE DUP-RSP

```

			.SBTTL	DUP.RSP RDRX INTERRUPT SERVICE ROUTINES	
000000	004137	000000G	DUP.RSP:	JSR R1,\$SAVE3	4946
000004	013701	000000G		MOV IPKT.ADDR,R1	4970
000010	005002			CLR R2	: COUNT
000012	010246		1\$:	MOV R2,-(SP)	: COUNT,*
000014	012746	000104		MOV #104,-(SP)	
000020	004737	000000G		JSR PC,BL\$MUL	
000024	022626			CMP (SP)+,(SP)+	
000026	026061	000012G 000012		CMP MSCP.PKT+12(R0),12(R1)	
000034	001024			BNE 2\$	
000036	026061	000014G 000014		CMP MSCP.PKT+14(R0),14(R1)	4971
000044	001020			BNE 2\$	
000046	026011	000000G		CMP MSCP.PKT(R0),(R1)	4972
000052	001415			BEQ 2\$	
000054	105760	000022G		TSTB MSCP.PKT+22(R0)	4973
000060	100412			BMI 2\$	
000062	126027	000011G 000002		CMPB MSCP.PKT+11(R0),#2	4974
000070	001006			BNE 2\$	
000072	105761	000022		TSTB 22(R1)	4975
000076	100003			BPL 2\$	
000100	010237	000000G		MOV R2,P_INDEX	: COUNT,*
000104	000406			BR 3\$: COUNT
000106	005202		2\$:	INC R2	: COUNT
000110	020227	000013		CMP R2,#13	: COUNT,*
000114	003736			BLE 1\$	
000116	005737	000000G		TST P_INDEX	: COUNT
000122	002013		3\$:	BGE 4\$	4982
000124	016146	000012		MOV 12(R1),-(SP)	: COUNT
000130	012746	000000G		MOV #DBM108,-(SP)	4985
000134	012746	000002		MOV #2,-(SP)	
000140	010600			MOV SP,R0	: SP,*
000142	104417			TRAP 17	
000144	062706	000006		ADD #6,SP	: COUNT
000150	000207			RTS PC	4984
000152	013746	000076'	4\$:	MOV ICTLR,-(SP)	: COUNT
000156	004737	000000G		JSR PC,GET.RETPKT	4989

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2:161 (50)

000162	010003		MOV	R0,R3	:	*,R.INDEX	
000164	005726		TST	(SP)+	:		
000166	005703		TST	R3	:	R.INDEX	
000170	002007		BGE	5\$:		
000172	012746	000000G	MOV	@DBM112,-(SP)	:		4991
000176	012746	000001	MOV	@1,-(SP)	:		
000202	010600		MOV	SP,R0	:	SP,*	
000204	104417		TRAP	17	:		
000206	000424		BR	7\$:		4989
000210	013701	000000G	5\$: MOV	IPKT.ADDR,R1	:	*,SRC.ADDR	4994
000214	062701	000006	ADD	@6,R1	:	*,SRC.ADDR	
000220	010346		MOV	R3,-(SP)	:	R.INDEX,*	4995
000222	012746	000060	MOV	@60,-(SP)	:		
000226	004737	000000G	JSR	PC,BL\$MUL	:		
000232	062700	000000G	ADD	@RETPKT,R0	:		
000236	010002		MOV	R0,R2	:	*,DST.ADDR	
000240	012700	000030	MOV	@30,R0	:	*,COUNT	4997
000244	012122		6\$: MOV	(R1)+,(R2)+	:	SRC.ADDR,DST.ADDR	4999
000246	005300		DEC	R0	:	COUNT	4997
000250	001375		BNE	6\$:		
000252	010316		MOV	R3,(SP)	:	R.INDEX,*	5004
000254	004737	000000G	JSR	PC,IN.IODQ	:		
000260	013700	000000G	7\$: MOV	P.INDEX,R0	:		5008
000264	002403		BLT	8\$:		
000266	010016		MOV	R0,(SP)	:		5010
000270	004737	000000G	JSR	PC,PUT.PKT	:		
000274	022626		8\$: CMP	(SP)+,(SP)+	:		4959
000276	000207		RTS	PC	:		4946

: Routine Size: 96 words, Routine Base: \$CODE\$ + 23214
: Maximum stack depth per invocation: 9 words

: 5013

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0417
Page 162
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (51)

```

: 5014 routine MSCP_RSP : novalue =
: 5015
: 5016 !
: 5017 !
: 5018 ! THIS ROUTINE IS CALLED BY POLL_RRING FOR EACH RESPONSE MESSAGE
: 5019 ! WHICH HAS A CONNECTION ID INDICATING A DISK MSCP ORIGINATOR
: 5020 ! (I.E., ALL EXCEPT DUP RESPONSES). ITS PURPOSE IS TO PASS
: 5021 ! CONTROL TO THE APPROPRIATE ROUTINE BASED ON THE MESSAGE TYPE
: 5022 ! FIELD (SEQUENTIAL, DATAGRAM, OR CREDIT NOTIFICATION).
: 5023 !
: 5024 ! IMPLICIT INPUTS:
: 5025 ! IPKT_ADDR - ADDRESS OF MSCP PACKET CONTAINING RESPONSE
: 5026 ! MESSAGE
: 5027 !
: 5028 selectoneu .IPKT_ADDR [MSGTYP] of
: 5029 set
: 5030
: 5031 [MT_SEQ] : SEQUEN ();
: 5032
: 5033 [MT_DG] : DATAGM ();
: 5034
: 5035 [otherwise] : PRINTF (DBM21, .IPKT_ADDR [MSGTYP]); ! "MESSAGE TYPE XX RECEIVED"
: 5036 tes;

```

```

000000 010146 .SBTTL MSCP.RSP RDRX INTERRUPT SERVICE ROUTINES
MSCP.RSP:
000002 013700 000000G MOV R1, -(SP) ; 5014
000006 116001 000010 MOV IPKT_ADDR, R0 ; 5028
000012 006201 MOVB 10(R0), R1
000014 006201 ASR R1
000016 006201 ASR R1
000020 006201 ASR R1
000022 042701 177760 BIC #177760, R1
000026 001003 BNE 1#
000030 004737 000000V JSR PC, SEQUEN ; 5031
000034 000417 BR 3# ; 5028
000036 020127 000001 1#: CMP R1, #1
000042 001003 BNE 2#
000044 004737 000000V JSR PC, DATAGM ; 5033
000050 000411 BR 3# ; 5028
000052 010146 2#: MOV R1, -(SP) ; 5035
000054 012746 000000G MOV #DBM21, -(SP)
000060 012746 000002 MOV #2, -(SP)
000064 010600 MOV SP, R0 ; SP, +
000066 104417 TRAP 17
000070 062706 000006 ADD #6, SP
000074 012601 3#: MOV (SP)+, R1 ; 5014
000076 000207 RTS PC

```

; Routine Size: 32 words, Routine Base: \$CODE\$ + 23514
; Maximum stack depth per invocation: 6 words

ZRQAM3
V01.2RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (52)SEQ 0418
Page 163

```

5037 routine SEQUEN : novalue =
5038
5039 !*
5040 ! THIS ROUTINE IS CALLED BY MSCP_RSP FOR EACH DISK MSCP RESPONSE MESSAGE
5041 ! WITH THE "SEQUENTIAL" MESSAGE TYPE. ITS GENERAL PURPOSE IS TO COPY THE
5042 ! CONTENTS OF THE MESSAGE PACKET INTO A RETURN PACKET SO THAT THE
5043 ! PACKET CAN BE RETURNED TO THE CONTROLLER. IN ADDITION,
5044 ! IF THE COMMAND WAS AN I/O TRANSFER (READ, WRITE, OR ACCESS), THEN SOME
5045 ! FIELDS OF THE COMMAND PACKET ARE COPIED INTO THE RETURN PACKET.
5046 !
5047 ! IMPLICIT INPUTS:
5048 ! ICTLR - INTERRUPTING CONTROLLER NUMBER
5049 ! IPKT_ADDR - ADDRESS OF MSCP PACKET CONTAINING RESPONSE
5050 !-
5051
5052 begin
5053
5054 local
5055     R_INDEX : signed word,
5056     SRC_ADDR,
5057     DST_ADDR,
5058     R_ADDR : ref block [RP_LEN, word] field (RP_FIELDS),
5059     TEMP_UNIT,
5060     SFT_ERR_PRINTED : byte initial (byte (FALSE));
5061
5062 incr COUNT from 0 to PKT_CNT - 1 do
5063
5064     if (.MSCP_PKT [.COUNT, CRN_LO] eql .IPKT_ADDR [CRN_LO]) and      ! IF THIS IS THE ASSOC CMD
5065         (.MSCP_PKT [.COUNT, CRN_HI] eql .IPKT_ADDR [CRN_HI]) and
5066         (.MSCP_PKT [.COUNT, PKT_LO] neq .IPKT_ADDR [PKT_LO]) and
5067         ((.MSCP_PKT [.COUNT, OPCODE] and OP_END) neq OP_END) and
5068         (.MSCP_PKT [.COUNT, MSGTYP] eql MT_SEQ) and
5069         ((.IPKT_ADDR [OPCODE] and OP_END) eql OP_END) and
5070         (.PKT_USE [.count] eql .ICTLR)                                ! don't want old packets from other controll
5071     then
5072         begin
5073             P_INDEX = .COUNT;          ! SET PKT NUMBER
5074             exitloop;
5075         end;
5076
5077     if .P_INDEX les 0                ! IF COMMAND NOT FOUND
5078     then
5079         begin
5080             PRINTF (DBM108, .IPKT_ADDR [CRN_LO]); ! UNKNOWN COMMAND REF. NUMBER
5081             return;
5082         end;
5083
5084     if (R_INDEX = GET_RETPKT (.ICTLR)) les 0 ! IF RETPKT IS NOT AVAILABLE
5085     then
5086         ! PRINTF (DBM22)                ! "SEQUEN: RETPKT NOT AVAILABLE"
5087     else
5088         begin
5089             SRC_ADDR = .IPKT_ADDR + 6; ! SET UP COPY (SKIP OVER PKT DESC)

```

```

:      5090      R_ADDR = DST_ADDR = RETPKT + (.R_INDEX * RP_LEN * 2);      ! START OF ALLOCATED RETPKT
:      5091
:      5092      incr COUNT from 1 to RP_LEN do
:      5093          begin
:      5094              .DST_ADDR = .SRC_ADDR;      ! COPY 1 WORD
:      5095              DST_ADDR = .DST_ADDR + 2;      ! ADVANCE DESTINATION ADDR
:      5096              SRC_ADDR = .SRC_ADDR + 2;      ! ADVANCE SOURCE ADDR
:      5097
:      5098              if .IPKT_ADDR [OPCODE] eq1 (OP_ONL or OP_END)      ! IF THIS IS THE ONLINE END MESSAGE
:      5099              then
:      5100
:      5101                  if .COUNT eq1 10      ! SKIP OVER RESERVED WORDS
:      5102                  then
:      5103                      SRC_ADDR = .SRC_ADDR + 4;      ! IN ONLINE END - MESSAGE
:      5104
:      5105                  end;      ! COPY LOOP
:      5106
:      5107      R_ADDR [CTLR] = .ICTLR;      ! LOAD CONTROLLER NUMBER INTO PACKET
:      5108
:      5109      if .P_INDEX geq 0      ! IF ASSOC. CMD PKT WAS FOUND
:      5110      then
:      5111
:      5112          if (.IPKT_ADDR [OPCODE] eq1 (OP_RD or OP_END)) or      ! IF END MESSAGE IS
:      5113              (.IPKT_ADDR [OPCODE] eq1 (OP_WRT or OP_END)) or      ! READ, WRITE, OR
:      5114              (.IPKT_ADDR [OPCODE] eq1 (OP_ACC or OP_END))      ! ACCESS
:      5115          then
:      5116              begin
:      5117                  R_ADDR [CMDMOD] = .MSCP_PKT [.P_INDEX, MODIFY];      ! COPY
:      5118                  R_ADDR [CBCNT_LO] = .MSCP_PKT [.P_INDEX, BC_LO];      ! RELEVANT
:      5119                  R_ADDR [CBCNT_HI] = .MSCP_PKT [.P_INDEX, BC_HI];      ! FIELDS
:      5120                  R_ADDR [LBN_LO] = .MSCP_PKT [.P_INDEX, LBN_L];      ! FROM
:      5121                  R_ADDR [LBN_HI] = .MSCP_PKT [.P_INDEX, LBN_H];      ! COMMAND
:      5122                  R_ADDR [BUFF_0] = .MSCP_PKT [.P_INDEX, BUF_0];      ! PACKET
:      5123                  R_ADDR [BUFF_1] = .MSCP_PKT [.P_INDEX, BUF_1];      ! TO RETPKT
:      5124              end;      ! IF ENCODED WAS READ, WRITE, OR ACCESS
:      5125
:      5126
:      5127      IN_IODQ (.R_INDEX);      ! PUT RETPKT INDEX INTO IODQ
:      5128      end;      ! IF RETPKT WAS ALLOCATED
:      5129
:      5130      if (.IPKT_ADDR [STATUS_CODE] neq ST_SUC) or
:      5131          (.IPKT_ADDR [STATUS_SUBCODE] neq 0)
:      5132      then
:      5133          LAST_PKT [.ICTLR, LAST_HRD_ERR] = HRD_OCCURED      ! SAVE ERROR CONDITION
:      5134      else
:      5135          LAST_PKT [.ICTLR, LAST_HRD_ERR] = HRD_NOT_OCCURED;      !
:      5136
:      5137      LAST_PKT [.ICTLR, LAST_CRN_LO] = .IPKT_ADDR [CRN_LO];      ! SAVE COMMAND REFERENCE NUMBER
:      5138      LAST_PKT [.ICTLR, LAST_CRN_HI] = .IPKT_ADDR [CRN_HI];      !
:      5139
:      5140      incr index from 0 to EP_CNT - 1 do      ! IF CORRESPONDING REF NUM HAD AN ERROR-LOG
:      5141
:      5142          if (.ELOG_PKT [.index, EL_CNTR] eq1 .ICTLR) and

```

```

:      5143      (.ELOG_PKT [.index, EL_CRN_LO] eq1 .IPKT_ADJR [CRN_LO]) and
:      5144      (.ELOG_PKT [.index, EL_CRN_HI] eq1 .IPKT_ADDR [CRN_HI]) and
:      5145      (.ELOG_PKT [.index, EL_CONTENTS] eq1 FULL)
:      5146      then
:      5147      begin
:      5148
:      5149      if .LAST_PKT [.ICTLR, LAST_HRD_ERR] eq1 HRD_NOT_OCCURED      ! IF SOFT ERROR OCCURED
:      5150      then
:      5151
:      5152      if .ELOG_PKT [.index, EL_FORMAT] lequ 4
:      5153      then
:      5154      begin
:      5155      SOFT_ERROR (.index);      ! UPATE SOFT ERROR COUNT
:      5156      TEMP_UNIT = .L$LUN;      ! SAVE UNIT NUMBER AS KNOWN TO DRS
:      5157
:      5158      incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
:      5159
:      5160      if (.ICST_ADDR [.OFFSET, D_DISK_NUM] eq1 .ELOG_PKT [.index, EL_DK_NUM]) and
:      5161      (.ICST_ADDR [.OFFSET, D_PRES] eq1 PRESENT)
:      5162      then
:      5163      begin
:      5164      L$LUN = .ICST_ADDR [.OFFSET, D_UNIT];      ! CORECT UNIT NUMBER FOR ERROR MESSAGE
:      5165      exitloop;
:      5166      end;
:      5167
:      5168      case .ELOG_PKT [.index, EL_FORMAT] from 0 to 4 of
:      5169      set
:      5170
:      5171      [0]:      ERRSOFT (60, 0, 0);      ! CONTROLLER ERROR
:      5172
:      5173      [1]:      ERRSOFT (61, 0, 0);      ! HOST MEMORY ACCESS ERROR
:      5174
:      5175      [2]:      ERRSOFT (62, 0, 0);      ! DISK TRANSFER ERROR
:      5176
:      5177      [3]:      ERRSOFT (63, 0, 0);      ! SDI ERROR
:      5178
:      5179      [4]:      ERRSOFT (64, 0, 0);      ! SMALL DISK ERROR
:      5180      tes;
:      5181
:      5182      L$LUN = .TEMP_UNIT;      ! RESTORE UNIT NUMBER
:      5183      SFT_ERR_PRINTED = TRUE;      ! SOFT ERROR PRINTOUT OCCURED
:      5184      end
:      5185
:      5186      else
:      5187      PRINTF (DBM109, .ELOG_PKT [.index, EL_FORMAT]);      ! UNKNOWN ERROR-LOG FORMAT
:      5188
:      5189      if not (.SFT_ERR_PRINTED)
:      5190      then
:      5191      PRINTB (CRLF);      ! EXTRA CARRIAGE-RETURN/LINE-FEED
:      5192
:      5193      EMS_EL (.index);      ! PRINT ERROR-LOG CONTENTS
:      5194      end;
:      5195

```

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Blues-16 V3-555
DISK:USER2:[DIETZ.RDRX]ZRQACO.BL2:161 (52)
SEQ 0421
Page 166

```

:      5196      if P_INDEX geq 0      ! IF ASSOC CMD PKT WAS FOUND
:      5197      then
:      5198          PUT_PKT (.P_INDEX); ! RETURN COMMAND PACKET TO POOL
:      5199
:      5200      end;                  ! ROUTINE DISK_RSP

```

```

000000 004137 000000G      .SBTTL SEQUEN: JSR R1,$SAVE5 ; 5037
000004 105046      CLR R1 ; SFT.ERR.PRINTED 5052
000006 013701 000000G      MOV IPKT.ADDR,R1 ; 5064
000012 005002      CLR R2 ; COUNT 5062
000014 010246      1$: MOV R2,-(SP) ; COUNT,* 5064
000016 012746 000104      MOV #104,-(SP)
000022 004737 000000G      JSR PC,BL$MUL
000026 022626      CMP (SP)+,(SP)+
000030 026061 000012G 000012      CMP MSCP.PKT+12(R0),12(R1)
000036 001031      BNE 2$
000040 026061 000014G 000014      CMP MSCP.PKT+14(R0),14(R1) ; 5065
000046 001025      BNE 2$
000050 026011 000000G      CMP MSCP.PKT(R0),(R1) ; 5066
000054 001422      BEQ 2$
000056 105760 000022G      TSTB MSCP.PKT+22(R0) ; 5067
000062 100417      BMI 2$
000064 132760 000360 000010G      BITB #360,MSCP.PKT+10(R0) ; 5068
000072 001013      BNE 2$
000074 105761 000022      TSTB 22(R1) ; 5069
000100 100010      BPL 2$
000102 116200 000000G      MOVB PKT.USE(R2),R0 ; *(COUNT),* 5070
000106 020037 000076'      CMP R0,ICTLR
000112 001003      BNE 2$
000114 010237 000000G      MOV R2,P.INDEX ; COUNT,* 5073
000120 000406      BR 3$ ; 5072
000122 005202      2$: INC R2 ; COUNT 5062
000124 020227 000013      CMP R2,#13 ; COUNT,*
000130 003731      BLE 1$
000132 005737 000000G      TST P.INDEX ; 5077
000136 002014      3$: BGE 4$
000140 016146 000012      MOV 12(R1),-(SP) ; 5080
000144 012746 000000G      MOV #DBM108,-(SP)
000150 012746 000002      MOV #2,-(SP)
000154 010600      MOV SP,R0 ; SP,*
000156 104417      TRAP 17
000160 062706 000006      ADD #6,SP ; 5077
000164 000137 025040'      JMP 30$ ; 5079
000170 013746 000076'      4$: MOV ICTLR,-(SP) ; 5084
000174 004737 000000G      JSR PC,GET.RETPKT
000200 010005      MOV R0,R5 ; *,R.INDEX
000202 005726      TST (SP)+ ; R.INDEX
000204 005705      TST R5
000206 002526      BLT 9$
000210 013704 000000G      MOV IPKT.ADDR,R4 ; *,SRC.ADDR 5089
000214 062704 000006      ADD #6,R4 ; *,SRC.ADDR

```


ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0422
Page 167
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (52)

000220	010546			MOV	R5, -(SP)			; R. INDEX, *	5090
000222	012746	000060		MOV	#60, -(SP)				
000226	004737	000000G		JSR	PC, BL \$MUL				
000232	062700	000000G		ADD	#RETPKT, R0				
000236	010003			MOV	R0, R3			; *, DST. ADDR	
000240	010002			MOV	R0, R2			; *, R. ADDR	
000242	013701	000000G		MOV	IPKT. ADDR, R1				5098
000246	012700	000001		MOV	#1, R0			; *, COUNT	5092
000252	012423			MOV	(R4)+, (R3)+			; SRC. ADDR, DST. ADDR	5094
000254	126127	000022	000211	CMPB	22(R1), #211				5098
000262	001005			BNE	6#				
000264	020027	000012		CMP	R0, #12			; COUNT, *	5101
000270	001002			BNE	6#				
000272	062704	000004		ADD	#4, R4			; *, SRC. ADDR	5103
000276	005200			INC	R0			; COUNT	5092
000300	020027	000030		CMP	R0, #30			; COUNT, *	
000304	003762			BLE	5#				
000306	013700	000076'		MOV	ICTLR, R0				5107
000312	042700	177760		BIC	#177760, R0				
000316	142762	000017	000002	BICB	#17, 2(R2)			; *, *(R. ADDR)	
000324	150062	000002		BISB	R0, 2(R2)			; *, *(R. ADDR)	
000330	013704	000000G		MOV	P. INDEX, R4				5109
000334	002447			BLT	8#				
000336	005000			CLR	R0				5112
000340	156100	000022		BISB	22(R1), R0				
000344	020027	000241		CMP	R0, #241				
000350	001406			BEQ	7#				
000352	020027	000242		CMP	R0, #242				5113
000356	001403			BEQ	7#				
000360	020027	000220		CMP	R0, #220				5114
000364	001033			BNE	8#				
000366	010416			MOV	R4, (SP)				5117
000370	012746	000104		MOV	#104, -(SP)				
000374	004737	000000G		JSR	PC, BL \$MUL				
000400	016062	000024G	000012	MOV	MSCP. PKT+24(R0), 12(R2)			; *, *(R. ADDR)	
000406	016062	000026G	000044	MOV	MSCP. PKT+26(R0), 44(R2)			; *, *(R. ADDR)	5118
000414	016062	000030G	000046	MOV	MSCP. PKT+30(R0), 46(R2)			; *, *(R. ADDR)	5119
000422	016062	000046G	000050	MOV	MSCP. PKT+46(R0), 50(R2)			; *, *(R. ADDR)	5120
000430	016062	000050G	000052	MOV	MSCP. PKT+50(R0), 52(R2)			; *, *(R. ADDR)	5121
000436	016062	000032G	000024	MOV	MSCP. PKT+32(R0), 24(R2)			; *, *(R. ADDR)	5122
000444	016062	000034G	000026	MOV	MSCP. PKT+34(R0), 26(R2)			; *, *(R. ADDR)	5123
000452	005726			TST	(SP)+				5116
000454	010516			MOV	R5, (SP)			; R. INDEX, *	5127
000456	004737	000000G		JSR	PC, IN. IODQ				
000462	022626			CMP	(SP)+, (SP)+				5088
000464	013746	000076'		MOV	ICTLR, -(SP)				5133
000470	012746	000006		MOV	#6, -(SP)				
000474	004737	000000G		JSR	PC, BL \$MUL				
000500	013702	000000G		MOV	IPKT. ADDR, R2				5130
000504	012703	000024		MOV	#24, R3				
000510	060203			ADD	R2, R3				
000512	132713	000037		BITB	#37, (R3)				
000516	001003			BNE	10#				

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0423
Page 168
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (52)

000520	032713	177740		BIT	#177740,(R3)	:	5131
000524	001404			BEQ	11#	:	
000526	012760	000001	000110'	10#:	MOV #1, LAST.PKT(R0)	:	5133
000534	000402			BR	12#	:	5130
000536	005060	000110'		11#:	CLR LAST.PKT(R0)	:	5135
000542	016260	000012	000112'	12#:	MOV 12(R2), LAST.PKT+2(R0)	:	5137
000550	016260	000014	000114'		MOV 14(R2), LAST.PKT+4(R0)	:	5138
000556	005003				CLR R3	:	5140
000560	010316			13#:	MOV R3, (SP)	:	INDEX,*
000562	012746	000102			MOV #102, -(SP)	:	INDEX,*
000566	004737	000000G			JSR PC, BL\$MUL		
000572	010005				MOV R0, R5		
000574	005726				TST (SP)+		
000576	005000				CLR R0		
000600	156500	000000G			BISB ELOG.PKT(R5), R0		
000604	020037	000076'			CMP R0, ICTLR		
000610	001170				BNE 27#		
000612	013700	000000G			MOV IPKT.ADDR, R0	:	5143
000616	026560	000006G	000012		CMP ELOG.PKT+6(R5), 12(R0)		
000624	001162				BNE 27#		
000626	026560	000010G	000014		CMP ELOG.PKT+10(R5), 14(R0)	:	5144
000634	001156				BNE 27#		
000636	126527	000001G	000001		CMPB ELOG.PKT+1(R5), #1	:	5145
000644	001152				BNE 27#		
000646	013716	000076'			MOV ICTLR, (SP)	:	5149
000652	012746	000006			MOV #6, -(SP)		
000656	004737	000000G			JSR PC, BL\$MUL		
000662	005726				TST (SP)+		
000664	005760	000110'			TST LAST.PKT(R0)		
000670	001122				BNE 25#		
000672	126527	000016G	000004		CMPB ELOG.PKT+16(R5), #4	:	5152
000700	101104				BHI 24#		
000702	010316				MOV R3, (SP)	:	INDEX,*
000704	004737	000000V			JSR PC, SOFT.ERROR		
000710	013702	000000G			MOV L\$LUN, R2	:	*,TEMP.UNIT
000714	012700	000006			MOV #6, R0	:	*,OFFSET
000720	010004				MOV R0, R4	:	OFFSET,*
000722	063704	000000G		14#:	ADD ICST.ADDR, R4		
000726	016546	000012G			MOV ELOG.PKT+12(R5), -(SP)		
000732	111401				MOVB (R4), R1		
000734	042701	177774			BIC #177774, R1		
000740	020126				CMP R1, (SP)+		
000742	001012				BNE 15#		
000744	032714	040000			BIT #40000, (R4)	:	5161
000750	001407				BEQ 15#		
000752	011401				MOV (R4), R1	:	5164
000754	000301				SWAB R1		
000756	042701	177760			BIC #177760, R1		
000762	010137	000000G			MOV R1, L\$LUN		
000766	000405				BR 16#	:	5163
000770	062700	000016		15#:	ADD #16, R0	:	*,OFFSET
000774	020027	000060			CMP R0, #60	:	OFFSET,*
001000	003747				BLE 14#		

001002	005000		16:	CLR	R0	:		5168
001004	156500	000016G		BISB	ELOG.PKT+16(R5),R0	:		
001010	006300			ASL	R0	:		
001012	066007	000000'		ADD	P.AAA(R0),PC	:	Case dispatch	
001016	104457		18:	TRAP	57	:		5171
001020	000074			.WORD	74	:		
001022	000000			.WORD	0	:		
001024	000000			.WORD	0	:		
001026	000423			BR	23:	:		5168
001030	104457		19:	TRAP	57	:		5173
001032	000075			.WORD	75	:		
001034	000000			.WORD	0	:		
001036	000000			.WORD	0	:		
001040	000416			BR	23:	:		5168
001042	104457		20:	TRAP	57	:		5175
001044	000076			.WORD	76	:		
001046	000000			.WORD	0	:		
001050	000000			.WORD	0	:		
001052	000411			BR	23:	:		5168
001054	104457		21:	TRAP	57	:		5177
001056	000077			.WORD	77	:		
001060	000000			.WORD	0	:		
001062	000000			.WORD	0	:		
001064	000404			BR	23:	:		5168
001066	104457		22:	TRAP	57	:		5179
001070	000100			.WORD	100	:		
001072	000000			.WORD	0	:		
001074	000000			.WORD	0	:		
001076	010237	000000G	23:	MOV	R2,L#LUN	:	TEMP.UNIT,*	5182
001102	112766	000001 000004		MOVB	#1,4(SP)	:	*,SFT.ERR.PRINTED	5183
001110	000412			BR	25:	:		5152
001112	005016		24:	CLR	(SP)	:		5187
001114	116516	000016G		MOVB	ELOG.PKT+16(R5),(SP)	:		
001120	012746	000000G		MOV	#DBM109,-(SP)	:		
001124	012746	000002		MOV	#2,-(SP)	:		
001130	010600			MOV	SP,R0	:	SP,*	
001132	104417			TRAP	17	:		
001134	022626			CMP	(SP)+,(SP)+	:		
001136	032766	000001 000004	25:	BIT	#1,4(SP)	:	*,SFT.ERR.PRINTED	5189
001144	001007			BNE	26:	:		
001146	012716	000000G		MOV	#CRLF,(SP)	:		5191
001152	012746	000001		MOV	#1,-(SP)	:		
001156	010600			MOV	SP,R0	:	SP,*	
001160	104414			TRAP	14	:		
001162	005726			TST	(SP)+	:		
001164	010316		26:	MOV	R3,(SP)	:	INDEX,*	5193
001166	004737	000000G		JSR	PC,EMS.EL	:		
001172	005203		27:	INC	R3	:	INDEX	5140
001174	020327	000013		CMP	R3,#13	:	INDEX,*	
001200	003002			BGT	28:	:		
001202	000137	024374'		JMP	13:	:		
001206	013700	000000G	28:	MOV	P.INDEX,R0	:		5196
001212	002403			BLT	29:	:		

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0425
Page 170
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (52)

001214	010016		MOV	RO,(SP)	:	5198
001216	004737	000000G	JSR	PC,PUT.PKT	:	
001222	022626		29\$: CMP	(SP)+,(SP)+	:	5052
001224	005726		30\$: TST	(SP)+	:	5037
001226	000207		RTS	PC	:	

; Routine Size: 332 words, Routine Base: \$CODE\$ + 23614
; Maximum stack depth per invocation: 13 words

000000 .PSECT \$PLIT\$, RO , D

000000	000000	P.AAA:			:	CASE Table for SEQUEN+1012	5168
000002	000012	17\$:	.WORD	0	:	[18\$]	
000004	000024		.WORD	12	:	[19\$]	
000006	000036		.WORD	24	:	[20\$]	
000010	000050		.WORD	36	:	[21\$]	
			.WORD	50	:	[22\$]	

```

: 5201 routine DATAGM : novalue =
: 5202
: 5203 !!+
: 5204 THIS ROUTINE HANDLES ALL DATAGRAM (ERROR LOG) MESSAGES RECEIVED FROM
: 5205 THE RDRX
: 5206
: 5207 IMPLICIT INPUTS:
: 5208 IPKT_ADDR - ADDRESS OF MSCP PACKET CONTAINING ERROR LOG
: 5209 MESSAGE
: 5210 ICST_ADDR - ADDRESS OF THE INTERRUPTING CONTROLLER'S CST
: 5211 !-
: 5212
: 5213 begin
: 5214
: 5215 local
: 5216 index : signed word initial (-1),
: 5217 SAVE_ADDR : ref block [EP_LEN, word] field (EP_FIELDS),
: 5218 SRC_ADDR,
: 5219 DST_ADDR,
: 5220 TEMP_UNIT,
: 5221 SFT_ERR_PRINTED : byte initial (byte (FALSE));
: 5222
: 5223 !
: 5224 ! FIND AN EMPTY SLOT IN THE ERROR-LOG PACKET SAVE AREA
: 5225 !
: 5226
: 5227 incr COUNT from 0 to EP_CNT - 1 do
: 5228
: 5229 if .ELOG_PKT [.COUNT, EL_CONTENTS] eq1 EMPTY ! IF EMPTY SLOT FOUND
: 5230 then
: 5231 begin
: 5232 index = .COUNT; ! SAVE INDEX INTO THE SAVE AREA
: 5233 exitloop;
: 5234 end;
: 5235
: 5236 !
: 5237 ! IF AN EMPTY SLOT FOUND, SAVE THE PACKET CONTENTS
: 5238 !
: 5239
: 5240 if .index geq 0
: 5241 then
: 5242 begin
: 5243 SAVE_ADDR = ELOG_PKT + (.index * EP_LEN * 2); ! ADDRESS OF THE SAVE AREA
: 5244 SAVE_ADDR [EL_CONTENTS] = FULL; ! MARK IT FULL
: 5245 SAVE_ADDR [EL_CNTR] = .ICTLR; ! OWNERSHIP
: 5246 SRC_ADDR = .IPKT_ADDR + 6; ! SETUP COPY ADDRESSES
: 5247 DST_ADDR = .SAVE_ADDR + 2; !
: 5248
: 5249 incr COUNT from 1 to ((.IPKT_ADDR [MSGLEN] + 1) / 2) + 2 do
: 5250 begin
: 5251 .DST_ADDR = ..SRC_ADDR; ! COPY A WORD
: 5252 SRC_ADDR = .SRC_ADDR + 2; ! UPDATE ADDRESS POINTERS
: 5253 DST_ADDR = .DST_ADDR + 2; !

```

```

: 5254         end;
: 5255
: 5256         end
: 5257     else
: 5258         begin
: 5259             PRINTF (DBM110);           ! IF EMPTY SLOT NOT FOUND, PRINT MESSAGE
: 5260             return;
: 5261         end;
: 5262
: 5263     !!
: 5264     !! CHECK IF THE CORRESPONDING RESPONSE HAS ALREADY BEEN RECEIVED
: 5265     !!
: 5266
: 5267     if (.SAVE_ADDR [EL_CRN_LO] eq1 .LAST_PKT [.ICTLR, LAST_CRN_LO]) and
: 5268         (.SAVE_ADDR [EL_CRN_HI] eq1 .LAST_PKT [.ICTLR, LAST_CRN_HI])
: 5269     then
: 5270         begin
: 5271
: 5272         if .LAST_PKT [.ICTLR, LAST_HRD_ERR] eq1 HRD_NOT_OCCURED ! IF SOFT ERROR HAD OCCURED
: 5273         then
: 5274
: 5275             if .SAVE_ADDR [EL_FORMAT] lequ 4
: 5276             then
: 5277                 begin
: 5278                     SOFT_ERROR (.index);           ! UPDATE SOFT ERROR COUNT
: 5279                     TEMP_UNIT = .L$LUN;           ! SAVE UNIT NUMBER AS KNOWN TO DRS
: 5280
: 5281                     incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5282
: 5283                         if (.ICST_ADDR [.OFFSET, D_DISK_NUM] eq1 .SAVE_ADDR [EL_DK_NUM]) and
: 5284                             (.ICST_ADDR [.OFFSET, D_PRES] eq1 PRESENT)
: 5285                         then
: 5286                             begin
: 5287                                 L$LUN = .ICST_ADDR [.OFFSET, D_UNIT]; ! CORRECT UNIT NUMBER FOR ERROR MESSAGE
: 5288                                 exitloop;
: 5289                             end;
: 5290
: 5291                     case .SAVE_ADDR [EL_FORMAT] from 0 to 4 of
: 5292                         set
: 5293
: 5294                         [0] :      ERRSOFT (60, 0, 0);           ! CONTROLLER ERROR
: 5295
: 5296                         [1] :      ERRSOFT (61, 0, 0);           ! HOST MEMORY ACCESS ERROR
: 5297
: 5298                         [2] :      ERRSOFT (62, 0, 0);           ! DISK TRANSFER ERROR
: 5299
: 5300                         [3] :      ERRSOFT (63, 0, 0);           ! SDI ERROR
: 5301
: 5302                         [4] :      ERRSOFT (64, 0, 0);           ! SMALL DISK ERROR
: 5303                     tes;
: 5304
: 5305                     L$LUN = .TEMP_UNIT;           ! RESTORE UNIT NUMBER
: 5306                     SFT_ERR_PRINTED = TRUE;       ! SOFT ERROR PRINTOUT OCCURED

```

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0428
Page 173
VAX-11 B11es-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (53)

```

:      5307                end
:      5308
:      5309                else
:      5310                PRINTF (DBM109, .SAVE_ADDR [EL_FORMAT]);           ! ERROR LOG FORMAT UNKNOWN
:      5311
:      5312                if not (.SFT_ERR_PRINTED)
:      5313                then
:      5314                PRINTB (CRLF);                                       ! EXTRA CARRIEGE-RETURN/LINE-FEED
:      5315
:      5316                EMS_EL (.index);                                     ! PRINT PACKET CONTENTS
:      5317                end;                                               ! CORRESPONDING RESPONSE RECEIVED
:      5318
:      5319                end;

```

```

025044                .SBTTL  DATAGM RDRX INTERRUPT SERVICE ROUTINES
                    .PSECT  $CODE$, RO
000000 004137 000000G  DATAGM: JSR    R1,$SAVE5                ;
000004 012705 177777    MOV    #-1,R5                ; *,INDEX                5201
000010 105046    CLR    -(SP)                ; SFT.ERR.PRINTED    5213
000012 005001    CLR    R1                    ; COUNT                5227
000014 010146    1$:  MOV    R1,-(SP)          ; COUNT,*            5229
000016 012746 000102    MOV    #102,-(SP)
000022 004737 000000G  JSR    PC,BL$MUL
000026 022626    CMP    (SP), (SP),
000030 105760 000001G  TSTB   ELOG.PKT+1(R0)
000034 001002    BNE    2$
000036 010105    MOV    R1,R5                ; COUNT,INDEX        5232
000040 000405    BR    3$                    ;
000042 005201    2$:  INC    R1                ; COUNT                5231
000044 020127 000013    CMP    R1,#13                ; COUNT,*            5227
000050 003761    BLE    1$
000052 005705    TST    R5                    ; INDEX                5240
000054 002446    3$:  BLT    6$
000056 010546    MOV    R5,-(SP)            ; INDEX,*            5243
000060 012746 000102    MOV    #102,-(SP)
000064 004737 000000G  JSR    PC,BL$MUL
000070 062700 000000G  ADD    #ELOG.PKT,R0
000074 010004    MOV    R0,R4                ; *,SAVE.ADDR
000076 111764 000001    MOVB   (PC),1(R4)           ; *,*(SAVE.ADDR)    5244
000102 113714 000076'  MOVB   ICTLR,(R4)          ; *,SAVE.ADDR        5245
000106 013700 000000G  MOV    IPKT,ADDR,R0        ;
000112 012702 000006    MOV    #6,R2                ; *,SRC.ADDR        5246
000116 060002    ADD    R0,R2                ; *,SRC.ADDR
000120 012703 000002    MOV    #2,R3                ; *,DST.ADDR        5247
000124 060403    ADD    R4,R3                ; SAVE.ADDR,DST.ADDR
000126 016016 000006    MOV    6(R0),(SP)          ;
000132 005216    INC    (SP)
000134 012746 000002    MOV    #2,-(SP)
000140 004737 000000G  JSR    PC,BL$DIV
000144 062700 000002    ADD    #2,R0
000150 005001    CLR    R1                    ; COUNT

```

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0429
Page 174
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (53)

000152	000401			BR	5:					
000154	012223			MOV	4:	(R2), (R3)		; SRC.ADDR, DST.ADDR		5251
000156	005201			INC	5:	R1		; COUNT		5249
000160	020100			CMP		R1, R0		; COUNT, *		
000162	003774			BLE		4:				
000164	062706	000006		ADD		#6, SP				5242
000170	000410			BR		7:				5240
000172	012746	000000G		MOV	6:	#DBM110, -(SP)				5259
000176	012746	000001		MOV		#1, -(SP)				
000202	010600			MOV		SP, R0		; SP, *		
000204	104417			TRAP		17				
000206	022626			CMP		(SP), (SP)				5240
000210	000560			BR		22:				5258
000212	013746	000076'		MOV	7:	ICTLR, -(SP)				5267
000216	012746	000006		MOV		#6, -(SP)				
000222	004737	000000G		JSR		PC, BL\$MUL				
000226	022626			CMP		(SP), (SP)				
000230	026460	000006	000112'	CMP		6(R4), LAST.PKT+2(R0)		; *(SAVE.ADDR), *		
000236	001145			BNE		22:				
000240	026460	000010	000114'	CMP		10(R4), LAST.PKT+4(R0)		; *(SAVE.ADDR), *		5268
000246	001141			BNE		22:				
000250	005760	000110'		TST		LAST.PKT(R0)				5272
000254	001120			BNE		20:				
000256	005001			CLR		R1				5275
000260	156401	000016		BISB		16(R4), R1		; *(SAVE.ADDR), *		
000264	020127	000004		CMP		R1, #4				
000270	101101			BHI		18:				
000272	010546			MOV		R5, -(SP)		; INDEX, *		5278
000274	004737	000000V		JSR		PC, SOFT.ERROR				
000300	013703	000000G		MOV		L\$LUN, R3		; *, TEMP.UNIT		5279
000304	012700	000006		MOV		#6, R0		; *, OFFSET		5281
000310	010002			MOV	8:	R0, R2		; OFFSET, *		5283
000312	063702	000000G		ADD		ICST.ADDR, R2				
000316	016446	000012		MOV		12(R4), -(SP)		; *(SAVE.ADDR), *		
000322	111246			MOV		(R2), -(SP)				
000324	042716	177774		BIC		#177774, (SP)				
000330	022626			CMP		(SP), (SP)				
000332	001012			BNE		9:				
000334	032712	040000		BIT		#40000, (R2)				5284
000340	001407			BEQ		9:				
000342	011246			MOV		(R2), -(SP)				5287
000344	000316			SWAB		(SP)				
000346	042716	177760		BIC		#177760, (SP)				
000352	012637	000000G		MOV		(SP), L\$LUN				
000356	000405			BR		10:				5286
000360	062700	000016		ADD	9:	#16, R0		; *, OFFSET		5281
000364	020027	000060		CMP		R0, #60		; OFFSET, *		
000370	003747			BLE		8:				
000372	006301			ASL	10:	R1				5291
000374	066107	000012'		ADD		P.AAB(R1), PC		; Case dispatch		
000400	104457			TRAP	12:	57				5294
000402	000074			.WORD		74				
000404	000000			.WORD		0				

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0430
Page 175
VAX-11 B11ss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (53)

000406	000000			.WORD	0			
000410	000423			BR	17:			5291
000412	104457		13:	TRAP	57			5296
000414	000075			.WORD	75			
000416	000000			.WORD	0			
000420	000000			.WORD	0			
000422	000416			BR	17:			5291
000424	104457		14:	TRAP	57			5298
000426	000076			.WORD	76			
000430	000000			.WORD	0			
000432	000000			.WORD	0			
000434	000411			BR	17:			5291
000436	104457		15:	TRAP	57			5300
000440	000077			.WORD	77			
000442	000000			.WORD	0			
000444	000000			.WORD	0			
000446	000404			BR	17:			5291
000450	104457		16:	TRAP	57			5302
000452	000100			.WORD	100			
000454	000000			.WORD	0			
000456	000000			.WORD	0			
000460	010337	000000G	17:	MOV	R3,L#LUN		: TEMP,UNIT,*	5305
000464	112766	000001 000002		MOVB	#1,2(SP)		: *,SFT.ERR.PRINTED	5306
000472	000410			BR	19:			5275
000474	010146		18:	MOV	R1,-(SP)			5310
000476	012746	000000G		MOV	#DBM109,-(SP)			
000502	012746	000002		MOV	#2,-(SP)			
000506	010600			MOV	SP,RO		: SP,*	
000510	104417			TRAP	17			
000512	022626			CMP	(SP)*,(SP)*			
000514	005726		19:	TST	(SP)*			5275
000516	032716	000001	20:	BIT	#1,(SP)		: *,SFT.ERR.PRINTED	5312
000522	001007			BNE	21:			
000524	012746	000000G		MOV	#CRLF,-(SP)			5314
000530	012746	000001		MOV	#1,-(SP)			
000534	010600			MOV	SP,RO		: SP,*	
000536	104414			TRAP	14			
000540	022626			CMP	(SP)*,(SP)*			
000542	010546		21:	MOV	R5,-(SP)		: INDEX,*	5316
000544	004737	000000G		JSR	PC,EMS.EL			
000550	005726			TST	(SP)*			5270
000552	005726		22:	TST	(SP)*			5201
000554	000207			RTS	PC			

: Routine Size: 183 words, Routine Base: #CODE# + 25044
: Maximum stack depth per invocation: 12 words

000012 .PSECT #PLIT#, RO, D

000012	000000	P.AAB:					: CASE Table for DATAGM-0374	5291
000014	000012	11:	.WORD	0			: [12#]	
			.WORD	12			: [13#]	

E2

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0431
Page 176
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (53)

000016 000024
000020 000036
000022 000050

.WORD 24
.WORD 36
.WORD 50

; [14#]
; [15#]
; [16#]

ZRQAM3
V01.2RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (54)SEQ 0432
Page 177

```

: 5320 routine SOFT_ERROR (INDEX) : novalue =
: 5321
: 5322 !+
: 5323 ! THIS ROUTINE UPDATES THE SOFT ERROR COUNT IN THE TALLY TABLE FOR EACH
: 5324 ! ERROR LOG MESSAGE RECEIVED
: 5325 !
: 5326 ! IMPLICIT INPUTS:
: 5327 ! ICST_ADDR - ADDRESS OF THE INTERRUPTING CONTROLLER'S CST
: 5328 !-
: 5329
: 5330 begin
: 5331
: 5332 local
: 5333 FOUND: byte initial (byte (FALSE)),
: 5334 SOFT_OCCURED : byte initial (byte (FALSE)),
: 5335 UNIT: word,
: 5336 ERROR_CODE : byte,
: 5337 ERROR_SUB : word,
: 5338 TALLY_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 5339 ELOG_ADDR : ref block [EP_LEN, word] field (EP_FIELDS);
: 5340
: 5341 ELOG_ADDR = ELOG_PKT + (.index * EP_LEN * 2); ! ADDRESS OF ERROR L
OG PACKET
: 5342 ERROR_CODE = .ELOG_ADDR [EL_CODE]; ! ERROR CODE
: 5343 ERROR_SUB = .ELOG_ADDR [EL_SUBCODE]; ! ERROR SUBCODE
: 5344
: 5345 incr OFFSET from (0 + OF_UN) to (3 * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5346
: 5347 if (.ICST_ADDR [.OFFSET, D_PRES] eq1 PRESENT) and ! MAP DISK NUMBER TO
UNIT NUMBER
: 5348 (.ICST_ADDR [.OFFSET, D_DISK_NUM] eq1 .ELOG_ADDR [EL_DK_NUM])
: 5349 then
: 5350 begin
: 5351 FOUND = TRUE;
: 5352 UNIT = .ICST_ADDR [.OFFSET, D_UNIT]; ! UNIT NUMBER OF DIS
K
: 5353
: 5354 exitloop;
: 5355 end;
: 5356 ! if (.ELOG_ADDR [EL_SUCCESS]) or
: 5357 ! (.ELOG_ADDR [EL_CONTINUE])
: 5358 ! then
: 5359 ! SOFT_OCCURED = TRUE; ! SOFT ERROR FLAG
: 5360
: 5361 if .FOUND ! IF UNIT NUMBER FOU
ND
: 5362 then
: 5363 begin
: 5364 TALLY_ADDR = TALLY + (.UNIT * TALLY_LEN * 2); ! ADDRESS OF TALLY T
ABLE
: 5365
: 5366 if .SOFT_OCCURED ! FOR SOFT ERRORS
: 5367 then
: 5368 selectoneu .ERROR_CODE of
: 5369 set
: 5370
: 5371 [ST_MFE]: TALLY_ADDR [ERR_SFT_SEK] = .TALLY_ADDR [ERR_SFT_SEK] + 1; ! SOFT - MEDIA FORMA
T
: 5372

```

ZRQAM3
V01.2RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES14-Dec-1983 16:12:07
14-Dec-1983 16:12:00VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (54)SEQ 0433
Page 178

```

: 5373          [ST_DAT]:      if .ERROR_SUB eq1 2                ! SOFT - DATA
: 5374          :              then
: 5375          :                  TALLY_ADDR [ERR_SFT_SEK] = .TALLY_ADDR [ERR_SFT_DAT] + 1
: 5376          :              else
: 5377          :                  TALLY_ADDR [ERR_SFT_DAT] = .TALLY_ADDR [ERR_SFT_DAT] + 1;
: 5378          :
: 5379          [ST_HST]:      TALLY_ADDR [ERR_SFT_HST] = .TALLY_ADDR [ERR_SFT_HST] + 1;        ! SOFT - HOST ACCESS
: 5380          :
: 5381          [ST_CNT]:      C_ERR_TBL [.ICTLR, C_ERR_SFT] = .C_ERR_TBL [.ICTLR, C_ERR_SFT] + 1;  ! SOFT - CONTROLLER
: 5382          :
: 5383          :
: 5384          [ST_DRV]:      if .ERROR_SUB eq1 3                ! SOFT - DRIVE
: 5385          :              then
: 5386          :                  TALLY_ADDR [ERR_SFT_SEK] = .TALLY_ADDR [ERR_SFT_SEK] + 1
: 5387          :              else
: 5388          :                  TALLY_ADDR [ERR_SFT_DRV] = .TALLY_ADDR [ERR_SFT_DRV] + 1;
: 5389          :
: 5390          :              tes
: 5391          :
: 5392          :              else
: 5393          :                  if (.ELOG_ADDR [EL_CRN_LO] eq1 0) and
: 5394          :                      (.ELOG_ADDR [EL_CRN_HI] eq1 0)
: 5395          :                      then
: 5396          :                          select oneu .ERROR_CODE of
: 5397          :                              set
: 5398          :                                  [ST_MFE]:      TALLY_ADDR [ERR_HRD_SEK] = .TALLY_ADDR [ERR_HRD_SEK] + 1;        ! HARD - MEDIA FORMA
:
: 5399          :
: 5400          :                                  [ST_DAT]:      if .ERROR_SUB eq1 2                ! HARD - DATA
: 5401          :                                      then
: 5402          :                                          TALLY_ADDR [ERR_HRD_SEK] = .TALLY_ADDR [ERR_HRD_SEK] + 1
: 5403          :                                      else
: 5404          :                                          TALLY_ADDR [ERR_HRD_DAT] = .TALLY_ADDR [ERR_HRD_DAT] + 1;
: 5405          :
: 5406          :                                  [ST_HST]:      TALLY_ADDR [ERR_HRD_HST] = .TALLY_ADDR [ERR_HRD_HST] + 1;        ! HARD - HOST ACCESS
: 5407          :
: 5408          :                                  [ST_CNT]:      C_ERR_TBL [.ICTLR, C_ERR_HRD] = .C_ERR_TBL [.ICTLR, C_ERR_HRD] + 1;  ! HARD - CONTROLLER
: 5409          :
: 5410          :                                  [ST_DRV]:      if .ERROR_SUB eq1 3                ! HARD - DRIVE
: 5411          :                                      then
: 5412          :                                          TALLY_ADDR [ERR_HRD_SEK] = .TALLY_ADDR [ERR_HRD_SEK] + 1
: 5413          :                                      else
: 5414          :                                          TALLY_ADDR [ERR_HRD_DRV] = .TALLY_ADDR [ERR_HRD_DRV] + 1;
: 5415          :
: 5416          :                                      tes;
: 5417          :
: 5418          :                                  end
: 5419          :
: 5420          :                                  else
: 5421          :                                      ! UNIT NOT FOUND
: 5422          :
: 5423          :                                      if .SOFT_OCCURED
: 5424          :                                          then
: 5425          :                                              C_ERR_TBL [.ICTLR, C_ERR_SFT] = .C_ERR_TBL [.ICTLR, C_ERR_SFT] + 1
:
: 5425          :                                  else

```

```

:      5426      C_ERR_TBL [.ICTLR, C_ERR_HRD] = .C_ERR_TBL [.ICTLR, C_ERR_HRD] + 1;
:      5427
:      5428      end;
! ROUTINE SOFT_ERROR

```

```

025622      .SBTTL  SOFT.ERROR RDRX INTERRUPT SERVICE ROUTINES
           .PSECT  $CODE$, RO

000000 004137 000000G      SOFT.ERROR:
000004 024646      JSR      R1,$SAVE5      ;
000006 105001      CMP      -(SP),-(SP)      ;
000010 105066 000002      CLRB   R1      ; FOUND      5320
000014 016646 000022      CLRB   2(SP)      ; SOFT.OCCURED      5330
000020 012746 000102      MOV     22(SP),-(SP)      ; INDEX,*      5341
000024 004737 000000G      MOV     #102,-(SP)
000030 062700 000000G      JSR     PC,BL$MUL
000034 010004      ADD     #ELOG.PKT,RO
000036 116400 000020      MOV     RO,R4      ; *,ELOG.ADDR
000042 042700 177740      MOVB   20(R4),RO      ; *(ELOG.ADDR),*      5342
000046 105002      BIC    #177740,RO
000050 050002      CLRB   R2      ; ERROR.CODE
000052 016403 000020      BIS    RO,R2      ; *,ERROR.CODE
000056 006203      MOV     20(R4),R3      ; *(ELOG.ADDR),ERROR.SUB      5343
000060 006203      ASR    R3      ; ERROR.SUB
000062 006203      ASR    R3      ; ERROR.SUB
000064 006203      ASR    R3      ; ERROR.SUB
000066 006203      ASR    R3      ; ERROR.SUB
000070 042703 174000      BIC    #174000,R3      ; ERROR.SUB
000074 012700 000006      MOV     #6,RO      ; *,ERROR.SUB
000100 010005      1#:   MOV     RO,R5      ; *,OFFSET      5345
000102 063705 000000G      ADD     ICST.ADDR,R5      ; OFFSET,*      5347
000106 032715 040000      BIT    #40000,(R5)
000112 001420      BEQ    2#
000114 016446 000012      MOV     12(R4),-(SP)      ; *(ELOG.ADDR),*      5348
000120 111546      MOVB   (R5),-(SP)
000122 042716 177774      BIC    #177774,(SP)
000126 022626      CMP    (SP)+,(SP)+
000130 001011      BNE    2#
000132 112701 000001      MOVB   #1,R1      ; *,FOUND      5351
000136 011546      MOV     (R5),-(SP)      ;
000140 000316      SWAB   (SP)      ;
000142 042716 177760      BIC    #177760,(SP)      ; *,UNIT
000146 012666 000004      MOV     (SP)+,4(SP)      ;
000152 000405      BR     3#      ;
000154 062700 000016      2#:   ADD     #16,RO      ; *,OFFSET      5350
000160 020027 000060      CMP    RO,#60      ; *,OFFSET      5345
000164 003745      BLE    1#      ;
000166 112766 000001 000006      3#:   MOVB   #1,6(SP)      ; *,SOFT.OCCURED      5359
000174 006001      ROR    R1      ; FOUND      5361
000176 103154      BCC    17#
000200 016616 000004      MOV     4(SP),(SP)      ; UNIT,*      5364
000204 012746 000070      MOV     #70,-(SP)

```

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0435
Page 180
VAX-11 Blues-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (54)

000210	004737	000000G		JSR	PC,BL#MUL		
000214	062700	000000G		ADD	#TALLY,R0		
000220	032766	000001	000010	BIT	#1,10(SP)	; *,SOFT.OCCURED	5366
000226	001455			BEQ	9#		
000230	120227	000005		CMPB	R2,#5	; ERROR.CODE,*	5368
000234	001444			BEQ	7#		5371
000236	120227	000010		CMPB	R2,#10	; ERROR.CODE,*	5368
000242	001014			BNE	4#		
000244	012701	000064		MOV	#64,R1		5375
000250	060001			ADD	R0,R1	; TALLY.ADDR,*	
000252	020327	000002		CMP	R3,#2	; ERROR.SUB,*	5373
000256	001065			BNE	10#		
000260	005004			CLR	R4		5375
000262	156104	000001		BISB	1(R1),R4		
000266	005204			INC	R4		
000270	110411			MOVB	R4,(R1)		
000272	000514			BR	16#		5373
000274	120227	000011	4#:	CMPB	R2,#11	; ERROR.CODE,*	5368
000300	001003			BNE	5#		
000302	105260	000067		INCB	67(R0)	; *(TALLY.ADDR)	5379
000306	000506			BR	16#		5368
000310	120227	000012	5#:	CMPB	R2,#12	; ERROR.CODE,*	
000314	001006			BNE	6#		
000316	013701	000076'		MOV	ICTLR,R1		5381
000322	006301			ASL	R1		
000324	105261	000001G		INCB	C.ERR.TBL+1(R1)		
000330	000475			BR	16#		5368
000332	120227	000013	6#:	CMPB	R2,#13	; ERROR.CODE,*	
000336	001072			BNE	16#		
000340	020327	000003		CMP	R3,#3	; ERROR.SUB,*	5384
000344	001003			BNE	8#		
000346	105260	000064	7#:	INCB	64(R0)	; *(TALLY.ADDR)	5386
000352	000464			BR	16#		5384
000354	105260	000066	8#:	INCB	66(R0)	; *(TALLY.ADDR)	5388
000360	000461			BR	16#		5368
000362	005764	000006	9#:	TST	6(R4)	; *(ELOG.ADDR)	5392
000366	001056			BNE	16#		
000370	005764	000010		TST	10(R4)	; *(ELOG.ADDR)	5393
000374	001053			BNE	16#		
000376	120227	000005		CMPB	R2,#5	; ERROR.CODE,*	5395
000402	001443			BEQ	14#		5398
000404	120227	000010		CMPB	R2,#10	; ERROR.CODE,*	5395
000410	001013			BNE	11#		
000412	012701	000060		MOV	#60,R1		5402
000416	060001			ADD	R0,R1	; TALLY.ADDR,*	
000420	020327	000002		CMP	R3,#2	; ERROR.SUB,*	5400
000424	001002			BNE	10#		
000426	105211			INCB	(R1)		5402
000430	000435			BR	16#		5400
000432	105261	000001	10#:	INCB	1(R1)		5404
000436	000432			BR	16#		5395
000440	120227	000011	11#:	CMPB	R2,#11	; ERROR.CODE,*	
000444	001003			BNE	12#		

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (54)

SEQ 0436

Page 181

000446	105260	000063		INCB	63(R0)		; *(TALLY.ADDR)	5406
000452	000424			BR	16\$;	5395
000454	120227	000012	12\$:	CMPB	R2,#12		; ERROR.CODE,*	
000460	001006			BNE	13\$;	
000462	013701	000076'		MOV	ICTLR,R1		;	5408
000466	006301			ASL	R1		;	
000470	105261	000000G		INCB	C.ERR.TBL(R1)		;	
000474	000413			BR	16\$;	5395
000476	120227	000013	13\$:	CMPB	R2,#13		; ERROR.CODE,*	
000502	001010			BNE	16\$;	
000504	020327	000003		CMP	R3,#3		; ERROR.SUB,*	5411
000510	001003			BNE	15\$;	
000512	105260	000060	14\$:	INCB	60(R0)		; *(TALLY.ADDR)	5413
000516	000402			BR	16\$;	5411
000520	105260	000062	15\$:	INCB	62(R0)		; *(TALLY.ADDR)	5415
000524	005726		16\$:	TST	(SP)+		;	5363
000526	000415			BR	19\$;	5361
000530	013700	000076'	17\$:	MOV	ICTLR,RO		;	5424
000534	006300			ASL	RO		;	
000536	062700	000000G		ADD	#C.ERR.TBL,RO		;	
000542	032766	000001 000006		BIT	#1,6(SP)		; *,SOFT.OCCURED	5422
000550	001403			BEQ	18\$;	
000552	105260	000001		INCB	1(R0)		;	5424
000556	000401			BR	19\$;	5422
000560	105210		18\$:	INCB	(R0)		;	5426
000562	062706	000010	19\$:	ADD	#10,SP		;	5320
000566	000207			RTS	PC		;	

; Routine Size: 188 words, Routine Base: #CODE# + 25622
; Maximum stack depth per invocation: 12 words

; 5429
; 5430 end
; 5431
; 5432 eludom

OTS external references

.GLOBL \$SAVE5, \$SAVE4, \$SAVE3, \$SAVE2
.GLOBL BL\$SHF, BL\$DIV, BL\$MOD, BL\$MUL

PSECT SUMMARY

Psect Name	Words	Attributes	LCL	REL	CON
\$GGG\$	315	RO, I			
\$CODE\$	5765	RO, I			
\$PLIT\$	10	RO, D			

K2

ZRQAM3
V01.2

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (54)

SEQ 0437
Page 182

LIBRARY STATISTICS

```

:
:
:
:   File
:
:
:
:   DISK$USER2:[DIETZ.RDRX]ZRQACO.L16;14
```

	----- Total	Symbols Loaded	----- Percent	Blocks Read
	400	309	77	113

COMMAND QUALIFIERS

```

:
:   BLISS /PDP11 ZRQACO.BL2/LIST=ZRQACO.LI2/OBJECT=ZRQACO.OB2/SOURCE=PAGE:53
```


ZRQAM4

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (55)

```

:      5433 module ZRQAM4 (
:      5434
:      5435 #title 'RD/RX EXERCISER'
:      5436             ident = 'V01.2',
:      5437             addressing_mode (absolute),
:      5438             environment (noeis)
:      5439             ) =
:      5440
:      5441 begin
:      5442
:      5443 #sbttl 'LASTAD AND SETUP'
:      5444
:      5445 library 'ZRQACO.L16';
:      5446
:      5447 require 'BLSMAC.REQ';           ! DIAGNOSTIC SUPERVISOR LIBRARY
:      6938
:      6939 LASTAD
:      6940
:      6941 BGNSETUP (3)
:      6942
:      P 6943         BGNPTAB
:      P 6944         INIT_IP_ADDR, INIT_INTR_VECT, INIT_BR_LEVEL, #0'000030', 0, RD_MAX_LBN
:      P 6945         ! IP, VECTOR, BR, DISK ADDR, START BLOCK, END BLOCK
:      6946         ENDP TAB
:      6947
:      P 6948         BGNPTAB
:      P 6949         INIT_IP_ADDR, INIT_INTR_VECT, INIT_BR_LEVEL, #0'000001', 0, RX_MAX_LBN
:      P 6950         ! IP, VECTOR, BR, DISK ADDR, START BLOCK, END BLOCK
:      6951         ENDP TAB
:      6952
:      P 6953         BGNPTAB
:      P 6954         INIT_IP_ADDR, INIT_INTR_VECT, INIT_BR_LEVEL, #0'000002', 0, RX_MAX_LBN
:      P 6955         ! IP, VECTOR, BR, DISK ADDR, START BLOCK, END BLOCK
:      6956         ENDP TAB
:      6957
:      6958 ENDSETUP

```

```

.TITLE ZRQAM4 RD/RX EXERCISER
.IDENT /V01.2/
.ENABL AMA

```

```

000000
000000 000064'
000002 000000C
000004 000030'
000006 000006
000010 172150
000012 000154
000014 000004
000016 000030
000020 000000

```

```

.PSECT $XYZ$, RO
BL$LAS: .WORD T$FREE
        .WORD <<T$FREE - <BL$LAS+4>>/2>
P.AAA:  .WORD L$LAST+24
        .WORD 6
P.AAB:  .WORD -5630
        .WORD 154
        .WORD 4
        .WORD 30
        .WORD 0

```

; Plit count word

ZRQAM4
V01.2

RD/RX EXERCISER
LASTAD AND SETUP

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0439
Page 184
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (55)

000022 052137
000024 000050'
000026 000006
000030 172150
000032 000154
000034 000004
000036 000001
000040 000000
000042 001437
000044 000000
000046 000006
000050 172150
000052 000154
000054 000004
000056 000002
000060 000000
000062 001437
000064 000000

.WORD 52137
P.AAC: .WORD L\$LAST+44
.WORD 6
P.AAD: .WORD -5630
.WORD 154
.WORD 4
.WORD 1
.WORD 0
.WORD 1437
P.AAE: .WORD 0
.WORD 6
P.AAF: .WORD -5630
.WORD 154
.WORD 4
.WORD 2
.WORD 0
.WORD 1437
T\$FREE:: .WORD 0

; Plit count word

; Plit count word

000004'
000003
000004'
000010'
000024'
000030'
000044'
000050'

L\$LAST== BL\$LAS+4
T\$PTHV== 3
\$LAS4= P.AAA
\$REM4= P.AAB
\$LAS3= P.AAC
\$REM3= P.AAD
\$LAS1= P.AAE
\$REM2= P.AAF

000000 000207

.SBTTL \$END.LINK LASTAD AND SETUP
\$END.LINK::
RTS PC ;

6937

; Routine Size: 1 word, Routine Base: \$XYZ\$ + 0066
; Maximum stack depth per invocation: 0 words

; 6959 end
; 6960
; 6961 eludom

PSECT SUMMARY

; Psect Name Words Attributes
; \$XYZ\$ 28 RO, I, LCL, REL, CON

LIBRARY STATISTICS

N2

ZRQAM4
V01.2

RD/RX EXERCISER
LASTAD AND SETUP

14-Dec-1983 16:12:07
14-Dec-1983 16:12:00

SEQ 0440
Page 185
VAX-11 Bliss-16 V3-555
DISK\$USER2:[DIETZ.RDRX]ZRQACO.BL2;161 (55)

File	----- Total	Symbols Loaded	----- Percent	Blocks Read
DISK\$USER2:[DIETZ.RDRX]ZRQACO.L16;14	400	7	1	15

COMMAND QUALIFIERS

BLISS /PDP11 ZRQACO.BL2/LIST=ZRQACO.LI2/OBJECT=ZRQACO.OB2/SOURCE=PAGE:53

6962
6963
Size: 5766 code + 352 data words
Run Time: 03:18.3
Elapsed Time: 06:22.2
Memory Used: 419 pages
Compilation Complete