

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

REM 8 IDENTIFICATION

PRODUCT CODE: AC 1400 MC

PRODUCT NAME: CORQADO RQ001 DEC/VX MODULE

PRODUCT DATE: 9 APRIL 1984

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: RICHARD DILLZ

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

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

COPYRIGHT (C) 1983, 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

VAX
DECUS

UNIBUS
DECTAPE

MASSBUS

519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572

000001

```

;SBTTL MODULE PRIVATE DATA
;*****
;MACRO DEFINITIONS
;*****
;SBTTL MACROS
;
; IN CASE NO SOB INSTRUCTION
;
NOSOB      *      1
;MACRO      SOB      REG,TO
;            DEC      REG
;            BNE      TO
;ENDM

;*****
;MACRO      PUSH      A,B,C,D,E
;            MOV      A,(R6)
;IF          NB      B
;            MOV      B,(R6)
;ENDC
;IF          NB      C
;            MOV      C,(R6)
;ENDC
;IF          NB      D
;            MOV      D,(R6)
;ENDC
;IF          NB      E
;            MOV      E,(R6)
;ENDC
;ENDM

;*****
;MACRO      POP      A,B,C,D,E
;            MOV      (R6)+,A
;IF          NB      B
;            MOV      (R6)+,B
;ENDC
;IF          NB      C
;            MOV      (R6)+,C
;ENDC
;IF          NB      D
;            MOV      (R6)+,D
;ENDC
;IF          NB      E
;            MOV      (R6)+,E
;ENDC
;ENDM
;*****

```

```

575      .SBITH EQUATES
576
577      000001      BIT00 = 1
578      000002      BIT01 = 2
579      000004      BIT02 = 4
580      000010      BIT03 = 10
581      000020      BIT04 = 20
582      000040      BIT05 = 40
583      000100      BIT06 = 100
584      000200      BIT07 = 200
585      000400      BIT08 = 400
586      001000      BIT09 = 1000
587      002000      BIT10 = 2000
588      004000      BIT11 = 4000
589      010000      BIT12 = 10000
590      020000      BIT13 = 20000
591      040000      BIT14 = 40000
592      100000      BIT15 = 100000
593
594      000002      UPPER = 2
595
596      ;
597      ; STATUS BITS
598      ;
599      000001      SR.PAK = BIT00 ; PRINT OUT PACKET, 1 = YES
600      000002      SR.RON = BIT01 ; 0 MEANS READ ONLY
601      ; 1 MEANS WRITE, READ, CHECK
602      000004      SR.REP = BIT02 ; REPORT ERROR AS THEY OCCUR 0 = REPORT,
603      ; 1 = DON'T REPORT
604      000010      SR.SUM = BIT03 ; REPORT ERRORS ON END OF PASS 0 = REPORT,
605      ; 1 = DON'T REPORT
606      002000      SR.SEQ = BIT10 ; DISK ADDRESSING 0 = SEQUENTIAL,
607      ; 1 = RANDOM
608      004000      SR.CMP = BIT11 ; DATA COMPARE 0 = DO DATA COMPARE,
609      ; 1 = DON'T DO DATA COMPARE
610
611      ;
612      ; ERROR BITS
613      ;
614      000000      ERR.0 = 0 ; NOT DEFINED
615      000001      ERR.1 = 1 ; DATA ERROR
616      000003      ERR.3 = 3 ; CONTROLLER NOT READY
617      000004      ERR.4 = 4 ; BLOCK NOT FOUND
618      000006      ERR.6 = 6 ; DRIVE NOT READY, OFF LINE OR NON EXISTENT
619      000032      ERR.32 = 32 ; NPR ERROR
620      000034      ERR.34 = 34 ; UNABLE TO INITIALIZE ERROR
621      000036      ERR.36 = 36 ; UNABLE TO READ ERROR
622      000037      ERR.37 = 37 ; UNABLE TO WRITE ERROR
623      000045      ERR.45 = 45 ; CRC ERROR
624      000051      ERR.51 = 51 ; SEEK ERROR

```

```

626          .SBTTL RQDX BIT DEFINITIONS
627
628          ;RQDXSA REGISTER UNIVERSAL READ BITS
629
630          004000          SA.S1= 004000          ;STEP 1 STATUS BIT
631          010000          SA.S2= 010000          ;STEP 2 STATUS BIT
632          020000          SA.S3= 020000          ;STEP 3 STATUS BIT
633          040000          SA.S4= 040000          ;STEP 4 STATUS BIT
634          100000          SA.ERR= 100000         ;ERROR INDICATOR
635
636          ;RQDXSA REGISTER ERROR STATUS BITS
637
638          003777          SA.ERC= 003777          ;ERROR CODE
639
640          ;RQDXSA REGISTER STEP ONE READ BITS
641
642          003400          SA.CTP= 003400          ;CONTROLLER TYPE
643          000400          SA.DIA= 000400          ;DIAG BIT IN RQDXSA
644          001000          SA.QB = 001000          ;SAME - DENOTES QBUS SUPPORT
645
646          ;RQDXSA REGISTER STEP ONE WRITE BITS
647
648          000177          SA.VEC= 000177          ;INTERRUPT VECTOR (DIVIDED BY 4)
649          000200          SA.INT= 000200          ;INTERRUPT ENABLE DURING INITIALIZATION
650          007400          SA.RSP= 007400          ;RESPONSE RING LENGTH
651          170000          SA.CMD= 170000          ;COMMAND RING LENGTH
652
653          ;RQDXSA REGISTER STEP TWO READ BITS
654
655          000177          SA.VCE= 000177          ;INTERRUPT VECTOR ECHO
656          000200          SA.IFE= 000200          ;INTERRUPT ENABLE ECHO
657
658          ;RQDXSA REGISTER STEP TWO WRITE BITS
659
660          000001          SA.PRQ= 000001          ;LOW ORDER MESSAGE RING BYTE ADDRESS
661          ;ENABLE VAX QIBUS ADAPTER PURGE INTERRUPT
662
663          ;RQDXSA REGISTER STEP THREE READ BITS
664
665          000017          SA.RSE= 000017          ;RESPONSE RING LENGTH ECHO
666          000360          SA.CME= 000360          ;COMMAND RING LENGTH ECHO
667
668          ;RQDXSA REGISTER STEP THREE WRITE BITS
669
670          040000          SA.LEC= 040000          ;HIGH ORDER MESSAGE RING BYTE ADDRESS
671          ;LAST FAILURE CODE REQUEST
672
673          ;RQDXSA REGISTER STEP FOUR READ BITS
674
675          ;RQDXSA REGISTER STEP FOUR WRITE BITS
676
677          000001          SA.GO= 0100          ;GO BIT TO START RQDX FIRMWARE

```



```

679          .SBTTL COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS
680
681          100000          RG.OWN= BIT15          ;SET WHEN RQDX OWNS RING
682          040000          RG.FLG= BIT14          ;FLAG BIT
683
684          ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
685
686          000010          HC.SIZ= 8.            ;SIZE OF HOST COMM AREA IN BYTES
687          000050          PKTSIZ= 48.           ;SIZE OF PACKETS IN BYTES
688
689          000000          HC.RES= 0.            ;RESPONCE RING START
690          000002          HC.RCT= 2.            ;RESPONCE RING CONTROL WORD
691          000004          HC.CMD= 4.            ;COMMAND RING START
692          000006          HC.CCT= 6.            ;CONTROL RING CONTROL WORD
693          000076          HC.RPK= RSPACK         ;START OF RESPONCE PACKET BUFFER
694          000356          HC.CPK= HC.RPK+PKTSIZ ;START OF COMMAND PACKET BUFFER
695
696          .SBTTL COMMAND PACKET OPCODES
697
698          000001          OP.ABO= 01            ;ABORT COMMAND
699          000020          OP.ACC= 20            ;ACCESS COMMAND
700          000010          OP.AVL= 10            ;AVAILABLE COMMAND
701          000021          OP.CCD= 21            ;COMPARE CONTROLLER DATA COMMAND
702          000040          OP.CMP= 40            ;COMPARE HOST DATA COMMAND
703          000013          OP.DAP= 13            ;DETERMINE ACCESS PATHS COMMAND
704          000022          OP.ERS= 22            ;ERASE COMMAND
705          000023          OP.FLU= 23            ;FLUSH COMMAND
706          000002          OP.GCS= 02            ;GET COMMAND STATUS COMMAND
707          000003          OP.GUS= 03            ;GET UNIT STATUS COMMAND
708          000011          OP.ONL= 11            ;ONLINE COMMAND
709          000041          OP.RD= 41              ;READ COMMAND
710          000024          OP.RPL= 24            ;REPLACE COMMAND
711          000004          OP.SCC= 04            ;SET CONTROLLER CHARACTERISTICS COMMAND
712          000012          OP.SUC= 12            ;SET UNIT CHARACTERISTICS COMMAND
713          000042          OP.WR= 42              ;WRITE COMMAND
714          000030          OP.MRD= 30            ;MAINTENANCE READ COMMAND
715          000031          OP.MWR= 31            ;MAINTENANCE WRITE COMMAND
716          000000          OP.END= 200          ;END PACKET FLAG
717          000100          OP.AVA= 100           ;AVAILABLE ATTENTION MESSAGE
718          000101          OP.ERL= 101           ;ERROR LOG ATTENTION MESSAGE
719          000102          OP.SHC= 102           ;SHADOW COPY COMPLETE ATTENTION MESSAGE
720          000102          OP.ACP= 102           ;ACCESS PATH ATTENTION MESSAGE
721
722          ; NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
723          ; PACKET FLAG TO THE COMMAND OPCODE. THE UNKNOWN COMMAND END PACKET CONTAINS
724          ; JUST THE END PACKET FLAG IN ITS OPCODE FIELD.

```

9.0 OPERATION OPTIONS

ONE OR MORE SOFTWARE SWITCH REGISTERS CAN BE USED BY THE MODULE PROGRAM GENERAL PURPOSE SWITCHES. THESE WORDS ARE USED TO DEFINE OR SPECIFY A UNIQUE DEVICE OPTION OR TO POINT TO A SPECIFIC ROUTINE IN THE MODULE. ANY OPTION MUST BE SPECIFIED BY THE OPERATOR BEFORE THE MODULE IS RUN. SWITCH REGISTER 1 HAS THE FOLLOWING CHARACTERISTICS (ALL BITS DEFAULT TO 0):

SR1

BIT 0	SET (1):	PRINT OUT PACKETS AS THEY ARE SENT AND RECEIVED (FOR MODULE DIAGNOSTIC PURPOSES)
	RESET (0):	SUPPRESS PRINTING OF PACKETS
BIT 1	SET (1):	PERFORM FULL TEST OF WRITE, READ, AND DATA CHECK ON ALL DEVICES WHOSE WRITE BITS ARE SET IN SW3 (EXCEPT THE LOAD UNIT)
	RESET (0):	ONLY READ FROM THE DESIRED UNITS USING THE MSCP READ FUNCTION.
BIT 2	SET (1):	DO NOT REPORT ERRORS AS THEY OCCUR
	RESET (0):	REPORT ERRORS AS THEY OCCUR.
BIT 3	SET (1):	DO NOT PRINT ERROR SUMMARY AT END OF PASS.
	RESET (0):	PRINT ERROR SUMMARY AT END OF PASS.
BIT 4	SET (1):	PERFORM TEST OF WRITE, AND READ DBN'S ON RD DEVICE IF SELECTED IN DVID1.
	RESET (0):	DO NOT ACCESS THE DBN'S OF THE RD.
BIT 10	SET (1):	SELECT RANDOM BLOCK ADDRESSING.
	RESET (0):	SELECT SEQUENTIAL BLOCK ADDRESSING.
BIT 11	SET (1):	BYPASS DATA COMPARE.
	RESET (0):	DO DATA COMPARE.

SR2 CONTAINS A NUMBER USED AS A RATIO BETWEEN RD AND RX ACCESS. FOR EVERY 1 ACCESS OF AN RX THERE WILL BE SR2 TO EACH RD. IF ITS CONTENTS ARE 0 (THE DEFAULT) THE VALUE 80 WILL BE USED.

208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255

```

2951 011172      040      111      116 MSG10: .ASCIZ ' IN STEP '
2952              .EVEN
2953 011204      045      052      052 MSG11: .ASCIZ '*** SOFT ERROR COUNT #'
2954              .EVEN
2955 011234      045      052      052 MSG12: .ASCIZ '*** HARD ERROR COUNT #'
2956              .EVEN
2957 011264      045      052      052 MSG13: .ASCIZ '*** CHECK DATA ERROR COUNT #'
2958              .EVEN
2959 011322      045      101      104 MSG14: .ASCIZ ' ADDR = '
2960              .EVEN
2961 011334      040      040      127 MSG16: .ASCIZ ' WAS FOUND OFFLINE.'
2962              .EVEN
2963 011362      045      122      121 MSG17: .ASCIZ 'RQDXSA IS NOT ZERO, = '
2964              .EVEN
2965 011412      045      122      121 MSG18: .ASCIZ 'RQDX IS GOING THROUGH INITIALIZATION'
2966              .EVEN
2967 011460      045      122      111 MSG19: .ASCIZ 'RING AREA NOT CLEARED DURING INITIALIZE'
2968              .EVEN
2969 011532      045      104      105 MSG20: .ASCIZ 'DEVICE ID BIT = '
2970              .EVEN
2971 011556      123      124      101 MSG21: .ASCIZ 'STATUS ENDCOD UNIT# '
2972              .EVEN
2973 011604      040      102      131 MSG22: .ASCIZ ' BYTECO HI LBN LO LBN EXTADR PHYADR'
2974              .EVEN
2975 011652      040      000          MSG23: .ASCIZ ' '
2976              .EVEN
2977 011654      040      104      102 MSG24: .ASCIZ ' DBN'
2978              .EVEN
2979 011662      040      041      040 MSG36: .ASCIZ ' ! UNITS WILL ONLY BE READ TESTED !'
2980              .EVEN
2981 011730      007      007      040 MSG37: .ASCII '<07><07>' ! CUSTOMER DATA WILL BE OVERWRITTEN !'
2982              .EVEN
2983 012002      040      055      055          .ASCIZ ' ..... '<07><07>'
2984              .EVEN
2985 012054      040      111      106 MSG40: .ASCII ' IF CUSTOMER DATA IS TO BE DESTROYED, SET BIT1 (NOT BIT0)'
2986              .EVEN
2987 012146      040      111      116          .ASCIZ ' IN SWITCH REGISTER 1(SR1) EQUAL TO 1.'
2988              .EVEN
2989 012216      045      115      117 MSG99: .ASCIZ 'MORE UNITS REQUESTED THAN EXIST'
2990              .EVEN
2991 012260      045      105      122 MSGD1: .ASCIZ 'ERRORS CAUSED DRIVE TO BE DROPPED'
2992              .EVEN
2993 012324      045      104      122 MSGD2: .ASCIZ 'DRIVE WAS NOT FOUND OR NOT AVAILABLE'
2994              .EVEN
2995 012374      045      104      126 MSGD3: .ASCIZ 'DVID1 BIT SET HIGHER THAN ACTUAL # OF DRIVES FOUND'
2996              .EVEN
2997 012462          RBUF: .BLKW 260.          ;THE READ BUFFER
2998              .END
000001

```

311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357

10.0 PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/V11 DOCUMENT.

B. NON STANDARD PRINTOUTS INCLUDE ERROR MESSAGES WHICH DUMP THE FOLLOWING:

FOR MSCP:

- 1) SUMMARY STATUS
- 2) FLAGS AND ENCODE
- 3) UNIT NUMBER
- 4) BYTE COUNT
- 5) HI 16-BIT LBN VALUE
- 6) LO 16-BIT LBN VALUE
- 7) EXTENDED ADDRESS
- 8) PHYSICAL ADDRESS

ALL VALUES EXCEPT FOR PASS, RUNTIME AND ERRCNT ARE PRINTED IN OCTAL. PASS, RUNTIME AND ERRCNT ARE PRINTED IN DECIMAL. EXAMPLE:

RQAA0 PA: 00064116 APC: 004630 PASS: 00000 ERRCNT: 00001
 CSRA: 174344 CSRC: 000000 ASTAT: 000006 ERRTP: 000006
 RUNTIME: 000:00:22

RQAA0 PA: 00064052 APC: 004564 PASS: 00000

STATUS ENCODE UNIT # BYTCO HI LBN LO LBN EXTADR PHYADR
 000006 000242 000005 000000 000003 116321 000001 062100

FOR DUP:

- 1) SUMMARY STATUS (FROM RESPONSE PACKET)
- 2) FLAGS AND ENCODE (FROM RESPONSE PACKET)
- 3) UNIT NUMBER
- 4) DIAGNOSTIC BLOCK NUMBER

RQAA0 PA: 00064116 APC: 004630 PASS: 00000 ERRCNT: 00001
 CSRA: 174344 CSRC: 000000 ASTAT: 000006 ERRTP: 000006
 RUNTIME: 000:00:22

STATUS ENCODE UNIT # DBN
 000006 000242 000005 000003 INCORRECT END CODE

```

871          .SBTTL ERROR LOG FLAGS
872
873          000200          EF.FRS= 000200          ;FIRST PACKET
874          000100          EF.LST= 000100          ;LAST PACKET
875          000001          EF.MIS= 000001          ;MESSAGE MISSING
876
877          ;
878          ;          ERROR LOG MESSAGE OFFSETS
879          ;
880          000000          L.EVNT= 0.              ;EVENT CODE
881          000001          L.SLOT= 2.              ;SLOT NUMBER
882          000004          L.CNTI= 4.              ;CONTROLLER IDENTIFIER
883          000014          L.CNTI= 12.             ;CONTROLLER SOFTWARE REVISION
884          000015          L.CHVR= 13.             ;CONTROLLER HARDWARE REVISION
885          000016          L.UNTI= 14.             ;UNIT IDENTIFIER
886          000026          L.USVR= 22.             ;UNIT SOFTWARE REVISION
887          000027          L.UHVR= 23.             ;UNIT HARDWARE REVISION
888          000030          L.ERLC= 24.             ;ERROR LOCATION
889          000034          L.CYL= 28.              ;CYLINDER
890          000040          L.GRP= 32.              ;GROUP
891          000041          L.TRCK= 33.              ;TRACK
892          000042          L.SCTR= 34.              ;SECTOR
893          000044          L.VSER= 36.              ;VOLUME SERIAL NUMBER
894          000050          L.DATA= 40.              ;EVENT DEPENDENT DATA
895
896          ;
897          ;          STATUS AND EVENT CODE DEFINITIONS
898          ;
899          000037          ST.MSK= 37              ;STATUS / EVENT CODE MASK
900          000040          ST.SUB= 40              ;SUB CODE MULTIPLIER
901          000000          ST.SUC= 0              ;SUCCESS
902          000001          ST.CMD= 1              ;INVALID COMMAND
903          000002          ST.ABO= 2              ;COMMAND ABORTED
904          000003          ST.OFL= 3              ;UNIT OFFLINE
905          000004          ST.AVL= 4              ;UNIT-AVAILABLE
906          000005          ST.MFE= 5              ;MEDIA ERROR
907          000006          ST.WPR= 6              ;WRITE PROTECTED
908          000007          ST.CMP= 7              ;COMPARE ERROR
909          000010          ST.DAT= 10             ;DATA ERROR
910          000011          ST.HST= 11             ;HOST BUFFER ACCESS ERROR
911          000012          ST.CNT= 12             ;CONTROLLER ERROR
912          000013          ST.DRV= 13             ;DRIVE ERROR
913          000037          ST.DIA= 37              ;MESSAGE FROM AN INTERNAL DIAGNOSTIC
914
915          ;
916          ;          SUBCODES FOR ST.OFL
917          ;
918          000040          SC.NVL= 40              ;NO VOLUME MOUNTED OR DRIVE DISABLED VIA RON STOP SWITCH
919          000100          SC.IOP= 100             ;UNIT INOPERATIVE
920          000400          SC.DIS= 400             ;UNIT DISABLED BY FIELD SERVICE OR INTERNAL DIAGNOSTIC
921          000200          SC.DUP= 200             ;DUPLICATE UNIT NUMBER
922
923          ;
924          ;          SUBCODES FOR ST.DRV
925          ;
926          000040          SC.SDI= 40              ;SDI RESPONSE TIME OFF
927          000100          SC.INV= 100             ;INVALID SDI RESPONSE
  
```

392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405

000000
 000000

12.0 BIBLIOGRAPHY

CXQUAAO 'DEC/11 USER'S MANUAL' SEPT 1978

CXQAFDO 'DEC/11 PROGRAMMERS'S GUIDE' SEPT 1978

8.

.SBTTL MODULE HEADER BLOCK

IOMODX <RDAD >,172150,154,4,0,0,40,104,RBUF,256.,256.

MODULE 150000,RDAD,172150,154,4,0,0,40,104,RBUF,256.,256.

.TITLE RDAD DEC/11 SYSTEM EXERCISER MODULE

.DOXCOM VERSION 6.4 28-JAN-82

.LIST BIN

BEGIN:

101 MODNAM: .ASCII /RDAD / ;MODULE NAME.

XFLAG: .BYTE OPEN

;USED TO KEEP TRACK OF WBUF USAGE

ADDR: 172150+0

;1ST DEVICE ADDR.

VECTOR: 154+0

;1ST DEVICE VECTOR.

BR1: .BYTE PRT1+0

;1ST BR LEVEL.

BR2: .BYTE PRT2+0

;2ND BR LEVEL.

DVID1: 0+1

;DEVICE INDICATOR 1.

SR1: OPEN

;SWITCH REGISTER 1

SR2: OPEN

;SWITCH REGISTER 2

SR3: OPEN

;SWITCH REGISTER 3

SR4: OPEN

;SWITCH REGISTER 4

STAT: 150000

;STATUS WORD.

INIT: START

;MODULE START ADDR.

SPOINT: MODSP

;MODULE STACK POINTER.

PASCNT: 0

;PASS COUNTER.

ICONT: 40

;# OF ITERATIONS PER PASS+40

ICOUNT: 0

;LOC TO COUNT ITERATIONS

SOFCNT: 0

;LOC TO SAVE TOTAL SOFT ERRORS

HRDCNT: 0

;LOC TO SAVE TOTAL HARD ERRORS

SOFPAS: 0

;LOC TO SAVE SOFT ERRORS PER PASS

HRDPAS: 0

;LOC TO SAVE HARD ERRORS PER PASS

SYSCNT: 0

;# OF SYS ERRORS ACCUMULATED

RANUM: 0

;HOLDS RANDOM # WHEN RAND MACRO IS CALLED

CONF IG:

;RESERVED FOR MONITOR USE

RES1: 0

;RESERVED FOR MONITOR USE

RES2: 0

;RESERVED FOR MONITOR USE

SVR0: OPEN

;LOC TO SAVE R0.

SVR1: OPEN

;LOC TO SAVE R1.

SVR2: OPEN

;LOC TO SAVE R2.

SVR3: OPEN

;LOC TO SAVE R3.

SVR4: OPEN

;LOC TO SAVE R4.

SVR5: OPEN

;LOC TO SAVE R5.

SVR6: OPEN

;LOC TO SAVE R6.

CDR: OPEN

;ADDR OF CURRENT CDR.

SEADR:

;ADDR OF GOOD DATA, OR

000000
 000000 122 121
 000003 104 040
 000005 000
 000006 172150
 000010 000154
 000012 200
 000013 000
 000014 000001
 000016 000000
 000020 000000
 000022 000000
 000024 000000

000026 150000
 000030 001002
 000032 000252
 000034 000000
 000036 000140
 000040 000000
 000042 000000
 000044 000000
 000046 000000
 000050 000000
 000052 000000
 000054 000000
 000056
 000056 000000
 000060 000000
 000062 000000
 000064 000000
 000066 000000
 000070 000000
 000072 000000
 000074 000000
 000076 000000
 000100 000000
 000102


```

407 *****
408 ;SETTL MODULE STORAGE AREA
409 ;      VERSION 2.0      FOR RELEASE
410 ;      XRDADO - THIS MODULE IS ADAPTED FREELY FROM XDUBC.
411 ;      THE GENERAL STRUCTURE IS THE SAME, BUT SOME DEVICE
412 ;      SPECIFIC CODE WAS ADDED.
413 ;
414 ;RQDXIP: .WORD 0          ;RQDX POLLING REG
415 000252 000000 RQDXSA: .WORD 0          ;RQDX STATUS REG
416 ;-----
417 ; THESE STORAGE AREAS ARE USED FOR COMMUNICATIONS WITH THE
418 ; CONTROLLER.
419 ;-----
420 000254 000000 CINTR: .WORD 0          ; COMMAND INTERRUPT INDICATOR
421 000256 000000 RINTR: .WORD 0          ; RESPONSE INTERRUPT INDICATOR
422 000260        RSPONC: .BLKW 2          ; MESSAGE RING
423 000264        COMMND: .BLKW 1          ; COMMAND RING
424 000270 000000        .WORD 0          ;
425 ;-----
426 000272 000000 RSPLN: .WORD 0          ; RESPONSE PACKET LENGTH
427 000274 000000 RSPVIR: .WORD 0        ; RESPONSE PACKET VIRTUAL CIRCUIT
428 000276        RSPACK: .BLKW 30       ; RESPONSE PACKET
429 ;-----
430 000372 000000 CMPLN: .WORD 0          ; COMMAND PACKET LENGTH
431 000374 000000 CMPVIR: .WORD 0        ; COMMAND PACKET VIRTUAL CIRCUIT
432 000376        CMPACK: .BLKW 30       ; COMMAND PACKET
433 ;-----
434 ; THESE STORAGE AREAS ARE TO KEEP TRACK OF THE ABOVE COMMUNICATION
435 ; AREAS.
436 ;-----
437 000472 000000 CMDREF: .WORD 0          ; COMMAND REFERENCE NUMBER
438 000474 000000 BYTCNT: .WORD 0        ; BYTE COUNT
439 000476 000000 DATLOC: .WORD 0        ; DATA LOCATION
440 000500 000000 ENDCOD: .WORD 0        ; END CODE EXPECTED
441 ;-----
442 ; ADDRESS MAPPING AREAS.
443 ;-----
444 000502 000264 VA: .WORD COMMND        ; GENERIC VIRTUAL ADDRESS FOR GETPA
445 000504 000000 PA: .WORD OPEN          ; GENERIC PHYSICAL ADDRESS
446 000506 000000 EA: .WORD OPEN          ; GENERIC EXTENDED ADDRESS
447 000510 000000 LO22: .WORD 0          ; 22 BIT LOW ORDER ADDRESS
448 000512 000000 HI22: .WORD 0          ; 22 BIT HIGH ORDER ADDRESS
449 ;-----
450 000514 000000 NUM: .WORD 0            ; ADDRESS USED IN OTOA
451 000516 000000 OLDPA: .WORD 0          ; THE OLD PHYSICAL ADDRESS
452 000520 000000 OLDEA: .WORD 0          ; THE OLD EXTENDED ADDRESS TO CHECK IF
453 ;      RQDX WILL BE REINITED
454 000522 000000 SAVPA: .WORD 0          ; SAVED PA NEEDED FOR DUP
455 000524 000000 SAVEA: .WORD 0          ; SAVED EA
456 ;-----

```

458					
459		000017		PRNUM = 15.	;PRINT MESSAGE EVERY 15TH TIME
460	000526	000017		PRMSG: .WORD PRNUM	;PRINT WORD SAVES THE VALUE TO CHECK FOR WHEN
461					; AN END OF PASS MESSAGE IS WRITTEN
462		027340		TIMER = 12000.	;TIMER TO WAIT 2 3 SECONDS AFTER DAP COMMAND
463					
464	000530			TEMP;	;LABEL FOR CONVENIENCE
465				ADR'X: .BLKB %	
466				.BYTE 0	
467				.ENDM	
468	000530			ADR1: .BLKB %	
	000536	000		.BYTE 0	
	000537			ADR2: .BLKB %	
	000545	000		.BYTE 0	
	000546			ADR3: .BLKB %	
	000554	000		.BYTE 0	
	000555			ADR4: .BLKB %	
	000563	000		.BYTE 0	
	000564			ADR5: .BLKB %	
	000572	000		.BYTE 0	
	000573			ADR6: .BLKB %	
	000601	000		.BYTE 0	
	000602			ADR7: .BLKB %	
	000610	000		.BYTE 0	
	000611			ADR8: .BLKB %	
	000617	000		.BYTE 0	
469				.EVEN	


```

471      .SBITL MORE MODULE STORAGE.
472
473 000620 000000      FLAGS: .WORD 0      ; MODULE FLAGS
474 000622 000000      ERRFLG: .WORD 0      ; ERROR FLAGS FROM DUP PART OF MODULE
475 000624 000000      DBN: .WORD 0      ; DBN FOR DUP
476 000626 000000      SEFLG: .WORD 0      ; SOFT_ERROR_FLAGS
477
478 000630 000000      XSECL: .WORD 0      ; CURRENT SECTOR LO ORDER ADDRESS FOR DX
479 000632 000000      XSECH: .WORD 0      ; CURRENT SECTOR HI ORDER ADDRESS FOR DX
480
481 000634 000000      RSECL: .WORD 0      ; CURRENT SECTOR LO ORDER ADDRESS FOR DR
482 000636 000000      RSECH: .WORD 0      ; CURRENT SECTOR HI ORDER ADDRESS FOR DR
483
484 000640 000000      UNSZL: .WORD 0      ; UNIT SIZE LO ORDER LIMIT FROM ONLINE CMND
485 000642 000000      UNSZH: .WORD 0      ; UNIT SIZE HI ORDER LIMIT
486
487 000644 000001      DEVICE: .WORD 1      ; DEVICE TO TEST
488 000646 000000      TRY: .WORD 0      ; NUMBER OF TRIES
489 000650 000000      UNITFL: .WORD 0      ; SAVE UNIT FLAGS
490 000652 000000      WORK: .WORD 0      ; TEMPORARY WORK AREA
491
492      TIMEOUT = 3000.      ; TIME OUT GADGE
493      RETIM = 4      ; RETRY LIMIT
494
495 000654 000000      ISRX: .WORD 0      ; BITMAP, SET IF DEVICE IS RX, CLEAR IF RD
496
497      $$ = 1      ; SET UP BIT MASK
498 000656 000020      TABLEW: .REPT 16.
499      .LIST      ;
500      .WORD 0, $$
501      $$ = $$ + 2
502      .LIST
503      .ENDR
504 000756      TEND:      ; END MARKER
505
506 000756 000000      RVA: .WORD 0      ; READ BUFFER STRUCTURE FOR 12 BIT ADDRESS
507 000760 000000      R18PA: .WORD 0
508 000762 000000      R18EA: .WORD 0
509 000764 000000      R22PA: .WORD 0
510 000766 000000      R22EA: .WORD 0
511
512 000770 000000      WVA: .WORD 0      ; WRITE BUFFER STRUCTURE FOR 20 BIT ADDRESS
513 000772 000000      W18PA: .WORD 0
514 000774 000000      W18EA: .WORD 0
515 000776 000000      W22PA: .WORD 0
516 001000 000000      W22EA: .WORD 0
517

```

519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573

000001

```
.SBTTL MODULE PRIVATE DATA
;*****
;*      MACRO DEFINITIONS
;*****
;SBTTL MACROS
;
; IN CASE NO SOB INSTRUCTION
;
NOSOB      *      1
.MACRO     SOB      REG,TO
          DEC      REG
          BNE      TO
.ENDM

;*****
.MACRO     PUSH      A,B,C,D,E
          MOV      A,(R6)
          IF      NB      B
          MOV      B,(R6)
          .ENDC
          IF      NB      C
          MOV      C,(R6)
          .ENDC
          IF      NB      D
          MOV      D,(R6)
          .ENDC
          IF      NB      E
          MOV      E,(R6)
          .ENDC
.ENDM

;*****
.MACRO     POP      A,B,C,D,E
          MOV      (R6)+,A
          IF      NB      B
          MOV      (R6)+,B
          .ENDC
          IF      NB      C
          MOV      (R6)+,C
          .ENDC
          IF      NB      D
          MOV      (R6)+,D
          .ENDC
          IF      NB      E
          MOV      (R6)+,E
          .ENDC
.ENDM
;*****
```

```

575          ,SBITH EQUATES
576
577          000001      BIT00 = 1
578          000002      BIT01 = 2
579          000004      BIT02 = 4
580          000010      BIT03 = 10
581          000020      BIT04 = 20
582          000040      BIT05 = 40
583          000100      BIT06 = 100
584          000200      BIT07 = 200
585          000400      BIT08 = 400
586          001000      BIT09 = 1000
587          002000      BIT10 = 2000
588          004000      BIT11 = 4000
589          010000      BIT12 = 10000
590          020000      BIT13 = 20000
591          040000      BIT14 = 40000
592          100000      BIT15 = 100000
593
594          000002      UPPER = 2
595
596          ;
597          ; STATUS BITS
598          ;
599          000001      SR.PAK = BIT00 ; PRINT OUT PACKET, 1 = YES
600          000002      SR.RON = BIT01 ; 0 MEANS READ ONLY
601          ; 1 MEANS WRITE, READ, CHECK
602          000004      SR.REP = BIT02 ; REPORT ERROR AS THEY OCCUR 0 = REPORT,
603          ; 1 = DON'T REPORT
604          000010      SR.SUM = BIT03 ; REPORT ERRORS ON END OF PASS 0 = REPORT,
605          ; 1 = DON'T REPORT
606          002000      SR.SEQ = BIT10 ; DISK ADDRESSING 0 = SEQUENTIAL,
607          ; 1 = RANDOM
608          004000      SR.CMP = BIT11 ; DATA COMPARE 0 = DO DATA COMPARE,
609          ; 1 = DON'T DO DATA COMPARE
610
611          ;
612          ; ERROR BITS
613          ;
614          000000      ERR.0 = 0 ; NOT DEFINED
615          000001      ERR.1 = 1 ; DATA ERROR
616          000003      ERR.3 = 3 ; CONTROLLER NOT READY
617          000004      ERR.4 = 4 ; BLOCK NOT FOUND
618          000006      ERR.6 = 6 ; DRIVE NOT READY, OFF LINE OR NON EXISTENT
619          000032      ERR.32 = 32 ; NPR ERROR
620          000034      ERR.34 = 34 ; UNABLE TO INITIALIZE ERROR
621          000036      ERR.36 = 36 ; UNABLE TO READ ERROR
622          000037      ERR.37 = 37 ; UNABLE TO WRITE ERROR
623          000043      ERR.43 = 43 ; CRC ERROR
624          000051      ERR.51 = 51 ; SEEK ERROR

```

```

626      .SRTIL RQDX BIT DEFINITIONS
627
628      ;RQDXSA REGISTER UNIVERSAL READ BITS
629
630          004000      SA.S1* 004000      ;STEP 1 STATUS BIT
631          010000      SA.S2* 010000      ;STEP 2 STATUS BIT
632          020000      SA.S3* 020000      ;STEP 3 STATUS BIT
633          040000      SA.S4* 040000      ;STEP 4 STATUS BIT
634          100000      SA.ERR* 100000     ;ERROR INDICATOR
635
636      ;RQDXSA REGISTER ERROR STATUS BITS
637
638          003777      SA.ERC* 003777     ;ERROR CODE
639
640      ;RQDXSA REGISTER STEP ONE READ BITS
641
642          003400      SA.CTP* 003400     ;CONTROLLER TYPE
643          000400      SA.DIA* 000400     ;DIAG BIT IN RQDXSA
644          001000      SA.QB * 001000     ;SAME - DENOTES QBUS SUPPORT
645
646      ;RQDXSA REGISTER STEP ONE WRITE BITS
647
648          000177      SA.VEC* 000177     ;INTERRUPT VECTOR (DIVIDED BY 4)
649          000200      SA.INT* 000200     ;INTERRUPT ENABLE DURING INITIALIZATION
650          007400      SA.RSP* 007400     ;RESPONCE RING LENGTH
651          170000      SA.CMD* 170000     ;COMMAND RING LENGTH
652
653      ;RQDXSA REGISTER STEP TWO READ BITS
654
655          000177      SA.VCE* 000177     ;INTERRUPT VECTOR ECHO
656          000200      SA.INE* 000200     ;INTERRUPT ENABLE ECHO
657
658      ;RQDXSA REGISTER STEP TWO WRITE BITS
659
660          000001      SA.PRG* 000001     ;LOW ORDER MESSAGE RING BYTE ADDRESS
661          000001      SA.PRG* 000001     ;ENABLE VAX UNIBUS ADAPTER PURGE INTERRUPT
662
663      ;RQDXSA REGISTER STEP THREE READ BITS
664
665          000017      SA.RSE* 000017     ;RESPONCE RING LENGTH ECHO
666          000360      SA.CME* 000360     ;COMMAND RING LENGTH ECHO
667
668      ;RQDXSA REGISTER STEP THREE WRITE BITS
669
670          040000      SA.LEC* 040000     ;HIGH ORDER MESSAGE RING BYTE ADDRESS
671          040000      SA.LEC* 040000     ;LAST FAILURE CODE REQUEST
672
673      ;RQDXSA REGISTER STEP FOUR READ BITS
674
675      ;RQDXSA REGISTER STEP FOUR WRITE BITS
676
677          000001      SA.GO* 0110        ;GO BIT TO START RQDX FIRMWARE

```

```

679      .SBTTL COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS
680
681      100000      RG.OWN= BIT15      ;SET WHEN RQDX OWNS RING
682      040000      RG.FLG= BIT14      ;FLAG BIT
683
684      ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
685
686      000010      HC.SIZ= 8.          ;SIZE OF HOST COMM AREA IN BYTES
687      000030      PKTSIZ= 48.         ;SIZE OF PACKETS IN BYTES
688
689      000000      HC.RES= 0.          ;RESPONCE RING START
690      000002      HC.RCT= 2.          ;RESPONCE RING CONTROL WORD
691      000004      HC.CMD= 4.          ;COMMAND RING START
692      000006      HC.CCT= 6.          ;CONTROL RING CONTROL WORD
693      000276      HC.RPK= RSPACK      ;START OF RESPONCE PACKET BUFFER
694      000356      HC.CPK= HC.RPK+PKTSIZ ;START OF COMMAND PACKET BUFFER
695

```

```

696      .SBTTL COMMAND PACKET OPCODES
697
698      000001      OP.ABO= 01          ;ABORT COMMAND
699      000020      OP.ACC= 20          ;ACCESS COMMAND
700      000010      OP.AVL= 10          ;AVAILABLE COMMAND
701      000021      OP.CCD= 21          ;COMPARE CONTROLLER DATA COMMAND
702      000040      OP.CMP= 40          ;COMPARE HOST DATA COMMAND
703      000013      OP.DAP= 13          ;DETERMINE ACCESS PATHS COMMAND
704      000022      OP.ERS= 22          ;ERASE COMMAND
705      000023      OP.FLU= 23          ;FLUSH COMMAND
706      000002      OP.GCS= 02          ;GET COMMAND STATUS COMMAND
707      000003      OP.GUS= 03          ;GET UNIT STATUS COMMAND
708      000011      OP.ONL= 11          ;ONLINE COMMAND
709      000041      OP.RD= 41           ;READ COMMAND
710      000024      OP.RPL= 24          ;REPLACE COMMAND
711      000004      OP.SCC= 04          ;SET CONTROLLER CHARACTERISTICS COMMAND
712      000012      OP.SUC= 12          ;SET UNIT CHARACTERISTICS COMMAND
713      000042      OP.WR= 42           ;WRITE COMMAND
714      000030      OP.MRD= 30          ;MAINTENANCE READ COMMAND
715      000031      OP.MWR= 31          ;MAINTENANCE WRITE COMMAND
716      000200      OP.END= 200         ;END PACKET FLAG
717      000100      OP.AVA= 100         ;AVAILABLE ATTENTION MESSAGE
718      000101      OP.ERL= 101         ;ERROR LOG ATTENTION MESSAGE
719      000102      OP.SHC= 102         ;SHADOW COPY COMPLETE ATTENTION MESSAGE
720      000102      OP.ACP= 102         ;ACCESS PATH ATTENTION MESSAGE
721

```

```

; NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
; PACKET FLAG TO THE COMMAND OPCODE. THE UNKNOWN COMMAND END PACKET CONTAINS
; JUST THE END PACKET FLAG IN ITS OPCODE FIELD.

```

226		.SBTTL COMMAND MODIFIERS	
227			
228	040000	MD.CMP= 040000	;COMPARE
229	100000	MD.EXP= 100000	;EXPRESS REQUEST
230	010000	MD.FRR= 010000	;FORCE ERROR
231	004000	MD.SCH= 004000	;SUPPRESS CACHING (HIGH SPEED)
232	002000	MD.SCL= 002000	;SUPPRESS CACHING (LOW SPEED)
233	001000	MD.SEC= 001000	;SUPPRESS ERROR CORRECTION
234	000400	MD.SER= 000400	;SUPPRESS ERROR RECOVERY
235	000200	MD.SSH= 000200	;SUPPRESS SHADOWING
236	000100	MD.WBN= 000100	;WRITE-BACK (NON VOLATILE)
237	000040	MD.WBV= 000040	;WRITE-BACK (VOLATILE)
238	000001	MD.SPD= 000001	;SPIN-DOWN
239	000001	MD.FEU= 000001	;FLUSH ENTIRE UNIT
240	000002	MD.VOL= 000002	;VOLATILE ONLY
241	000001	MD.NXU= 000001	;NEXT UNIT
242			
243		.SBTTL END PACKET FLAGS	
244			
245	000200	EF.BBR= 000200	;BAD BLOCK REPORTED
246	000100	EF.BBU= 000100	;BAD BLOCK UNREPORTED
247	000040	EF.LOG= 000040	;ERROR LOG GENERATED
248	000020	EF.SEX= 000020	;SERIOUS EXCEPTION
249			
250			
251		.SBTTL UNIT FLAGS	
252			
253	000001	UF.CMR= 000001	;COMPARE READS
254	000002	UF.CMW= 000002	;COMPARE WRITES
255	010000	UF.RPI= 010000	;HOST INITIATED BAD BLOCK REPLACEMENT
256	040000	UF.INA= 040000	;INACTIVE SHADOW SET UNIT
257	000200	UF.RMV= 000200	;REMOVABLE MEDIA
258	004000	UF.SCH= 004000	;SUPPRESS CACHING (HIGH SPEED)
259	002000	UF.SCL= 002000	;SUPPRESS CACHING (LOW SPEED)
260	000040	UF.WBN= 000040	;WRITE-BACK (NON VOLATILE)
261	020000	UF.WPH= 020000	;WRITE-PROTECT(HARDWARE)
262	010000	UF.WPS= 010000	;WRITE-PROTECT(SOFTWARE OR VOLUME)
263	000004	UF.S76= 000004	;576 BYTE SECTORS


```

765          .SBTTL CONTROLLER FLAGS
766
767          000000      CF.AVL= 000200      ;ENABLE AVAILABLE ATTENTION MESSAGES
768          000100      CF.MSC= 000100      ;ENABLE MISCELLANEOUS ERROR LOG MESSAGES
769          000040      CF.OTH= 000040      ;ENABLE OTHER HOST'S ERROR LOG MESSAGES
770          000020      CF.IHS= 000020      ;ENABLE THIS HOST'S ERROR LOG MESSAGES
771          000000      CF.SHD= 000000      ;SHADOWING
772          000001      CF.S76= 000001      ;576 BYTE SECTORS
773
774          .SBTTL COMMAND PACKET OFFSETS
775
776          ; GENERIC COMMAND PACKET OFFSETS:
777          000000      P.CRF= 0.             ;COMMAND REFERENCE NUMBER
778          000004      P.UNIT= 4.            ;UNIT NUMBER
779          000010      P.OPCD= 8.            ;OPCODE
780          000010      P.MOD= 10.            ;MODIFIERS
781          000014      P.BCNT= 12.           ;BYTE COUNT
782          000020      P.BUFF= 16.           ;BUFFER DESCRIPTOR
783          000020      P.ADPA= 16.           ;BUFFER'S PHYSICAL ADDRESS (P.BUFF)
784          000020      P.ADEA= 18.           ;BUFFER'S EXTENDED ADDRESS (P.BUFF+2)
785          000034      P.LBN= 28.            ;LOGICAL BLOCK NUMBER
786          000040      P.SETW= 32.           ;SOFTWARE WORDS
787
788          ; ABORT AND GET COMMAND STATUS COMMAND PACKET OFFSETS:
789          000014      P.OTRF= 12.           ;OUTSTANDING REFERENCE NUMBER
790
791          ; ONLINE AND SET UNIT CHARACTERISTICS COMMAND PACKET OFFSETS:
792          000014      P.UNFL= 14.            ;UNIT FLAGS
793          000020      P.HSTI= 16.            ;HOST IDENTIFIER
794          000024      P.UNIT= 20.            ;UNIT IDENTIFIER
795          000034      P.ELGF= 28.            ;ERROR LOG FLAGS
796          000040      P.SHUN= 32.            ;SHADOW UNIT
797          000042      P.CPSP= 34.            ;COPY SPEED
798
799          ; REPLACE COMMAND PACKET OFFSETS:
800          000014      P.RBN= 12.             ;REPLACEMENT BLOCK NUMBER
801
802          ; SET CONTROLLER CHARACTERISTICS COMMAND PACKET OFFSETS:
803          000014      P.VRSN= 12.            ;MSCP VERSION
804          000016      P.CNFI= 14.            ;CONTROLLER FLAGS
805          000020      P.HIMO= 16.            ;HOST TIMEOUT
806          000022      P.USEF= 18.            ;USE FRACTION
807          000024      P.TIME= 20.            ;QUAD WORD TIME AND DATE
808
809          ; MAINTENANCE READ AND MAINTENANCE WRITE COMMAND PACKET OFFSETS:
810          000034      P.RGID= 28.            ;REGION ID
811          000040      P.RGOF= 32.            ;REGION OFFSET

```

```

813          .SBTTL END PACKET OFFSETS
814
815          ; GENERIC END PACKET OFFSETS:
816          000000          P.CRF= 0.          ;COMMAND REFERENCE NUMBER
817          000004          P.UNIT= 4.          ;UNIT NUMBER
818          000010          P.OPCD= 8.          ;OPCODE (ALSO CALLED ENDCODE)
819          000011          P.FLAGS= 9.          ;END PACKET FLAGS
820          000012          P.STS= 10.          ;MODIFIERS
821          000014          P.BCNT= 12.          ;BYTE COUNT
822          000034          P.FBBK= 28.          ;FIRST BAD BLOCK
823          000040          P.SETW= 32.          ;SOFTWARE WORDS
824
825          ; GET COMMAND STATUS END PACKET OFFSETS:
826          000014          P.OTRF= 12.          ;OUTSTANDING REFERENCE NUMBER
827          000020          P.CMST= 16.          ;COMMAND STATUS
828
829          ; GET UNIT STATUS END PACKET OFFSETS:
830          000014          P.MLUN= 12.          ;MULTI-UNIT CODE
831          000016          P.UNFL= 14.          ;UNIT FLAGS
832          000020          P.HSTI= 16.          ;HOST IDENTIFIER
833          000024          P.UNIT= 20.          ;UNIT IDENTIFIER
834          000034          P.DEV= 28.          ;MEDIA TYPE IDENTIFIER
835          000040          P.SHUN= 32.          ;SHADOW UNIT
836          000042          P.SHST= 34.          ;SHADOW STATUS
837          000044          P.TRCK= 36.          ;TRACK SIZE
838          000046          P.GRP= 38.          ;GROUP SIZE
839          000050          P.CYL= 40.          ;CYLINDER SIZE
840          000054          P.RCTS= 44.          ;RCT TABLE SIZE
841          000056          P.RBNS= 46.          ;RBN / TRACK
842          000057          P.RCTC= 47.          ;RCT COPIES
843
844          ; ONLINE AND SET UNIT CHARACTERISTICS
845          000014          P.MLUN= 12.          ;MULTI-UNIT CODE
846          000016          P.UNFL= 14.          ;UNIT FLAGS
847          000020          P.HSTI= 16.          ;HOST IDENTIFIER
848          000021          P.UNIT= 20.          ;UNIT IDENTIFIER
849          000040          P.SHUN= 32.          ;SHADOW UNIT
850          000044          P.ONSZ= 36.          ;UNIT SIZE
851          000050          P.VSER= 40.          ;VOLUME SERIAL NUMBER
852
853          ; SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS:
854          000014          P.VRSN= 12.          ;MSCP VERSION
855          000016          P.CNTR= 14.          ;CONTROLLER FLAGS
856          000020          P.CTMO= 16.          ;CONTROLLER TIMEOUT
857          000022          P.CNCL= 18.          ;CONTROLLER COMMAND LIMIT
858          000024          P.CNTI= 20.          ;CONTROLLER ID
859          000034          P.MEDI= 28.          ;MEDIA TYPE
860          000042          P.SHST= 34.          ;SHADOW STATUS
861
862          ; ERROR LOG ATTENTION MESSAGE PACKET OFFSETS:
863          000000          P.CRF= 0.          ;COMMAND REFERENCE NUMBER
864          000004          P.UNIT= 4.          ;UNIT NUMBER
865          000006          P.CNT= 6.          ;COUNT
866          000010          P.OPCD= 8.          ;OPCODE
867          000011          P.FLAGS= 9.          ;ERROR LOG FLAGS
868          000012          P.SIZE= 10.          ;SIZE OR OFFSET
869          000014          P.LGDT= 12.          ;START OF ERROR LOG DATA

```



```

871      .SBTTL ERROR LOG FLAGS
872
873      000200      EF.FRS= 000200      ;FIRST PACKET
874      000100      EF.LST= 000100      ;LAST PACKET
875      000001      EF.MIS= 000001      ;MESSAGE MISSING
876
877      ;
878      ;
879      ;
880      000000      L.EVNT= 0.            ;EVENT CODE
881      000001      L.SLOT= 2.            ;SLOT NUMBER
882      000004      L.CNTI= 4.            ;CONTROLLER IDENTIFIER
883      000014      L.CNTI= 12.           ;CONTROLLER SOFTWARE REVISION
884      000015      L.CHVR= 13.           ;CONTROLLER HARDWARE REVISION
885      000016      L.UNIT= 14.           ;UNIT IDENTIFIER
886      000026      L.USVR= 22.           ;UNIT SOFTWARE REVISION
887      000027      L.UHVR= 23.           ;UNIT HARDWARE REVISION
888      000030      L.ERLC= 24.           ;ERROR LOCATION
889      000034      L.CYL= 28.            ;CYLINDER
890      000040      L.GRP= 32.            ;GROUP
891      000041      L.TRCK= 33.            ;TRACK
892      000042      L.SCTR= 34.            ;SECTOR
893      000044      L.VSER= 36.            ;VOLUME SERIAL NUMBER
894      000050      L.DATA= 40.           ;EVENT DEPENDENT DATA
895
896      ;
897      ;
898      ;
899      ;
900      000037      ST.MSK= 37             ;STATUS / EVENT CODE MASK
901      000040      ST.SUB= 40             ;SUB-CODE MULTIPLIER
902      000000      ST.SUC= 0              ;SUCCESS
903      000001      ST.CMD= 1              ;INVALID COMMAND
904      000002      ST.ABO= 2              ;COMMAND ABORTED
905      000003      ST.OFL= 3              ;UNIT-OFFLINE
906      000004      ST.AVL= 4              ;UNIT-AVAILABLE
907      000005      ST.MFE= 5              ;MEDIA ERROR
908      000006      ST.WPR= 6              ;WRITE PROTECTED
909      000007      ST.CMP= 7              ;COMPARE ERROR
910      000010      ST.DAT= 10             ;DATA ERROR
911      000011      ST.HST= 11             ;HOST BUFFER ACCESS ERROR
912      000012      ST.CNT= 12             ;CONTROLLER ERROR
913      000013      ST.DRV= 13             ;DRIVE ERROR
914      000037      ST.DIA= 37             ;MESSAGE FROM AN INTERNAL DIAGNOSTIC
915
916      ;
917      ;
918      ;
919      ;
920      ;
921      ;
922      ;
923      ;
924      ;
925      ;
926      ;
927      ;
928      ;
929      ;
930      ;
931      ;
932      ;
933      ;
934      ;
935      ;
936      ;
937      ;
938      ;
939      ;
940      ;
941      ;
942      ;
943      ;
944      ;
945      ;
946      ;
947      ;
948      ;
949      ;
950      ;
951      ;
952      ;
953      ;
954      ;
955      ;
956      ;
957      ;
958      ;
959      ;
960      ;
961      ;
962      ;
963      ;
964      ;
965      ;
966      ;
967      ;
968      ;
969      ;
970      ;
971      ;
972      ;
973      ;
974      ;
975      ;
976      ;
977      ;
978      ;
979      ;
980      ;
981      ;
982      ;
983      ;
984      ;
985      ;
986      ;
987      ;
988      ;
989      ;
990      ;
991      ;
992      ;
993      ;
994      ;
995      ;
996      ;
997      ;
998      ;
999      ;
1000     ;
  
```

.SBTTL DUP DEFINITIONS

ALL REMOTE PROGRAMS (ROM BASED OR MEDIA BASED) START WITH A HEADER,
 WHICH LOOKS LIKE THIS:

```

15                                     0
+-----+-----+
|                                     | 00. N.BYTE
+-----+-----+
|          BYTE COUNT          | 02.
+-----+-----+
|                                     | 04. N.OVRL
+-----+-----+
|          OVERLAY BYTE COUNT    | 06.
+-----+-----+
|                                     | 08. N.NAME
+-----+-----+
|          ASCII NAME            | 10.
+-----+-----+
|                                     | 12.
+-----+-----+
|          TIME OUT          | 14. N.FLAG N.TOUT
+-----+-----+

```

LOCAL PROGRAMS HAVE AN EXTENDED HEADER, WHICH INCLUDES:

```

+-----+-----+
|                                     | 16. N.AVER (H.ADRS)
+-----+-----+
|          ASCII VERSION #        | 18.
+-----+-----+
|          SPACE                   | 20.
+-----+-----+
|          DIALOG          | STAND ALONE | 22. N.ALON N.DLOG
+-----+-----+
|          BINARY VERSION #       | 24. N.BVER
+-----+-----+
|                                     | 26. N.ADRS
+-----+-----+
|          PROGRAM CODE          |
+-----+-----+

```

PROGRAM CODE FOR LOCAL PROGRAMS STARTS AT N.ADRS, WHEREAS CODE FOR
 HOST SUPPLIED PROGRAMS STARTS AT H.ADRS.

;OTHER USEFUL CONSTANTS.

```

17'400      HIBYTE = 17'400      ;MASK FOR THE HIGH ORDER BYTE.
000017      LONIBL = 017        ;MASK FOR THE LOW ORDER NIBBLE.
000020      HINIBL = 20         ;MULTIPLIER TO SHIFT DATA TO HIGH NIBBLE.

```

```

984      .SBTTL  OFFSET AND FLAG DEFINITIONS
985
986      ; OFFSET DEFINITIONS.
987      ; -----
988
989      ; GENERIC COMMAND/RESPONSE PACKET.
990
991      000011      P.RSVD= 9.          ;RESERVED FIELD.
992
993      ; GET DUST STATUS RESPONSE.
994
995      000014      S.EXT= 12.          ;PROGRAM EXTENSION.
996      000017      S.FLAG= 15.         ;FLAGS.
997      000020      S.PRGS= 16.         ;PROGRESS INDICATOR.
998      000024      S.TOUT= 20.         ;TIME-OUT VALUE FOR LOCAL PROGRAMS.
999
1000     ; EXECUTE SUPPLIED PROGRAM COMMAND.
1001
1002     000034      P.OVRL= 28.          ;OVERLAY BUFFER DESCRIPTOR.
1003
1004     ; EXECUTE LOCAL PROGRAM COMMAND.
1005
1006     000014      L.NAME= 12.          ;PROGRAM NAME.
1007
1008     ; EXECUTE LOCAL PROGRAM RESPONSE.
1009
1010     000014      L.VERS= 12.          ;PROGRAM VERSION.
1011     000016      L.TOUT= 14.          ;TIME OUT IN SECS.
1012     000017      L.FLAG= 15.          ;FLAGS.
1013
1014     ; LOCAL PROGRAM'S HEADER TABLE.
1015
1016     000000      N.BYTE= 0             ;BYTE COUNT.
1017     000004      N.OVRL= 4             ;OVERLAY BYTE COUNT.
1018     000010      N.NAME= 8.            ;ASCII NAME.
1019     000016      N.FLAG= 14.           ;FLAGS.
1020     000017      N.TOUT= 15.           ;TIME OUT VALUE.
1021     000020      N.AVER= 16.           ;ASCII VERSION NUMBER.
1022     000026      N.ALON= 22.           ;STAND-ALONE FLAG.
1023     000027      N.DLOG= 23.           ;STANDARD DIALOG FLAG.
1024     000030      N.BVER= 24.           ;BINARY VERSION NUMBER.
1025     000032      N.ADRS= 26.           ;START OF PROGRAM PROPER.
1026
1027     ; HOST SUPPLIED PROGRAM'S HEADER TABLE.
1028
1029     000020      H.ADRS= N.AVER        ;START OF PROGRAM PROPER.
1030
1031     ; SEND DATA TEXT.
1032
1033     000000      S.TEXT= 0             ;ASCII TEXT.
1034
1035     ; RECEIVE DATA TEXT.
1036
1037     000000      R.MESG= 0             ;MESSAGE NUMBER.
1038     000001      R.TYPE= 1             ;TYPE.
1039     000002      R.TEXT= 2             ;ASCII TEXT.
1040

```

```

1041      ; FLAGS AND MODIFIERS.
1042      ; -----
1043      ;
1044      ; GET DUST STATUS RESPONSE.
1045      ;
1046      000000      SE.IDL = 0      ; SERVER IDLE.
1047      000001      SE.DIS = BIT0   ; PROGRAM EXECUTION DISABLES ALL OTHER SERVERS.
1048      000002      SE.LOC = BIT1   ; LOCAL LOAD MEDIA AVAILABLE.
1049      000004      SE.SUP = BIT2   ; WILL NOT ACCEPT EXECUTE SUPPLIED PROGRAM.
1050      000010      SE.ACT = BIT3   ; SERVER ACTIVE.
1051      ;
1052      ; EXECUTE LOCAL PROGRAM OPCODE MODIFIER.
1053      ;
1054      000001      LM.ALO = BIT0    ; ALLOW STAND-ALONE OPERATION OF LOCAL PROGRAM.
1055      ;
1056      ; REMOTE PROGRAM HEADER FLAGS.
1057      ;
1058      000001      PH.ALO = BIT0    ; STANDALONE.
1059      000002      PH.OVR = BIT1    ; NEEDS OVERLAYS FROM THE HOST.
1060      000004      PH.RW = BIT2     ; OVERLAY IS R/W.
1061      000010      PH.DLG = BIT3    ; PROGRAM USES STANDARD DUP DIALOG.
1062      ;
1063      ; CODES, TYPES, ETC.
1064      ; -----
1065      ;
1066      ; OP CODES
1067      ;
1068      000001      OP.GDS = 1      ; GET DUST STATUS
1069      000002      OP.ESP = 2      ; EXECUTE SUPPLIED PROGRAM
1070      000003      OP.ELP = 3      ; EXECUTE LOCAL PROGRAM
1071      000004      OP.SND = 4      ; SEND DATA
1072      000005      OP.RCD = 5      ; RECEIVE DATA
1073      000006      OP.ABP = 6      ; ABORT PROGRAM
1074      ;
1075      ; STATUS CODES.
1076      ;
1077      000000      ST.SUC = 0      ; SUCCESS.
1078      000001      ST.CMD = 1      ; ILLEGAL COMMAND.
1079      000002      ST.NRA = 2      ; NO REGION AVAILABLE.
1080      000003      ST.NRS = 3      ; NO REGION SUITABLE.
1081      000004      ST.NON = 4      ; PROGRAM NOT KNOWN.
1082      000005      ST.FAL = 5      ; LOAD FAILURE.
1083      000006      ST.ALO = 6      ; STANDALONE.
1084      ;
1085      ; RECEIVE DATA MESSAGE TYPES.
1086      ;
1087      000020      R1.QUE = 1*HINIBL ; QUESTION.
1088      000040      R1.DEF = 2*HINIBL ; DEFAULT QUESTION.
1089      000060      R1.INF = 3*HINIBL ; INFORMATION.
1090      000100      R1.TER = 4*HINIBL ; TERMINATION.
1091      000120      R1.FAT = 5*HINIBL ; FATAL ERROR.
1092      000140      R1.SPL = 6*HINIBL ; SPECIAL.
  
```

1093		:		
1095		:	COMMAND LENGTHS (IN BYTES.)	
1096		:		
1097	000014		.C.ABO= 12.	;ABORT PROGRAM.
1098	000014		.C.DUS= 12.	;GET DUST STATUS.
1099	000050		.C.SUP= 40.	;EXECUTE SUPPLIED PROGRAM.
1100	000022		.C.LOC= 18.	;EXECUTE LOCAL PROGRAM.
1101	000034		.C.SND= 28.	;SEND DATA.
1102	000034		.C.RCD= 28.	;RECEIVE DATA.
1103		:		
1104		:	RESPONSE LENGTHS (IN BYTES.)	
1105		:		
1106	000014		.R.ABO= 12.	;ABORT PROGRAM.
1107	000026		.R.DUS= 22.	;GET DUST STATUS.
1108	000014		.R.SUP= 12.	;EXECUTE SUPPLIED PROGRAM.
1109	000020		.R.LOC= 16.	;EXECUTE LOCAL PROGRAM.
1110	000020		.R.SND= 16.	;SEND DATA.
1111	000020		.R.RCD= 16.	;RECEIVE DATA.
1112				

1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148

```
.SBTTL  MODULE CODE
*****
:
:   INIT VALUES
:   INIT RQDX
:   XFER TO DISK?
:       F FOR J = 1,CYCLE LIMIT
:           MAINTENANCE WRITE
:           MAINTENANCE READ
:           CHECK DATA?
:               T CHECK
:       NEXT J
:       T FOR J = 1,CYCLE LIMIT
:           GET UNIT STATUS
:           IF DRIVE IS NOT AVAILABLE, WAIT UNTIL IT IS
:           DRIVE THERE?
:           F DROP
:               ALL DRIVES DROPPED?
:                   T DROP MODULE
:                   F
:           T ONLINE
:           ONLINE?
:               T PICK BLOCK  IF RANDOM, GET RAND @ MOD *
:                   ELSE INCREMENT
:                   IF LBN > LIMIT THEN LBN = 0
:               WRITE
:               READ
:               CHECK DATA ?
:                   T CHECK
:               AVAILABLE DRIVE(I)
:           F TRY TO BRING ONLINE AGAIN
:       NEXT J
*****
```


1150
1151
1152
1153
1154
1155
1156
1157
1158

1159 001002
1160 001002 042767 100000 177634
1161 001010 005067 177130
1162 001014 012767 000017 177504
1163 001022 016767 176766 177614
1164 001030 012767 177777 177620
1165 001036 005067 177430
1166 001042 104417 000000
1167 001046 016767 177002 177554
1168 001054 005067 177552
1169 001060 016767 176770 177546
1170 001066 005067 177544
1171 001072 016767 176710 177152
1172 001100 062767 000002 177144
1173 001106 005067 177404
1174 001112 005067 177402
1175
1176 001116 052767 000001 177474

```

*****
: START SEQUENCE
:
: IF THE CODE IS RESTARTED, CLEAR THE OLD ADDRESSES SO THE
: THE RQDX WILL GET REINITED.
:
*****
: START:
: BIC      %BIT15,DVICE      ; MUST BE CLEAR FOR LOOPING
: CLR      CDECT             ; CLEAR DATA CHECK ERROR COUNT
: MOV      %PRNUM,PRNMSG     ; INITIALIZE PRINT WORD
: MOV      DVID1,DVICE       ; DVICE HAS DESIRED BITS SET
: MOV      %1,TABLEW        ; FORCE TABLE SET UP FIRST TIME THROUGH
: CLR      CMDREF            ; COMMAND REF # = 0
: RAND%,BEGIN
: MOV      RANUM,XSECT       ;
: CLR      XSECT             ; FOR RESTARTING (INITIAL SECTOR ADDR)
: MOV      RANUM,RSECT       ;
: CLR      RSECT             ; FOR RESTARTING (INITIAL SECTOR ADDR)
: MOV      ADDR,RQDXSA        ; STORE IN SA REG
: ADD      %2,RQDXSA         ; RQDXSA HAS PROPER ADDRESS
: CLR      OLPA              ; OLD PHYSICAL ADDRESS CLEARED
: CLR      OLDEA             ; OLD EXTENDED ADDRESS CLEARED
:          ; THIS WILL FORCE A RQDX REINIT TO TAKE PLACE
: BIS      %BIT0,FLAGS       ; FLAGS = START

```

```

1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190 001124
1191 001124 012767 000260 177350
1192 001132 104415 000000 000502
1193 001140 032767 001000 176710
1194 001146 001404
1195 001150 104416 000000 000504
1196 001156 000416
1197 001160 016767 177320 177322 1$:
1198 001166 006267 177314
1199 001172 006267 177310
1200 001176 006267 177304
1201 001202 006267 177300
1202 001206 016767 177274 177276
1203 001214 026767 177270 177274 2$:
1204 001222 001004
1205 001224 026767 177262 177266
1206 001232 001412
1207
1208 001234 016767 177250 177254
1209 001242 016767 177244 177250
1210 001250 004767 000550
1211 001254 005067 177366
1212 001260
1213 001260 032767 000001 177332
1214 001266 001411
1215 001270 042767 000001 177322
1216 001276 016767 177214 177216
1217 001304 016767 177210 177212
1218 001312
1219 001312 032767 000010 176476
1220 001320 001006
1221 001322 026767 177200 176504
1222 001330 001002
1223 001332 004767 000404
1224 001336
1225
1226 001336 012777 004770 176444
1227 001344 012767 012462 177130
1228 001352 104415 000000 000502
1229 001360 032767 001000 176470
1230 001366 001420
1231 001370 104416 000000 000504
1232 001376 016767 177102 176522
1233 001404 016767 177072 176516
1234 001412 016767 177072 177344

*****
: RESTART SEQUENCE
:
: CHECK THE ADDRESS OF THE RINGS TO SEE IF THEY WERE RELOCATED
: IF THEY WERE, REINIT THE RQDX.
:
: GET THE NEW ADDRESSES. IF THE DISKLESS OPERATION IS DESIRED
: THEN DO THE MAINTENANCE WRITE AND READ. ELSE DO THE WRITE
: AND READ WITH A DRIVE.
:
*****
RESTR1:
MOV 0RSPONC,VA ; DID THE RINGS RELOCATE?
GETPA$,BEGIN,VA ; GET PHYSICAL ADDRESS FROM 16 BIT VA
BIT 0ADDR22,CONFIG ; CHECK FOR 22BIT ADDRESSING
BEQ 1$ ; IF 18BIT ADDRESSING THEN BRANCH
MAP22$,BEGIN,PA ; GET 22-BIT ADDR FROM 18 BIT ADDR
BR 2$
MOV PA,LO22 ; MOV 18BIT ADDRESS INTO 22BIT SPACE
ASR EA ; ADJUST EXTENDED BITS SO THEY LINE
ASR EA ; UP AT BITS 0 & 1
ASR EA
MOV EA,HI22
CMP LO22,OLDEA ; IF OLD PHYS ADDR NOT = NEW ONE?
BNE RESTR2 ; OR
CMP HI22,OLDEA ; IF OLD EXTN ADDR NOT = NEW ONE?
BEQ RESTR1 ; THEN

RESTR2:
MOV LO22,OLDEA ; SET THE OLD RING ADDR
MOV HI22,OLDEA ; AND THE OLD EXTENDED ADDR
CALL INIPRT ; AND INIT THE RQDX
CLR TRY ; CLEAR RETRY COUNT
ENDIF
RESTR1:
BIT 0BIT0,FLAGS ; IF FLAGS = START (ONLY DO DUP WHEN LOCATED HERE)
BEQ 1$ ; THEN
BIC 0BIT0,FLAGS ; FLAGS = START CLEARED
MOV OLDEA,SAVEA ; SAVE EA
MOV OLDEA,SAVEA ; SAVE EA
ENDIF
1$:
BIT 0SR,SUM,SRI ; IF WE WANT THE REPORT?
BNE 2$ ; THEN
CMP PRNMSG,PASCNT ; IF DO WE PRINT?
BNE 2$ ; THEN (ONLY IF PASS COUNT IS RIGHT)
CALL PRSTS ; CALL PRINT EOP STATUS
2$:
ENDIF
ENDIF
MOV 0NTRPT,0VECTOR ; GET VECTOR ADDRESS SET POINTER
MOV 0RBUF,VA ; MAP THIS TO 22 BITS
GETPA$,BEGIN,VA ; GET PHYSICAL ADDRESS FROM 16 BIT VA
BIT 0ADDR22,CONFIG ; CHECK FOR 22BIT ADDRESSING
BEQ 22$ ; BRANCH IF 18 BIT ADDRESSING
MAP22$,BEGIN,PA ; GET 22 BIT ADDR FROM 18 BIT ADDR
MOV PA,RBUFPA ; FILL IN BUFFER ADDRESS FOR CKDATA
MOV EA,RBUFPA
MOV LO22,R22PA ; FILL IN BUFFER ADDRESS FOR READ CMD

```



```

1235 001420 016767 177066 177340      MOV      HI22,R22EA      ;
1236 001426 000424                    BR          33$          ;
1237 001430 016767 177050 176470 22$: MOV      PA,RBUFPA      ; FILL IN BUFFER ADDRESS FOR CKDATA
1238 001436 016767 177044 176464      MOV      EA,RBUFPA      ;
1239 001444 016767 177034 177312      MOV      PA,R22PA      ; FILL IN 18BIT ADDRESS FOR READ CMD
1240 001452 006267 177030      ASR      EA      ; ADJUST EXTENDED BITS
1241 001456 006267 177024      ASR      EA
1242 001462 006267 177020      ASR      EA
1243 001466 006267 177014      ASR      EA
1244 001472 016767 177010 177266      MOV      EA,R22EA      ; FILL IN BUFFER ADDRESS FOR READ CMD
1245 001500 004767 001326 33$: CALL     SETUP      ; FIND DRIVES/SET UP TABLE
1246 001504 005767 177134      TST      DVICE      ; IF NO DRIVES TO DO
1247 001510 001002      BNE      3$      ; : THEN
1248 001512 104410 000000      ENDS$,BEGIN
1249 001516                    3$:      ;
1250 001516 026767 176774 176776      CMP      OLDDPA,SAVPA      ; IF OLD_PHYS_ADDR = SAVE_PHYS_ADDR
1251 001524 001026      BNE      4$      ; AND
1252 001526 026767 176766 176770      CMP      OLDEA,SAVEA      ; IF OLD_EXTN_ADDR = SAVE_EXTN_ADDR
1253 001534 001022      BNE      4$      ; : THEN
1254 001536 012704 000656      MOV      @TABLEW,R4      ; : R4 = ADRS OF TABLE OF UNITNO AND PORTID
1255 001542 036467 000002 177074      BIT      2(R4),DVICE      ; : IF DRIVE NOT DROPPED
1256 001550 001414      BEQ      4$      ; : AND
1257 001552 032767 000001 177074      BIT      @1,ISRX      ; : IF RD = AVAILABLE
1258 001560 001010      BNE      4$      ; : AND
1259 001562 032767 000020 176226      BIT      @BIT4,SR1      ; : IF DUP OPERATIONS = SELECTED
1260 001570 001404      BEQ      4$      ; : : THEN
1261 001572 004767 004074      CALL     DODUP      ; : : CALL DO DUP
1262 001576 004767 000222      CALL     INIPRT      ; : : CALL INIT THE RQDX (MUST DO AFTER DUP)
1263                    ; : ENDIF
1264 001602                    4$:      ; ENDIF
1265 001602                    LOOP1: ; BGND0
1266 001602 012702 000001      MOV      @1,R2      ; : SCALING FACTOR FOR RD
1267 001606 026767 177032 177040      CMP      DVICE,ISRX      ; : IF RD
1268 001614 001405      BEQ      1$      ; : : THEN
1269 001616 016702 176176      MOV      SR2,R2      ; : : LOAD SCALING FACTOR
1270 001622 001002      BNE      1$      ; : :
1271 001624 012702 000120      MOV      @80.,R2      ; : : DEFAULT RD SCALING FACTOR TO 80.
1272 001630                    1$:      ; : ENDIF
1273                    ; : BGND0
1274 001630 012704 000656      MOV      @TABLEW,R4      ; : : R4 = ADRS OF TABLE OF UNITNO AND PORTID
1275 001634                    2$:      ; : : BGND0
1276 001634 036467 000002 177002      BIT      2(R4),DVICE      ; : : : IF DRIVE NOT DROPPED
1277 001642 001424      BEQ      10$      ; : : : THEN
1278 001644 036467 000002 177002      BIT      2(R4),ISRX      ; : : : : IF DRIVE IS AN RX (ACCESS ONLY ONCE PER LOOP)
1279 001652 001403      BEQ      4$      ; : : : : AND
1280 001654 022702 000001      CMP      @1,R2      ; : : : : IF LAST TIME THROUGH
1281 001660 001015      BNE      10$      ; : : : : : THEN
1282 001662 010246      MOV      R2,(SP)      ; : : : : : R2 IS KILLED IN CYCLED AND DROPT
1283 001664 016746 176134      MOV      SR4,(SP)      ; : : : : : CONSECUTIVE ACCESSSES TO DO TO A DEVICE
1284 001670                    5$:      ; : : : : : BGND0
1285 001670 004767 001504      CALL     CYCLED      ; : : : : : : DO A DISK CYCLE OPERATION
1286 001674 103402      BCS      9$      ; : : : : : : EXIT ON FAILURE
1287 001676 005316      DEC      (SP)      ; : : : : : : DECREMENT ACCESS CTR
1288 001700 003373      BGT      5$      ; : : : : : : DOUNTIL (IF 0 ONLY DO ONCE)
1289 001702 005324      DEC      (SP)      ; : : : : : : DON'T TOUCH C BIT
1290 001704 012602      MOV      (SP)+,R2      ; : : : : : :
1291                    ; : : : : : : IF CYCLE NOT SUCCESSFUL

```

```

1292 001706 103002          BCC      10$
1293 001710 004767 002310    JSR      PC,DROP1
1294
1295
1296 001714          10$:
1297 001714 062704 000004    ADD      #4,R4
1298 001720 026467 000002 176716  CMP      2(R4),DEVICE
1299 001726 101742          BLOS     2$
1300 001730 005302          DEC      R2
1301 001732 001336          BNE      1$
1302 001734 104413 000000'    ENDIT$,BEGIN
1303 001740 000720          BR       LOOP1
1304
1305
; : : : : : : THEN
; : : : : : : CALL DROP DRIVE
; : : : : : : ENDF
; : : : : : : ENDF
; : : : : : : ENDF
; : : : : : : POINT TO NEXT ENTRY OF THE TABLE
; : : : : : : DUNTIL TABLE_ENTRY = DEVICE
; : : : : : : DECREMENT SCALE_FACTOR
; : : : : : : DUNTIL SCALE_FACTOR = 0
; : : : : : : SIGNAL END OF ITERATION.
; : : : : : : MONITOR SHALL TEST END OF PASS
; : : : : : : ENDDO

```

```

1307 ;*****
1308 ;
1309 ; PRINT EOP STATUS
1310 ;
1311 ;*****
1312 ;
1313 ;
1314 001742 PRSTS: ADD @PRNUM,PRNMSG ; PRINT WORD IS INCREMENT
1315 001742 062767 000017 176556 ;*****
1316 ;
;*****
; CONVERT SOFCNT TO ASCII AND
; STORE AT ADR2
001750 104421 000000' 000042' BTOD$,BEGIN,SOFCNT,ADR2
001756 000537'
;*****
;
1317 001760 105067 176560 CLR8 ADR2+5
1318 ;*****
; CONVERT HRDCNT TO ASCII AND
; STORE AT ADR3
001764 104421 000000' 000044' BTOD$,BEGIN,HRDCNT,ADR3
001772 000546'
;*****
;
1319 001774 105067 176553 CLR8 ADR3+5
1320 ;*****
; CONVERT CDERCT TO ASCII AND
; STORE AT ADR1
002000 104421 000000' 000144' BTOD$,BEGIN,CDERCT,ADR1
002006 000530'
;*****
;
1321 002010 105067 176521 CLR8 ADR1+5
1322 002014 104403 000000' 007764' MSGN$,BEGIN,ERRPAS ; ASCII MESSAGE CALL WITH COMMON HEADER
1323 002022 000207 RETURN
1324 ;
1325 ;
1326 ;
1327 ;
1328 ;

```

H/5

1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350

1351 002024
1352 002024 005077 175756
1353 002030 012767 000260 176444
1354 002036 104415 000000 000504
1355 002044 032767 001000 176004
1356 002052 001404
1357 002054 104416 000000 000504
1358 002062 000416
1359 002064 016767 176414 176416 11:
1360 002072 006267 176410
1361 002076 006267 176404
1362 002102 006267 176400
1363 002106 006267 176374
1364 002112 016767 176370 176372
1365 002120 005004 22:
1366 002122 012702 000001
1367 002126 012701 027340
1368 002132 017700 176114 1:
1369 002136 032700 100000
1370 002142 001003
1371 002144 005301
1372 002146 001371
1373 002150 000404
1374 002152 012703 004000 2:
1375 002156 000167 000474
1376
1377
1378 002162 032700 004000 4:
1379 002166 001003
1380 002170 000167 000456
1381
1382 002174 016705 175610 5:
1383 002200 006205
1384 002202 006205
1385 002204 052705 100200
1386 002210 010500

```
*****
INITIALIZE THE RQDX
DO THE 4 STEPS FOR INITIALIZING THE RQDX.
STEP 1 - CHECK FOR ERROR, STEP 1
SEND VECTOR/4, INTERRUPT ENABLE, RING LEN'S = 0
STEP 2 - CHECK VECTOR ECHO, INTERRUPT ECHO,
ERROR, STEP 2
SEND PHYSICAL ADDRESS & BURGE = 0
STEP 3 - CHECK RING LEN = 0, ERROR, STEP 3
SEND EXTENDED ADDRESS BITS
STEP 4 - CHECK STEP 4
SEND LFAIL = 0, GO AND BURST
*****

INIPRT:
CLR @ADDR
MOV @RSPONC,VA
GETPA$,BEGIN,VA
BIT @ADDR22,CONFIG
BEQ 11$
MAP2$,BEGIN,PA
BR 22$
MOV PA,LO22
ASR EA
ASR EA
ASR EA
ASR EA
MOV EA,H122
CLR R4
MOV @1,R2
MOV @TIMER,R1
MOV @RQDXSA,RO
BIT @SA,ERR$,RO
BNE 2$
DEC R1
BNE 1$
BR 4$
MOV @SA,S1,R5
JMP ERROR1
;4$: CMP @SA,S1+SA,QB+SA,DIA$,RO
; BEQ 5$
; BIT @SA,S1,RO
; BNE 5$
; JMP ERROR3
5$: MOV VECTOR,R5
ASR R5
ASR R5
BIS @SA,INT+BIT15$,R5
MOV R5,RO

; INITIALIZE
; WRITE TO RQDXIP TO INIT RQDX
; VA = RSPONC
; GET PHYSICAL ADDRESS FROM 16 BIT VA
; CHECK FOR 22BIT ADDRESSING
; IF 18BIT ADDRESSING THEN BRANCH
; GET 22 BIT ADDR FROM 18-BIT ADDR
; MOV 18BIT ADDRESS INTO 22BIT SPACE
; ADJUSTED EXTENDED BITS

; R4 IS USED IF AN ERROR IS DETECTED
; R2 = STEP INDICATOR REG FOR MSG'S
; SET TIME OUT LIMIT
; RO HAS RQDXSA DATA
; CHECK FOR ERROR
; IF FOUND, GET OUT OF LOOP
; TIME OUT?
; IF NOT, LOOP
; IF DONE, CONTINUE
; R3 = STEP 1 BIT
; IF HERE, ERROR
; DID DATA COMPARE PROPERLY?
; IF SO, CONTINUE
; STEP 1 BIT SET?
; IF NE, YES
; REPORT ERROR
***** STEP 1
; VECTOR GIVEN
; SET TO APPROPRIATE VALUE
; VECTOR/4
; ENABLE INTERRUPTS, SET RING LEN'S ARE 0
; STORE R5 IN RO FOR SUBROUTINE
```

1387	002212	012703	004000		MOV	%SA,S1,R3		; R3 HAS STEP BIT FOR SUBROUTINE
1388	002216	004767	000332		JSR	PC,SNDSIP		; SEND STEP DATA
1389	002222	042705	100000		BIC	%BIT15,R5		; CLEAR MSB FOR COMPARE DATA
1390	002226	042700	000200		BIC	%BIT07,R0		; IF BIT07 ONLY SET
1391	002232	001404			BEQ	6\$		; : THEN
1392	002234	052700	010200		BIS	%<SA,S2+BIT07>,R0		; : SET R0 TO REPORT THE ERROR
1393	002240	000167	000406		JMP	ERROR3		; : EXIT BLOCK TO REPORT ERROR
1394								; ENDIF
1395								; ***** STEP 2
1396	002244	016700	176240	6\$:	MOV	L022,R0		; RO GETS PHYSICAL ADDRESS
1397	002250	004767	000300		JSR	PC,SNDSIP		; SEND STEP DATA
1398	002254	042705	177400		BIC	%177400,R5		; HIGH BYTE CLEARED
1399	002260	020500			CMP	R5,R0		; CHECK ECHO DATA
1400	002262	001402			BEQ	7\$		; IF OK, SKIP
1401	002264	000167	000362		JMP	ERROR3		; IF NOT, REPORT ERROR
1402								; ***** STEP 3
1403	002270	016700	176216	7\$:	MOV	H122,R0		; ADJUST THE EXTENDED ADDRESS BITS
1404	002274	004767	000254		JSR	PC,SNDSIP		; SEND STEP DATA
1405	002300	012700	000254		MOV	%RSPONC+4,R0		; RO -> RING ENVELOP
1406								; ***** STEP 4
1407	002304			8\$:				; BGNDD
1408	002304	005720			TST	(R0)+		; : IF RING ENTRY NOT = 0?
1409	002306	001154			BNE	ERROR5		; : : THEN -> ERROR
1410	002310	022700	000266		CMP	%RSPONC+6,R0		; DUNTIL R0 POINT PAST THE RINGS?
1411	002314	001373			BNE	8\$		
1412	002316	012777	000001	175726	MOV	%SA,G0,%RQDXSA		; SEND DATA TO RQDX/DONE WITH THE INIT
1413	002324	012767	000376	176150	MOV	%CMPACK,VA		; GET COMMAND PACKET PA AND EA
1414	002332	104415	000000	000502	GETPA\$,BEGIN,VA			; GET PHYSICAL ADDRESS FROM 16-BIT VA
1415	002340	032767	001000	175510	BIT	%ADDR22,CONFIG		; CHECK FOR 22BIT ADDRESSING
1416	002346	001404			BEQ	8E\$		; IF 18BIT ADDRESSING THEN BRANCH
1417	002350	104416	000000	000504	MAP22\$,BEGIN,PA			; GET 22-BIT ADDR FROM 18-BIT ADDR
1418	002356	000416			BR	99\$		
1419	002360	016767	176120	176122	88\$:	MOV	PA,L022	; MOV 18BIT ADDRESS INTO 22BIT SPACE
1420	002366	006267	176114		ASR	EA		; ADJUST EXTENDED BITS
1421	002372	006267	176110		ASR	EA		
1422	002376	006267	176104		ASR	EA		
1423	002402	006267	176100		ASR	EA		
1424	002406	016767	176074	176076	MOV	EA,H122		
1425	002414	016767	176070	175642	99\$:	MOV	L022,COMMND	; STORE ADDRESS IN THE RING
1426	002422	016767	176064	175636	MOV	H122,COMMND+2		; MOVE ADJUSTED EA INTO RING
1427	002430	012767	000276	176044	MOV	%RSPACK,VA		; GET RESPONSE PACKET PA AND EA
1428	002436	104415	000000	000502	GETPA\$,BEGIN,VA			; GET PHYSICAL ADDRESS FROM 16-BIT VA
1429	002444	032767	001000	175404	BIT	%ADDR22,CONFIG		; CHECK FOR 22BIT ADDRESSING
1430	002452	001404			BEQ	66\$		; IF 18BIT ADDRESSING THEN BRANCH
1431	002454	104416	000000	000504	MAP22\$,BEGIN,PA			; GET 22-BIT ADDR FROM 18-BIT ADDR
1432	002462	000416			BR	77\$		
1433	002464	016767	176014	176016	66\$:	MOV	PA,L022	; MOV 18BIT ADDRESS INTO 22BIT SPACE
1434	002472	006267	176010		ASR	EA		
1435	002476	006267	176004		ASR	EA		
1436	002502	006267	176000		ASR	EA		
1437	002506	006267	175774		ASR	EA		
1438	002512	016767	175770	175772	MOV	EA,H122		
1439	002520	016767	175764	175552	77\$:	MOV	L022,RSPONC	; STORE ADDRESS IN THE RING
1440	002526	016767	175760	175526	MOV	H122,RSPONC+2		; MOVE ADJUSTED EA INTO RING
1441	002534	012777	004770	175246	MOV	%INTRPT,%VECTOR		; STORE INTERRUPT ADDRESS IN VECTOR
1442	002542	005067	176100		CLR	TRY		; CLEAR TRY SO DRIVE WILL GO BACK ONLINE IF NEEDED
1443	002546	004767	002042		CALL	SCC		; SET CONTROLLER CHARACTERISTICS EVERY TIME WE RE IN

IT RQDX1

04
RQAD DEC N11 SYSTEM EXERCISER M MACRO M1200 09-APR 84 16:31 PAGE 30-2
MODULE CODE

SEQ 35

1444 000552 000207

RTS PC

EXIT INITIALIZE


```

1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459 002554 016701 175230
1460 002560 012721 002600'
1461 002564 116711 175222
1462 002570 010077 175456
1463
1464 002574 104400 000060'
1465
1466 002600
1467 002600 000004 000000' 002606'

1468 002606
1469 002606 017700 175440
1470 002612 032700 100000
1471 002616 001017
1472 002620 005202
1473 002622 006303
1474 002624 030300
1475 002626 001002
1476 002630 000167 000020
1477 002634
1478 002634 040300
1479 002636 000207

; *****
;
; SEND STEP DATA
;
; INPUT: R0 HAS DATA TO BE SENT TO RQDX FOR STEP
;        R3 HAS PREVIOUS STEP FLAG SET
;
; OUTPUT: R0 HAS DATA SENT FROM RQDX TO HOST FOR ECHO AND NEXT STEP
;         R3 HAS CURRENT STEP FLAG SET
; *****
;
SNDSTP: MOV VECTOR,R1 ;
        MOV $INTA,(R1),+ ; SET UP INTERRUPT HANDLER ADDRESS
        MOVB BR1,(R1) ; SET PRIORITY LEVEL
        MOV R0,@RQDXSA ; SEND STEP1 WRITE FORMATTED DATA
        EXIT$,BEGIN ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

INTA:
; -----
PIRQ$,BEGIN,3$ ; QUEUE UP TO CONTINUE AT 3$ AND RTI
; -----
3$:
MOV @RQDXSA,R0 ; GET STEP N FORMATTED DATA
BIT $SA.ERR,R0 ; IF NO HARDWARE ERROR
BNE ERROR1 ; : THEN
INC R2 ; SET STEP REGISTER
ASL R3 ; R3 HAS STEP BIT PROPERLY SET
BIT R3,R0 ; IF STEP N NOT SET?
BNE 4$ ; : THEN
JMP ERROR2 ; : NOT CORRECT STEP, ERROR
4$:
BIC R3,R0 ; CLEAR THE STEP BIT, FOR COMPARE
RTS PC ; RETURN

```

1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509

1510
1511

1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526

002640				
002640	104403	000000'	010044'	
002646	104410	000000'		
002652	005204			
002654	005204			
002656	005204			
002660	010267	175630		
002664	104420	000000'	000514'	
002672	000537'			
002674	017767	175352	175612	
002702	104420	000000'	000514'	
002710	000530'			
002712	105067	175620		
002716	005304			
002720	001003			
002722	104403	000000'	007656'	
002730	005304			1\$:
002732	001003			
002734	104403	000000'	007702'	
002742	005304			2\$:
002744	001003			
002746	104403	000000'	007710'	
002754				3\$:

```

*****
:
: ERROR 1
: PRINT AN ERROR REPORTED BY THE RQDX DIAGNOSTICS
:
: ERROR 2
: PRINT VALUE OF THE RQDXSA WHEN THE STEP BIT WAS NOT SET
:
: ERROR 3
: PRINT VALUE OF THE RQDXSA WHEN THE ECHO WAS NOT SET CORRECTLY
:
: INPUT:  R0 -> RQDXSA
:         R2 = STEP COUNT
:
: OUTPUT: THE RETRY COUNT IS INCREMENTED
:         IF THE RETRY COUNT > RETRY LIMIT, END MODULE
:
: ERROR 5
: RING WASN'T ALL ZERO -> FATAL ERROR -> DROP MODULE
:
*****
ERROR5: MSGN$,BEGIN,ZERO      ;ASCII MESSAGE CALL WITH COMMON HEADER
:      END$,BEGIN          ;
:
ERROR3: INC      R4          ;R4 = 3 FOR ERROR3
ERROR2: INC      R4          ;R4 = 2 FOR ERROR2
ERROR1: INC      R4          ;R4 = 1 FOR ERROR1
MOV     R2,NUM             ;STORE STEP REG IN A NUMBER FOR CONVERT
:*****
:      ;CONVERT NUM TO ASCII AND
:      ;STORE AT ADR2
:
:*****
MOV     @RQDXSA,NUM        ;STORE VALUE IN A NUMBER
:*****
:      ;CONVERT NUM TO ASCII AND
:      ;STORE AT ADR1
:
:*****
CLRB    ADR2-1            ;
:
DEC      R4                ;ERROR 1?
BNE     1$                ;IF NOT, CHECK IF IT IS THE NEXT ERROR
MSGN$,BEGIN,INITE1        ;ASCII MESSAGE CALL WITH COMMON HEADER
:
DEC      R4                ;ERROR 2?
BNE     2$                ;IF NOT, CHECK IF IT IS THE NEXT ERROR
MSGN$,BEGIN,INITE2        ;ASCII MESSAGE CALL WITH COMMON HEADER
:
DEC      R4                ;ERROR 3?
BNE     3$                ;IF NOT, CHECK IF IT IS THE NEXT ERROR
MSGN$,BEGIN,INITE3        ;ASCII MESSAGE CALL WITH COMMON HEADER
:
*****

```


Address	Hex	Hex	Hex	Hex	Assembly	Comment
	002754	104420	000000'	000006'	OTOA\$,BEGIN,ADDR,ADR3	;CONVERT ADDR TO ASCII AND ;STORE AT ADR3
	002762	000546'				
1527	002764	104405	000000'	000000	HRDR\$,BEGIN,NULL	
1528	002772	104403	000000'	007664'	MSGN\$,BEGIN,INITER	;ASCII MESSAGE CALL WITH COMMON HEADER
1529	003000	005267	175642		INC TRY	;INCREMENT RETRY COUNT
1530	003004	022767	000004	175634	CMP @RLIM,TRY	;IS THE RETRY COUNT EXCEEDED?
1531	003012	001402			BEQ 6\$	;IF SO, END IT
1532	003014	000167	175762		JMP START	;IF NOT, TRY AGAIN
1533	003020					
	003020	104403	000000'	010040'	MSGN\$,BEGIN,ABORT	;ASCII MESSAGE CALL WITH COMMON HEADER
1534	003026	104410	000000'		END\$,BEGIN	
1535						

```

1537 ; *****
1538 ;
1539 ;   SET UP
1540 ;
1541 ;   GO FIND OUT WHAT DRIVES ARE OUT THERE.
1542 ;   A TABLE IS FILLED WITH UNIT NUMBERS(MAX IS 16)
1543 ;
1544 ;   THIS SHOULD ONLY BE DONE AT THE VERY BEGINNING OF RUNNING
1545 ;   THIS DECX MODULE; THEN NOT RUN AGAIN.
1546 ;
1547 ;   INPUT:  DVICE HAS APPROPRIATE BITS SET.  THE # OF BITS =
1548 ;           # OF DRIVES WANTED TO TEST.
1549 ;           POSITION OF BITS = WHICH DRIVE IN THE SYSTEM IS DESIRED.
1550 ; *****
1551 ;
1552 ;
1553 ; SETUP:
1554 003032 005767 175620      TST    TABLEW      ; ONLY DO SETUP THE FIRST TIME THROUGH
1555 003036 002133          BGE    100$          ; (USE DRIVES FOUND AT BEGINNING)
1556          CALL    SCC          ; CALL SET_CONTROLLER_CHARACTERISTICS
1557 003040 012703 000001      MOV    #1,R3        ; INITIAL PORTID VALUE
1558 003044 012704 000656      MOV    #TABLEW,R4    ; R4 -> TABLEW
1559 003050 005014          CLR    (R4)          ;
1560 003052          1$:        ;
1561 003052 012746 000024      MOV    #24, -(SP)    ; WORK = RETRY LIMIT
1562 003056 004767 001474      3$:    JSR    PC,GTSTAT ; GET STATUS, GET NEXT UNIT NUMBER
1563 003062 103011          RCC    6$          ; OK, CONTINUE
1564 003064 005316          DEC    -(SP)          ;
1565 003066 001373          BNE    3$          ;
1566 003070 005726          TST    (SP)+          ;
1567 003072 056467 000002 175544      BIS    2(R4),DVICE ; ASSURE REPORT WILL BE PRINTED
1568 003100 004767 001130      JSR    PC,DROP2    ; DROP THE DRIVE
1569 003104 000476          BR     177$          ; TRY NEXT UNIT
1570 003106 005726          6$:    TST    (SP)+    ; PO: RETRY COUNT OFF STACK
1571 003110 016714 175166      7$:    MOV    P,UNIT+RSPACK,(R4) ; UNIT NUMBER FROM RESPONSE PACKET
1572 003114 001015          BNE    10$          ; UNIT NUMBER RETURNED, CONTINUE
1573 003116 022703 000001      CMP    #1,R3      ; FIRST TIME THROUGH MAY BE UNIT 0
1574 003122 001412          BEQ    10$          ;
1575          ;
1576          ; NO UNITS ARE LEFT, BUT BITS ARE STILL SET IN DVICE
1577          ;
1578 003124 030367 175514      8$:    BIT    R3,DVICE ; IS THIS ONE SET?
1579 003130 001402          BEQ    9$          ; NO, CONTINUE
1580 003132 004767 001106      JSR    PC,DROP3    ; DROP UNIT AND WARN
1581 003136 062704 000004      9$:    ADD    #4,R4 ; MAINTAIN TABLE POINTER FOR DROP3
1582 003142 006303          ASL    R4          ; TRY NEXT BIT
1583 003144 001367          BNE    8$          ; IF ANY ARE LEFT, TRY AGAIN
1584 003146 000467          BR     100$         ; OTHERWISE, EXIT
1585          ;
1586 003150 126727 175154 000006 10$:  CMPB   P,UNIT+6+RSPACK,#6 ; MEDIA ID IS 6 FOR RD, 7 FOR RX
1587 003156 001416          BEQ    17$          ; NO - IT IS AN RX
1588 003160 050367 175470      BIS    R3,ISRX    ; YES - SET RD BIT
1589 003164 122737 000025 000041      CMPB   #25,#41 ; IS LOAD MEDIUM DU?
1590 003172 001010          BNE    17$          ; NO - SKIP CHECKING
1591 003174 121437 000040      CMPB   (R4),#40    ;
1592 003200 001005          BNE    17$          ;
1593 003202 040367 174614      BIC    R3,SR3    ; CLEAR WRITE BIT FOR LOAD DEVICE

```

```

1594 003206 104403 000000' 010074' MSGN$,BEGIN,BUNIT ;ASCII MESSAGE CALL WITH COMMON HEADER
1595 003214 036467 000002' 175422' 173: BIT 2(R4),DVICE ; ONLY PRINT MESSAGE IF IT IS VALID
1596 003222 001427 BEQ 177$ ; UNIT NOT TO BE TESTED
1597 003224 011467 175264 MOV (R4),NUM ; WE HAVE A GOOD UNIT PRINT MESSAGE
1598 ;*****
;CONVERT NUM TO ASCII AND
;STORE AT ADR1
003230 104420 000000' 000514' OTOA$,BEGIN,NUM,ADR1
003236 000530'
;*****
1599 003240 105067 175272 CURR ADR2 1 ;
1600 003244 032767 000002' 174544 173: BIT 0BIT1,SR1 ; NO READ ACCESS - READ ONLY MESSAGE
1601 003252 001410 BEQ 175$ ;
1602 003254 036467 000002' 174540 BIT 2(R4),SR3 ; UNIT HAS WRITE TURNED OFF
1603 003262 001404 BEQ 175$ ;
1604 003264 104403 000000' 010050 MSGN$,BEGIN,RDWR1 ;ASCII MESSAGE CALL WITH COMMON HEADER
1605 003272 000403 BR 177$ ;
1606 003274 104403 000000' 010062' MSGN$,BEGIN,RDWR1 ;ASCII MESSAGE CALL WITH COMMON HEADER
1607 003302 062704 000004 177$: ADD 04,R4 ; BUMP R4 TO NEXT TABLE POINTER
1608 003306 016414 177774 MOV 4(R4),(R4) ; SET NEXT UNIT NUMBER > CURRENT ONE
1609 003312 005214 INC (R4) ;
1610 003314 006503 ASL R3 ; NEXT BIT IN DVICE
1611 003316 020367 175322 CMP R3,DVICE ; IF 0, WE'RE DONE
1612 003322 101653 BLOS 1$ ; (BIT 15 IN DVICE IS EXPLICITLY 0)
1613 003324 005014 CLR (R4) ; TAKE EXTRA NUMBER OUT OF TABLE
1614 003326 000207 100$: RTS PC ; ...AND BACK.

```

```

1616
1617
1618
1619
1620
1621
1622 003330 022700 000003
1623 003334 001403
1624 003336 022700 000013
1625 003340 001012
1626
1627 003344
1628
1629
1630
1631 003344 032767 000740 174736
1632 003350 001004
1633
1634
1635 003354 032767 177000 174726
1636 003360 001000
1637
1638 003364 000241
1639 003366 000403
1640
1641 003370 000261
1642
1643 003374 004767 002040
1644 003376 000207
1645

```

```

*****
;
; ISTOP1
; TEST TO SEE WHAT KIND OF AN OFFLINE CONDITION HAS OCCURED.
;
*****
ISTOFL: CMP     051,OFL,R0      ; WAS THE DRIVE FOUND OFF LINE?
        BEQ     10$           ; CHECK WHAT KIND OF OFFLINE
        CMP     051,DRV,R0     ; WAS IT A DRIVE ERROR?  > SDI?
        BNE     13$           ; IF IT WAS NOT, ERROR (DROP DRIVE)

10$:    ; IF ANY OF FOLLOWING BITS SET
        ; NO VOLUME MOUNTED, UNIT DISABLED BY FIELD SERVICE
        ; OR DUPLICATE UNIT NUMBER OR UNIT INOPERATIVE
        BIT     0<SC,NVL+SC,DIS+SC,DUP+SC,IOP>,P,STS+RSPACK
        BNE     12$           ; IF SO, EXIT
                                ; ANY OTHER DATA?

12$:    ;
        CLC
        BR      14$           ; CLEAR CARRY NO ERROR

13$:    SEC
                                ; SET CARRY, DRIVE WAS FOUND TO BE OFFLINE
                                ; OR ANOTHER ERROR
                                ; REPORT ERROR
                                ; RETURN

14$:    JBR     PC,ERRORH
        RTS     PC

```

```

1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664 003400
1665
1666
1667 003400 005767 175242
1668 003404 100420
1669 003406 004767 001240
1670 003412 103571
1671 003414 016767 174724 175200
1672 003422 016767 174714 175210
1673 003430 001006
1674 003432 005767 175204
1675 003436 001760
1676 003440 012767 100000 175200
1677 003446
1678 003446 016700 174470
1679 003452 005001
1680 003454
1681 003454 005201
1682 003456 162700 000400
1683 003462 100374
1684 003464 160167 175150
1685
1686 003470 012702 000630
1687 003474 036467 000002 175152
1688 003502 001001
1689 003504 022222
1690 003506 000240
1691 003510 004767 000264
1692 003514 004767 001004
1693 003520 103732
1694 003522 022700 000004
1695 003526 001727
1696 003530 000240
1697 003532 032767 000002 174256
1698 003540 001506
1699 003542 036467 000002 174252
1700 003550 001502
1701 003552 104414 000004
1702 003556 016767 174352 175200
1703 003564 016767 174316 175202

*****
CYCLE DISK
*****
DO GET STATUS COMMANDS TO ASSURE THAT THE DRIVE
IS AVAILABLE (FOR DUAL PORTING)
CHECK DRIVE TO BE ONLINE
IF TRUE
    PICK THE BLOCK
    WRITE
    READ
    DATA CHECK
    MAKE THE DRIVE AVAILABLE
ELSE DROP DRIVE
*****

CYCLED:
    CHECK IF WE DO ONLINE FOR THE FIRST TIME.
    IF TRY HAS NOT SET MSB
    THEN
        DO AND ONLINE COMMAND
        IF CARRY WAS SET, TRY AGAIN
        IS THE UNIT SIZE HI ADDRESS
        GET UNIT SIZE/IS IT > 0?
        IF NOT ZERO, CONTINUE WITH ITERATION
        IS UNSZ ALSO 0?
        IF 0, TRY TO BRING ONLINE AGAIN
        TRY = MSB SET (INITIAL ONLINE IS DONE)
    ENDIF
    WBUF$Z IN RO AS A LIMIT
    R1 = 0 OF BLOCKS
    BGND0
    INCREMENT THE # OF BLOCKS
    DECREMENT A BLOCK
    DOUNTIL # OF BLOCKS < 0
    ADJUST THE UNIT SIZE
    ( NOW PICK WHICH BLOCK TO WRITE TO )
    RX LOW SECTOR NUMBER
    BUMP TO DSCL
    ELSE SELECT A BLOCK TO TEST
    DID WE NOT GET THE DRIVE ONLINE?
    IF NOT, GO BACK TO TOP AND TRY AGAIN
    IS IT AVAILABLE?
    IF SO, GO BACK TO TOP AND TRY AGAIN
    IF WRITE ACCESS
    AND
    IF UNIT WRITE IS DESIRED
    THEN
        GET WRITE BUFFER INFORMATION
        COPY 18 BIT ADDRESS

```

```

1704 003572 032767 001000 174256      BIT      #ADDR22,CONF IG      ; : CHECK FOR 22BIT ADDRESSING
1705 003600 001404                      BEQ      111$              ; : IF 18BIT ADDRESSING THEN BRANCH
1706 003602 104416 000000' 000772'      MAP22$, BEGIN,W18PA      ; : GET 22-BIT ADDR FROM 18-BIT ADDR
1707 003610 000416                      BR      222$              ; :
1708 003612 016767 175154 175156 111$:  MOV      W18PA,W22PA      ; : MOV 18BIT ADDRESS INTO 22BIT SPACE
1709 003620 006267 175150                      ASR      W18EA      ; : ADJUST EXTENDED BITS
1710 003624 006267 175144                      ASR      W18EA
1711 003630 006267 175140                      ASR      W18EA
1712 003634 006267 175134                      ASR      W18EA
1713 003640 016767 175130 175132      MOV      W18EA,W22EA
1714
1715 003646 004767 000544                222$: JSR      PC,WRITE      ; : WRITE THE DATA
1716 003652 103003                      BCC      19$              ; : IF NOT OK
1717
1718 003654 004767 001556                19$: JSR      PC,ERRORH      ; : : HARD ERROR (BCS 22$ )
1719 003660 000445                      BR      22$              ; : : EXIT
1720 003662                      19$:      ; : : EXIT
1721 003662 004767 000600                19$: CALL     READ      ; : READ A BLOCK
1722 003666 103003                      BCC      195$             ; : IF NOT OK
1723
1724 003670 004767 001542                195$: JSR      PC,ERRORH      ; : : HARD ERROR
1725 003674 000437                      BR      22$              ; : : EXIT
1726 003676                      195$:      ; : : EXIT
1727 003676 032767 004000 174112      BIT      #SR,CMP,SR1      ; : IF DO A DATA COMPARE
1728 003704 001053                      BNE      22$              ; : : THEN
1729 003706 104412 000000' 000126'      CDATA$,BEGIN,RBUFA      ; REQUEST FOR MONITOR TO CHECK DATA
1730 003714 003716'                      ; : IF ERROR, CONTINUE
1731 003720 000240                      ; :
1732 003722 000240                      ; :
1733
1734 003724 000423                      BR      22$              ; :
1735 003726 011467 174562                197$: MOV      (R4),NUM      ; :
1736
;*****
;CONVERT NUM TO ASCII AND
;STORE AT ADR1
003732 104420 000000' 000514'      OTOA$,BEGIN,NUM,ADR1
003740 000530'
;*****
1737 003742 105067 174570                20$: CLRB      ADRP-1      ; :
1738 003746 104403 000000' 010116'      MSGN$,BEGIN,CHKRPT      ; ASCII MESSAGE CALL WITH COMMON HEADER
1739 003754 000407                      BR      20$              ; : : ENDIF
1740 003756 000240                      NOP      ; : : ELSE
1741 003760 004767 000502                20$: CALL     READ      ; : READ ONLY - NO DATA CHECK
1742 003764 103003                      BCC      22$              ; : IF ERROR
1743
1744 003766 004767 001444                22$: JSR      PC,ERRORH      ; : : HARD ERROR
1745 003772 000400                      BR      22$              ; : : AND EXIT; BCS 22$
1746 003774                      22$:      ; : : ENDIF
1747
1748 003774 000341                      CCR      PC      ; : EVERYTHING WAS OK
1749 003776 000207                24$: RTS      PC      ; : RETURN
1750
1751

```



```

1753 ;*****
1754 ;
1755 ;   PICK A BLOCK TO WRITE TO.
1756 ;
1757 ;   PICK THE NEXT SEQUENTIAL BLOCK (DEFAULT) OR TAKE ONE AT RANDOM.
1758 ;
1759 ;   OUTPUT: FILL SECH & SECL (CURRENT SECTOR ADDR)
1760 ;
1761 ;*****
1762 004000 PICKBK:
1763 004000 032267 002000 174010      BIT      #SR,SEQ,SR1      ;CHECK SR1 FOR RANDOM ACCESS MODE
1764 004006 001020                      BNF      RANACC          ;BR IF RANDOM ACCESS
1765 ;
1766 ;   GENERATE DISK ADDRESS BY SEQUENTIAL ADDRESSING
1767 ;
1768 004010 SEQACC:
1769 004010 005212      INC      (R2)          ;INCREMENT THE SECTOR ADDRESS
1770 004012 001408      BEQ      16$          ;BR IF ZERO
1771 004014 021267 174620      CMP      (R2),UNS2L      ;OVER LIMIT?
1772 004020 103412      BLO      18$          ;BR IF LOWER
1773 004022 000402      BR      17$          ;SKIP THE INCREMENT
1774 004024 16$:
1775 004024 005262 000002      INC      2(R2)          ;INCREMENT SECTOR HIGH ADDRESS
1776 004030 17$:
1777 004030 026267 000002 174604      CMP      2(R2),UNS2H      ;OVER LIMIT?
1778 004036 103403      BLO      18$          ;BR IF LOWER
1779 004040 005012      CLR      (R2)          ;RESET THE STARTING SECTOR ADDRESS
1780 004042 005062 000002      CLR      2(R2)
1781 004046 18$:
1782 004046 000207      RTS      PC
1783 ;

```

```

1785 ;*****
1786 ;
1787 ; GENERATE DISK ADDRESS BY RANDOM ADDRESSING
1788 ;
1789 ;*****
1790 ;
1791 004050 RANACC:
1792 004050 104417 000000' RAND$,BEGIN
1793 004054 016746 173774 MOV RANNUM, -(SP) ;GENERATE THE SECTOR ADDRESS
1794 004060 104417 000000' RAND$,BEGIN
1795 004064 016746 173764 MOV RANNUM, (SP) ;GENERATE THE SECTOR ADDRESS
1796 ;
1797 ; ADJUST HI ADDRESS FIRST
1798 ;
1799 004070 000241 CLC ; CLEAR CARRY FOR ROTATE
1800 004072 042716 100000 BIC #100000,(SP) ; CLEAR UPPER BIT MAKES SURE VALUE'S +
1801 004076 012662 000000' MOV (SP)+,2(R2) ; STORE IN SECTOR HI ADDRESS
1802 004102 005767 174534 TST UNSZH ; IS THE MAX SIZE 0?
1803 004106 001430 BEQ 3$ ; IF 0, GET LOW SECTOR ADDRESS
1804 ;
1805 ; *** UNSZH > 0 IF CODE FALLS THROUGH HERE
1806 ;
1807 004110 016700 174526 MOV UNSZH,R0 ; R0 = MAX VALUE
1808 004114 005100 COM R0 ; R0 COMPLEMENT, NOW FIND MS ZERO
1809 004116 012701 100000 MOV #100000,R1 ; R1 IS INDEX INTO MAX VALUE
1810 004122 030100 1$: BIT R1,R0 ; HAVE 0 YET?
1811 004124 001403 BEQ 2$ ; IF 1ST 0 REACHED, CLEAR REST OF THE BITS
1812 004126 000241 CLC ; CLEAR CARRY FOR ROR
1813 004130 006001 ROR R1 ; POINT TO NEXT BIT
1814 004132 000773 BR 1$ ; BRANCH TO TEST AGAIN
1815 004134 040100 2$: BIC R1,R0 ; CLEAR REST OF THE BITS
1816 004136 000241 CLC ; CLEAR CARRY FOR ROR
1817 004140 006001 ROR R1 ; IF R1 ROTATES INTO CARRY, R1 = 0
1818 004142 001374 BNE 2$ ; IF R1 NOT 0, MORE BITS TO CLEAR
1819 004144 040062 000000' BIC R0,2(R2) ; CLEAR UPPER BITS OF HIGH SECTOR VALUE
1820 004150 026267 000002 174464 CMP 2(R2),UNSZH ; IF THE HIGH SECTOR VALUE > MAX VALUE?
1821 004156 002420 BLT 7$ ; IF <, EXIT
1822 004160 001405 BEQ 4$ ; IF =, TEST LOW ORDER VALUE
1823 004162 006262 000002 ASR 2(R2) ; SECH = SECH/2 - CAN'T BE > MAX NOW
1824 004166 000414 BR 7$ ;EXIT
1825 ;
1826 ; GET LOW SECTOR ADDRESS
1827 ;
1828 004170 005062 000002 3$: CLR 2(R2) ;CLEAR HI SECTOR SIZE
1829 004174 005767 173440 4$: TST UNSZL ;IS THE HIGHEST POSSIBLE = 0?
1830 004200 001406 BEQ 5$ ;IF TRUE, DON'T DO LOOP
1831 004202 166716 174464 5$: SUB UNSZL,(SP) ; ELSE, SECU = SECU - UNSZL (ADJUSTED)
1832 004206 103375 BCC 7$ ; IF UNSZL > SECU, LOOP
1833 004210 066716 174424 ADD UNSZL,(SP) ; ELSE SUBTRACTED ONCE TOO OFTEN
1834 004214 000401 BR 7$ ; AND EXIT
1835 004216 005016 6$: CLR (SP) ;HIGHEST POSSIBLE = 0
1836 004220 012612 7$: MOV (SP)+,(R2) ;SAVE TO SECTOR ADDRESS
1837 004222 000207 RTS PC ; RETURN
1838

```



```

1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853 004224
1854 004224 012767 000001 174262
1855 004232 000407
1856 004234
1857 004234 012767 000002 174252
1858 004242 000403
1859 004244
1860 004244 012767 000003 174242
1861 004252 010046
1862 004254 016700 174234
1863 004260 0364 000002 174256
1864 004266 001401
1865 004270 046467 000001 174346
1866 004276 046467 000001 174350
1867 004304 011467 174204
1868

;*****
; DROP A DRIVE
;
; A DRIVE WOULDN'T RESPOND, DROP IT. SET THIS UP IN DEVICE.
;
; INPUT: UNITNO = UNIT NUMBER OF DRIVE TO DROP
;         PORTID = BIT SET TO DROP DRIVE
;
; OUTPUT: DEVICE HAS A BIT CLEARED. THE BIT POSITION = THE DRIVE
;*****
DROP1:
MOV     01,NUM
BR      DROP4
DROP2:
MOV     02,NUM
BR      DROP4
DROP3:
MOV     03,NUM
MOV     R0, -(SP)
MOV     NUM,R0
BIT     2(R4),DEVICE ;DON'T DROP DEVICE TWICE
BEQ     10$
BIC     2(R4),DEVICE ;DROP THE DRIVE
BIC     2(R4),ISRX ;MAINTAIN ACCURACY
MOV     (R4),NUM ;MOVE IT INTO A LOCATION
;*****
; CONVERT NUM TO ASCII AND
; STORE AT ADDR?
BTOD$,BEGIN,NUM,ADR?
;*****
CLRB    ADR?+5
MOV     2(R4),NUM
BIC     0BIT15,NUM ;IF BOOT DEVICE
;*****
; CONVERT NUM TO ASCII AND
; STORE AT ADR1
OTOA$,BEGIN,NUM,ADR1
;*****
CLRB    ADR1+5
DEC     R0 ;DROPPED FOR WHICH ERROR?
BNE     1$ ;IF NOT FOR ERRORS, CONTINUE
MSGN$,BEGIN,DRP1 ;ASCII MESSAGE CALL WITH COMMON HEADER
BR      10$
1$: DEC     R0 ;WAS UNIT NOT FOUND?(NON-EXISTENT UNIT)
BNE     2$ ;IF NOT, CONTINUE
MSGN$,BEGIN,DRP2 ;ASCII MESSAGE CALL WITH COMMON HEADER
BR      10$
2$: MSGN$,BEGIN,DRP3 ;ASCII MESSAGE CALL WITH COMMON HEADER
; ACTUAL UNITS FOUND
10$: MOV     -(SP),R0
RTS     PC

```

1887

1888

1889

1890

1891

1892

1893

1894

1895

1896

1897

1898

1899

1900

1901

1902

1903

1904

1905

1906

1907

1908

1909

1910

1911

1912

1913

1914

1915

1916

1917

1918

1919

1920

1921

1922

1923

1924

1925

1926

1927

1928

1929

1930

1931

1932

1933

004416 004767 000714
 004422 012767 000042 173756
 004430 016700 173506
 004434 016767 174335 173754
 004442 016767 174332 173750
 004450 000423

```
*****
;
; WRITE
;
; SET UP OP CODE, MODIFIERS, BUFFER SIZE (BYTE COUNT),
; BUFFER DESCRIPTOR (PHYSICAL AND EXTENDED ADDRESS)
; LET READ SET SIMILAR DATA IN THE PACKET:
; DISK ADDRESS AND CYLINDER ID (LOGICAL BLOCK NUMBER),
; THEN SEND THE PACKET.
;
*****
```

```
WRITE: CALL CLR ;CLEAR PACKETS
;MOV #OP.WR,P.OPCD+CMACK ;SET THE OPCODE
WRITEA: MOV WBUF SZ,RO ;STORE THE BUFFER SIZE IN WORDS
;MOV W22PA,P.ADEA+CMACK ;SET THE BUFFER DESCRIPTOR(PA)
;MOV W22EA,P.ADEA+CMACK ;SET THE BUFFER DESCRIPTOR(EA)
;BR READA ;
```

```
*****
;
; READ, ACCESS
;
; SET UP OP CODE, MODIFIERS, BUFFER SIZE (BYTE COUNT),
; BUFFER DESCRIPTOR (PHYSICAL AND EXTENDED ADDRESS),
; DISK ADDRESS AND CYLINDER ID (LOGICAL BLOCK NUMBER),
; THEN SEND THE PACKET.
;
*****
```

```
ACCESS: JSR PC,CLR ;CLEAR PACKETS
;MOV #OP.ACC,P.OPCD+CMACK ;SET THE OPCODE
;BR ACREAD ;COMMON ACCESS/READ CODE

READ: JSR PC,CLR ;CLEAR PACKETS
;MOV #OP.RD,P.OPCD+CMACK ;SET THE OPCODE
ACREAD: MOV RBUF SZ,RO ;STORE THE BUFFER SIZE IN WORDS
;MOV R22EA,P.ADEA+CMACK ;SET THE BUFFER DESCRIPTOR
;MOV R22PA,P.ADEA+CMACK ;
READA: MOV #32,RSPLEN ;SET RESPONSE PACKET LENGTH
;MOV #32,CMPLN ;SET COMMAND PACKET LENGTH
;ASL RO ;MAKE IT NUMBER OF BYTES
;MOV RO,P.BCNT+CMACK ;SET READ BUFFER SIZE
;MOV CR,P.LBN+CMACK ;SET LOGICAL BLOCK NUMBER
;MOV CR,P.LBN+2+CMACK ;
;BR SEND ;SEND THE PACKET
```

1935

1936

1937

1938

1939

1940

1941

1942

1943

1944

1945 004556 004767 000554

1946 004562 012767 000003 173516

1947 004570 012767 000001 173513

1948 004576 012767 000060 173456

1949 004604 012767 000014 173560

1950 004612 000434

1951

1952

1953

1954

1955

1956

1957

1958

1959

1960

1961

1962

1963

1964 004614

1965 004614 004767 000516

1966 004620 012767 000000 173544

1967 004626 012767 000034 173436

1968 004634 012767 000004 173544

1969 004642 012767 000120 173544

1970 004650 000415

1971

1972

1973

1974

1975

1976

1977

1978

1979

1980

1981

1982 004652 004767 000460

1983 004656 012767 000054 173406

1984 004664 012767 000044 173500

1985 004672 012767 000011 173506

1986 004700 011467 173532

1987

```
*****
:
:   GET UNIT STATUS
:
:   SET OPCODE AND MODIFIER (FOR THEN NEXT UNIT
:   THEN SEND THE PACKET
:
:*****
```

```
GTSTA: CALL CLR ;CLEAR PACKETS
: MOV #0P,GUS,P.OPCD+CMACK ;SET THE OPCODE
: MOV #MD,NXU,P.MOD+CMACK ;CLEAR MODIFIERS
: MOV #48,,RSLEN ;SET RESPONSE PACKET LENGTH
GTSTAA: MOV #12,,CMLEN ;SET COMMAND PACKET LENGTH
: BR SEND ;SEND THE PACKET
```

```
*****
:
:   SET CONTROLLER CHARACTERISTICS
:
:   SET OP CODE AND CONTROLLER FLAG (ENABLE ERROR LOGS MSGS)
:   CLEAR MSCP VERSION, HOST TIMEOUT, USE FRACTION,
:   AND ALL OF QUAD WORD TIME AND DATE.
:   THEN SEND PACKET
:
:*****
```

```
SCC:
: CALL CLR ;GO CLEAR THE COMMAND PACKET
: MOV #32,,CMLEN ;SET UP COMMAND PACKET LENGTH
: MOV #28,,RSLEN ;SET UP RESPONSE PACKET LENGTH
: MOV #0P,SCC,P.OPCD+CMACK ;SET THE OPCODE
: MOV #120,P.CNTR+CMACK ;SET CONTROLLER FLAGS = ERROR LOGS
: BR SEND ;SEND THE PACKET
```

```
*****
:
:   ONLINE
:
:   SET OPCODE, MODIFIERS, UNIT ID, HOST ID
:   SHADOW UNIT, ERROR FLAGS
:   THEN SEND PACKET
:
:*****
```

```
ONLINE: CALL CLR ;CLEAR PACKETS
: MOV #44,,RSLEN ;SET RESPONSE PACKET LENGTH
: MOV #36,,CMLEN ;SET COMMAND PACKET LENGTH
: MOV #0P,ONL,P.OPCD+CMACK ;SET THE OPCODE
: MOV (R4),P.SHUN+CMACK ;SHADOW UNIT UNITNO.
: ;SEND THE PACKET
```

```

1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009 004704
      004704 010046
2010 004706 005067 173714
2011 004712 005267 173554
2012 004716 001775
2013 004720 016767 173546 173450
2014 004726 011467 173450
2015 004732 042767 040000 173326
2016 004740 052767 100000 173320
2017 004746 052767 140000 173306
2018 004754 004767 002470
2019 004760 005777 173022
2020 004764
2021 004764 104400 000000
2022 004770
2023      004770 000004 000000 004776
2024 004776
2025 004776 005067 173254
2026 005002 004767 002460
2027 005006
2028 005006 016700 173262
2029 005012 042700 177417
2030 005016 022700 000020
2031 005022 001006
2032 005024 005267 173576
2033 005030 052767 140000 173224
2034 005036 000752
2035 005040
2036 005040 004767 000004
2037 005044
      005044 016600
2038 005046 000207
2039

```

```

*****
:
: SEND - SEND A PACKET
: INTERP - WAIT FOR AN INTERRUPT
:
: SET UP THE COMMAND REFERENCE NUMBER AND UNITNO IN THE PACKET
: SET OWN, CLEAR FLAG IN THE COMMAND RING (FOR RQDX)
: SET OWN & FLAG IN MESSAGE RING (FOR INTERRUPTS BY RQDX)
: AFTER INTERRUPT, MAKE SURE THE PACKET WAS PROCESSED (NO HARD
: OR SOFT ERRORS) THEN RETURN TO CYCLED.
:
: INPUT:  CMPACK IS FILLED EXCEPT FOR CMDREF & UNITNO
:         INTERRUPT VECTOR AND BR LEVEL ARE ESTABLISHED
:
: OUTPUT: MSPACK IS FILLED
:         CLEAR CARRY IF COMMAND PACKET WAS OK
:         ELSE GO DO A HARD/SOFT ERROR
:
*****
:
: SEND:
:      MOV      R0, -(R5)
:      CLR      SETFLG
:      INC      CMDREF
:      BEQ      1$
:      MOV      CMDREF,P.CRF+CMPPACK
:      MOV      (R4),P.UNIT+CMPPACK
:      BIC      @RG.FLG,COMMND*2
:      BIS      @RG.OWN,COMMND*2
:      BIS      @RG.FLG!RG.OWN,RSPONC*2
:      JSR      PC,OUTPKT
:      IST      @ADDR
:
: WAIT:
:      EXIT$,BEGIN
:
: NTRUPT:
:      PIRQ$,BEGIN,1$
:
: 1$:
:      CLR      RINTR
:      JSR      PC,INPKT
:
: 2$:
:      MOV      RSPVIR,R0
:      BIC      @177417,R0
:      CMP      @20,R0
:      BNE      10$
:      INC      SETFLG
:      BIS      @RG.FLG!RG.OWN,RSPONC*2
:      BR       WAIT
:
: 10$:
:      CALL     ERRCHK
:
: 99$:
:      MOV      (R5)+,R0
:      RTS      PC
:
:      ; CLEAR SOFT ERROR FLAG
:      ; NEW COMMAND REFERENCE NUMBER
:      ; COMMAND REF # CANNOT = 0
:      ; SET COMMAND REF NUMBER
:      ; SET UNIT NUMBER
:      ; CLEAR FLAG
:      ; SET OWN FOR COMMAND RING
:      ; SET OWN AND FLAG FOR MESSAGE RING
:      ; PRINT PACKET (DIAGNOSTICS)
:      ; FORCE POLLING TO PACKET
:      ; BGND0
:      ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
:
:      ; -----
:      ; QUEUE UP TO CONTINUE AT 1$ AND RTI
:
:      ; -----
:
:      ; :
:      ; : CLEAR INTERRUPT FLAG
:      ; : PRINT PACKET (DIAGNOSTICS)
:      ; :
:      ; : GET RESPONSE
:      ; : CLEAR ANY CREDITS
:      ; : IF RESPONSE = ERROR LOG
:      ; :
:      ; : INCREMENT SOFT ERROR FLAG
:      ; : SET OWN AND FLAG FOR MESSAGE RING
:      ; : DO UNTIL
:      ; :
:      ; : CALL ERROR CHECK
:      ; :
:      ; : RETURN

```

```
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054 005050
2055 005050 005067 173032
2056 005054 016700 173230
2057 005060 042700 177740
2058 005064 001513
2059
2060
2061 005066 122700 000013
2062 005072 001004
2063 005074 012767 000006 173004
NON EXISTENT
2064 005102 000474
2065 005104 3$:
2066 005104 122700 000012
2067 005110 001004
2068 005112 012767 000003 172766
2069 005120 000465
2070 005122 4$:
2071 005122 122700 000011
2072 005126 001004
2073 005130 012767 000032 172750
2074 005136 000456
2075 005140 5$:
2076 005140 122700 000010
2077 005144 001004
2078 005146 012767 000001 172732
2079 005154 000447
2080 005156 6$:
2081 005156 122700 000006
2082 005162 001004
2083 005164 012767 000006 172714
NON EXISTENT
2084 005172 000440
2085 005174 8$:
2086 005174 122700 000004
2087 005200 001010
2088 005202 022767 000003 173176
2089 005210 001437
2090 005212 012767 000006 172666
R NON EXISTENT
2091 005220 000425
2092 005222 9$:
2093 005222 122700 000003
2094 005226 001025
2095 005230 022767 000011 173150
2096 005236 001410
2097 005240 022767 000003 173140

*****
:
: ERROR CHECK
:
: CHECK FOR HARD OR SOFT ERRORS THEN RETURN
:
: INPUT: R0 = STATUS OF CONTROLLER
:
: OUTPUT: CLEAR CARRY IF COMMAND PACKET WAS OK
: ELSE GO DO A HARD/SOFT ERROR
:
*****
ERRCHK:
CLR ERRTP ; CLEAR ERROR_TYPE
MOV P,STS+RSPACK,R0 ; GET SUCCESS
BIC #177740,R0 ; CLEAR UPPER 11 BITS OF SUB-STATUS
BEQ 15$ ; IF SUCCESS NOT = 0
; THEN
; SELECT STATUS
; CASE DRIVE ERROR:
; ERROR_TYPE = DRIVE NOT READY, OFF LINE OR
3$: BR 12$
; CASE CONTROLLER ERROR
; ERROR_TYPE = CONTROLLER NOT READY
4$: CMPB #ST,CNT,R0
BNE 4$
MOV #ERR.3,ERRTP
BR 12$
; CASE HOST BUFFER ACCESS ERROR:
; ERROR_TYPE = NPR ERROR
5$: CMPB #ST,HST,R0
BNE 5$
MOV #ERR.32,ERRTP
BR 12$
; CASE DATA ERROR:
; ERROR_TYPE = DATA ERROR
6$: CMPB #ST,DA1,R0
BNE 6$
MOV #ERR.1,ERRTP
BR 12$
; CASE WRITE PROTECTED:
; ERROR_TYPE = DRIVE NOT READY, OFF LINE OR
8$: BR 12$
; CASE STILL AVAILABLE:
; IF COMMAND NOT GET UNIT STATUS
; ERROR TYPE = DRIVE NOT READY, OFF LINE O
9$: CMPB #ST,AVL,R0
BNE 9$
CMP #OP,GUS,P,OPCD+CMPPACK
BEQ 14$
MOV #ERR.6,ERRTP
BR 12$
; CASE UNIT OFFLINE:
; IF AN ONLINE COMMAND
; OR
; IF GET UNIT STATUS COMMAND
```

```

2098 005246 001404      BEQ      10$
2099 005250 022767 000042 173130    CMP      #OP.WR,P.OPCD+CMPPACK
2100 005256 001002      BNE      11$
2101 005260 000261      SEC
BLE DRIVE
2102 005262 000424      BR       99$
2103 005264      11$:
2104 005264 012767 000006 172614    MOV      #ERR.6,ERRTYP
INE OR NON EXISTEN
T
2105 005272 000400      BR       12$
2106
2107 005274      12$:
2108 005274 004767 000136    CALL      ERRORH
2109 005300 000415      BR       99$
2110
2111 005302      13$:
2112 005302 004767 000170    CALL      ERRORS
2113
2114 005306 000412      BR       99$
2115
2116 005310      14$:
2117 005310 000241      CLC
2118 005312 000410      BR       99$
2119 005314      15$:
2120 005314 005767 173306    TST      SFIFLG
2121 005320 001405      BEQ      99$
2122 005322 005067 173300    CLR      SFIFLG
2123 005326 004767 000144    CALL      ERRORS
2124 005332 000241      CLC
2125
2126
2127 005334 000207      99$: RETURN
2128
2129

```

```

: : : OR
: : : IF A WRITE COMMAND
: : : : THEN
: : : : SET CARRY TO DROP DRIVE/AWAIT AVAILA
: : : : EXIT
: : : : ELSE
: : : : ERROR_TYPE = DRIVE NOT READ, OFF L
: : :
: : : ENDIF
: : : ENDSELECT
: : : *** HARD ERROR EXIT ->
: : : AND HARD ERROR
: : :
: : : *** SOFT ERROR EXIT -> ERROR TYPE = 0
: : : ERROR WITH ERRTP = 0 & IS A SOFT ERROR
: : : ST.CMP,ST.MFE,.ST.ABO,ST.CMD
: : :
: : : *** SUCCESSFUL EXIT
: : : CLEAR CARRY 'CAUSE PACKET IS OK
: : :
: : : ELSE
: : : IF SOFT_ERROR_FLAG NOT = 0
: : : : THEN
: : : : : CLEAR SOFT_ERROR_FLAG
: : : : : CALL ERROR_SOFT
: : : : : CLEAR CARRY - DON'T DROP DRIVE
: : : : ENDIF
: : : ENDIF
:

```



```

2131
2132
2133
2134
2135
2136
2137
2138
2139 005336
2140 005336 010246
2141 005340 017767 172706 173146
2142
2143 005346 001417
2144
;*****
;
; CLEAR
; IF SA = ERROR REPORT & REINIT
; CLEAR PACKETS
;
;*****
CLR:
MOV R2, -(SP)
MOV @RQDXSA, NUM ; IS RQDXSA NOT ZERO/STORE IN NUM IF TRUE
; IF RQDXSA IS NOT ZERO
DEQ 5$ ; THEN
;*****
; CONVERT NUM TO ASCII AND
; STORE AT ADR1
OTOA$, BEGIN, NUM, ADR1
;*****
2145 005360 104403 000000' 010004' MSGN$, BEGIN, SANOTO ; ASCII MESSAGE CALL WITH COMMON HEADER
2146 005366 010346 MOV R3, -(SP) ; : SAVE R3
2147 005370 01446 MOV R4, -(SP) ; : SAVE R4
2148 005372 004767 174426 JSR PC, INIPRT ; : RE INIT RQDXSA
2149 005376 012604 MOV (SP)+, R4 ; : RESTORE R3
2150 005400 012603 MOV (SP)+, R3 ; : RESTORE R4
2151 ; JSR PC, SCC ; : SET CONTROLLER CHARS AGAIN
2152 005402 005267 172436 INC HRDCNT ; : INCREMENT HARD ERROR COUNT
2153 ; ; THIS WILL CAUSE ANOTHER CALL TO CLRPAK
2154 005406 5$; ; ENDF
2155 005406 004767 000004 CALL CLRPAK ; CALL CLEAR_PACKETS
2156 005412 012602 MOV (SP)+, R2 ; POP
2157 005414 000207 RTS PC ; RETURN
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171 005416 CLRPAK:
2172 005416 012702 000100 MOV #4, R2 ; R2 = # OF WORDS TO CLEAR
2173 005422 012705 000272' MOV @RSPLN, R5 ; R5 = RSPLN, 1ST WORD TO CLEAR
2174 005426 ; BGNDO
2175 005426 005025 6$; CLR (R5)+ ; : CLEAR WORD
2176 005430 005302 DEC R2 ; : DECREMENT COUNTER
2177 005432 001375 BNE 6$ ; DOUNTIL = ZERO
2178 005434 000207 RETURN

```

```

2180
2181
2182
2183
2184
2185
2186 005436
2187 005436 032767 000004 172352
2188 005444 001403
2189 005446 005267 172352
2190 005452 000407
2191 005454
2192 005454 004767 000056
2193
2194 005466 004767 000070
2195 005472
2196 005472 000261
2197 005474 000207
2198
2199
2200
2201
2202
2203
2204
2205 005476
2206 005476 032767 000004 172352
2207 005504 001403
2208 005506 005267 172330
2209 005512 000407
2210 005514
2211 005514 004767 000016
2212
2213 005526 004767 000030
2214 005532
2215 005532 000261
2216 005534 000207
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226 005536
2227 005536 016767 172510 172334
2228 005544 016767 172540 172332
2229 005552 016767 172474 172322
2230 005560 000207
2231

```

```

*****
| HARD ERROR CARRY WILL BE SET
| *****
ERRORH:
| BIT 0SR,REP,SR1 | IF REPORT ERROR = NO
| BEQ 7$ | THEN
| INC HRDCNT | INCREMENT THE HARD ERROR
| BR 8$ |
7$:
| JSR PC,SETTAB | SET UP TABLE
| *****
| HRDEF$,BEGIN,NOU
| *****
8$:
| JSR PC,PRINTE |
| SEC |
| RTS PC | RETURN TO CYCLED
| *****
| SOFT ERROR CARRY WILL BE SET
| *****
ERRORS:
| BIT 0SR,REP,SR1 | IF REPORT ERROR = NO
| BEQ 9$ | THEN
| INC SOFCNT | INCREMENT THE SOFT ERROR
| BR 10$ |
9$:
| CALL SETTAB | SET UP TABLE
| *****
| SOFER$,BEGIN,NOU
| *****
10$:
| CALL PRINTE |
| SEC |
| RTS PC | SET CARRY
| RETURN TO CYCLED
| *****
| SETTAB
| *****
| SET UP A TABLE OF VALUES FOR A SOFT OR HARD ERROR
| *****
SETTAB:
| MOV RQDXSA,CSRA | SET UP CONTROL STATUS REG REPORT
| MOV PNTGREGPC,ASTAT | SET UP STATUS
| MOV BRQDXSA,ACSR | REPORT WHAT IS STATUS REG
| RTS PC | RETURN

```


2233
2234
2235
2236
2237
2238
2239 005562
2240

```
*****
: PRINT EXTENDED ERROR (STATUS, OPCODE, UNIT NUMBER, BYTE COUNT,
: LBN AND ADDRESS)
: *****
PRINT:
```

005562 104420 000000 000310
005570 000530

```
*****
: CONVERT P,STS+RSPACK TO ASCII AND
: STORE AT ADR1
OTOA$,BEGIN,P,STS+RSPACK,ADR1
```

2241

005572 104420 000000 000306
005600 000537

```
*****
: CONVERT P,OPCD+RSPACK TO ASCII AND
: STORE AT ADR2
OTOA$,BEGIN,P,OPCD+RSPACK,ADR2
```

2242

005602 104420 000000 000302
005610 000546

```
*****
: CONVERT P,UNIT+RSPACK TO ASCII AND
: STORE AT ADR3
OTOA$,BEGIN,P,UNIT+RSPACK,ADR3
```

2243

005612 104420 000000 000312
005620 000555

```
*****
: CONVERT P,BCNT+RSPACK TO ASCII AND
: STORE AT ADR4
OTOA$,BEGIN,P,BCNT+RSPACK,ADR4
```

2244

005622 104420 000000 000434
005630 000564

```
*****
: CONVERT P,LBN+2+CMPPACK TO ASCII AND
: STORE AT ADR5
OTOA$,BEGIN,P,LBN+2+CMPPACK,ADR5
```

2245

005632 104420 000000 000432
005640 000573

```
*****
: CONVERT P,LBN+CMPPACK TO ASCII AND
: STORE AT ADR6
OTOA$,BEGIN,P,LBN+CMPPACK,ADR6
```

2246

005642 104420 000000 000420
005650 000602

```
*****
: CONVERT P,ADDA+CMPPACK TO ASCII AND
: STORE AT ADR7
OTOA$,BEGIN,P,ADDA+CMPPACK,ADR7
```

2247

005652 104420 000000 000416
005660 000611

```
*****
: CONVERT P,ADPA+CMPPACK TO ASCII AND
: STORE AT ADR8
OTOA$,BEGIN,P,ADPA+CMPPACK,ADR8
```

2248 005662 104403 000000 010226
2249 005670 000207

```
*****
MSGN$,BEGIN,BANLRI ASCII MESSAGE CALL WITH COMMON HEADER
RTS PC RETURN
```

```

2251 *****
2252
2253 DO DUP OPERATIONS
2254
2255 *****
2256
2257
2258 005672 DODUP;
2259 005672 1$:
2260
2261 005672 004767 000462 CALL GTDUST
2262
2263 005676 132767 000010 172373 B118 056,ACT,RESPONC+S,FLAG
2264 005704 001401 BEQ 2$
2265 005706 103371 BCC 1$
2266 005710 103423 2$: BCS 99$
2267
2268 005712 104417 000000 RAND$,BEGIN
2269 005716 016767 172132 172700 MOV RANUM,DBN
2270 005724 042767 177560 172672 B1C 0177560,DBN
2271 005732 004767 000240 CALL REDDBN
2272 005736 10 410 BCS 99$
2273
2274 005740 004767 000016 CALL WRITDBN
2275 005744 10 405 BCS 99$
2276
2277 005746 004767 000224 CALL REDDBN
2278
2279 005752 000402 BR 99$
2280 005754 10$:
2281 005754 004767 001220 CALL DUPERR
2282
2283 005760 000207 99$: RETURN
2284

```

```

2286
2287
2288
2289
2290
2291
2292 005762
2293 005762 012700 006476
2294 005766 004767 000424
2295 005772 103500
2296
2297
2298
2299 005774 012767 000002 172472
2300 006002 012767 012462 172466
2301 006010 004767 000556
2302 006014 103467
2303
2304
2305 006016 022767 010006 004436
2306 006024 001406
2307 006026 012767 000006 172566
2308 006034 004767 001140
2309 006040 000455
2310 006042
2311 006042 011467 004414
2312 006046 016767 172552 004410
2313 006054 012767 125125 004404
2314 006062 012767 000006 172404
2315 006070 012767 012462 172400
2316 006076 004767 000410
2317 006102 103434
2318
2319
2320 006104 012767 000004 172362
2321 006112 012767 012462 172356
2322 006120 004767 000446
2323 006124 103423
2324
2325
2326 006126 022767 030003 004326
2327 006134 001406
2328 006136 012767 000006 172456
2329 006144 004767 001030
2330 006150 000411
2331 006152
2332 006152 005767 004306
2333 006156 001406
2334 006160 012767 000006 172434
2335 006166 004767 001006
2336 006172 000400
2337
2338 006174
2339 006174 000207
2340

```

```

*****
: DO WRITE DBN
*****
:
WRTDBN:
MOV     @WDBN,R0
CALL    EXLCPG
BCS     99$
: SETUP = WRITE DBN
: CALL EXECUTE LOCAL PROGRAM
: IF ERROR
: : THEN
: : EXIT BLOCK
: ENDIF
: SET BYTE COUNT = 2
: SET DATA LOCATION
: CALL RECIEVE DATA
: IF ERROR
: : THEN -> EXIT BLOCK
: ENDIF
: IF RESPONSE NOT = OK
: : THEN
: : SET ERROR
: : CALL DUP_ERROR
: : EXIT BLOCK
: ENDIF
: SET UNIT NO
: SET DBN
: SET PATTERN
: SET BYTE COUNT = 6
: SET DATA LOCATION
: CALL SEND DATA
: IF ERROR
: : THEN -> EXIT BLOCK
: ENDIF
: SET BYTE COUNT = 4
: SET DATA LOCATION
: CALL RECIEVE DATA
: IF ERROR
: : THEN -> EXIT BLOCK
: ENDIF
: IF RESPONSE NOT = OK
: : THEN
: : SET ERROR
: : CALL DUP_ERROR
: : EXIT BLOCK
: ELSE
: : IF ERROR SET
: : : THEN
: : : SET ERROR
: : : CALL DUP_ERROR
: : : EXIT BLOCK
: : ENDIF
: ENDIF
:
99$: RETURN

```

```

2342 ;*****
2343 ;
2344 ; DO READ DBN
2345 ;
2346 ;
2347 ;*****
2348 ;
2349 REDDBN:
2350 006176 012700 006504 MOV #RDBN,R0 ; SETUP = READ DBN
2351 006202 004767 000210 CALL EXLCPG ; CALL EXECUTE_LOCAL PROGRAM
2352 006206 103463 BCS 99$ ; IF ERROR
2353 ; ; THEN -> EXIT BLOCK
2354 ; ENDF
2355 006210 012767 000002 172256 MOV #2,BYTCNT ; SET BYTE COUNT = 2
2356 006216 012767 012462 172252 MOV #RBUF,DATLOC ; SET DATA LOCATION
2357 006224 004767 000342 CALL RCVDAT ; CALL RECIEVE DATA
2358 006230 103452 BCS 99$ ; IF ERROR
2359 ; ; THEN -> EXIT BLOCK
2360 ; ENDF
2361 006232 022767 010005 004222 CMP #BIT12+5,RBUF ; IF RESPONSE NOT = OK
2362 006240 001406 BEQ 2$ ; ; THEN
2363 006242 012767 000006 172352 MOV #6,ERRFLG ; ; SET ERROR
2364 006250 004767 000724 CALL DUPERR ; ; CALL DUP_ERROR
2365 006254 000440 BR 99$ ; ; EXIT BLOCK
2366 006256 2$:
2367 006256 011467 004200 MOV (R4),RBUF ; SET UNIT NO
2368 006262 016767 172336 004174 MOV DBN,RBUF+2 ; SET DBN
2369 006270 012767 000004 172176 MOV #4,BYTCNT ; SET BYTE COUNT = 4
2370 006276 012767 012462 172172 MOV #RBUF,DATLOC ; SET DATA LOCATION
2371 006304 004767 000202 CALL SNDDAT ; CALL SEND DATA
2372 006310 103422 BCS 99$ ; IF ERROR
2373 ; ; THEN -> EXIT BLOCK
2374 ; ENDF
2375 006312 012767 001002 172154 MOV #514,BYTCNT ; SET BYTE COUNT = 514
2376 006320 012767 012462 172150 MOV #RBUF,DATLOC ; SET DATA LOCATION
2377 006326 004767 000240 CALL RCVDAT ; CALL RECIEVE DATA
2378 006332 103411 BCS 99$ ; IF ERROR
2379 ; ; THEN -> EXIT BLOCK
2380 ; ENDF
2381 006334 022767 060002 004120 CMP #BIT14+BIT13+2,RBUF ; IF R_BUF NOT = VALID
2382 006342 001405 BEQ 10$ ; ; THEN
2383 006344 012767 000006 172250 MOV #6,ERRFLG ; ; SET ERROR
2384 006352 004767 000622 CALL DUPERR ; ; CALL DUP_ERROR
2385 006356 10$:
2386 006356 000207 99$: RETURN ; ENDF
2387 ;

```

2389

2390

2391

2392

2393

2394

2395

2396

2397 006360

2398 006360 004767 177032

2399 006364 012767 000014 172000

2400 006372 012767 000026 171672

2401 006400 012767 000001 172000

2402 006406 012767 000201 172064

2403 006414 000516

2404

2405

2406

2407

2408

2409

2410

2411

2412

2413

2414 006416

2415 006416 004767 176774

2416 006422 012767 000022 171742

2417 006430 012767 000020 171634

2418 006436 012767 000003 171742

2419 006444 012767 000001 171736

2420 006452 012067 171734

2421 006456 012067 171732

2422 006462 011067 171730

2423 006466 012767 000203 172004

2424 006474 000466

2425

2426 006476 127 122 124

006501 104 102 116

2427 006504 122 105 104

006507 104 102 116

2428

2429

:

GET DUST STATUS

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

GET DUST STATUS

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

GET DUST STATUS

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

GET DUST STATUS

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

GET DUST STATUS

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

:

```

2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443 006512
2444 006512 004767 176700
2445 006516 016767 171750 171652
2446 006524 012767 000034 171640
2447 006532 012767 000020 171532
2448 006540 012767 000004 171640
2449 006546 016767 171722 171636
2450 006554 016767 171716 171634
2451 006562 012767 000204 171710
2452
2453 006570 000430
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467 006572
2468 006572 004767 176620
2469 006576 016767 171670 171572
2470 006604 012767 000034 171560
2471 006612 012767 000020 171452
2472 006620 012767 000005 171560
2473 006626 016767 171642 171556
2474 006634 016767 171636 171554
2475 006642 012767 000205 171630
2476
2477 006650 000400
2478
2479
2480

```

```

*****
:
: SEND DATA
:
: INPUT:
:          OPCODE =
:          BYTE_COUNT =
:          LOCATION =
:
: *****
:
: SNDDAT:
:          JSR      PC,CLRPAK          ; GO CLEAR THE COMMAND PACKET
:          MOV      CMDREF,P.CRF+CMACK ; SET COMMAND REFERENCE
:          MOV      C.C.SND,CMPLN      ; SET UP COMMAND PACKET LENGTH
:          MOV      R.R.SND,RSPLN      ; SET UP RESPONSE PACKET LENGTH ??????
:          MOV      ROP,SND,P.OPCD+CMACK ; SET THE OPCODE
:          MOV      BYTCNT,CMACK+14     ; SET BYTE_COUNT
:          MOV      DATLOC,CMACK+20     ; SET DATA_LOCATION
:          MOV      ROP,SND!OP.END,ENDCOD ; SET END CODE FOR CHECK
:
:          BR       SNDDUP              ; SEND THE PACKET
:
: *****
:
: RECEIVE DATA
:
: INPUT:
:          OPCODE =
:          BYTE_COUNT =
:          LOCATION =
:
: *****
:
: RCVDAT:
:          JSR      PC,CLRPAK          ; GO CLEAR THE COMMAND PACKET
:          MOV      CMDREF,P.CRF+CMACK ; SET COMMAND REFERENCE
:          MOV      C.C.RCD,CMPLN      ; SET UP COMMAND PACKET LENGTH
:          MOV      R.R.RCD,RSPLN      ; SET UP RESPONSE PACKET LENGTH ???
:          MOV      ROP,RCD,P.OPCD+CMACK ; SET THE OPCODE
:          MOV      BYTCNT,CMACK+14     ; SET BYTE_COUNT
:          MOV      DATLOC,CMACK+20     ; SET DATA_LOCATION
:          MOV      ROP,RCD!OP.END,ENDCOD ; SET END CODE FOR CHECK
:
:          BR       SNDDUP              ; SEND THE PACKET
:
: *****

```


2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521

006652 005267 171614
006656 001775
006660 016767 171606 171510
006666 011467 171510
006672 042767 040000 171366
006700 052767 100000 171360
006706 042767 040000 171346
006714 052767 100000 171340
006722 012767 001001 171444
006730 004767 000514
006734 005777 171046
006740 004767 000020
006744 004767 000516
006750 004767 000076
006754 103002
006756 004767 000216
006762 000207

```

*****
: SNDDUP - SEND A DUP PACKET
:
: SET UP THE COMMAND REFERENCE NUMBER AND UNITNO IN THE PACKET
: SET OWN, CLEAR FLAG IN THE COMMAND RING (FOR RQDX)
: SET OWN & FLAG IN MESSAGE RING
: AFTER DONE, MAKE SURE THE PACKET WAS PROCESSED (NO HARD
: OR SOFT ERRORS) THEN RETURN.
:
: INPUT: CMPACK IS FILLED EXCEPT FOR CMDREF & UNITNO
: INTERRUPT VECTOR AND BR LEVEL ARE ESTABLISHED
:
: OUTPUT: MSPACK IS FILLED
: CLEAR CARRY IF COMMAND PACKET WAS OK
: ELSE GO DO A HARD/SOFT ERROR
:
*****
SNDDUP: INC      CMDREF      ; NEW COMMAND REFERENCE NUMBER
          BEQ      SNDDUP      ; COMMAND REF # CANNOT = 0
          MOV      CMDREF,P.CRF+CMPPACK ; SET COMMAND REF NUMBER
          MOV      (R4),P.UNIT+CMPPACK ; SET UNIT NUMBER
          BIC      *RG.FLG,COMMND+2    ; CLEAR FLAG
          BIS      *RG.OWN,COMMND+2    ; SET OWN FOR COMMAND RING
          BIC      *RG.FLG,RSPONC+2    ; CLEAR FLAG
          BIS      *RG.OWN,RSPONC+2    ; SET OWN FOR RESPONSE RING
          MOV      *1001,CMPVIR        ; SET DUP COMMAND
          JSR      PC,OUTPKT           ; PRINT PACKET (DIAGNOSTICS)
          TST      @ADDR               ; FORCE POLLING TO PACKET
          :
          : -----
          CALL     POLL                ; CALL POLL FOR DONE ( )
          JSR      PC,INPKT             ; PRINT PACKET (DIAGNOSTICS)
          CALL     RSPCHK               ; CALL RESPONSE CHECK
          BCC      99$                 ; IF ERROR
          : THEN
          : : CALL DUP ERROR
          : : ENDF
          99$: RETURN

```

2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537

2538
2539
2540
2541
2542
2543

2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555

2556
2557
2558

```

: *****
: POLL      WAIT FOR DONE
:
:   AFTER DONE MAKE SURE THE PACKET WAS PROCESSED (NO HARD
:   OR SOFT ERRORS) THEN RETURN.
:
:   INPUT:
:
:   OUTPUT:
:
: *****
: POLL:
:     MOV     R4, -(R6)
:     MOV     R5, -(R6)
:     CLC
:     MOV     #100, R4      ; CLEAR ERROR FLAG
:                               ; SET TIME_OUT = 100
1$:     MOV     #10, R5      ; BGND0
:                               ; : SET DOWN_COUNTER = 10
2$:     MOV     #10, R5      ; : BGND0
:                               ; :
:     BREAK$, BEGIN          ; TEMPORARY RETURN TO MONITOR....
:     BREAK$, BEGIN          ; THEN CONTINUE AT NEXT INSTRUCTION.
:     DEC     R5              ; : : DECREMENT DOWN_COUNTER
:     BNE     2$              ; : DOUNTIL DOWN_COUNTER = 0
:     DEC     R4              ; : DECREMENT TIME_OUT
:                               ; :
:     BIT     @RG.OWN, RSPONC+2 ; DOUNTIL RESPONSE = DONE (WE OWN)
:     BEQ     99$             ; OR
:     BIT     @SA.ERR, @RQDXSA ; SA = ERROR
:     BNE     99$             ; OR
:     IST     R4              ; TIME_OUT = 0
:     BNE     1$              ;
:                               ;
99$:     MOV     (R6)+, R5
:     MOV     (R6)+, R4
:     RETURN
:

```


MODULE CODE

```

2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572 007052
2573 007052 000241
2574 007054 005067 171026
2575 007060 005067 171536
2576 007064 026767 171216 171406
2577 007072 001405
2578 007074 000261
2579 007076 012767 000001 171516
2580 007104 000434
2581 007106 026767 171164 171356 1$:
2582 007114 001405
2583 007116 000261
2584 007120 012767 000002 171474
2585 007126 000423
2586 007130 016700 171154 2$:
2587 007134 042700 177740
2588 007140 001405
2589
2590 007142 000261
2591 007144 012767 000003 171450
2592 007152 000411
2593 007154 032777 100000 171070 3$:
2594 007162 001405
2595 007164 000261
2596 007166 012767 000004 171426
2597 007174 000400
2598 007176 000207
2599

```

```

*****
:
: CHECK RESPONSE AND COMMAND PACKETS
:
: INPUT:
:      END_CODE
:
: OUTPUT:
:      ERROR
:
*****
:
: RSPCHK:
:
: CLR
: CLR ERRTP
: CLR ERRFLG
: CMP RSPACK+P.OPCD,ENDCOD
: BEQ 1$
: SEC
: MOV #1,ERRFLG
: BR 99$
: CMP RSPACK,CMDREF
: BEQ 2$
: SEC
: MOV #2,ERRFLG
: BR 99$
: MOV P.STS+RSPACK,R0
: BIC #17740,R0
: BEQ 3$
:
: SEC
: MOV #3,ERRFLG
: BR 99$
: BIT #5A,ERR,&RQDXSA
: BEQ 99$
: SEC
: MOV #4,ERRFLG
: BR 99$
: 99$: RETURN
:
: CLEAR ERROR
: ERROR TYPE = 0
: ERROR FLAGS = 0
: IF RESPONSE_CODE NOT = END_CODE
: : THEN
: : SET ERROR
: : END CODE DOES NOT MATCH
: ENDIF
: IF CMD_REF_NUMBER NOT = REF_NUMBER
: : THEN
: : SET ERROR
: : COMMAND_REF_NO NOT MATCH RESPONSE
: ENDIF
: GET STATUS SUCCESS
: CLEAR UPPER 11 BITS OF SUB-STATUS
: IF STATUS NOT = 0
: : THEN
: : SET ERROR
: : COMMAND NOT SUCCESSFUL
: ENDIF
: IF SA = ERROR
: : THEN
: : SET ERROR
: : SA = ERROR
: ENDIF
:

```

```

2601 ;*****
2602 ;
2603 ;   DUP HARD ERROR           CARRY WILL BE SET
2604 ;
2605 ;*****
2606 ;
2607 007200 DUPERR:
2608 007200 032767 000004 170610 BIT    #SR.REP,SR1    ; IF REPORT ERROR = NO
2609 007206 001403          BELQ    7$              ; : THEN
2610 007210 005267 170630 INC     HRLCNT          ; : INCREMENT THE HARD ERROR
2611 007214 000465          BR      99$              ; :
2612 007216          7$:                          ; : ELSE
2613 007216 004767 176314 CALL    SETTAB          ; : SET UP TABLE
2614          007222 104405 000000' 000000 ;*****
          ;HDRER$,BEGIN,NULL ;
          ;*****
2615 007230 004767 000140 CALL    DERRPT          ; : CALL DUP_ERROR_REPORT
2616          ; : SELECT ERROR_FLAGS
2617 007234 005367 171362 DEC     ERRFLG          ; : : CASE END CODE ERROR:
2618 007240 001004          BNE     10$              ; :
2619 007242 104402 000000' 010126' MSGS$,BEGIN,DUPMS1 ; ASCII MESSAGE CALL WITH NO HEADER
2620 007250 000447          BR      99$              ; :
2621 007252 005367 171344 10$: DEC     ERRFLG          ; : : CASE COMMAND REF ERROR:
2622 007256 001004          BNE     12$              ; :
2623 007260 104402 000000' 010140' MSGS$,BEGIN,DUPMS2 ; ASCII MESSAGE CALL WITH NO HEADER
2624 007266 000440          BR      99$              ; :
2625 007270 005367 171326 12$: DEC     ERRFLG          ; : : CASE NO SUCCESS:
2626 007274 001004          BNE     14$              ; :
2627 007276 104402 000000' 010152' MSGS$,BEGIN,DUPMS3 ; ASCII MESSAGE CALL WITH NO HEADER
2628 007304 000431          BR      99$              ; :
2629 007306 005367 171310 14$: DEC     ERRFLG          ; : : CASE SA ERROR:
2630 007312 001004          BNE     16$              ; :
2631 007314 104402 000000' 010162' MSGS$,BEGIN,DUPMS4 ; ASCII MESSAGE CALL WITH NO HEADER
2632 007322 000422          BR      99$              ; :
2633 007324 005367 171272 16$: DEC     ERRFLG          ; : : CASE SEND ERROR:
2634 007330 001004          BNE     18$              ; :
2635 007332 104402 000000' 010172' MSGS$,BEGIN,DUPMS5 ; ASCII MESSAGE CALL WITH NO HEADER
2636 007340 000413          BR      99$              ; :
2637 007342 005367 171254 18$: DEC     ERRFLG          ; : : CASE RECEIVE ERROR:
2638 007346 001004          BNE     50$              ; :
2639 007350 104402 000000' 010204' MSGS$,BEGIN,DUPMS6 ; ASCII MESSAGE CALL WITH NO HEADER
2640 007356 000404          BR      99$              ; :
2641 007360          50$:                          ; : : DEFAULT : DUP ERROR
2642 007360 104402 000000' 010216' MSGS$,BEGIN,DUPMSF ; ASCII MESSAGE CALL WITH NO HEADER
2643 007366 000400          BR      99$              ; :
2644          ; : ENDSELECT
2645          99$:                          ; :
2646 007370          SEC                      ; SET ERROR_EXIT
2647 007372 000207          RTS     PC              ; RETURN TO CYCLED
2648

```

```

*****
;
;   DUP ERROR REPORT
;
;   INPUT:
;
;   OUTPUT: REPORT ERROR
;
;           PRINT STATUS, OPCODE, UNIT NUMBER, DBN )
;
*****
DERRPT:
; *****
;                               ;CONVERT P.STS+RSPACK TO ASCII AND
;                               ;STORE AT ADR1
OTOA$,BEGIN,P.STS+RSPACK,ADR1
; *****
; *****
;                               ;CONVERT P.OPCD+RSPACK TO ASCII AND
;                               ;STORE AT ADR2
OTOA$,BEGIN,P.OPCD+RSPACK,ADR2
; *****
MOV      (R4),NUM                ; GET UNIT #
; *****
;                               ;CONVERT NUM TO ASCII AND
;                               ;STORE AT ADR3
OTOA$,BEGIN,NUM,ADR3
; *****
; *****
;                               ;CONVERT DBN TO ASCII AND
;                               ;STORE AT ADR4
OTOA$,BEGIN,DBN,ADR4
; *****
MSGN$,BEGIN,BANER2              ;ASCII MESSAGE CALL WITH COMMON HEADER
RETURN                          ;

```

```

2672 ;*****
2673 ;
2674 ; INPKT AND OUTPKT - PRINT RESPONSE AND COMMAND PACKETS
2675 ;
2676 ;*****
2677 007450 010246 OUTPKT: MOV R2, -(SP)
2678 007452 012702 000376' MOV @CMPACK, R2
2679 007456 012767 052517 001074 MOV @"OU, RY1
2680 007464 000406 BR COM
2681 007466 010246 INPKT: MOV R2, -(SP)
2682 007470 012702 000276' MOV @RSPACK, R2
2683 007474 012767 047111 001056 MOV @"IN, RY1
2684
2685 007502 032767 000001 170306 COM: BIT @SR.PAK, SR1 ;PRINT PACKET INFORMATION?
2686 007510 001460 BEQ 10$
2687 007512 010046 MOV R0, -(SP)
2688 007514 012700 000004 MOV @4, R0
2689
2690 007520 104403 000000' 010102' MSGN$, BEGIN, PAKHEB ;ASCII MESSAGE CALL WITH COMMON HEADER
2691 007526 012267 170762 1$: MOV (R2)+, NUM
2692 ;*****
;CONVERT NUM TO ASCII AND
;STORE AT ADR3
007532 104420 000000' 000514' OTOA$, BEGIN, NUM, ADR3
007540 000546
;*****
2693 007542 012267 170746 MOV (R2)+, NUM
2694 ;*****
;CONVERT NUM TO ASCII AND
;STORE AT ADR4
007546 104420 000000' 000514' OIOA$, BEGIN, NUM, ADR4
007554 000555'
;*****
2695 007556 012267 170732 MOV (R2)+, NUM
2696 ;*****
;CONVERT NUM TO ASCII AND
;STORE AT ADR5
007562 104420 000000' 000514' OIOA$, BEGIN, NUM, ADR5
007570 000564'
;*****
2697 007572 012267 170716 MOV (R2)+, NUM
2698 ;*****
;CONVERT NUM TO ASCII AND
;STORE AT ADR6
007576 104420 000000' 000514' OTOA$, BEGIN, NUM, ADR6
007604 000573'
;*****
2699 007606 012267 170702 MOV (R2)+, NUM
2700 ;*****
;CONVERT NUM TO ASCII AND
;STORE AT ADR7
007612 104420 000000' 000514' OIOA$, BEGIN, NUM, ADR7
007620 000602'
;*****
2701 007622 012267 170666 MOV (R2)+, NUM
2702 ;*****
;CONVERT NUM TO ASCII AND

```

	007636	104400	000000	000514	UTOA\$,BEGIN,NUM,ADDR	STORE AT ADDR
	007634	000611				
2103	007636	104400	000000	010240	*****	
2104					MSG\$,BEGIN,PACKET	ASCII MESSAGE CALL WITH NO HEADER
2105	007644	005300			DEC	R0
	007646	001307			BNE	1\$
2106	007650	012600			MOV	(SP)+,R0
2107	007652	012602	10\$:		MOV	(SP)+,R2
2108	007654	000207			RTS	PC
2109						

OP.AVI = 000010	PRTY5 = 000240	QMON22 = 000010	SA.CME = 000360	ST.ABO = 000002
OP.CCD = 000021	PRTY6 = 000300	RANACC = 004050R	SA.CTP = 003400	ST.ALO = 000006
OP.CMP = 000040	PRTY7 = 000340	RAND\$ = 104417	SA.DIA = 000400	ST.AVL = 000004
OP.DAP = 000013	PS = 177776	RANNUM = 000054R	SA.ERC = 003777	ST.CMD = 000001
OP.ELP = 000003	PSW = 177776	RBUF = 012462R	SA.ERR = 100000	ST.CMP = 000007
OP.END = 000200	PUSH = 005746	RBUFEA = 000130R	SA.GO = 000001	ST.CNT = 000012
OP.ERL = 000101	PUSH2 = 024646	RBUFEA = 000126R	SA.INE = 000200	ST.DAT = 000010
OP.ERS = 000022	PWRFLG = 000002	RBUFSZ = 000132R	SA.INT = 000200	ST.DIA = 000037
OP.ESP = 000002	P.ADEA = 000022	RBUFEA = 000124R	SA.LFC = 040000	ST.DRV = 000013
OP.FLL = 000023	P.ADPA = 000020	RCVDAT = 006572R	SA.PRG = 000001	ST.FAL = 000005
OP.GCS = 000002	P.BCNT = 000014	RDBN = 006504R	SA.QB = 001000	ST.HST = 000011
OP.GDS = 000001	P.BUFF = 000020	RDONLY = 010062R	SA.RSE = 000017	ST.MFE = 000003
OP.GUS = 000003	P.CMST = 000020	RDWRT = 010050R	SA.RSP = 007400	ST.MSK = 000037
OP.MRD = 000030	P.CNCL = 000022	READ = 004466R	SA.S1 = 004000	ST.NON = 000004
OP.MWR = 000031	P.CNT = 000006	READA = 004520R	SA.S2 = 010000	ST.NRA = 000002
OP.ONL = 000011	P.CNTF = 000016	REDDBN = 006176R	SA.S3 = 020000	ST.NRS = 000003
OP.RCD = 000005	P.CNTI = 000024	RESTR1 = 001124R	SA.S4 = 040000	ST.OFL = 000003
OP.RD = 000041	P.CPSP = 000042	RESTR2 = 001260R	SA.VCE = 000177	ST.SUB = 000040
OP.RPL = 000024	P.CRF = 000000	RESTR2 = 001234R	SA.VEC = 000177	ST.SUC = 000000
OP.SCC = 000004	P.CTMO = 000020	RES1 = 000056R	SBADR = 000102R	ST.WPR = 000006
OP.SHC = 000102	P.CYL = 000050	RES2 = 000060R	SCC = 004614R	SVR0 = 000062R
OP.SND = 000004	P.DEV = 000034	RG.FLG = 040000	SC.DIS = 000400	SVR1 = 000064R
OP.SUC = 000012	P.ELGF = 000034	RG.OWN = 100000	SC.DUP = 000200	SVR2 = 000066R
OP.WR = 000042	P.FBBK = 000034	RH70 = 001000	SC.INV = 000100	SVR3 = 000070R
OTOA\$ = 104420	P.FLGS = 000011	RINTR = 000256R	SC.IOP = 000100	SVR4 = 000072R
OUTPKT = 007450R	P.GRP = 000046	RLIM = 000004	SC.NVL = 000040	SVR5 = 000074R
PA = 000504R	P.HSTI = 000020	RQDXSA = 000252R	SC.STO = 000040	SVR6 = 000076R
PACKET = 010240R	P.HTMO = 000020	RSECH = 000636R	SEND = 004704R	SYSCNT = 000052R
PAKHEO = 010102R	P.LBN = 000034	RSECL = 000634R	SEQACC = 004010R	S.EXT = 000014
PARPRE = 002000	P.LGDT = 000014	RSPACK = 000276R	SETTAB = 005536R	S.FLAG = 000017
PASCNT = 000034R	P.MEDI = 000034	RSPCHK = 007052R	SETUP = 003032R	S.PRG\$ = 000020
PDPF11 = 000002	P.MLUN = 000014	RSPLN = 000272R	SE.ACT = 000010	S.TEXT = 000000
PDPFSI = 020000	P.MOD = 000012	RSPONC = 000260R	SE.DIS = 000001	S.TOUT = 000024
PDP44 = 100000	P.OPCD = 000010	RSPVIR = 000274R	SE.IDL = 000000	TABLEW = 000656R
PDP60 = 004000	P.OTRF = 000014	RSTRT = 000112R	SE.LOC = 000002	TEMP = 000530R
PDP70 = 010000	P.OVRL = 000034	RVA = 000756R	SE.SUP = 000004	TEND = 000756R
PH.ALO = 000001	P.RBN = 000014	RY0 = 010550R	SE.TELG = 000626R	TIMER = 027340
PH.DLG = 000010	P.RBNS = 000056	RY1 = 010560R	SNDDAT = 006512R	TIMOUT = 005670
PH.OVR = 000002	P.RCTC = 000057	RY3 = 010564R	SNDDUP = 006652R	TRPDEF = 000023
PH.RW = 000004	P.RCTS = 000054	RY4 = 010574R	SNDSIP = 002554R	TRY = 000646R
PICKBK = 004000R	P.RGID = 000034	RY5 = 010612R	SOFCNT = 000042R	TSTOFL = 003330R
PIRQ\$ = 000004	P.RGOF = 000040	RY6 = 010642R	SOFR\$ = 104406	UF.CMR = 000001
PKYSIZ = 000060	P.RSVD = 000011	RY7 = 010670R	SOFPAS = 000046R	UF.CMW = 000002
POLL = 006764R	P.SFTW = 000040	RY8 = 010714R	SPOINT = 000032R	UF.INA = 040000
POPSP = 005726	P.SHST = 000042	R.MESG = 000000	SPSIZ = 000040	UF.RMV = 000200
POPSP2 = 022626	P.SHUN = 000040	R.TEXT = 000002	SR.CMP = 004000	UF.RPL = 010000
PRHMS\$ = 000002	P.STS = 000012	R.TYPE = 000001	SR.PAK = 000001	UF.SCH = 004000
PRINTE = 005562R	P.SZOF = 000012	R18EA = 000762R	SR.REP = 000004	UF.SCL = 002000
PRNMSG = 000526R	P.TIME = 000024	R18PA = 000760R	SR.RON = 000002	UF.WBN = 000040
PRNUM = 000017	P.TRCK = 000044	R22EA = 000766R	SR.SEQ = 002000	UF.WPF = 020000
PRTSTS = 001742R	P.UNFL = 000016	R22PA = 000764R	SR.SUM = 000010	UF.WPS = 010000
PRTT = 000000	P.UNIT = 000004	R6 = 000006	SR1 = 000016R	UF.S76 = 000004
PRTT0 = 000000	P.UNS7 = 000044	R7 = 000007	SR2 = 000020R	UNIDFF = 010014R
PRTT1 = 000040	P.UNTI = 000024	SANOT0 = 010004R	SR3 = 000022R	UNITEL = 000650R
PRTT2 = 000100	P.USFF = 000022	SAVEA = 000524R	SR4 = 000024R	UN57H = 000642R
PRTT3 = 000140	P.VRSN = 000014	SAVEPA = 000522R	START = 001002R	UN57I = 000640R
PRTT4 = 000200	P.VSER = 000050	SA.CMD = 170000	STAT = 000026R	UPPER = 000002

USTACK= 000001	WBUFQ 000140R	W18EA 000774R	.C.ABO= 000014	.R.RCD= 000020
VA 000502R	WBUF5Z 000142R	W18PA 000772R	.C.DUS= 000014	.R.SND= 000020
VECTOR 000010R	WDBN 006476R	W22EA 001000R	.C.LOC= 000022	.R.SUP= 000014
WAIT 004764R	WDFR 000116R	W22PA 000776R	.C.RCD= 000034	.T.DFF= 000040
WARN1 010024R	WDT0 000114R	XFLAG 000005R	.C.SND= 000034	.T.FAT= 000120
WARN2 010030R	WORK 000652R	XSECH 000632R	.C.SUP= 000050	.T.INF= 000060
WARN3 010034R	WRITE 004416R	XSECL 000630R	.R.ABO= 000014	.T.QUE= 000020
WASADR 000104R	WRITEA 004430R	ZERO 010044R	.R.DUS= 000026	.T.SPI= 000140
WBUFEA 000136R	WRIDBN 005762R	\$\$ = 000000	.R.LOC= 000020	.T.TER= 000100
WBUFEA 000134R	WVA 000770R			

.ABS. 000000 000
013472 001

ERRORS DETECTED: 0
VIRTUAL MEMORY USED: 13793 WORDS (54 PAGES)
DYNAMIC MEMORY: 19748 WORDS (75 PAGES)
ELAPSED TIME: 00:02:14
XRQADO, XRQADO/CR/-SP=DDXCOM, XRQADO

DEC/X11 SYSTEM EXERS....B1
 DEC/X11 SYSTEM EXERS....C1
 DEC/X11 SYSTEM EXERS....D1
 DEC/X11 SYSTEM EXERS....E1
 DEC/X11 SYSTEM EXERS....F1
 DEC/X11 SYSTEM EXERS....G1
 DEC/X11 SYSTEM EXERS....H1
 DEC/X11 SYSTEM EXERS....I1
 DEC/X11 SYSTEM EXERS....J1
 DEC/X11 SYSTEM EXERS....K1
 MODULE HEADER BLOCK L1
 MODULE HEADER BLOCK M1
 MODULE STORAGE AREA N1

MODULE CODE B5
 MODULE CODE C5
 MODULE CODE D5
 MODULE CODE E5
 MODULE CODE F5
 MODULE CODE G5
 MODULE CODE H5
 MODULE CODE I5
 MODULE CODE J5
 MODULE CODE K5
 MODULE CODE L5
 MODULE CODE M5
 MODULE CODE N5

MORE MODULE STORAGE B2
 MODULE PRIVATE DATA C2
 EQUATES D2
 RQDX BIT DEFINITIONS....E2
 COMMAND/MESSAGE DESC....F2
 COMMAND MODIFIERS G2
 CONTROLLER FLAGS H2
 END PACKET OFFSETS I2
 ERROR LOG FLAGS J2
 DUP DEFINITIONS K2
 OFFSET AND FLAG DEFI....L2
 OFFSET AND FLAG DEFI....M2
 OFFSET AND FLAG DEFI....N2

MODULE CODE B6
 MODULE MESSAGES C6
 MODULE MESSAGES D6
 MODULE MESSAGES E6
 MODULE MESSAGES F6
 MORE MODULE MESSAGES....G6
 MORE MODULE MESSAGES....H6
 SYMBOL TABLE I6
 SYMBOL TABLE J6
 SYMBOL TABLE K6

MODULE CODE J3
 MODULE CODE C3
 MODULE CODE D3
 MODULE CODE E3
 MODULE CODE F3
 MODULE CODE G3
 MODULE CODE H3
 MODULE CODE I3
 MODULE CODE J3
 MODULE CODE K3
 MODULE CODE L3
 MODULE CODE M3
 MODULE CODE N3

MODULE CODE B4
 MODULE CODE C4
 MODULE CODE D4
 MODULE CODE E4
 MODULE CODE F4
 MODULE CODE G4
 MODULE CODE H4
 MODULE CODE I4
 MODULE CODE J4
 MODULE CODE K4
 MODULE CODE L4
 MODULE CODE M4
 MODULE CODE N4

ZRQAM1

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX 11 B1199 16 V4.0 579
DISK\$USER2:[POWER5]ZRQADO.BL1;6

SEQ 0001
Page 1
(1)

```

: 0001 0 module ZRQAM1 (
: 0002 0
: 0003 0 *title 'RD/RX EXERCISER'
: 0004 0 ident = 'V01.6',
: 0005 0 addressing_mode (absolute),
: 0006 0 environment (noeig)
: 0007 0 )
: 0008 0
: 0009 1 begin
: 0010 1
: 0011 1
: C 0012 1 *
: C 0013 1 IDENTIFICATION
: C 0014 1
: C 0015 1
: C 0016 1 PRODUCT CODE: AC-1398D-MC
: C 0017 1
: C 0018 1 PRODUCT NAME: CZRQADO RQDX1/RUX50 EXERCISER
: C 0019 1
: C 0020 1 PRODUCT DATE: 11 APR 84
: C 0021 1
: C 0022 1 MAINTAINER: DIAGNOSTIC ENGINEERING
: C 0023 1
: C 0024 1 AUTHOR: RAVINDER K. KARWAN
: C 0025 1 BOB POWERS
: C 0026 1
: C 0027 1
: C 0028 1 Copyright (C) 1983, 1984
: C 0029 1
: C 0030 1 Digital Equipment Corporation, Maynard, Massachusetts 01754
: C 0031 1
: C 0032 1 This software is furnished under a license for use only on a single
: C 0033 1 computer system and may be copied only with the inclusion of the
: C 0034 1 above copyright notice. This software, or any other copies thereof,
: C 0035 1 may not be provided or otherwise made available to any other person
: C 0036 1 except for use on such system and to one who agrees to these license
: C 0037 1 terms. Title to and ownership of the software shall at all times
: C 0038 1 remain in DEC.
: C 0039 1
: C 0040 1 the information in this document is subject to change without notice
: C 0041 1 and should not be construed as a commitment by Digital Equipment
: C 0042 1 Corporation.
: C 0043 1
: C 0044 1 DEC assumes no responsibility for the use or reliability of its
: C 0045 1 software on equipment which is not supplied by DEC.
: C 0046 1
: C 0047 1
: C 0048 1 The following are trademarks of Digital Equipment Corporation:
: C 0049 1
: C 0050 1 DIGITAL POP UNIBUS MASSBUS
: C 0051 1 DEC DECUS DECTAPE

```

C1

ZRQAM1
VOL.6

RD-RX EXERCISER

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX-11 B1199 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0002
Page 2
(2)

: C 0052 1
: C 0053 1
: C 0054 1
: C 0055 1
: C 0056 1
: C 0057 1
: C 0058 1
: C 0059 1
: C 0060 1
: C 0061 1
: C 0062 1
: C 0063 1
: C 0064 1
: C 0065 1
: C 0066 1
: C 0067 1
: C 0068 1
: C 0069 1
: C 0070 1
: C 0071 1
: C 0072 1
: C 0073 1
: C 0074 1
: C 0075 1
: C 0076 1
: C 0077 1
: C 0078 1
: C 0079 1
: C 0080 1
: C 0081 1
: C 0082 1
: C 0083 1
: C 0084 1
: C 0085 1
: C 0086 1
: C 0087 1
: C 0088 1
: C 0089 1
: C 0090 1
: C 0091 1
: C 0092 1
: C 0093 1
: C 0094 1

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
1.6	MEMORY MAP
2.0	OPERATING INSTRUCTIONS
2.1	HARDWARE QUESTIONS
2.2	SOFTWARE QUESTIONS
3.0	ERROR TYPES
3.1	ERROR INFORMATION
3.2	INITIALIZATION ERRORS
3.3	EXERCISER ERRORS
3.4	ERROR LOG MESSAGES
3.5	MSCP ERRORS
3.6	SAMPLE ERROR STATEMENT
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	TEST SUMMARY
5.1	INITIALIZATION SUBTEST
5.2	EXERCISER
5.3	DROP UNIT SUMMARY
6.0	ERROR LIST
7.0	DATA PATTERNS

D1

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0 579
DISK\$USER2:([POWER5])ZRQAD0.BL1;6

SEQ 0003
Page 3
(3)

```

: C 0095 1
: C 0096 1
: C 0097 1
: C 0098 1
: C 0099 1
: C 0100 1
: C 0101 1
: C 0102 1
: C 0103 1
: C 0104 1
: C 0105 1
: C 0106 1
: C 0107 1
: C 0108 1
: C 0109 1
: C 0110 1
: C 0111 1
: C 0112 1
: C 0113 1
: C 0114 1
: C 0115 1
: C 0116 1
: C 0117 1
: C 0118 1
: C 0119 1
: C 0120 1
: C 0121 1
: C 0122 1
: C 0123 1
: C 0124 1
: C 0125 1
: C 0126 1
: C 0127 1
: C 0128 1
: C 0129 1
: C 0130 1
: C 0131 1
: C 0132 1
: C 0133 1
: C 0134 1
: C 0135 1
: C 0136 1
: C 0137 1
: C 0138 1
: C 0139 1
: C 0140 1
: C 0141 1
: C 0142 1
: C 0143 1
: C 0144 1
: C 0145 1

1.0 GENERAL INFORMATION
.....

1.1 PROGRAM ABSTRACT
.....

This program will functionally verify and exercise RQDX1
or RUX50 Controller/Disk Drive subsystems. It is designed
to verify that the subsystem is functioning correctly and
operating within design specifications.

1.2 SYSTEM REQUIREMENTS
.....

1.2.1 HARDWARE REQUIREMENTS
.....

LSI - 11/23 processor with 28K or more of memory, console
device (eg. VT100) and RQDX1 or RUX50 controller board and
attached RD51 or RD52 WINCHESTER drive(s) and RX-50 FLOPPY
drive(s)

1.2.2 SOFTWARE REQUIREMENTS
.....

This diagnostic is designed to run with the Diagnostic
Supervisor as described in paragraph 2.0.

1.3 RELATED DOCUMENTS AND STANDARDS
.....

XXDP, SUPERVISOR/USERS MANUAL CHQUN
UQSSP UNIBUS/Q BUS STORAGE SYSTEMS PORT
MSCP MASS STORAGE SYSTEM PROTOCOL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
.....

NONE

1.5 ASSUMPTIONS
.....

The hardware, other than the subsystem being tested, is
assumed to work properly. False errors may be reported if
the processor, memory, etc., do not function properly.

```

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0004
Page 4
(4)

1.6 MEMORY MAP

Memory layout on 28k machine - XXDP environment

ADDRESS

160000

XXDP MONITOR

XXDP TTY I/O AREA

DRS

121000

EXERCISER

2000

STACKS

400

VECTOR AREA

0

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

: C 0136 1
: C 0137 1
: C 0188 1
: C 0189 1
: C 0190 1
: C 0191 1
: C 0192 1
: C 0193 1
: C 0194 1
: C 0195 1
: C 0196 1
: C 0197 1
: C 0198 1
: C 0199 1
: C 0200 1
: C 0201 1
: C 0202 1
: C 0203 1
: C 0204 1
: C 0205 1
: C 0206 1
: C 0207 1
: C 0208 1
: C 0209 1
: C 0210 1
: C 0211 1
: C 0212 1
: C 0213 1
: C 0214 1
: C 0215 1
: C 0216 1
: C 0217 1
: C 0218 1
: C 0219 1
: C 0220 1
: C 0221 1
: C 0222 1
: C 0223 1
: C 0224 1
: C 0225 1
: C 0226 1
: C 0227 1
: C 0228 1
: C 0229 1
: C 0230 1
: C 0231 1
: C 0232 1
: C 0233 1
: C 0234 1

2.0 OPERATING INSTRUCTIONS

This is a Rev C Supervisor Diagnostic: for operating instructions, please see chapter 5 of XXDP: operator's manual. They are no longer included in the diagnostic because it is desired that a change in those instructions not require a re-assembly of all Supervisor Diagnostics.

2.1 HARDWARE QUESTIONS

The following series of questions collect the parameters necessary to identify each disk subsystem.

Hardware Configuration Questions

The program will ask the following questions in response to a START command (non-script).

1. CHANGE HW (L) Y ?

Answer NO to use the pre built answers for all hardware questions. This program will be released pre-built to test three units with default answers shown below. The pre built answers may be changed at any time with the setup utility. Answer YES if you want all the hardware questions to be asked.

2. NUMBER OF UNITS (D) ?

No default. Answer with the number of disk drive units to be exercised or tested. This answer will determine how many times the following questions are asked. A range of 1 to 4 units may be specified. A unit number will be assigned sequentially from 0 by the Diagnostic supervisor for each unit.

3. IP ADDRESS (U) 172150 ?

Enter the address of the IP register of one RQDX1 or RUX50 as addressed by the processor with memory management turned off. The program expects an even 16 bit address in the range of 160000 to 177744. 172150 is the default.

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-577
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0006
Page 6
(6)

4. VECTOR ADDRESS (0) 154 ?
Answer with the interrupt vector of the same RQDX1 or RUX50 controller described in the above question. A vector address in the range of 4 to 774 may be specified, 154 is the default.

5. BR LEVEL (0) 4 ?
Answer with the bus request interrupt level used by the above controller. Levels 4 through 7 are acceptable, 4 is the default.

6. DRIVE NUMBER (0) 0 ?
Enter the logical unit number for one drive associated with the IP address above. Drive numbers are in the range of 0 through 15. The number entered here must match the unit plug on the front panel of the drive, and must be within the range implied by the jumper (LJNO ?) on the RQDX1 or RUX50 controller board. 0 is the default answer.

7. TEST ENTIRE CUSTOMER DATA AREA OF THIS DISK (1) Y ?
This question is asked to give the opportunity of limiting the addressing range over which the testing will be performed. An affirmative answer will cause no limits to be imposed for the unit in question. A negative answer will cause limits to be imposed, as defined by the following four questions.

8. LOWER OCTAL WORD OF BEGINNING LBN ADDRESS (0) 0 ?
Enter in octal the less significant 16 bit word of the lowest LBN address in the test range. The value may be from 000000 to 177777.

9. HIGHER OCTAL WORD OF BEGINNING LBN ADDRESS (0) 0 ?
Enter in octal the more significant 16-bit word of the lowest LBN address in the test range.

10. LOWER OCTAL WORD OF ENDING LBN ADDRESS (0) 150477 ?
Enter in octal the less significant 16-bit word of the highest LBN address in the test range. 150477 is the highest LBN address for an RQ52.

H1

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11ss 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0007 /
Page (6)

: C 0288 1
: C 0289 1
: C 0290 1
: C 0291 1
: C 0292 1
: C 0293 1
: C 0294 1
: C 0295 1
: C 0296 1
: C 0297 1
: C 0298 1
: C 0299 1
: C 0300 1
: C 0301 1
: C 0302 1
: C 0303 1
: C 0304 1

11. HIGHER OCTAL WORD OF ENDING LBN ADDRESS (0) 0?

Enter in octal the more significant 16-bit word of the
highest LBN address in the test range.

Note:

The four previous questions are usually software
Parameter questions, but since three different disk
drives exist on the 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
drives. The program will check for block number bounds,
and, if they are exceeded, will assign the maximum
bounds for that drive.

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11 Apr-1984 11:45:02

VAX-11 Bliss 16 v4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0008
Page 8
(7)

; C 0305 1
; C 0306 1
; C 0307 1
; C 0308 1
; C 0309 1
; C 0310 1
; C 0311 1
; C 0312 1
; C 0313 1
; C 0314 1
; C 0315 1
; C 0316 1

12. EXERCISE ON CUSTOMER DATA AREA ON THIS DISK UNIT (1) ?

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 (1) ?

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

ZRGAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B11-1.1b V4.0 579
DISK\$USER2:[POWERS]ZRGADO.BL1;6SEQ 0009
Page 9
(8)

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. ENTER TIME AS HHMM (EXAMPLE: 1305) (D) 0 ?

Enter the time of day (in 24 hour format). DRS does not like to see leading zeros in numeric values entered. For example, for 14 minutes past midnight, you would enter 14, and for 30 minutes past 3 in the afternoon, enter 1530.

3. 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.

4. 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.

: C 0317 1
: C 0318 1
: C 0319 1
: C 0320 1
: C 0321 1
: C 0322 1
: C 0323 1
: C 0324 1
: C 0325 1
: C 0326 1
: C 0327 1
: C 0328 1
: C 0329 1
: C 0330 1
: C 0331 1
: C 0332 1
: C 0333 1
: C 0334 1
: C 0335 1
: C 0336 1
: C 0337 1
: C 0338 1
: C 0339 1
: C 0340 1
: C 0341 1
: C 0342 1
: C 0343 1
: C 0344 1
: C 0345 1
: C 0346 1
: C 0347 1
: C 0348 1
: C 0349 1
: C 0350 1
: C 0351 1
: C 0352 1
: C 0353 1
: C 0354 1
: C 0355 1
: C 0356 1
: C 0357 1
: C 0358 1
: C 0359 1
: C 0360 1
: C 0361 1
: C 0362 1

: C 0363 1
 : C 0364 1
 : C 0365 1
 : C 0366 1
 : C 0367 1
 : C 0368 1
 : C 0369 1
 : C 0370 1
 : C 0371 1
 : C 0372 1
 : C 0373 1
 : C 0374 1
 : C 0375 1
 : C 0376 1
 : C 0377 1
 : C 0378 1
 : C 0379 1
 : C 0380 1
 : C 0381 1
 : C 0382 1
 : C 0383 1
 : C 0384 1
 : C 0385 1
 : C 0386 1
 : C 0387 1
 : C 0388 1
 : C 0389 1
 : C 0390 1
 : C 0391 1
 : C 0392 1
 : C 0393 1
 : C 0394 1
 : C 0395 1
 : C 0396 1
 : C 0397 1
 : C 0398 1
 : C 0399 1
 : C 0400 1
 : C 0401 1
 : C 0402 1
 : C 0403 1
 : C 0404 1
 : C 0405 1
 : C 0406 1
 : C 0407 1
 : C 0408 1
 : C 0409 1
 : C 0410 1
 : C 0411 1
 : C 0412 1
 : C 0413 1

5. PERCENTAGE OF 'FIXED DISK' 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 RD51/52s (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 RD51/52s. The numbers associated with usage are adjusted internally by the program according to drive type and percentage.

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. REWRITE BLOCKS WHEN "FORCED ERROR" DETECTED ON READS (L) Y ?

On encountering a bad block on the RD51 or RD52 disk (during either a read or a write operation), the RQDX1 or RUX50 controller will revector the logical block to another physical location on the disk. This operation is transparent to the user. However, if the revectoring was done subsequent to a write operation (i.e. the write operation detected the bad block), the data is flagged with a "Forced Error" code, signifying that the data at the revector location is suspect. The controller returns an error code whenever the block is re-read. Answer 'Yes' to the question to force a WRITE operation on the same block whenever a "Forced Error" flag is detected on a read. This is to avoid the same error code (the "Forced Error") being reported for the same block repeatedly. The re-write will, however, take place only if writes are enabled for the particular disk unit.

8. HALT ON BAD BLOCK HARD ERRORS (0s 35, 38) (L) Y ?

When the Exerciser is run with the DRS "Halt on Error" switch set (eg. START/FLAGS:H0E), the Exerciser halts on encountering ANY error. If it is desired that the testing continue on a bad block error, even with the H0E switch set, answer No to the question.

9. HALT ON OTHER HARD ERRORS (0s 31, 34, 36, 37, 39, 45) (L) Y ?

This question is similar to question 8, but refers to non-bad block type of Hard Errors.

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0011
Page 11
(10)

; C 0414 1
; C 0415 1
; C 0416 1
; C 0417 1
; C 0418 1
; C 0419 1
; C 0420 1
; C 0421 1
; C 0422 1
; C 0423 1
; C 0424 1
; C 0425 1
; C 0426 1
; C 0427 1
; C 0428 1
; C 0429 1
; C 0430 1
; C 0431 1
; C 0432 1
; C 0433 1
; C 0434 1
; C 0435 1
; C 0436 1
; C 0437 1
; C 0438 1
; C 0439 1
; C 0440 1
; C 0441 1
; C 0442 1
; C 0443 1
; C 0444 1
; C 0445 1
; C 0446 1
; C 0447 1
; C 0448 1
; C 0449 1
; C 0450 1
; C 0451 1
; C 0452 1
; C 0453 1
; C 0454 1
; C 0455 1
; C 0456 1
; C 0457 1
; C 0458 1
; C 0459 1
; C 0460 1
; C 0461 1
; C 0462 1
; C 0463 1
; C 0464 1
; C 0465 1
; C 0466 1

10. HALT ON SOFT ERRORS (Qs 50-54) (L) N ?

This question is similar to question 8, but refers to Soft Errors.

11. COUNT EACH RETRY AS A SEPARATE SOFT ERROR (L) N ?

On encountering any error on a read/write, the controller retries the operation a number of times. If the operation is eventually successful, this is reported as a Soft Error. The error log packet contains the number of retries performed before the operation was successful. Normally, the whole sequence of retries is classified as one Soft Error. Answer yes to the question if it is desired to count each internal retry attempt as a separate Soft Error.

12. RANDOM SEEK MODE (L) Y ?

Answer YES to cause block numbers to be chosen randomly. Answer NO to cause block numbers to be selected sequentially.

13. UNITS TO BE SELECTED AT RANDOM (NO, IMPLIES SEQUENTIAL) (L) N ?

This question is optionally asked if the answer to the previous question is N[o]. The selection of units for sequential operations is affected by the answer to this question. If the default answer is chosen (N[o]), then units shall be selected in a predetermined manner in accordance with the typical seek time margins for each drive. If the alternate answer is chosen (Y[es]), then the units will be chosen at random in accordance with the percentages specified in Software question 4.

14. READ COMPARES PERFORMED AT THE CONTROLLER (L) Y ?

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.

15. 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.

M

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0012
Page 12
(11)

; C 0467 1
; C 0468 1
; C 0469 1
; C 0470 1
; C 0471 1
; C 0472 1
; C 0473 1
; C 0474 1
; C 0475 1
; C 0476 1
; C 0477 1
; C 0478 1
; C 0479 1
; C 0480 1
; C 0481 1
; C 0482 1
; C 0483 1
; C 0484 1
; C 0485 1
; C 0486 1
; C 0487 1
; C 0488 1
; C 0489 1
; C 0490 1
; C 0491 1
; C 0492 1
; C 0493 1
; C 0494 1
; C 0495 1
; C 0496 1
; C 0497 1
; C 0498 1

16. CHECK ALL WRITES AT HOST BY READING (L) Y ?

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 drive write-compare operations are suspect.

17. 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.

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

There are 21 pre-defined 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).

19. 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).

N1

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0013
Page 13
(12)

; C 0499 1
; C 0500 1
; C 0501 1
; C 0502 1
; C 0503 1
; C 0504 1
; C 0505 1
; C 0506 1
; C 0507 1
; C 0508 1
; C 0509 1
; C 0510 1
; C 0511 1
; C 0512 1
; C 0513 1
; C 0514 1
; C 0515 1
; C 0516 1
; C 0517 1
; C 0518 1
; C 0519 1
; C 0520 1
; C 0521 1
; C 0522 1
; C 0523 1
; C 0524 1
; C 0525 1
; C 0526 1
; C 0527 1
; C 0528 1
; C 0529 1
; C 0530 1
; C 0531 1
; C 0532 1
; C 0533 1
; C 0534 1
; C 0535 1
; C 0536 1
; C 0537 1
; C 0538 1
; C 0539 1
; C 0540 1
; C 0541 1
; C 0542 1
; C 0543 1
; C 0544 1
; C 0545 1

3.0 ERROR TYPES

This program has four types of error classifications;
system fatal, drive 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.

DRIVE FATAL ERRORS

Drive fatal errors are a result of:

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

HARD ERRORS

Hard errors are a result of:

1. retries of a soft error on *
2. a non-recoverable error
3. a soft error if retries 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.

3.1 ERROR INFORMATION

All general error messages will include the type of error (system-fatal, drive-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 drive-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.

Drive-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 drive present)
2. Vector Test failure
3. BR Level Test failure
4. Initialization sequence failure
5. Online failed
6. Access failed

: C 0595 1
: C 0596 1
: C 0597 1
: C 0598 1
: C 0599 1
: C 0600 1
: C 0601 1
: C 0602 1
: C 0603 1
: C 0604 1
: C 0605 1
: C 0606 1
: C 0607 1
: C 0608 1
: C 0609 1
: C 0610 1
: C 0611 1
: C 0612 1
: C 0613 1
: C 0614 1
: C 0615 1
: C 0616 1
: C 0617 1
: C 0618 1
: C 0619 1
: C 0620 1
: C 0621 1
: C 0622 1
: C 0623 1
: C 0624 1
: C 0625 1
: C 0626 1
: C 0627 1
: C 0628 1
: C 0629 1
: C 0630 1
: C 0631 1
: C 0632 1
: C 0633 1
: C 0634 1
: C 0635 1
: C 0636 1
: C 0637 1
: C 0638 1
: C 0639 1
: C 0640 1
: C 0641 1

3.5 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 controller. 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.

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:([POWERS])ZRQADO.BL 1:6

SEQ 0016
Page 16
(15)

: C 0642 1
: C 0643 1
: C 0644 1
: C 0645 1
: C 0646 1
: C 0647 1
: C 0648 1
: C 0649 1
: C 0650 1
: C 0651 1
: C 0652 1
: C 0653 1
: C 0654 1
: C 0655 1
: C 0656 1
: C 0657 1
: C 0658 1
: C 0659 1
: C 0660 1
: C 0661 1
: C 0662 1
: C 0663 1
: C 0664 1
: C 0665 1
: C 0666 1
: C 0667 1
: C 0668 1
: C 0669 1
: C 0670 1
: C 0671 1
: C 0672 1

3.6 SAMPLE 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
INVALID COMMAND	!MAJOR STATUS CODE RECEIVED BACK
SUB-CODE XXXX	!SUB-CODE OF GIVEN COMMAND
COMMAND: READ	!COMMAND GIVEN TO DRIVE
LBN: XXXXX	!LOGICAL BLOCK NUMBER GIVEN
BYTE COUNT IN COMMAND XXXXX	!NUMBER OF BYTES WANTED TO READ
ACTUAL # OF BYTES TRANSFERRED XXXXX	!NUMBER OF BYTES ACTUALLY READ

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.

E2

JRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1199-16 V4.0 579
DISK\$USER2:[POWERS]ZRQADC.BL1;6

SEQ 0017
Page 17
(16)

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.

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.

UNT	#	OF	BYTES	#	OF	BYTES	--	HRD	ERS	--	SFT	ERS	
#	TYPE	READS	READ	WRITES	WRITTEN	DAT	SEK	DRV	HST	DAT	SEK	DRV	HST
X	XXXX	XXXX	XXXX	XXXXXX	XXXXXXX	X	X	X	X	X	X	X	X
:	::	::	::	:::	::::	:	:	:	:	:	:	:	:
.	::	::	::	:::	::::	.	.	.	.	.	.	.	.

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bli99-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0018
Page 18
(17)

5.0 TEST SUMMARY

This 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 drive 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 drive-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.

JRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0019
Page 19
(18)

: C 0757 1
: C 0758 1
: C 0759 1
: C 0760 1
: C 0761 1
: C 0762 1
: C 0763 1
: C 0764 1
: C 0765 1
: C 0766 1
: C 0767 1
: C 0768 1
: C 0769 1
: C 0770 1
: C 0771 1
: C 0772 1
: C 0773 1
: C 0774 1
: C 0775 1
: C 0776 1
: C 0777 1
: C 0778 1
: C 0779 1
: C 0780 1
: C 0781 1
: C 0782 1
: C 0783 1
: C 0784 1
: C 0785 1

If a read/write error occurs during this test, then the controller will initiate an appropriate number of re-tries. 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.

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 drive 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.

ZRGAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGAD0.BL1;6SEQ 0020
Page 20
(19)

6.0 ERROR CODES GENERATED BY THIS EXERCISER

SYSTEM FATAL ERRORS

1 More than 4 units specified

DRIVE 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 either the inner or the outer track failed. Innermost or outermost track's header may be corrupted.

18 Unit went off line.

19 Drive type not known. The version of the Exerciser being run does not support this disk type.

; C 0786 1
; C 0787 1
; C 0788 1
; C 0789 1
; C 0790 1
; C 0791 1
; C 0792 1
; C 0793 1
; C 0794 1
; C 0795 1
; C 0796 1
; C 0797 1
; C 0798 1
; C 0799 1
; C 0800 1
; C 0801 1
; C 0802 1
; C 0803 1
; C 0804 1
; C 0805 1
; C 0806 1
; C 0807 1
; C 0808 1
; C 0809 1
; C 0810 1
; C 0811 1
; C 0812 1
; C 0813 1
; C 0814 1
; C 0815 1
; C 0816 1
; C 0817 1
; C 0818 1
; C 0819 1
; C 0820 1
; C 0821 1
; C 0822 1
; C 0823 1
; C 0824 1
; C 0825 1
; C 0826 1
; C 0827 1
; C 0828 1
; C 0829 1
; C 0830 1
; C 0831 1
; C 0832 1
; C 0833 1
; C 0834 1

TP

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0021
Page 21
(20)

: C 0835	1	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 0836	1			
: C 0837	1			
: C 0838	1			
: C 0839	1			
: C 0840	1	21	Controller returned wrong 'end code' for the 'Set Controller Characteristics' command.	Problem with the Controller microcode or the port/DMA interface.
: C 0841	1			
: C 0842	1			
: C 0843	1			
: C 0844	1	22	Failed to send 'On-line' command	Either the unit is off-line or the diagnostic is corrupted because of any problems with its RAM.
: C 0845	1			
: C 0846	1			
: C 0847	1			
: C 0848	1			
: C 0849	1	23	Controller returned wrong 'end code' for the 'On-line' command.	Problem with the Controller's microcode or the port/DMA interface.
: C 0850	1			
: C 0851	1			
: C 0852	1			
: C 0853	1	24	Drive went to the 'Available' state.	---
: C 0854	1			
: C 0855	1			
: C 0856	1		HARD ERRORS	
: C 0857	1		-----	
: C 0858	1			
: C 0859	1	31	Controller received an invalid command.	The diagnostic is corrupted because of any problems with its RAM, or there is a problem with the Controller microcode (RAM or ROM) or there is problem with the port/DMA interface.
: C 0860	1			
: C 0861	1			
: C 0862	1			
: C 0863	1			
: C 0864	1			
: C 0865	1			
: C 0866	1			
: C 0867	1			
: C 0868	1	32	Command aborted by the Controller.	Command timed-out in the Controller.
: C 0869	1			
: C 0870	1			
: C 0871	1	35	Media format error.	---
: C 0872	1			
: C 0873	1	36	Drive write protected.	---
: C 0874	1			
: C 0875	1	37	Controller read or write compare error.	---
: C 0876	1			
: C 0877	1			
: C 0878	1	38	Data error.	CRC error in the data field of a disk block.
: C 0879	1			
: C 0880	1			
: C 0881	1	39	Host buffer access error	---
: C 0882	1			
: C 0883	1	40	Controller error.	Difficult to categorize without looking at the error sub-code or any associated error-log message.
: C 0884	1			
: C 0885	1			
: C 0886	1			
: C 0887	1			

ZRQAM1
V01.6

RD/RX EXERCISER

JP

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0022
Page 22
(21)

: C 0888	1	41 Drive error.	See #40.
: C 0889	1		
: C 0890	1	42 Host write compare error.	Error detected when Host
: C 0891	1		CPU compared the data
: C 0892	1		written and read back. May
: C 0893	1		be a problem with the Host
: C 0894	1		or Controller RAM.
: C 0895	1		
: C 0896	1	43 Message from internal diagnostics	See #40.
: C 0897	1		
: C 0898	1	44 Duplicate unit number detected	---
: C 0899	1	by the Controller.	
: C 0900	1		
: C 0901	1	45 Unknown end code received.	Problem with the Control-
: C 0902	1		ler microcode or the port/
: C 0903	1		DMA interface.
: C 0904	1		
: C 0905	1		
: C 0906	1	SOFT ERRORS	
: C 0907	1	-----	
: C 0908	1		
: C 0909	1	50 Controller error.	See error-log packet for
: C 0910	1		details as the exact cause
: C 0911	1		may not be evident.
: C 0912	1		
: C 0913	1	51 Host memory access error.	See #50.
: C 0914	1		
: C 0915	1	52 Disk transfer error.	See #50.
: C 0916	1		
: C 0917	1	53 'Standard Disk Interconnect'	See #50.
: C 0918	1	error.	
: C 0919	1		
: C 0920	1	54 'Small Disk' error.	See #50.

K2

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0023
Page 23
(22)

	7.0	DATA PATTERNS		
		HEX	OCTAL	BINARY
		---	-----	-----
: C 0921 1				
: C 0922 1				
: C 0923 1				
: C 0924 1				
: C 0925 1				
: C 0926 1				
: C 0927 1				
: C 0928 1	Pattern 1		R A N D O M	N U M B E R S
: C 0929 1				
: C 0930 1	Pattern 2	0000	000000	0 000 000 000 000 000
: C 0931 1				
: C 0932 1	Pattern 3	FFFF	177777	1 111 111 111 111 111
: C 0933 1				
: C 0934 1	Pattern 4	8B8B	105613	1 000 101 110 001 011
: C 0935 1				
: C 0936 1	Pattern 5	3333	031463	0 011 001 100 110 011
: C 0937 1				
: C 0938 1	Pattern 6	3091	030221	0 011 000 010 010 001
: C 0939 1				
: C 0940 1	Pattern 7	0001	000001	0 000 000 000 000 001
: C 0941 1		0003	000003	0 000 000 000 000 011
: C 0942 1		0007	000007	0 000 000 000 000 111
: C 0943 1		000F	000017	0 000 000 000 001 111
: C 0944 1		001F	000037	0 000 000 000 011 111
: C 0945 1		003F	000077	0 000 000 000 111 111
: C 0946 1		007F	000177	0 000 000 001 111 111
: C 0947 1		00FF	000377	0 000 000 011 111 111
: C 0948 1		01FF	000777	0 000 000 111 111 111
: C 0949 1		03FF	001777	0 000 001 111 111 111
: C 0950 1		07FF	003777	0 000 011 111 111 111
: C 0951 1		0FFF	007777	0 000 111 111 111 111
: C 0952 1		1FFF	017777	0 001 111 111 111 111
: C 0953 1		3FFF	037777	0 011 111 111 111 111
: C 0954 1		7FFF	077777	0 111 111 111 111 111
: C 0955 1		FFFF	177777	1 111 111 111 111 111
: C 0956 1				
: C 0957 1	Pattern 8	FFFE	177776	1 111 111 111 111 110
: C 0958 1		FFFC	177774	1 111 111 111 111 100
: C 0959 1		FFF8	177770	1 111 111 111 111 000
: C 0960 1		FFF0	177760	1 111 111 111 110 000
: C 0961 1		FFE0	177740	1 111 111 111 100 000
: C 0962 1		FFC0	177700	1 111 111 111 000 000
: C 0963 1		FF80	177600	1 111 111 110 000 000
: C 0964 1		FF00	177400	1 111 111 100 000 000
: C 0965 1		FE00	177000	1 111 111 000 000 000
: C 0966 1		FC00	176000	1 111 110 000 000 000
: C 0967 1		F800	174000	1 111 100 000 000 000
: C 0968 1		F000	170000	1 111 000 000 000 000
: C 0969 1		E000	160000	1 110 000 000 000 000
: C 0970 1		C000	140000	1 100 000 000 000 000
: C 0971 1		8000	100000	1 000 000 000 000 000
: C 0972 1		0000	000000	0 000 000 000 000 000

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0024
Page 24
(23)

: C 0973	1	Pattern 9	0000	000000	0 000	000	000	000	000
: C 0974	1		0000	000000	0 000	000	000	000	000
: C 0975	1		0000	000000	0 000	000	000	000	000
: C 0976	1		FFFF	177777	1 111	111	111	111	111
: C 0977	1		FFFF	177777	1 111	111	111	111	111
: C 0978	1		FFFF	177777	1 111	111	111	111	111
: C 0979	1		0000	000000	0 000	000	000	000	000
: C 0980	1		0000	000000	0 000	000	000	000	000
: C 0981	1		FFFF	177777	1 111	111	111	111	111
: C 0982	1		FFFF	177777	1 111	111	111	111	111
: C 0983	1		0000	000000	0 000	000	000	000	000
: C 0984	1		FFFF	177777	1 111	111	111	111	111
: C 0985	1		0000	000000	0 000	000	000	000	000
: C 0986	1		FFFF	177777	1 111	111	111	111	111
: C 0987	1		0000	000000	0 000	000	000	000	000
: C 0988	1		FFFF	177777	1 111	111	111	111	111
: C 0989	1								
: C 0990	1	Pattern 10	B6D9	133331	1 011	011	011	011	001
: C 0991	1								
: C 0992	1	Pattern 11	5555	052525	0 101	010	101	010	101
: C 0993	1		5555	052525	0 101	010	101	010	101
: C 0994	1		5555	052525	0 101	010	101	010	101
: C 0995	1		AAAA	125252	1 010	101	010	101	010
: C 0996	1		AAAA	125252	1 010	101	010	101	010
: C 0997	1		AAAA	125252	1 010	101	010	101	010
: C 0998	1		5555	052525	0 101	010	101	010	101
: C 0999	1		5555	052525	0 101	010	101	010	101
: C 1000	1		AAAA	125252	1 010	101	010	101	010
: C 1001	1		AAAA	125252	1 010	101	010	101	010
: C 1002	1		5555	052525	0 101	010	101	010	101
: C 1003	1		AAAA	125252	1 010	101	010	101	010
: C 1004	1		5555	052525	0 101	010	101	010	101
: C 1005	1		AAAA	125252	1 010	101	010	101	010
: C 1006	1		5555	052525	0 101	010	101	010	101
: C 1007	1		AAAA	125252	1 010	101	010	101	010

M2

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr 1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6SEQ 0025
Page 25
(24)

: C 1008 1	Pattern 12	2D2D	026455	0 010 110 100 101 101
: C 1009 1		2D2D	026455	0 010 110 100 101 101
: C 1010 1		2D2D	026455	0 010 110 100 101 101
: C 1011 1		D2D2	151322	1 101 001 011 010 010
: C 1012 1		D2D2	151322	1 101 001 011 010 010
: C 1013 1		D2D2	151322	1 101 001 011 010 010
: C 1014 1		2D2D	026455	0 010 110 100 101 101
: C 1015 1		2D2D	026455	0 010 110 100 101 101
: C 1016 1		D2D2	151322	1 101 001 011 010 010
: C 1017 1		D2D2	151322	1 101 001 011 010 010
: C 1018 1		2D2D	026455	0 010 110 100 101 101
: C 1019 1		2D2D	026455	0 010 110 100 101 101
: C 1020 1		D2D2	151322	1 101 001 011 010 010
: C 1021 1		2D2D	026455	0 010 110 100 101 101
: C 1022 1		D2D2	151322	1 101 001 011 010 010
: C 1023 1		2D2D	026455	0 010 110 100 101 101
: C 1024 1		D2D2	151322	1 101 001 011 010 010
: C 1025 1		2D2D	026455	0 010 110 100 101 101
: C 1026 1		D2D2	151322	1 101 001 011 010 010
: C 1027 1		2D2D	026455	0 010 110 100 101 101
: C 1028 1				
: C 1029 1	Pattern 13	6DB6	066666	0 110 110 110 110 110
: C 1030 1				
: C 1031 1	Pattern 14	0001	000001	0 000 000 000 000 001
: C 1032 1		0002	000002	0 000 000 000 000 010
: C 1033 1		0004	000004	0 000 000 000 000 100
: C 1034 1		0008	000010	0 000 000 000 001 000
: C 1035 1		0010	000020	0 000 000 000 010 000
: C 1036 1		0020	000040	0 000 000 000 100 000
: C 1037 1		0040	000100	0 000 000 001 000 000
: C 1038 1		0080	000200	0 000 000 010 000 000
: C 1039 1		0100	000400	0 000 000 100 000 000
: C 1040 1		0200	001000	0 000 001 000 000 000
: C 1041 1		0400	002000	0 000 010 000 000 000
: C 1042 1		0800	004000	0 000 100 000 000 000
: C 1043 1		1000	010000	0 001 000 000 000 000
: C 1044 1		2000	020000	0 010 000 000 000 000
: C 1045 1		4000	040000	0 100 000 000 000 000
: C 1046 1		8000	100000	1 000 000 000 000 000

N?

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0026
Page 26
(25)

: C 1047	1	Pattern 15	FFFE	177776	1 111 111 111 111 110
: C 1048	1		FFFD	177775	1 111 111 111 111 101
: C 1049	1		FFFB	177773	1 111 111 111 111 011
: C 1050	1		FFF7	177767	1 111 111 111 110 111
: C 1051	1		FFEF	177757	1 111 111 111 101 111
: C 1052	1		FFDF	177737	1 111 111 111 011 111
: C 1053	1		FFBF	177677	1 111 111 110 111 111
: C 1054	1		FF7F	177577	1 111 111 101 111 111
: C 1055	1		FEFF	177377	1 111 111 011 111 111
: C 1056	1		FCFF	176777	1 111 110 111 111 111
: C 1057	1		FBFF	175777	1 111 101 111 111 111
: C 1058	1		F7FF	173777	1 111 011 111 111 111
: C 1059	1		FFFF	167777	1 110 111 111 111 111
: C 1060	1		DFFF	157777	1 101 111 111 111 111
: C 1061	1		BFFF	137777	1 011 111 111 111 111
: C 1062	1		7FFF	077777	0 111 111 111 111 111
: C 1063	1				
: C 1064	1	Pattern 16	B6D9	133331	1 011 011 011 011 001
: C 1065	1		B6D9	133331	1 011 011 011 011 001
: C 1066	1		B6D9	133331	1 011 011 011 011 001
: C 1067	1		DB6C	155554	1 101 101 101 101 100
: C 1068	1		DB6C	155554	1 101 101 101 101 100
: C 1069	1		DB6C	155554	1 101 101 101 101 100
: C 1070	1		B6D9	133331	1 011 011 011 011 001
: C 1071	1		B6D9	133331	1 011 011 011 011 001
: C 1072	1		DB6C	155554	1 101 101 101 101 100
: C 1073	1		DB6C	155554	1 101 101 101 101 100
: C 1074	1		B6D9	133331	1 011 011 011 011 001
: C 1075	1		DB6C	155554	1 101 101 101 101 100
: C 1076	1		B6D9	133331	1 011 011 011 011 001
: C 1077	1		DB6C	155554	1 101 101 101 101 100
: C 1078	1		B6D9	133331	1 011 011 011 011 001
: C 1079	1		DB6C	155554	1 101 101 101 101 100

{ 3 }

ZRQAM1
V01.6

RD/RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Release 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 002 /
Page 27
(26)

	Pattern 17	(LBN)*	(LBN)	(LBN)
: C 1080 1		8D36	106466	1 000 110 100 110 110
: C 1081 1		8D36	106466	1 000 110 100 110 110
: C 1082 1		72C9	071311	0 111 001 011 001 001
: C 1083 1		72C9	071311	0 111 001 011 001 001
: C 1084 1		72C9	071311	0 111 001 011 001 001
: C 1085 1		8D36	106466	1 000 110 100 110 110
: C 1086 1		8D36	106466	1 000 110 100 110 110
: C 1087 1		8D36	106466	1 000 110 100 110 110
: C 1088 1		8D36	106466	1 000 110 100 110 110
: C 1089 1		72C9	071311	0 111 001 011 001 001
: C 1090 1		72C9	071311	0 111 001 011 001 001
: C 1091 1		72C9	071311	0 111 001 011 001 001
: C 1092 1		72C9	071311	0 111 001 011 001 001
: C 1093 1		72C9	071311	0 111 001 011 001 001
: C 1094 1		72C9	071311	0 111 001 011 001 001
: C 1095 1		8D36	106466	1 000 110 100 110 110
: C 1096 1		8D36	106466	1 000 110 100 110 110
: C 1097 1		8D36	106466	1 000 110 100 110 110
: C 1098 1		8D36	106466	1 000 110 100 110 110
: C 1099 1		8D36	106466	1 000 110 100 110 110
: C 1100 1		8D36	106466	1 000 110 100 110 110

* This word position contains the number of the logical block to be written.

	Pattern 18	(LBN)	(LBN)	(LBN)
: C 1107 1		8D36	106466	1 000 110 100 110 110
: C 1108 1		72C9	071311	0 111 001 011 001 001
: C 1109 1		8D36	106466	1 000 110 100 110 110
: C 1110 1		8D36	106466	1 000 110 100 110 110
: C 1111 1		8D36	106466	1 000 110 100 110 110
: C 1112 1		72C9	071311	0 111 001 011 001 001
: C 1113 1		72C9	071311	0 111 001 011 001 001
: C 1114 1		72C9	071311	0 111 001 011 001 001
: C 1115 1		72C9	071311	0 111 001 011 001 001
: C 1116 1		72C9	071311	0 111 001 011 001 001
: C 1117 1		8D36	106466	1 000 110 100 110 110
: C 1118 1		8D36	106466	1 000 110 100 110 110
: C 1119 1		8D36	106466	1 000 110 100 110 110
: C 1120 1		8D36	106466	1 000 110 100 110 110
: C 1121 1		72C9	071311	0 111 001 011 001 001
: C 1122 1		72C9	071311	0 111 001 011 001 001
: C 1123 1		72C9	071311	0 111 001 011 001 001
: C 1124 1		72C9	071311	0 111 001 011 001 001
: C 1125 1		72C9	071311	0 111 001 011 001 001
: C 1126 1		72C9	071311	0 111 001 011 001 001
: C 1127 1		72C9	071311	0 111 001 011 001 001

C3

ZRQADM
VOL.6

RD-RX EXERCISER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1195 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL 1:6

SEQ 0028
Page 28
(27)

: C 1128 1	Pattern 19	(LBN)	(LBN)	(LBN)	
: C 1129 1		8999	134631	1 011 100 110 011 001	
: C 1130 1		8999	134631	1 011 100 110 011 001	
: C 1131 1		4666	043146	0 100 011 001 100 110	
: C 1132 1		4666	043146	0 100 011 001 100 110	
: C 1133 1		4666	043146	0 100 011 001 100 110	
: C 1134 1		8999	134631	1 011 100 110 011 001	
: C 1135 1		8999	134631	1 011 100 110 011 001	
: C 1136 1		8999	134631	1 011 100 110 011 001	
: C 1137 1		8999	134631	1 011 100 110 011 001	
: C 1138 1		4666	043146	0 100 011 001 100 110	
: C 1139 1		4666	043146	0 100 011 001 100 110	
: C 1140 1		4666	043146	0 100 011 001 100 110	
: C 1141 1		4666	043146	0 100 011 001 100 110	
: C 1142 1		4666	043146	0 100 011 001 100 110	
: C 1143 1		8999	134631	1 011 100 110 011 001	
: C 1144 1		8999	134631	1 011 100 110 011 001	
: C 1145 1		8999	134631	1 011 100 110 011 001	
: C 1146 1		8999	134631	1 011 100 110 011 001	
: C 1147 1		8999	134631	1 011 100 110 011 001	
: C 1148 1		8999	134631	1 011 100 110 011 001	
: C 1149 1				1 011 100 110 011 001	
: C 1150 1	Pattern 20	8999	134631	1 011 100 110 011 001	
: C 1151 1		(LBN)	(LBN)	(LBN)	
: C 1152 1		4666	043146	0 100 011 001 100 110	
: C 1153 1		8999	134631	1 011 100 110 011 001	
: C 1154 1		8999	134631	1 011 100 110 011 001	
: C 1155 1		8999	134631	1 011 100 110 011 001	
: C 1156 1		4666	043146	0 100 011 001 100 110	
: C 1157 1		4666	043146	0 100 011 001 100 110	
: C 1158 1		4666	043146	0 100 011 001 100 110	
: C 1159 1		4666	043146	0 100 011 001 100 110	
: C 1160 1		8999	134631	1 011 100 110 011 001	
: C 1161 1		8999	134631	1 011 100 110 011 001	
: C 1162 1		8999	134631	1 011 100 110 011 001	
: C 1163 1		8999	134631	1 011 100 110 011 001	
: C 1164 1		8999	134631	1 011 100 110 011 001	
: C 1165 1		4666	043146	0 100 011 001 100 110	
: C 1166 1		4666	043146	0 100 011 001 100 110	
: C 1167 1		4666	043146	0 100 011 001 100 110	
: C 1168 1		4666	043146	0 100 011 001 100 110	
: C 1169 1		4666	043146	0 100 011 001 100 110	
: C 1170 1		4666	043146	0 100 011 001 100 110	
: C 1171 1				0 100 011 001 100 110	
: C 1172 1	Pattern 21	(LBN)	(LBN)	(LBN)	
: C 1173 1					
: C 1174 1					

)

13

ZRQAM1
V01.6

RDRX EXERCISER
PROGRAM HEADER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1199 16 V4.0-579
DISK\$USER2:([POWERS])ZRQADO,BL116

SEQ 0029
Page 29
(28)

```

: 1175 1 *sbttl 'PROGRAM HEADER'
: 1176 1
: 1177 1 library 'ZRQADO.L16';
: 1178 1
: 1179 1 require 'BLSMAC.REU';
: 2670 1
: 2671 1 literal
: 2672 1     DS$NBR OF TESTS = 1;
: 2673 1
: 2674 1 EQUALS;
: 2675 1
: 2676 1 POINTER (ALL);
: 2677 1
: 2678 1 !
: 2679 1 ! THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: 2680 1 ! THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
: 2681 1 !
: 2682 1
: 2683 1 HEADER (%ascii'ZRQA , %ascii'D', %ascii'O', 32767, 1, PRI00);

```

```

! RDRX EXERCISER GLOBAL LIBRARY
! DIAGNOSTIC SUPERVISOR LIBRARY
! NUMBER OF TESTS IN THIS DIAGNOSTIC

```

E-3

ZRQAM1
V01.6

RD/RX EXERCISER
DISPATCH TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0030
Page 30
(29)

```
: 2684 1 *sbttl 'DISPATCH TABLE'
: 2685 1
: 2686 1 !
: 2687 1 ! THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
: 2688 1 ! IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
: 2689 1 !
: 2690 1
: 2691 1 DISPATCH (DS$NBR_OF_TESTS);
```


ZRQAM1
V01.6

RD/RX EXERCISER
GLOBAL DATA SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWER5]ZRQADO.BL1;6

SEQ 0031
Page 31
(30)

```

: 2692 1 *sbttl 'GLOBAL DATA SECTION'
: 2693 1
: 2694 1
: 2695 1 !
: 2696 1 ! THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
: 2697 1 ! IN MORE THAN ONE TEST.
: 2698 1 !
: 2699 1
: 2700 1 psect
: 2701 1     global = $FFF$ (read, write, noexecute, global, concatenate);
: 2702 1
: 2703 1 global
: 2704 1     CST : blockvector [MAX_CTLR, CST_LEN, word] field (CST_FIELDS),
: 2705 1             ! RUN-TIME CONTROLLER STATUS TABLES
: 2706 1     CST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 2707 1             ! CONTROLLER STATUS TABLE ADDRESS OF "CURRENT" CONTROLLER
: 2708 1     DCT : blockvector [MAX_CTLR, DCT_LEN, word] field (DCT_FIELDS),
: 2709 1             ! DRIVER CONTROLLER TABLES
: 2710 1     DCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 2711 1             ! ADDRESS OF "CURRENT" DRIVER CONTROLLER TABLE
: 2712 1     RDRX_ADDR : ref rdx field (RC_REG),
: 2713 1             ! DEVICE ADDRESS OF "CURRENT" CONTROLLER
: 2714 1     IRDRX_ADDR : ref rdx field (RC_REG),
: 2715 1             ! DEVICE ADDRESS OF INTERRUPTING CONTROLLER
: 2716 1
: 2717 1     BST : BLOCKVECTOR [MAX_UNITS, 2, WORD],
: 2718 1             !CONTAINS LO, HI LBN FIELDS FOR SEQUENTIAL
: 2719 1             !I/O TRANSFER FOR EACH UNIT.
: 2720 1     TALLY : vector [MAX_UNITS * TALLY_LEN, word] field (T_FIELDS),
: 2721 1             ! STATISTICS TABLES
: 2722 1     T_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 2723 1             ! ADDRESS OF STATISTICS TABLE (TALLY) FOR CURRENT UNIT
: 2724 1
: 2725 1     DUPPKT : BLOCK [257, WORD] FIELD (DP_FIELDS),
: 2726 1             !INFO FROM RECEIVE + SEND CMDS
: 2727 1     RK_SGN : VECTOR [MAX_UNITS, BYTE, SIGNED] INITIAL (BYTE (REP
: 2728 1             MAX_UNITS OF (1))),
: 2729 1             !CURRENT TRACK DIRECTION
: 2730 1     RDM_CNT : WORD INITIAL (RDM_LEN),
: 2731 1             !NO OF RANDOM NOS
: 2732 1     RANDOM : VECTOR [RDM_LEN, WORD],
: 2733 1             !RANDOM NO. TABLE
: 2734 1             //TOGETHER
: 2735 1
: 2736 1     C_ERR_TBL : blockvector [MAX_CTLR, C_ERR_LEN, word] field (C_ERR_FIELDS),
: 2737 1             ! STATISTICS TABLE FOR CONTROLLER ERRORS
: 2738 1     MSCP_PKT : blockvector [PKT_CNT, PKT_LEN, word] field (PKT_FIELDS),
: 2739 1             ! MSCP PACKET POOL
: 2740 1     IPKT_ADDR : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 2741 1             ! ADDRESS OF AN MSCP PACKET (INTERUPT PROCESSING)
: 2742 1     PKT_USE : vector [PKT_CNT, byte, signed],
: 2743 1             ! MSCP PACKET POOL ALLOCATION TABLE
: 2744 1     RETPKT : blockvector [RP_CNT, RP_LEN, word] field (RP_FIELDS),
: 2745 1             ! RETURN PACKET POOL
: 2746 1     RP_USE : vector [RP_CNT, byte, signed],
: 2747 1             ! RETURN PACKET POOL ALLOCATION TABLE
: 2748 1     RP_IDX : word,
: 2749 1             ! CURRENT RETURN PACKET INDEX
: 2750 1     RP_ADDR : ref block [RP_LEN, word] field (RP_FIELDS),

```

ZRGAM1
VOL.6RD/RX EXERCISER
GLOBAL DATA SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX 11 Bliss-16 V4.0 579
DISK\$USER2:[POWERS]ZRGADO.BL1;6SEQ 0032
Page 32
(30)

```

: 2745 1      ! CURRENT RETURN PACKET ADDRESS
: 2746 1      ELOG_PKT : blockvector [EP_CNT + 1, EP_LEN, word] field (EP_FIELDS),
: 2747 1      ! ERROR-LOG PACKET SAVE AREA
: 2748 1      BUFF_ADDR : vector [MAX_BUF_CNT],
: 2749 1      BUFF_OWN : vector [MAX_BUF_CNT, byte, signed],
: 2750 1      IODQ : vector [IODQ_LEN, byte],
: 2751 1      IODQ_IN : word,
: 2752 1      IODQ_OUT : word,
: 2753 1      ENTRY_REASON : byte,
: 2754 1      EOP_FLAG : byte,
: 2755 1      DUP_FLAGS : word,
: 2756 1      CCTLR : word,
: 2757 1      CDISK : word,
: 2758 1      CUOFF : word,
: 2759 1      CTLR_CNT : word,
: 2760 1      DUR : vector [MAX_UNITS, byte],
: 2761 1      QIO : vector [MAX_CTLR, byte],
: 2762 1      FREE_MEM_ADDR,
: 2763 1      BYTS_PER_QIO : word,
: 2764 1      ST_CODE : word,
: 2765 1      SB_CODE : word,
: 2766 1      STEP : word,
: 2767 1      OF_RC : signed word,
: 2768 1      SA_REG : word,
: 2769 1      CMD_TIME : word,
: 2770 1      NEX : word,
: 2771 1      CRN_LOW : word,
: 2772 1      CRN_HIGH : word,
: 2773 1      CREDIT_BAL : word,
: 2774 1      NEXT_PKT_USE : byte,
: 2775 1      HOURS : byte,
: 2776 1      MINUTES : byte,
: 2777 1      CLK_TICKS : word,
: 2778 1      CLK_PRESENT : byte,
: 2779 1      HOE_FLAG : byte,
: 2780 1
: 2781 1      S_PATTERN : word,
: 2782 1      S_DUPPKT : word,
: 2783 1      P_INDEX : signed word,
: 2784 1      FORCED_ERROR : byte,
: 2785 1      FER_LBN : word,
: 2786 1      FER_BC : word,
: 2787 1      INIT_OCCURED : byte initial (byte (FALSE)),
: 2788 1      ADDR_VECT_OK : byte initial (byte (FALSE));
: 2789 1
: 2790 1      ERR_TBL;

! TABLE OF I/O BUFFER DESCRIPTORS
! I/O BUFFER OWNERSHIP (CONTROLLER NUMBER)
! I/O DONE QUEUE - CIRCULAR QUEUE OF RETPKT INDECS
! I/O DONE QUEUE IN POINTER
! I/O DONE QUEUE OUT POINTER
! CURRENT OPERATOR COMMAND
! END-OF-PASS FLAG
! DUP FLAGS
! ZZZ
! NUMBER OF "CURRENT" CONTROLLER
! CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
! CURRENT UNIT CST OFFSET
! TOTAL NUMBER OF CONFIGURED CONTROLLERS
! DROP UNIT REASON
! NUMBER OF OUTSTANDING QIOs PER CONTROLLER
! START OF FREE MEMORY
! SIZE (BYTES) OF AN I/O BUFFER
! CURRENT STATUS CODE
! CURRENT SUB CODE
! CURRENT STEP IN HARD INIT
! OFFSET (0 OR 2) TO READ IP OR SA
! STORAGE FOR SA REGISTER READS AND WRITES
! COMMAND TIMEOUT VALUE (IN SECONDS)
! NON-EXISTENT MEMORY TRAP INDICATOR
! COMMAND REF NUMBER OF LAST COMMAND SENT
! COMMAND REF NUMBER (HI ORDER)
! CREDIT BALANCE
! POINTER TO NEXT ENTRY IN PKT_USE TABLE
! TIME OF DAY (HOURS)
! TIME OF DAY (MINUTES)
! TIME OF DAY (LINE-CLOCK TICKS)
! FLAG INDICATES IF LINE-CLOCK PRESENT
! FLAG INDICATES IF "HALT ON ERROR" FLAG SET
! PATTERN FOR DUP WRITES
! ZZZ
! DBN BYTE COUNTER
! ZZZ
! CURRENT MESSAGE PACKET INDEX
! ZZZ
! "FORCED ERROR" DETECTED IN LAST READ
! LBN OF THE "FORCED ERROR" BLOCK
! BYTE COUNT OF THE "FORCED ERROR" BLOCK
! EXERCISER INITIALIZATION COMPLETE
! FLAG INDICATES IF ADDRESS/VECTOR TEST PASSED

```


ZRGAM1
VOL.6RD/RX EXERCISER
GLOBAL TEXT SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL1;6SEQ 0033
Page 33
(31)

```

: 2791 1 *sbttl 'GLOBAL TEXT SECTION'
: 2792 1
: 2793 1
: 2794 1 ! THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: 2795 1 ! MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: 2796 1 ! MORE THAN ONE TEST.
: 2797 1
: 2798 1
: 2799 1 global bind
: 2800 1
: 2801 1 ! HARDWARE DIALOG
: 2802 1
: 2803 1 HWQ1 = uplit (*asciz'IP address'),
: 2804 1 HWQ2 = uplit (*asciz'Vector'),
: 2805 1 HWQ3 = uplit (*asciz'BR level'),
: 2806 1 HWQ4 = uplit (*asciz'Drive number'), !ZZZ
: 2807 1 HWQ5 = uplit (*asciz'Test entire customer area of this disk'), !ZZZ
: 2808 1 HWQ6A = uplit (*asciz'Lower octal word of beginning LBN address'), !ZZZ
: 2809 1 HWQ6B = uplit (*asciz'Higher octal word of beginning LBN address'), !ZZZ
: 2810 1 HWQ7A = uplit (*asciz'Lower octal word of ending LBN address'), !ZZZ
: 2811 1 HWQ7B = uplit (*asciz'Higher octal word of ending LBN address'), !ZZZ
: 2812 1 HWQ8 = uplit (*asciz'Write on customer data area on this disk'),
: 2813 1 HWQ9 = uplit (*asciz'** WARNING - CUSTOMER DATA AREA MAY BE OVERWRITTEN! ... CONFIRM'),
: 2814 1 HWQ10 = uplit (*asciz'Also run DUP exerciser'), !ZZZ
: 2815 1 HWQ11 = uplit (*asciz'Write on diagnostic area'), !ZZZ
: 2816 1
: 2817 1 ! SOFTWARE DIALOG
: 2818 1
: 2819 1 SWQ1 = uplit (*asciz'Hard Error limit'),
: 2820 1 SWQ2 = uplit (*asciz'Transfer limit in megabytes (0 for "Quick pass")'),
: 2821 1 SWQ4 = uplit (*asciz'Random seek mode'),
: 2822 1 SWQ7 = uplit (*asciz'Read-compares performed at the Controller'),
: 2823 1 SWQ9 = uplit (*asciz'Write-compares performed at the Controller'),
: 2824 1 SWQ10 = uplit (*asciz'Check all Writes at least by reading'),
: 2825 1 SWQ11 = uplit (*asciz'User-defined data pattern'),
: 2826 1 SWQ12 = uplit (*asciz'Select pre-defined data pattern (0 for Sequential selection)'),
: 2827 1 SWQ13 = uplit (*asciz'Number of words in data pattern (16 maximum)'),
: 2828 1 SWQ14 = uplit (*asciz'Pattern value (No leading zeros allowed)'),
: 2829 1 SWQ15 = uplit (*asciz'Clear statistical tables after printing'),
: 2830 1 SWQ17 = uplit (*asciz'Percentage of "Fixed Disk" operations out of total operations'),
: 2831 1 SWQ19 = uplit (*asciz'Units to be selected at random (No, implies sequential)'),
: 2832 1 SWQ20 = uplit (*asciz'Want to rewrite blocks when "Forced Error" detected on reads'),
: 2833 1 SWQ21 = uplit (*asciz'Do you want to halt on other Hard Errors (0s 31-34, 36-37, 39-45)'),
: 2834 1 SWQ22 = uplit (*asciz'Do you want to halt on Soft Errors (0s 50-54)'),
: 2835 1 SWQ23 = uplit (*asciz'Do you want to halt on Bad-block Hard Errors (0s 35, 38)'),
: 2836 1 SWQ24 = uplit (*asciz'Enter time as HHMM (Example: 1305 - No leading zeros allowed)'),
: 2837 1 SWQ25 = uplit (*asciz'Count each retry on a Read/Write error as a separate Soft Error'),
: 2838 1 SWM1 = uplit (*asciz'The remaining questions only apply to unprotected disks'),
: 2839 1 NULL = uplit (*asciz'),
: 2840 1
: 2841 1
: 2842 1 ! THE FOLLOWING DBMs ARE DEBUG MESSAGES, AND SHOULD BE REMOVED BEFORE
: 2843 1 ! RELEASING THE PROGRAM. THEY INCLUDE THE NAMES OF EACH ROUTINE, PLUS

```

ZRDAMI
V01.6RD/RX EXERCISER
GLOBAL TEXT SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRDADO.BL1;6SEQ 0034
Page 34
(31)

```

: 2844 1 ! FORMAT STATEMENTS FOR PRINTING OUT OTHER INFORMATION.
: 2845 1 !
: 2846 1
: 2847 1 DBM5 = uplit (%asciz' %N% A** Drop unit #D2'),
: 2848 1 DBM12 = uplit (%asciz' %N% A** PROC_RETPKT: Conn ID #06#A received'),
: 2849 1 DBM15 = uplit (%asciz' %N% A** Multi-drive test'),
: 2850 1 DBM18 = uplit (%asciz' %N% A** FATAL_ERROR: RETPKT not available'),
: 2851 1 DBM19 = uplit (%asciz' %N% A** FSET_UPAR: Can't find disk #D3#A in CST #D1'),
: 2852 1 DBM20 = uplit (%asciz' %N% A** Bad conn ID #06#A received from #06'),
: 2853 1 DBM21 = uplit (%asciz' %N% A** Message type #02#A received in MSCP packet'),
: 2854 1 DBM22 = uplit (%asciz' %N% A** SEQUEN: RETPKT not available'),
: 2855 1 DBM23 = uplit (%asciz' %N% A** Error in SET_CTLR_CHAR'),
: 2856 1 DBM25 = uplit (%asciz' %N% A** Ctlr timeout = #D3#A, seconds'),
: 2857 1 DBM26 = uplit (%asciz' %N% A** Error in UNIT_INIT'),
: 2858 1 DBM27 = uplit (%asciz' %N% A** UNIT_INIT: RETPKT has bad ENDCODE'),
: 2859 1 DBM28A = uplit (%asciz' %N% A** Unit size (Lo) = #D5#A,'),
: 2860 1 DBM28B = uplit (%asciz' %N% A** Unit size (Hi) = #D5#A,'),
: 2861 1 DBM29 = uplit (%asciz' %N% A** ACCESS: RETPKT has bad ENDCODE'),
: 2862 1 DBM32 = uplit (%asciz' %N% A** QIO_UNIT: CST #D1#A no unit selected'),
: 2863 1 DBM101 = uplit (%asciz' %N% A** Unit # is: #06'),
: 2864 1 DBM104 = uplit (%asciz' %N% A** Removable disk is selected'),
: 2865 1 DBM105 = uplit (%asciz' %N% A** Fixed disk is selected'),
: 2866 1 DBM107 = uplit (%asciz' %N% A** Illegal function: #06'),
: 2867 1 DBM108 = uplit (%asciz' %N% A** Command ref # #06#A/#06#A (Oct) not sent by Host'),
: 2868 1 DBM109 = uplit (%asciz' %N% A** Unknown Error Log format #03#A received'),
: 2869 1 ! DBM110 = uplit (%asciz' %N% A** Error-Log save area full'),
: 2870 1 DBM111 = uplit (%asciz' %N% A** Op-code #03#A, End-code #03#A for ref # #06#A/#06#A (8)'),
: 2871 1 DBM112 = uplit (%asciz' %N% A** Cmd-bc #06#A/#06#A Rsp-bc #06#A/#06#A for #06#A/#06#A (8)'),
: 2872 1 DBM120 = uplit (%asciz' %N% A** Response already received for cmd #06#A/#06#A (8)'),
: 2873 1 DBM121 = uplit (%asciz' %N% A** Failure to send command after # #06#A/#06#A (8)'),
: 2874 1 !
: 2875 1 ! DROP UNIT MESSAGES
: 2876 1 !
: 2877 1 DU_MSG = uplit (%asciz' %N% AUNIT#D2#A DROPPED - '),
: 2878 1 DU_RSN = uplit (
: 2879 1 uplit (%asciz' %AUSER COMMAND#N'),
: 2880 1 uplit (%asciz' %ACONFIGURATION ERROR#N'),
: 2881 1 uplit (%asciz' %AINIT ERROR#N'),
: 2882 1 uplit (%asciz' %ATRANSFER LIMIT REACHED#N'),
: 2883 1 uplit (%asciz' %AERROR LIMIT REACHED#N'),
: 2884 1 uplit (%asciz' %AUNRECOVERABLE DRIVE ERROR#N'),
: 2885 1 uplit (%asciz' %AUNRECOVERABLE CONTROLLER ERROR#N'),
: 2886 1 uplit (%asciz' %AFAILED TO COME ONLINE#N'),
: 2887 1 uplit (%asciz' %AFAILED TO ACCESS EITHER FIRST OR LAST TRACK DURING INIT#N'),
: 2888 1 uplit (%asciz' %ADISK WRITE PROTECTED#N'),
: 2889 1 uplit (%asciz' %ACOMMAND TIME OUT#N')) : vector [11],
: 2890 1 !
: 2891 1 ! SYSTEM MESSAGES (PRINTF)
: 2892 1 !
: 2893 1 MSG_01 = uplit (%asciz' %N% APOWER DELAY - WAITING'),
: 2894 1 MSG_02 = uplit (%asciz' %N% AFUNCTIONAL TEST STARTED'),
: 2895 1 MSG_03 = uplit (%asciz' %N% N% AEXERCISER STARTED#N'),
: 2896 1 !

```

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL1:6

SEQ 0035
Page 35
(31)

ZRQAM1
VOL.6

RD/RX EXERCISER
GLOBAL TEXT SECTION

```

2897 1 1 REPORT MESSAGES (PRINTS)
2898 1
2899 1 RPT1 = uplit (%asciz'%N%N%AUNT DSK%$8%A# OF # BYTES # OF # BYTES'),
2900 1 RPT2 = uplit (%asciz'%A --HARD ERRORS-- --SOFT ERRORS--'),
2901 1 RPT3 = uplit (%asciz'%N%A # ^ TYPE READS READ WRITES WRITTEN'),
2902 1 RPT4 = uplit (%asciz'%A SEK DAT DRV HST SEK DAT DRV HST'),
2903 1 RPT5 = uplit (%asciz'%N%A-----'),
2904 1 RPT6 = uplit (%asciz'%A-----'),
2905 1 RPT7 = uplit (%asciz'%N%D2%D4$S2$T'),
2906 1 RPT8 = uplit (%asciz'%D4$Z3%D3$A,$Z3$A,$Z3$'),
2907 1 RPT9 = uplit (%asciz'%D4%D4%D4%D4%D4%D4%D4'),
2908 1 RPT10 = uplit (%asciz'%N%A . . CNIR . . . . .'),
2909 1 RPT11 = uplit (%asciz'%A . . %D4$A . . . %D4$A .'),
2910 1
2911 1 GENERAL ERROR MESSAGES
2912 1
2913 1 SYSTEM FATAL (ERRSF)
2914 1
2915 1 EGS_01 = uplit (%asciz%'TOO MANY UNITS'),
2916 1 EGS_02 = uplit (%asciz%'NOT ENOUGH FREE MEMORY FOR ALLOCATING READ/WRITE BUFFERS'),
2917 1
2918 1 DRIVE FATAL (ERRDF)
2919 1
2920 1 EGD_10 = uplit (%asciz%'REGISTER EXISTENCE TEST FAILED'),
2921 1 EGD_11 = uplit (%asciz%'VECTOR TEST FAILED'),
2922 1 EGD_12 = uplit (%asciz%'BR LEVEL TEST FAILED'),
2923 1 EGD_13 = uplit (%asciz%'INIT SEQUENCE FAILED'),
2924 1 EGD_14 = uplit (%asciz%'FATAL CONTROLLER ERROR'),
2925 1 EGD_15 = uplit (%asciz%'ONLINE FAILED'),
2926 1 EGD_16 = uplit (%asciz%'WRITE-PROTECT CONFLICT'),
2927 1 EGD_17 = uplit (%asciz%'ACCESS FAILED'),
2928 1 EGD_18 = uplit (%asciz%'FATAL I/O ERROR'),
2929 1 EGD_19 = uplit (%asciz%'CONTROLLER TIMEOUT'),
2930 1 EGD_19 = uplit (%asciz%'DISK TYPE UNKNOWN TO EXERCISER'),
2931 1 EGD_20 = uplit (%asciz%'FAILED TO SEND SET-CONTROLLER-CHARACTERISTICS COMMAND'),
2932 1 EGD_21 = uplit (%asciz%'SET-CONTROLLER-CHARACTERISTICS RESPONSE HAS BAD ENDCODE OR FLAGS IN ERROR'),
2933 1 EGD_22 = uplit (%asciz%'FAILED TO SEND ON LINE COMMAND'),
2934 1 EGD_23 = uplit (%asciz%'ON-LINE RESPONSE HAS BAD ENDCODE'),
2935 1 EGD_24 = uplit (%asciz%'ON-LINE RESPONSE HAS UNKNOWN DEVICE'),
2936 1
2937 1 HARD or SOFT (ERRHRD or ERRSOF)
2938 1
2939 1 EGH_30 = uplit (%asciz%'I/O REQUEST FAILED'),
2940 1
2941 1 BASIC ERROR MESSAGES (PRINTB)
2942 1
2943 1 SYSTEM FATAL (ERRSF)
2944 1
2945 1 EBS_01 = uplit (%asciz%'AMORE THAN %D2$A UNITS SPECIFIED'),
2946 1
2947 1 DRIVE FATAL (ERRDF)
2948 1
2949 1 EBD_10 = uplit (%asciz%'$A NO RESPONSE AT ADDRESS %D6$'),

```

ZRGAM1
V01.6RD/RX EXERCISER
GLOBAL TEXT SECTION11 Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL1;6SEQ 0036
Page 36
(31)

```

: 2950 1      EBD_12 = uplit (*asciz' *A* INCORRECT BR LEVEL FOR DRIVE *06'),
: 2951 1      EBD_13 = uplit (*asciz' *A* STEP *01* READ ERROR'),
: 2952 1      EBD_14 = uplit (*asciz' *A* BAD SA CODE FROM DRIVE *06'),
: 2953 1      EBD_18 = uplit (*asciz' *A* DISK *02* WENT OFFLINE'),
: 2954 1      EBD_19 = uplit (*asciz' *A* DRIVE *06* NOT PROCESSING COMMAND PACKETS'),
: 2955 1      EBD_24 = uplit (*asciz' *A* DISK *02* WENT TO THE "AVAILABLE" STATE'),
: 2956 1      !
: 2957 1      !
: 2958 1      !      HARD or SOFT (ERRHRD or ERRSOFT)
: 2959 1      !
: 2960 1      EH_0 = UPLIT (*asciz' - UNRECOGNIZED MESSAGE TYPE'),          !???
: 2961 1      EH_1 = UPLIT (*asciz' - UNRECOGNIZED CONNECTION ID'),        !???
: 2962 1      EH_2 = UPLIT (*asciz' - UNRECOGNIZED RETURN MESSAGE'),        !???
: 2963 1      EH_3 = UPLIT (*asciz' - UNRECOGNIZED RETURN PACKET'),        !???
: 2964 1      EH_4 = UPLIT (*asciz' - UNRECOGNIZED CRN'),                  !???
: 2965 1      EH_5 = UPLIT (*asciz' - UNRECOGNIZED OPCODE'),              !???
: 2966 1      EH_6 = UPLIT (*asciz' - MSCP STATUS CODE ERR'),              !???
: 2967 1      EH_7 = UPLIT (*asciz' - DUP STATUS CODE ERR'),              !???
: 2968 1      EH_8 = UPLIT (*asciz' - UNRECOGNIZED STATUS CODE'),          !???
: 2969 1      EH_9 = UPLIT (*asciz' - LBN HOST COMPARE ERR'),              !???
: 2970 1      EH_10 = UPLIT (*asciz' - DBN HOST COMPARE ERR'),             !???
: 2971 1      EH_12 = UPLIT (*asciz' - UNABLE TO LOAD DUP MEDIA'),         !???
: 2972 1      EH_13 = UPLIT (*asciz' - ERR IN DUP PKT WHEN USING CTLR LC PROG'), !???
: 2973 1      !
: 2974 1      ERR_00 = uplit (*asciz' *A* DISK *02'),
: 2975 1      ERR_COD = uplit (
: 2976 1          uplit (*asciz' *A* INVALID COMMAND'),
: 2977 1          uplit (*asciz' *A* COMMAND ABORTED'),
: 2978 1          uplit (*asciz' *A* UNIT OFFLINE'),
: 2979 1          uplit (*asciz' *A* TRANSITION TO AVAILABLE STATE'),
: 2980 1          uplit (*asciz' *A* MEDIA FORMAT ERROR'),
: 2981 1          uplit (*asciz' *A* WRITE-PROTECTED'),
: 2982 1          uplit (*asciz' *A* DEVICE COMPARE ERROR'),
: 2983 1          uplit (*asciz' *A* DATA ERROR'),
: 2984 1          uplit (*asciz' *A* HOST BUFFER ACCESS ERROR'),
: 2985 1          uplit (*asciz' *A* CONTROLLER ERROR'),
: 2986 1          uplit (*asciz' *A* DRIVE ERROR'),
: 2987 1          uplit (*asciz' *A* MESSAGE FROM INTERNAL DIAGNOSTICS'),
: 2988 1          uplit (*asciz' *A* HOST COMPARE ERROR'),
: 2989 1          uplit (*asciz' *A* COMMAND TIMEOUT')) : vector [14],
: 2990 1      !
: 2991 1      !      ERROR LOG MESSAGE (ERRSOFT)
: 2992 1      !
: 2993 1      ELG_00 = uplit (*asciz' *A* ERROR LOG MESSAGE RECEIVED: *N'),
: 2994 1      ELG_FMT = uplit (
: 2995 1          uplit (*asciz' *A* CONTROLLER ERROR *N'),
: 2996 1          uplit (*asciz' *A* HOST MEMORY ACCESS ERROR *N'),
: 2997 1          uplit (*asciz' *A* DISK *02* - DISK TRANSFER ERROR *N'),
: 2998 1          uplit (*asciz' *A* DISK *02* - "STANDARD DISK INTERCONNECT" ERROR *N'),
: 2999 1          uplit (*asciz' *A* DISK *02* - "SMALL DISK" ERROR *N')) : vector [5],
: 3000 1      !
: 3001 1      !      EXTENDED ERROR MESSAGES (PRINTX)
: 3002 1      !

```

ZRQAM1
V01.6

RD/RX EXERCISER
GLOBAL TEXT SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0037
Page 3/
(31)

```

: 3003 1 EX_SA = uplit (ASCII'NNA SA: #06'),
: 3004 1 EX_SC = uplit (ASCII'NNA STATUS CODE: #02'),
: 3005 1 EX_SBO = uplit (ASCII'#04'),
: 3006 1 EX_SB = uplit (ASCII'NNA SUB_CODE: '),
: 3007 1 EX_CMD = uplit (ASCII'NNA COMMAND: '),
: 3008 1 EX_RD = uplit (ASCII'AREAD'),
: 3009 1 EX_WRT = uplit (ASCII'AWRITE'),
: 3010 1 EX_CMP = uplit (ASCII'A-COMPARE'),
: 3011 1 EX_ONL = uplit (ASCII'AONLINE'),
: 3012 1 EX_ACC = uplit (ASCII'AACCESS'),
: 3013 1 EX_OP = uplit (ASCII'#03'),
: 3014 1 EX_BB = uplit (ASCII'NNA BAD BLOCK (Host replaceable): #D5#A. (OCT #06#A)'),
: 3015 1 EX_BB1 = uplit (ASCII'NNA 1st BAD BLOCK (Host replaceable): #D5#A. (OCT #06#A)'),
: 3016 1 EX_BBU = uplit (ASCII'NNA BAD BLOCK REPORTED (Replaced): #D#A. (OCT #06#A)'),
: 3017 1 EX_LBN = uplit (ASCII'NNA LBN: #D5#A. (OCT #06#A)'),
: 3018 1 EX_PBN = uplit (ASCII'NNA PBN: #D5#A. (OCT #06#A)'),
: 3019 1 EX_LBR = uplit (ASCII'NNA LBN: (READ) #D5#A. (OCT #06#A)'),
: 3020 1 EX_LBW = uplit (ASCII'NNA LBN: (WRITE) #D5#A. (OCT #06#A)'),
: 3021 1 EX_RBY = uplit (ASCII'NNA REPLACEMENT BLOCK NO. #D5#A. (OCT #06#A)'),
: 3022 1 EX_CBC = uplit (ASCII'NNA BYTE COUNT IN COMMAND: #D5#A.'),
: 3023 1 EX_CBR = uplit (ASCII'NNA BYTE COUNT IN READ COMMAND: #D5#A.'),
: 3024 1 EX_CBW = uplit (ASCII'NNA BYTE COUNT IN WRITE COMMAND: #D5#A.'),
: 3025 1 EX_BC = uplit (ASCII'NNA ACTUAL # OF BYTES TRANSFERRED: #D5#A.'),
: 3026 1 EX_BD = uplit (ASCII'NNA I/O BUFFER ADDRESS (32 bits): #06#A #06'),
: 3027 1 EX_BDR = uplit (ASCII'NNA I/O BUFFER ADDRESS FOR READ (32 bits): #06#A #06'),
: 3028 1 EX_BDW = uplit (ASCII'NNA I/O BUFFER ADDRESS FOR WRITE (32 bits): #06#A #06'),
: 3029 1 EX_RP = uplit (ASCII'NNA CONTENTS OF COMMAND/RESPONSE PACKET SAVE AREA: #N'),
: 3030 1 EX_WRD = uplit (ASCII'NA #06'),
: 3031 1 EX_TIM = uplit (ASCII'NNA TIME: #Z2#A;#Z2#A HOURS#N'),
: 3032 1
: 3033 1
: 3034 1 XX13 = UPLIT (ASCII'NNA * DISK : #D2'),
: 3035 1 XX23 = UPLIT (ASCII'NNA ADBN: #D5#A. (OCT #06#A)'),
: 3036 1 XX32 = UPLIT (ASCII'NNA ABYTE NUMBER: #D3'),
: 3037 1 XX33 = UPLIT (ASCII'NNA RANDOM WRITTEN WORD : #B16'),
: 3038 1 XX34 = UPLIT (ASCII'NNA RANDOM READ WORD bin: #B16#A oct: #U6'),
: 3039 1
: 3040 1
: 3041 1 ! CONFIGURATION ERROR MESSAGES (PRINTF)
: 3042 1 !
: 3043 1 CER_01 = uplit (ASCII'NNA DUPLICATE UNIT: #D2#A AT IP: #06'),
: 3044 1 CER_02 = uplit (ASCII'NNA MORE THAN #D1#A DIFFERENT IP ADDRESSES'),
: 3045 1 !
: 3046 1 ! ERROR/EVENT SUB CODES (PRINTX)
: 3047 1 !
: 3048 1 SC_SDI = uplit (ASCII'ASPIN-DOWN IGNORED'),
: 3049 1 SC_CON = uplit (ASCII'ASTILL CONNECTED'),
: 3050 1 SC_DUP = uplit (ASCII'ADUPLICATE UNIT NUMBER'),
: 3051 1 SC_ONL = uplit (ASCII'AALREADY ONLINE'),
: 3052 1 SC_SON = uplit (ASCII'ASTILL ONLINE'),
: 3053 1 SC_UNK = uplit (ASCII'AUNIT UNKNOWN OR ONLINE TO ANOTHER CONTROLLER'),
: 3054 1 SC_VOL = uplit (ASCII'ANO VOLUME MOUNTED OR DRIVE DISABLED BY SWITCH'),
: 3055 1 SC_IOP = uplit (ASCII'AUNIT INOPERATIVE (RD51/52 write fault)'),
```

ZRQAM1
VO1.6RD/RX EXERCISER
GLOBAL TEXT SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0038
Page 38
(31)

```

: 3056 1 SC_DIS = uplit (asciz' AUNIT DISABLED BY FIELD SERVICE OR INTERNAL DIAGNOSTICS'),
: 3057 1 SC_FER = uplit (asciz' A"FORCED ERROR" DETECTED WHILE ACCESSING FCT OR RCT'),
: 3058 1 SC_FE2 = uplit (asciz' ASECTOR HAD BEEN WRITTEN WITH "FORCED ERROR" MODIFIER'),
: 3059 1 SC_ISH = uplit (asciz' AFCT OR RCT UNREADABLE - INVALID SECTOR HEADER'),
: 3060 1 SC_IS2 = uplit (asciz' AHEADER COMPARE ERROR (Valid header not found)'),
: 3061 1 SC_DST = uplit (asciz' AFCT OR RCT UNREADABLE - DATA SYNC TIMEOUT'),
: 3062 1 SC_DS2 = uplit (asciz' ADATA SYNC NOT FOUND (Data sync timeout)'),
: 3063 1 SC_ECC = uplit (asciz' AFCT OR RCT UNREADABLE - UNCORRECTABLE ECC ERROR'),
: 3064 1 SC_ECD = uplit (asciz' AUNCORRECTABLE ECC ERROR'),
: 3065 1 SC_RCT = uplit (asciz' ARCT CORRUPTED'),
: 3066 1 SC_FUL = uplit (asciz' ANO REPLACEMENT BLOCK AVAILABLE (RCT full)'),
: 3067 1 SC_576 = uplit (asciz' ADISK NOT FORMATTED WITH 512 BYTE SECTORS'),
: 3068 1 SC_FCT = uplit (asciz' ADISK NOT FORMATTED OR FCT CORRUPTED'),
: 3069 1 SC_EC1 = uplit (asciz' AONE SYMBOL ECC ERROR'),
: 3070 1 SC_EC2 = uplit (asciz' ATWO SYMBOL ECC ERROR'),
: 3071 1 SC_EC3 = uplit (asciz' ATHREE SYMBOL ECC ERROR'),
: 3072 1 SC_EC4 = uplit (asciz' AFOUR SYMBOL ECC ERROR'),
: 3073 1 SC_EC5 = uplit (asciz' AFIVE SYMBOL ECC ERROR'),
: 3074 1 SC_EC6 = uplit (asciz' ASIX SYMBOL ECC ERROR'),
: 3075 1 SC_EC7 = uplit (asciz' ASEVEN SYMBOL ECC ERROR'),
: 3076 1 SC_EC8 = uplit (asciz' AEIGHT SYMBOL ECC ERROR'),
: 3077 1 SC_EC9 = uplit (asciz' ACORRECTABLE ERROR IN ECC FIELD'),
: 3078 1 SC_SWP = uplit (asciz' AUNIT SOFTWARE WRITE PROTECTED'),
: 3079 1 SC_HWP = uplit (asciz' AUNIT HARDWARE WRITE PROTECTED'),
: 3080 1 SC_ODA = uplit (asciz' AODD TRANSFER ADDRESS'),
: 3081 1 SC_ODB = uplit (asciz' AODD BYTE COUNT'),
: 3082 1 SC_NXM = uplit (asciz' ANON-EXISTENT HOST MEMORY'),
: 3083 1 SC_PAR = uplit (asciz' AHOST MEMORY PARITY ERROR'),
: 3084 1 SC_CTO = uplit (asciz' ACOMMAND TIMEOUT OR RETRY LIMIT EXCEEDED'),
: 3085 1 SC_SDS = uplit (asciz' ASERIALIZER/DESERIALIZER OVERRUN OR UNDERRUN'),
: 3086 1 SC_EDC = uplit (asciz' A"ERROR DETECTION CODE" ERROR'),
: 3087 1 SC_IDS = uplit (asciz' AINCONSISTENT INTERNAL DATA STRUCTURE'),
: 3088 1 SC_SRT = uplit (asciz' ADRIVE COMMAND TIMEOUT (No response or seek incomplete)'),
: 3089 1 SC_SRI = uplit (asciz' ACONTROLLER DETECTED TRANSMISSION OR PROTOCOL ERROR'),
: 3090 1 SC_POE = uplit (asciz' APOSITION ERROR (Mis-seek)'),
: 3091 1 SC_RDY = uplit (asciz' ALOST READ/WRITE READY DURING/BETWEEN TRANSFERS'),
: 3092 1 SC_CLK = uplit (asciz' ADRIVE CLOCK DROPOUT'),
: 3093 1 SC_RSP = uplit (asciz' ALOST RECEIVER READY BETWEEN SECTORS'),
: 3094 1 SC_SUR = uplit (asciz' ADRIVE DETECTED ERROR'),
: 3095 1 SC_PSP = uplit (asciz' ACONTROLLER DETECTED PULSE OR STATE PARITY ERROR'),
: 3096 1 !
: 3097 1 ! CONTROLLER GENERIC ERROR CODES
: 3098 1 !
: 3099 1 CNTR_ERR = uplit (
: 3100 1     uplit (asciz' ACONTROLLER TIMEOUT'),
: 3101 1     uplit (asciz' AENVELOPE/PACKET READ ERROR (Parity or timeout)'),
: 3102 1     uplit (asciz' AENVELOPE/PACKET WRITE ERROR (Parity or timeout)'),
: 3103 1     uplit (asciz' ACONTROLLER ROM AND RAM PARITY ERROR'),
: 3104 1     uplit (asciz' ACONTROLLER RAM PARITY ERROR'),
: 3105 1     uplit (asciz' ACONTROLLER ROM PARITY ERROR'),
: 3106 1     uplit (asciz' ARING READ ERROR (Parity or timeout)'),
: 3107 1     uplit (asciz' ARING WRITE ERROR (Parity or timeout)'),
: 3108 1     uplit (asciz' INTERRUPT MASTER FAILURE'),

```


ZRQAM1
V01.6RD/RX EXERCISER
GLOBAL TEXT SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0039
Page 39
(31)

```

: 3109 1      uplit (*asciz'AHOST ACCESS TIMEOUT (Higher level protocol dependent)'),
: 3110 1      uplit (*asciz'ACREDIT LIMIT EXCEEDED'),
: 3111 1      uplit (*asciz'AQ-BUS MASTER ERROR'),
: 3112 1      uplit (*asciz'ACONTROLLER FATAL ERROR'),
: 3113 1      uplit (*asciz'AINSTRUCTION LOOP TIMEOUT'),
: 3114 1      uplit (*asciz'AILLEGAL VIRTUAL CIRCUIT ID'),
: 3115 1      uplit (*asciz'AINERRUPT VECTOR ILLEGAL'),
: 3116 1      uplit (*asciz'AMAINTENANCE READ/WRITE INVALID REGION IDENTIFIER'),
: 3117 1      uplit (*asciz'AMAINTENANCE WRITE LOAD TO NON-LOADABLE CONTROLLER'),
: 3118 1      uplit (*asciz'ACONTROLLER RAM ERROR (Non parity)'),
: 3119 1      uplit (*asciz'AINIT SEQUENCE ERROR'),
: 3120 1      uplit (*asciz'AHIGHER LEVEL PROTOCOL INCOMPATIBILITY ERROR'),
: 3121 1      uplit (*asciz'APURGE/POLL HARDWARE FAILURE'),
: 3122 1      uplit (*asciz'AMAPPING REGISTER READ FAILURE (Parity or timeout)') : vector [23],
: 3123 1      !
: 3124 1      ! RD/RX CONTROLLER DEPENDENT ERRORS CODES
: 3125 1      !
: 3126 1      RDRX_ERR = uplit (
: 3127 1          uplit (*asciz'AT11 CPU FAILURE'),
: 3128 1          uplit (*asciz'ANON-PARITY RAM ERROR'),
: 3129 1          uplit (*asciz'ASTATE MACHINE FAILURE - T11 ADDRESS REGISTER'),
: 3130 1          uplit (*asciz'ASTATE MACHINE FAILURE - Q-BUS ADDRESS REGISTER'),
: 3131 1          uplit (*asciz'ASTATE MACHINE FAILURE - CRC REGISTER'),
: 3132 1          uplit (*asciz'ASTATE MACHINE FAILURE - SERIALIZER/DESERIALIZER REGISTER'),
: 3133 1          uplit (*asciz'ASTATE MACHINE FAILURE - WRONG HARDWARE VERSION') : vector [7],
: 3134 1      !
: 3135 1      ! PRINTOUTS THAT FAKE THE DRS ERROR MESSAGES
: 3136 1      !
: 3137 1      DF_MSG = uplit (*asciz'N#AZRQA DEV FTL  #Z5#A ON UNIT #Z2#A TST 001 SUB 000 PC: #06'),
: 3138 1      HRD_MSG = uplit (*asciz'N#AZRQA HRD ERR  #Z5#A ON UNIT #Z2#A TST 001 SUB 000 PC: #06'),
: 3139 1      SFT_MSG = uplit (*asciz'N#AZRQA SFT ERR  #Z5#A ON UNIT #Z2#A TST 001 SUB 000 PC: #06#N'),
: 3140 1      HRD_SUB = uplit (*asciz'N#AI/O REQUEST FAILED#N'),
: 3141 1      !
: 3142 1      ! MISCELLANEOUS
: 3143 1      !
: 3144 1      SPACE4 = uplit (*asciz'S4'),
: 3145 1      CRLF = uplit (*asciz'N'),
: 3146 1      DASH = uplit (*asciz'A - '),
: 3147 1      ASTERISK = uplit (*asciz'A* ');

```

PROGRAM
V01.6

RD-RX EXERCISER
DEFAULT HARDWARE P-TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11-B1159-14 V4.0 5%
DISK\$USER2:[POWER5]ZRGADO.BL 1:6

SEQ 0040
Page 40
(32)

```

: 3148 1  *sbttl 'DEFAULT HARDWARE P-TABLE'
: 3149 1
: 3150 1
: 3151 1  !
: 3152 1  ! THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: 3153 1  ! THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: 3154 1  ! IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
: 3155 1  ! AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
: 3156 1  !
: 3157 1  BGNHW (DEPTBL);
: 3158 1
: 3159 1  global
: 3160 1      HWPT_IP_ADDR : word initial (INIT_IP_ADDR),      ! IP ADDRESS
: 3161 1      HWPT_VECTOR : word initial (INIT_INTR_VECT),    ! VECTOR ADDRESS
: 3162 1      HWPT_BR_LEVEL : word initial (INIT_BR_LEVEL),    ! BR LEVEL
: 3163 1      HWPT_DISK : word initial ('0'100340'),          ! PROTECT, WHOLE DISK, DUP WT, RUN DUP   ???
: 3164 1      ! DUP WT, DK 0   ???
: 3165 1      HWPTS0_LBN : word initial (0),                    ! STARTING TRACK LO  ???
: 3166 1      HWPTS1_LBN : word initial (0),                    ! STARTING TRACK HI  ???
: 3167 1      HWPTEO_LBN : word initial ('0'177777'),          ! ENDING TRACK LO   ???
: 3168 1      HWPTET_LBN : word initial (0),                    ! ENDING TRACK HI   ???
: 3169 1
: 3170 1  ENDHW;

```


[4]

ZRQ4M1
VOL.6

RD RX EXERCISER
SOFTWARE P TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1199 16 V4.0 579
DISK\$USER2:([POWERS])ZRQ4D0.BL1;6

SEQ.0041
Page 41
(33)

```

: 3171 1 *sbttl 'SOFTWARE P TABLE
: 3172 1
: 3173 1
: 3174 1 !
: 3175 1 ! THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
: 3176 1 ! PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE
: 3177 1 ! SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
: 3178 1 ! AT RUN TIME.
: 3179 1 !
: 3180 1 BGNSW (SEPTBL);
: 3181 1
: 3182 1 global
: 3183 1 SWP_ERROR : word initial (32), ! HARD ERROR LIMIT FOR DROPPING UNIT
: 3184 1 SWP_XFER : word initial (0), ! XFER LIMIT, DEFAULT : QUICK PASS !ZZZ
: 3185 1 SWP_FLAGS : word initial (SWP_RDM or SWP_CRC or SWP_HWC or SWP_FER or SWP_HRD or SWP_BLK), ! FLAGS (SEE DOCUMENTATION)
: 3186 1 SWP_DPAT : word initial (0), ! DATA PATTERN NUMBER
: 3187 1 SWP_RAT : word initial (99), ! RD51/52 OPERATION RATIO
: 3188 1 SWP_TIME : word initial (0), ! START TIME (HHMM)
: 3189 1 DUPROUND : word initial (11), ! NO OF 1/0s PER DBN TEST ZZZ
: 3190 1
: 3191 1 ! THE NEXT TWO LOCATIONS SHOULD BE TOGETHER
: 3192 1
: 3193 1 SWP_UCNT : word initial (MAX_UDP_CNT), ! USER DATA PATTERN COUNT
: 3194 1 SWP_UDPAT : vector (MAX_UDP_CNT, word), ! USER DATA PATTERN
: 3195 1
: 3196 1 ENDSW;

```

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX-11 B1199 16 V4.0 579
DISK\$USER2:[POWERS]ZRQAU0.BL1;6

SEQ 0042
Page 42
(34)

```

: 3197 1 *sbttl 'PROTECTION TABLE
: 3198 1
: 3199 1
: 3200 1 ! THIS TABLE IS USED BY THE RUNTIME SERVICES
: 3201 1 ! TO PROTECT THE LOAD MEDIA.
: 3202 1 !
: 3203 1
: 3204 1 BGNPROT (0, -1, 6);
: 3205 1
: 3206 1 !1ST ARG : OFFSET INTO P-TABLE FOR CSR ADDRESS
: 3207 1 !2ND ARG : OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
: 3208 1 !3RD ARG : OFFSET INTO P-TABLE FOR DRIVE NUMBER
: 3209 1
: 3210 1 ENDPROT;
: 3211 1 end
: 3212 1
: 3213 0 eludom

```

.TITLE ZRQAM1 RD/RX EXERCISER
.IDENT /V01.6/
.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 104 .ASCII /D/
000011 060 .ASCII /O/
000012 000000G L$UNIT::,WORD T$PTHV
000014 07777 L$TIME::,WORD 77777
000016 000000G L$HPCP::,WORD L$HARD
000020 000000G L$SPCP::,WORD L$SOFT
000022 022176 L$HPTP::,WORD L$HW
000024 022222 L$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$ENVI::,WORD 0
000046 000000 L$EXPI::,WORD 0
000050 L$MRGV::
000050 003 .BYTE 3
000051 003 .BYTE 3
000052 000000 L$EF::,WORD 0

```

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B1199 16 V4.0-5/9
DISK\$USER2:[POWERS]ZRQAD0.BL1:6SEQ 0043
Page 43
(34)

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 022304
000114 000000
000116 000000
000120 000000
000122 000001
000124 000000G

000126
000130
000132
000134
000136 111 120 040
000141 141 144 144
000144 162 145 163
000147 163 000 000
000152 126 145 143
000155 164 157 162
000160 000 000
000162 102 122 040
000165 154 145 166
000170 145 154 000
000173 000
000174 104 162 151
000177 166 145 040
000202 156 165 155
000205 142 145 162
000210 000 000
000212 124 145 163
000215 164 040 145
000220 156 164 151
000223 162 145 040
000226 143 165 163
000231 164 157 155
000234 145 162 040
000237 141 162 145
000242 141 040 157
000245 145 040 164
000250 150 151 163

L\$SPC:: .WORD 0
L\$DEVP:: .WORD 0
L\$REPP:: .WORD L\$DVT/P
L\$EXP4:: .WORD L\$RPT
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::
L\$DISPATCH:: .WORD T1
ERRTYP:: .BLKW 1
ERRNBR:: .BLKW 1
ERRMSG:: .BLKW 1
ERRBLK:: .BLKW 1
P.AAA: .ASCII /IP /
P.AAA: .ASCII /add/
P.AAA: .ASCII /res/
P.AAA: .ASCII /s/<00><00>
P.AAB: .ASCII /vec/
P.AAB: .ASCII /tor/
P.AAB: .ASCII <00><00>
P.AAC: .ASCII /BR /
P.AAC: .ASCII /lev/
P.AAC: .ASCII /el/<00>
P.AAC: .ASCII <00>
P.AAD: .ASCII /Dri/
P.AAD: .ASCII /ve /
P.AAD: .ASCII /num/
P.AAD: .ASCII /ber/
P.AAD: .ASCII <00><00>
P.AAE: .ASCII /Tes/
P.AAE: .ASCII /t e/
P.AAE: .ASCII /nti/
P.AAE: .ASCII /ne /
P.AAE: .ASCII /cus/
P.AAE: .ASCII /tom/
P.AAE: .ASCII /er/
P.AAE: .ASCII /are/
P.AAE: .ASCII /a o/
P.AAE: .ASCII /t t/
P.AAE: .ASCII /his/

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX 11 B11gs-16 V4.0-579
DISK:USER2:[POWERS]ZRQAD0.BL1,6SEQ 0044
Page 44
(34)

000253	040	144	151	.ASCII	/ di/
000256	163	153	000	.ASCII	/sk/<00>
000261	000			.ASCII	<00>
000262	114	157	167	P.AAF:	.ASCII /Low/
000265	145	162	040		.ASCII /er /
000270	157	143	164		.ASCII /oct/
000273	141	154	040		.ASCII /al /
000276	167	157	162		.ASCII /wor/
000301	144	040	157		.ASCII /d o/
000304	146	040	142		.ASCII /f b/
000307	145	147	151		.ASCII /egi/
000312	156	156	151		.ASCII /nni/
000315	156	147	040		.ASCII /ng /
000320	114	102	116		.ASCII /LBN/
000323	040	141	144		.ASCII / ad/
000326	144	162	145		.ASCII /dre/
000331	163	163	000		.ASCII /ss/<00>
000334	110	151	147	P.AAG:	.ASCII /Hig/
000337	150	145	162		.ASCII /her/
000342	040	157	143		.ASCII / oc/
000345	164	141	154		.ASCII /tal/
000350	040	167	157		.ASCII / wo/
000353	162	144	040		.ASCII /rd /
000356	157	146	040		.ASCII /of /
000361	142	145	147		.ASCII /beg/
000364	151	156	156		.ASCII /inn/
000367	151	156	147		.ASCII /ing/
000372	040	114	102		.ASCII / LB/
000375	116	040	141		.ASCII /N a/
000400	144	144	162		.ASCII /ddr/
000403	145	163	163		.ASCII /ess/
000406	000	000			.ASCII <00><00>
000410	114	157	167	P.AAH:	.ASCII /Low/
000413	145	162	040		.ASCII /er /
000416	157	143	164		.ASCII /oct/
000421	141	154	040		.ASCII /al /
000424	167	157	162		.ASCII /wor/
000427	144	040	157		.ASCII /d o/
000432	146	040	145		.ASCII /f e/
000435	156	144	151		.ASCII /ndi/
000440	156	147	040		.ASCII /ng /
000443	114	102	116		.ASCII /LBN/
000446	040	141	144		.ASCII / ad/
000451	144	162	145		.ASCII /dre/
000454	163	163	000		.ASCII /ss/<00>
000457	000				.ASCII <00>
000460	110	151	147	P.AAI:	.ASCII /Hig/
000463	150	145	162		.ASCII /her/
000466	040	157	143		.ASCII / oc/
000471	164	141	154		.ASCII /tal/
000474	040	167	157		.ASCII / wo/
000477	162	144	040		.ASCII /rd /
000502	157	146	040		.ASCII /of /

G4

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11 Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0045
Page 45
(34)

000505	145	156	144	.ASCII	/end/
000510	151	156	147	.ASCII	/ing/
000513	040	114	102	.ASCII	/LB/
000516	116	040	141	.ASCII	/N a/
000521	144	144	162	.ASCII	/ddr/
000524	145	163	163	.ASCII	/ess/
000527	000			.ASCII	<00>
000530	127	162	151	P.AAJ:	.ASCII /Wri/
000533	164	145	040	.ASCII	/te /
000536	157	156	040	.ASCII	/on /
000541	143	165	163	.ASCII	/cus/
000544	164	157	155	.ASCII	/tom/
000547	145	162	040	.ASCII	/er /
000552	144	141	164	.ASCII	/dat/
000555	141	040	141	.ASCII	/a a/
000560	162	145	141	.ASCII	/nea/
000563	040	157	156	.ASCII	/ on/
000566	040	164	150	.ASCII	/ th/
000571	151	163	040	.ASCII	/is /
000574	144	151	163	.ASCII	/dis/
000577	153	000	000	P.AAK:	.ASCII /k/<00><00>
000602	052	052	040	.ASCII	/++ /
000605	127	101	122	.ASCII	/WAR/
000610	116	111	116	.ASCII	/NIN/
000613	107	040	055	.ASCII	/G /
000616	040	103	125	.ASCII	/CU/
000621	123	124	117	.ASCII	/STO/
000624	115	105	122	.ASCII	/MER/
000627	040	104	101	.ASCII	/DA/
000632	124	101	040	.ASCII	/TA /
000635	101	122	105	.ASCII	/ARE/
000640	101	040	115	.ASCII	/A M/
000643	101	131	040	.ASCII	/AY /
000646	102	105	040	.ASCII	/BE /
000651	117	126	105	.ASCII	/OVL/
000654	122	127	122	.ASCII	/RWR/
000657	111	124	124	.ASCII	/III/
000662	105	116	041	.ASCII	/EN!/
000665	040	056	056	.ASCII	/.../
000670	056	040	103	.ASCII	/..C/
000673	117	116	106	.ASCII	/ONE/
000676	111	122	115	.ASCII	/IRM/
000701	000			.ASCII	<00>
000702	101	154	163	P.AAL:	.ASCII /Als/
000705	157	040	162	.ASCII	/o n/
000710	165	156	040	.ASCII	/un /
000713	104	125	120	.ASCII	/DUP/
000716	040	145	170	.ASCII	/e.../
000721	145	162	143	.ASCII	/enc/
000724	151	163	145	.ASCII	/ise/
000727	162	000	000	P.AAM:	.ASCII /r/<00><00>
000732	127	162	151	.ASCII	/Wri/
000735	164	145	040	.ASCII	/te /

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO,BL1;6SEQ 0046
Page 46
(34)

000740	157	156	040	.ASCII	/on /
000743	144	151	141	.ASCII	/dia/
000746	147	156	157	.ASCII	/gno/
000751	163	164	151	.ASCII	/sti/
000754	143	040	141	.ASCII	/c a/
000757	162	145	141	.ASCII	/rea/
000762	000	000		.ASCII	<00><00>
000764	110	141	162	P.AAN:	.ASCII /Har/
000767	144	040	105	.ASCII	/d E/
000772	162	162	157	.ASCII	/rro/
000775	162	040	154	.ASCII	/r l/
001000	151	155	151	.ASCII	/imi/
001003	164	000	000	P.AAO:	.ASCII /t/<00><00>
001006	124	162	141	.ASCII	/Tra/
001011	156	163	146	.ASCII	/nsf/
001014	145	162	040	.ASCII	/er /
001017	154	151	155	.ASCII	/lim/
001022	151	164	040	.ASCII	/it /
001025	151	156	040	.ASCII	/in /
001030	155	145	147	.ASCII	/meg/
001033	141	142	171	.ASCII	/aby/
001036	164	145	163	.ASCII	/tes/
001041	040	050	060	.ASCII	/ (0/
001044	040	146	157	.ASCII	/ fo/
001047	162	040	042	.ASCII	/r "/
001052	121	165	151	.ASCII	/Qui/
001055	143	153	040	.ASCII	/ck /
001060	160	141	163	.ASCII	/pas/
001063	163	042	051	.ASCII	/s")/
001066	000	000		.ASCII	<00><00>
001070	122	141	156	P.AAP:	.ASCII /Ran/
001073	144	157	155	.ASCII	/dom/
001076	040	163	145	.ASCII	/ se/
001101	145	153	040	.ASCII	/ek /
001104	155	157	144	.ASCII	/mod/
001107	145	000	000	P.AAQ:	.ASCII /e/<00><00>
001112	122	145	141	.ASCII	/Rea/
001115	144	055	143	.ASCII	/d c/
001120	157	155	160	.ASCII	/omp/
001123	141	162	145	.ASCII	/are/
001126	163	040	160	.ASCII	/s p/
001131	145	162	146	.ASCII	/erf/
001134	157	162	155	.ASCII	/orm/
001137	145	144	040	.ASCII	/ed /
001142	141	164	040	.ASCII	/at /
001145	164	150	145	.ASCII	/the/
001150	040	103	157	.ASCII	/ Co/
001153	156	164	162	.ASCII	/ntr/
001156	157	154	154	.ASCII	/oll/
001161	145	162	000	P.AAR:	.ASCII /er/<00>
001164	127	162	151	.ASCII	/Wri/
001167	164	145	055	.ASCII	/te /
001172	143	157	155	.ASCII	/com/

ZRGAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL1;6SEQ 004 /
Page 47
(34)

001175	160	141	162	.ASCII	/par/
001200	145	163	040	.ASCII	/es /
001203	160	145	162	.ASCII	/per/
001206	146	157	162	.ASCII	/for/
001211	155	145	144	.ASCII	/med/
001214	040	141	164	.ASCII	/ at/
001217	040	164	150	.ASCII	/ th/
001222	145	040	103	.ASCII	/e C/
001225	157	156	164	.ASCII	/ont/
001230	162	157	154	.ASCII	/rol/
001233	154	145	162	.ASCII	/ler/
001236	000	000		.ASCII	<00><00>
001240	103	150	145	P.AAS: .ASCII	/Che/
001243	143	153	040	.ASCII	/ck /
001246	141	154	154	.ASCII	/all/
001251	040	127	162	.ASCII	/ Wr/
001254	151	164	145	.ASCII	/ite/
001257	163	040	141	.ASCII	/s a/
001262	164	040	110	.ASCII	/t H/
001265	157	163	164	.ASCII	/ost/
001270	040	142	171	.ASCII	/ by/
001273	040	162	145	.ASCII	/ re/
001276	141	144	151	.ASCII	/adi/
001301	156	147	000	.ASCII	/ng/<00>
001304	125	163	145	P.AAT: .ASCII	/Use/
001307	162	055	144	.ASCII	/r d/
001312	145	146	151	.ASCII	/efi/
001315	156	145	143	.ASCII	/ned/
001320	040	144	141	.ASCII	/ da/
001323	164	141	040	.ASCII	/ta /
001326	160	141	163	.ASCII	/pat/
001331	164	145	162	.ASCII	/ter/
001334	156	000		.ASCII	/n/<00>
001336	123	145		P.AAU: .ASCII	/Sel/
001341	145	143	164	.ASCII	/ect/
001344	040	160	162	.ASCII	/ pr/
001347	145	055	144	.ASCII	/e-d/
001352	145	146	151	.ASCII	/efi/
001355	156	145	144	.ASCII	/ned/
001360	040	144	141	.ASCII	/ da/
001363	164	141	040	.ASCII	/ta /
001366	160	141	164	.ASCII	/pat/
001371	164	145	162	.ASCII	/ter/
001374	156	040	050	.ASCII	/n (/
001377	060	040	146	.ASCII	/o f/
001402	157	162	040	.ASCII	/or /
001405	123	145	161	.ASCII	/heq/
001410	165	145	156	.ASCII	/uen/
001413	164	151	141	.ASCII	/tia/
001416	154	040	163	.ASCII	/l s/
001421	145	154	145	.ASCII	/ele/
001424	143	164	151	.ASCII	/cti/
001427	157	156	051	.ASCII	/on)/

ZRQAN1
VOL.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6SEQ 0048
Page 48
(34)

001432	000	000		P.AAV:	.ASCII	<00><00>
001434	116	165	155		.ASCII	/Num/
001437	142	145	162		.ASCII	/ber/
001442	040	157	146		.ASCII	/ of/
001445	040	167	157		.ASCII	/ wo/
001450	162	144	163		.ASCII	/rds/
001453	040	151	156		.ASCII	/ in/
001455	040	144	141		.ASCII	/ da/
001461	164	141	040		.ASCII	/ta /
001464	160	141	164		.ASCII	/pat/
001467	164	145	162		.ASCII	/ter/
001472	156	040	050		.ASCII	/n (/
001475	061	066	040		.ASCII	/16 /
001500	155	141	170		.ASCII	/max/
001503	151	155	165		.ASCII	/imu/
001506	155	051	000		.ASCII	/m)/<00>
001511	000			P.AAW:	.ASCII	<00>
001512	120	141	164		.ASCII	/Pat/
001515	162	145	162		.ASCII	/ter/
001520	156	040	166		.ASCII	/n v/
001523	141	154	165		.ASCII	/alu/
001526	145	040	050		.ASCII	/e (/
001531	116	157	040		.ASCII	/No /
001534	154	145	141		.ASCII	/lea/
001537	144	151	156		.ASCII	/din/
001542	147	040	172		.ASCII	/g z/
001545	145	162	157		.ASCII	/ero/
001550	163	040	141		.ASCII	/s a/
001553	154	154	157		.ASCII	/llo/
001556	167	145	144		.ASCII	/wed/
001561	051	000	000	P.AAX:	.ASCII	/)/<00><00>
001564	103	154	145		.ASCII	/Cle/
001567	141	162	040		.ASCII	/ar /
001572	163	164	141		.ASCII	/sta/
001575	164	151	163		.ASCII	/tis/
001600	164	151	143		.ASCII	/tic/
001603	141	154	040		.ASCII	/hl /
001606	164	141	142		.ASCII	/tab/
001611	154	145	163		.ASCII	/len/
001614	040	141	146		.ASCII	/ af/
001617	164	145	162		.ASCII	/ter/
001622	040	160	162		.ASCII	/pr/
001625	151	156	164		.ASCII	/int/
001630	151	156	147		.ASCII	/ing/
001633	000			P.AAY:	.ASCII	<00>
001634	120	145	162		.ASCII	/Per/
001637	143	145	156		.ASCII	/cen/
001642	164	141	147		.ASCII	/tag/
001645	145	040	157		.ASCII	/e o/
001650	146	040	042		.ASCII	/f /
001653	106	151	170		.ASCII	/f /
001656	145	144	040		.ASCII	/ed /
001661	104	151	163		.ASCII	/Dis/

K4

ZRQAM1
VOL.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

001664	153	042	040	.ASCII	/k" /
001667	157	160	145	.ASCII	/ope/
001672	162	141	164	.ASCII	/rat/
001675	151	157	156	.ASCII	/ion/
001700	163	040	157	.ASCII	/s o/
001703	165	164	040	.ASCII	/ut /
001706	157	146	040	.ASCII	/of /
001711	164	157	164	.ASCII	/tot/
001714	141	154	040	.ASCII	/al /
001717	157	160	145	.ASCII	/ope/
001722	162	141	164	.ASCII	/rat/
001725	151	157	156	.ASCII	/ion/
001730	163	000		.ASCII	/s'<00>
001732	125	156	151	P.AAZ: .ASCII	/Uii/
001735	164	163	040	.ASCII	/t /
001740	164	157	040	.ASCII	/ti /
001743	142	145	040	.ASCII	/bi /
001746	163	145	154	.ASCII	/sel/
001751	145	143	164	.ASCII	/ect/
001754	145	144	040	.ASCII	/ed /
001757	141	164	040	.ASCII	/at /
001762	162	141	156	.ASCII	/ran/
001765	144	157	155	.ASCII	/dom/
001770	040	050	116	.ASCII	/ (N/
001773	157	054	040	.ASCII	/o, /
001776	151	155	160	.ASCII	/imp/
002001	154	151	145	.ASCII	/lie/
002004	163	040	163	.ASCII	/s s/
002007	145	161	165	.ASCII	/equ/
002012	145	156	164	.ASCII	/ent/
002015	151	141	154	.ASCII	/ial/
002020	051	000		.ASCII	/)/<00>
002022	127	141	156	P.ABA: .ASCII	/Wan/
002025	164	040	164	.ASCII	/t t/
002030	157	040	162	.ASCII	/o n/
002033	145	167	162	.ASCII	/ewr/
002036	151	164	145	.ASCII	/ite/
002041	040	142	154	.ASCII	/ bl/
002044	157	143	153	.ASCII	/ock/
002047	163	040	167	.ASCII	/s w/
002052	150	145	156	.ASCII	/hen/
002055	040	042	106	.ASCII	/ "v/
002060	157	162	143	.ASCII	/orc/
002063	145	144	040	.ASCII	/ed /
002066	105	162	162	.ASCII	/enn/
002071	157	162	042	.ASCII	/on"/
002074	040	144	145	.ASCII	/ de/
002077	164	145	143	.ASCII	/tec/
002102	164	145	144	.ASCII	/ted/
002105	040	157	156	.ASCII	/ on/
002110	040	167	145	.ASCII	/ re/
002113	141	144	163	.ASCII	/nds/
002116	000	000		.ASCII	<00><00>

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11 Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0050
Page 50
(34)

002120	104	157	040	P.ABB:	.ASCII	/Do /
002123	171	157	165		.ASCII	/you/
002126	040	167	141		.ASCII	/ wa/
002131	156	164	040		.ASCII	/nt /
002134	164	157	040		.ASCII	/to /
002137	150	141	154		.ASCII	/hal/
002142	164	040	157		.ASCII	/t o/
002145	156	040	157		.ASCII	/n o/
002150	164	150	145		.ASCII	/the/
002153	162	040	110		.ASCII	/r H/
002156	141	162	144		.ASCII	/ard/
002161	040	105	162		.ASCII	/ Er/
002164	162	157	162		.ASCII	/ror/
002167	163	040	050		.ASCII	/s (/
002172	043	163	040		.ASCII	/ts /
002175	063	061	055		.ASCII	/31-/
002200	063	064	054		.ASCII	/34,/
002203	040	063	066		.ASCII	/ 36/
002206	055	063	067		.ASCII	/ 37/
002211	054	040	063		.ASCII	/, 3/
002214	071	055	064		.ASCII	/9-4/
002217	065	051	000		.ASCII	/5)/<00>
002222	104	157	040	P.ABC:	.ASCII	/Do /
002225	171	157	165		.ASCII	/you/
002230	040	167	141		.ASCII	/ wa/
002233	156	164	040		.ASCII	/nt /
002236	164	157	040		.ASCII	/to /
002241	150	141	154		.ASCII	/hal/
002244	164	040	157		.ASCII	/t o/
002247	156	040	123		.ASCII	/n S/
002252	157	146	164		.ASCII	/oft/
002255	040	105	162		.ASCII	/ Er/
002260	162	157	162		.ASCII	/ror/
002263	163	040	050		.ASCII	/s (/
002266	043	163	040		.ASCII	/ts /
002271	065	060	055		.ASCII	/50-/
002274	065	064	051		.ASCII	/54)/
002277	000				.ASCII	<00>
002300	104	157	040	P.ABD:	.ASCII	/Do /
002303	171	157	165		.ASCII	/you/
002306	040	167	141		.ASCII	/ wa/
002311	156	164	040		.ASCII	/nt /
002314	164	157	040		.ASCII	/to /
002317	150	141	154		.ASCII	/hal/
002322	164	040	157		.ASCII	/t o/
002325	156	040	102		.ASCII	/n B/
002330	141	144	055		.ASCII	/ad /
002333	142	154	157		.ASCII	/blo/
002336	143	153	040		.ASCII	/ck /
002341	110	141	162		.ASCII	/Har/
002344	144	040	105		.ASCII	/d t/
002347	162	162	157		.ASCII	/rro/
002352	162	163	040		.ASCII	/rs /

M4

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-10 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0051
Page 51
(34)

002355	050	043	163	.ASCII	/(#s/
002360	040	063	065	.ASCII	/ 35/
002363	054	040	063	.ASCII	/, 3/
002366	070	051	000	.ASCII	/8)/<00>
002371	000			.ASCII	<00>
002372	105	156	164	P.ABE: .ASCII	/Ent/
002375	145	162	040	.ASCII	/er /
002400	164	151	155	.ASCII	/tim/
002403	145	040	141	.ASCII	/e a/
002406	163	040	110	.ASCII	/s H/
002411	110	115	115	.ASCII	/HMM/
002414	040	050	105	.ASCII	/ (E/
002417	170	141	155	.ASCII	/xam/
002422	160	154	145	.ASCII	/ple/
002425	072	040	061	.ASCII	/: 1/
002430	063	060	065	.ASCII	/305/
002433	040	055	040	.ASCII	/ - /
002436	040	116	157	.ASCII	/ No/
002441	040	154	145	.ASCII	/ le/
002444	141	144	151	.ASCII	/adi/
002447	156	147	040	.ASCII	/ng /
002452	172	145	162	.ASCII	/zer/
002455	157	163	040	.ASCII	/os /
002460	141	154	154	.ASCII	/all/
002463	157	167	144	.ASCII	/owd/
002466	051	000		.ASCII	/)/<00>
002470	103	157	165	P.ABF: .ASCII	/Cou/
002473	156	164	040	.ASCII	/nt /
002476	145	141	143	.ASCII	/eac/
002501	150	040	162	.ASCII	/h r/
002504	145	164	162	.ASCII	/etr/
002507	171	040	157	.ASCII	/y o/
002512	156	040	141	.ASCII	/n a/
002515	040	122	145	.ASCII	/ Re/
002520	141	144	057	.ASCII	/ad, <57>
002523	127	162	151	.ASCII	/Wri/
002526	164	145	040	.ASCII	/te /
002531	145	162	162	.ASCII	/err/
002534	157	162	040	.ASCII	/or /
002537	141	163	040	.ASCII	/as /
002542	141	040	163	.ASCII	/a s/
002545	145	160	145	.ASCII	/epe/
002550	162	141	164	.ASCII	/nat/
002553	145	040	123	.ASCII	/e S/
002556	157	146	164	.ASCII	/oft/
002561	040	105	162	.ASCII	/ Er/
002564	162	157	162	.ASCII	/nor/
002567	000			.ASCII	<00>
002570	124	150	145	P.ABG: .ASCII	/The/
002573	040	162	145	.ASCII	/ re/
002576	155	141	151	.ASCII	/mai/
002601	156	151	156	.ASCII	/nin/
002604	147	040	161	.ASCII	/g q/

N4

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWER\$]ZRQADO.BL1;6SEQ 0052
Page 52
(34)

002607	165	145	163	.ASCII	/ues/
002612	164	151	157	.ASCII	/tio/
002615	156	163	040	.ASCII	/ns /
002620	157	156	154	.ASCII	/onl/
002623	171	040	141	.ASCII	/y a/
002626	160	160	154	.ASCII	/ppl/
002631	171	040	164	.ASCII	/y t/
002634	157	040	165	.ASCII	/o u/
002637	156	160	162	.ASCII	/npr/
002642	157	164	145	.ASCII	/ote/
002645	143	164	145	.ASCII	/cte/
002650	144	040	144	.ASCII	/d d/
002653	151	163	153	.ASCII	/isk/
002656	163	000		.ASCII	/s/<00>
002660	000	000		P.ABH:	.ASCII <00><00>
002662	045	116	045	P.ABI:	.ASCII /N#/
002665	101	052	052		.ASCII /A**/
002670	040	104	162		.ASCII / Dr/
002673	157	160	040		.ASCII /op /
002676	165	156	151		.ASCII /uni/
002701	164	040	045		.ASCII /t #/
002704	104	062	000		.ASCII /D2/<00>
002707	000				.ASCII <00>
002710	045	116	045	P.ABJ:	.ASCII /N#/
002713	101	052	052		.ASCII /A**/
002716	040	120	122		.ASCII / PR/
002721	117	103	137		.ASCII /OC /
002724	122	105	124		.ASCII /REI/
002727	120	113	124		.ASCII /PKI/
002732	072	040	103		.ASCII /: C/
002735	157	156	156		.ASCII /onn/
002740	040	111	104		.ASCII / ID/
002743	040	045	117		.ASCII / #0/
002746	066	045	101		.ASCII /6#A/
002751	040	162	145		.ASCII / re/
002754	143	145	151		.ASCII /cei/
002757	166	145	144		.ASCII /ved/
002762	000	000			.ASCII <00><00>
002764	045	116	045	P.ABK:	.ASCII /N#/
002767	101	052	052		.ASCII /A**/
002772	040	115	165		.ASCII / Mu/
002775	154	164	151		.ASCII /lti/
003000	055	144	162		.ASCII /-dr/
003003	151	166	145		.ASCII /ive/
003006	040	164	145		.ASCII / te/
003011	163	164	000		.ASCII /st/<00>
003014	045	116	045	P.ABL:	.ASCII /N#/
003017	101	052	052		.ASCII /A**/
003022	040	106	101		.ASCII / FA/
003025	124	101	114		.ASCII /TAL /
003030	137	105	122		.ASCII / ER/
003033	122	117	122		.ASCII /RQH/
003036	072	040	122		.ASCII /: R/

BL

ZRQAM1
V01.6

RD RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B11a 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL116

SEQ 0053
Page 53
(34)

003041	105	124	120	.ASCII	/ETP/
003044	113	124	040	.ASCII	/KI /
003047	156	157	164	.ASCII	/not/
003052	040	141	166	.ASCII	/ av/
003055	141	151	154	.ASCII	/nil/
003060	141	142	154	.ASCII	/abl/
003063	145	000	000	.ASCII	/e/<00><00>
003066	045	116	045	P,ABM: .ASCII	/Nw/
003071	101	052	052	.ASCII	/A++/
003074	040	106	123	.ASCII	/FS/
003077	105	124	137	.ASCII	/ET /
003102	125	120	101	.ASCII	/UPA/
003105	122	072	040	.ASCII	/R: /
003110	103	141	156	.ASCII	/Can/
003113	047	164	040	.ASCII	/t /
003116	146	151	156	.ASCII	/fin/
003121	144	040	144	.ASCII	/d d/
003124	151	163	153	.ASCII	/sk/
003127	040	045	104	.ASCII	/ #D/
003132	063	045	101	.ASCII	/3#A/
003135	040	151	156	.ASCII	/ in/
003140	040	103	123	.ASCII	/CS/
003143	124	040	045	.ASCII	/T #/
003146	104	061	000	.ASCII	/D1/<00>
003151	000			.ASCII	<00>
003152	045	116	045	P,ABN: .ASCII	/Nw/
003155	101	052	052	.ASCII	/A++/
003160	040	102	141	.ASCII	/Ba/
003163	144	040	143	.ASCII	/d c/
003166	157	156	156	.ASCII	/onn/
003171	040	111	104	.ASCII	/ID/
003174	040	045	117	.ASCII	/ #O/
003177	066	045	101	.ASCII	/6#A/
003202	040	162	145	.ASCII	/ re/
003205	143	145	151	.ASCII	/cei/
003210	166	145	144	.ASCII	/ved/
003213	040	146	162	.ASCII	/fr/
003216	157	155	040	.ASCII	/om /
003221	045	117	066	.ASCII	/#06/
003224	000	000		.ASCII	<00><00>
003226	045	116	045	P,ABU: .ASCII	/Nw/
003231	101	052	052	.ASCII	/A++/
003234	040	115	145	.ASCII	/Me/
003237	163	163	141	.ASCII	/ssa/
003242	147	145	040	.ASCII	/ge /
003245	164	171	160	.ASCII	/typ/
003250	145	040	045	.ASCII	/e #/
003253	117	062	045	.ASCII	/U2#/
003256	101	040	162	.ASCII	/A n/
003261	145	143	145	.ASCII	/ece/
003264	151	166	145	.ASCII	/ive/
003267	144	040	151	.ASCII	/d /
003272	156	040	115	.ASCII	/n M/

C1

ZRQAM1
VOL.6

RD RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1133 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0054
Page 54
(34)

003275	123	103	120		.ASCII	/SCP/
003300	040	160	141		.ASCII	/ pa/
003303	143	153	145		.ASCII	/cke/
003306	164	000			.ASCII	/t/<00>
003310	045	116	045	P.ABP:	.ASCII	/N#
003313	101	052	052		.ASCII	/A**/
003316	040	123	105		.ASCII	/ SE/
003321	121	125	105		.ASCII	/QUE/
003324	116	072	040		.ASCII	/N: /
003327	122	105	124		.ASCII	/RET/
003332	120	113	124		.ASCII	/PKI/
003335	040	156	157		.ASCII	/ no/
003340	164	040	141		.ASCII	/t a/
003343	166	141	151		.ASCII	/vai/
003346	154	141	142		.ASCII	/lab/
003351	154	145	000		.ASCII	/le/<00>
003354	045	116	045	P.ABQ:	.ASCII	/N#
003357	101	052	052		.ASCII	/A**/
003362	040	105	162		.ASCII	/ Er/
003365	162	157	162		.ASCII	/non/
003370	040	151	156		.ASCII	/ in/
003373	040	123	105		.ASCII	/ SE/
003376	124	137	103		.ASCII	/I C/
003401	124	114	122		.ASCII	/TLR/
003404	137	103	110		.ASCII	/ CH/
003407	101	122	000		.ASCII	/AR/<00>
003412	045	116	045	P.ABR:	.ASCII	/N#
003415	101	052	052		.ASCII	/A**/
003420	040	103	164		.ASCII	/ Ct/
003423	154	162	040		.ASCII	/lr /
003426	164	151	155		.ASCII	/tim/
003431	145	157	165		.ASCII	/eou/
003434	164	040	075		.ASCII	/t . /
003437	040	045	104		.ASCII	/ dD/
003442	063	045	101		.ASCII	/3#A/
003445	056	040	163		.ASCII	/ . g/
003450	145	143	157		.ASCII	/eco/
003453	156	144	163		.ASCII	/nds/
003456	000	000			.ASCII	<00><00>
003460	045	116	045	P.ABS:	.ASCII	/N#
003463	101	052	052		.ASCII	/A**/
003466	040	105	162		.ASCII	/ Er/
003471	162	157	162		.ASCII	/non/
003474	040	151	156		.ASCII	/ in/
003477	040	125	116		.ASCII	/ UN/
003502	111	124	137		.ASCII	/IT /
003505	111	116	111		.ASCII	/INI/
003510	124	000			.ASCII	/I/<00>
003512	045	116	045	P.ABT:	.ASCII	/N#
003515	101	052	052		.ASCII	/A**/
003520	040	125	116		.ASCII	/ UN/
003523	111	124	137		.ASCII	/IT /
003526	111	116	111		.ASCII	/INI/

D5

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1:6

SEQ 0055
Page 55
(34)

003531	124	072	040
003534	122	105	124
003537	120	113	124
003542	040	150	141
003545	163	040	142
003550	141	144	040
003553	105	116	104
003556	103	117	104
003561	105	000	000
003564	045	116	045
003567	101	052	052
003572	040	125	156
003575	151	164	040
003600	163	151	172
003603	145	040	050
003606	114	157	051
003611	040	075	040
003614	045	104	065
003617	045	101	056
003622	000	000	
003624	045	116	045
003627	101	052	052
003632	040	125	156
003635	151	164	040
003640	163	151	172
003643	145	040	050
003646	110	151	051
003651	040	075	040
003654	045	104	065
003657	045	101	056
003662	000	000	
003664	045	116	045
003667	101	052	052
003672	040	101	103
003675	103	105	123
003700	123	072	040
003703	122	105	124
003706	120	113	124
003711	040	150	141
003714	163	040	142
003717	141	144	040
003722	105	116	104
003725	103	117	104
003730	105	000	
003732	045	116	045
003735	101	052	052
003740	040	121	111
003743	117	137	125
003746	116	111	124
003751	072	040	103
003754	123	124	040
003757	045	164	061
003762	045	101	040

	.ASCII	/T:/
	.ASCII	/RET/
	.ASCII	/PKT/
	.ASCII	/ha/
	.ASCII	/sb/
	.ASCII	/ad/
	.ASCII	/END/
	.ASCII	/COD/
	.ASCII	/E/<00><00>
P.ABU:	.ASCII	/N#
	.ASCII	/A**/
	.ASCII	/Un/
	.ASCII	/it/
	.ASCII	/siz/
	.ASCII	/e(/
	.ASCII	/Lo)/
	.ASCII	/./
	.ASCII	/D5/
	.ASCII	/A./
	.ASCII	<00><00>
P.ABV:	.ASCII	/N#
	.ASCII	/A**/
	.ASCII	/Un/
	.ASCII	/it/
	.ASCII	/siz/
	.ASCII	/e(/
	.ASCII	/Hi)/
	.ASCII	/./
	.ASCII	/D5/
	.ASCII	/A./
	.ASCII	<00><00>
P.ABW:	.ASCII	/N#
	.ASCII	/A**/
	.ASCII	/AC/
	.ASCII	/CES/
	.ASCII	/S:/
	.ASCII	/RET/
	.ASCII	/PKT/
	.ASCII	/ha/
	.ASCII	/sb/
	.ASCII	/ad/
	.ASCII	/END/
	.ASCII	/COD/
	.ASCII	/E/<00><00>
P.ABX:	.ASCII	/N#
	.ASCII	/A**/
	.ASCII	/DI/
	.ASCII	/DUP/
	.ASCII	/NII/
	.ASCII	/:C/
	.ASCII	/ST/
	.ASCII	/D1/
	.ASCII	/A/

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11gs-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0056
Page 56
(34)

003765	156	157	040	.ASCII	/no /
003770	165	156	151	.ASCII	/uni /
003773	164	040	163	.ASCII	/t s/
003776	145	154	145	.ASCII	/ele/
004001	143	164	145	.ASCII	/cte/
004004	144	000		.ASCII	/d/<00>
004006	045	116	045	P.ABY:	.ASCII /Nw/
004011	101	052	052		.ASCII /A**/
004014	040	125	156		.ASCII / Un/
004017	151	164	040		.ASCII /it /
004022	043	040	151		.ASCII /# i/
004025	163	072	040		.ASCII /s: /
004030	045	117	066		.ASCII /#06/
004033	000				.ASCII <00>
004034	045	116	045	P.ABZ:	.ASCII /Nw/
004037	101	052	052		.ASCII /A**/
004042	040	122	145		.ASCII / Re/
004045	155	157	166		.ASCII /mov/
004050	141	142	154		.ASCII /obl/
004053	145	040	144		.ASCII /e d/
004056	151	163	153		.ASCII /isk/
004061	040	151	163		.ASCII / is/
004064	040	163	145		.ASCII / se/
004067	154	145	143		.ASCII /lec/
004072	164	145	144		.ASCII /ted/
004075	000				.ASCII <0C>
004076	045	116	045	P.ACA:	.ASCII /Nw/
004101	101	052	052		.ASCII /A**/
004104	040	106	151		.ASCII / fi/
004107	170	145	144		.ASCII /xed/
004112	040	144	151		.ASCII / di/
004115	163	153	040		.ASCII /sk /
004120	151	163	040		.ASCII /is /
004123	163	145	154		.ASCII /sel/
004126	145	143	164		.ASCII /ect/
004131	145	144	000		.ASCII /ed/<00>
004134	045	116	045	P.ACB:	.ASCII /Nw/
004137	101	052	052		.ASCII /A**/
004142	040	111	154		.ASCII / ll/
004145	154	145	147		.ASCII /leg/
004150	141	154	040		.ASCII /al /
004153	146	165	156		.ASCII /fun/
004156	143	164	151		.ASCII /cti/
004161	157	156	072		.ASCII /on: /
004164	040	045	117		.ASCII / #0/
004167	066	000	000		.ASCII /#<00><00>
004172	045	116	045	P.ACC:	.ASCII /Nw/
004175	101	052	052		.ASCII /A**/
004200	040	103	157		.ASCII / Co/
004203	155	155	141		.ASCII /mma/
004206	156	144	040		.ASCII /nd /
004211	162	145	146		.ASCII /ref/
004214	040	043	040		.ASCII / # /

F5

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0057
Page 57
(34)

004217	045	117	066	.ASCII	/06/
004222	045	101	057	.ASCII	/A/<57>
004225	045	117	066	.ASCII	/06/
004230	045	101	040	.ASCII	/A /
004233	050	117	143	.ASCII	/(0c/
004236	164	051	040	.ASCII	/t) /
004241	156	157	164	.ASCII	/not/
004244	040	163	145	.ASCII	/ ge/
004247	156	164	040	.ASCII	/nt /
004252	142	171	040	.ASCII	/by /
004255	110	157	163	.ASCII	/Hos/
004260	164	000		.ASCII	/t/<00>
004262	045	116	045	P.ACD: .ASCII	/N# /
004265	101	052	052	.ASCII	/A**/
004270	040	125	156	.ASCII	/ Un/
004273	153	156	157	.ASCII	/kno/
004276	167	156	040	.ASCII	/wn /
004301	105	162	162	.ASCII	/Err/
004304	157	162	040	.ASCII	/or /
004307	114	157	147	.ASCII	/Log/
004312	040	146	157	.ASCII	/ fo/
004315	162	155	141	.ASCII	/rma/
004320	164	040	045	.ASCII	/t #/
004323	117	063	045	.ASCII	/03# /
004326	101	040	162	.ASCII	/A r/
004331	145	143	145	.ASCII	/ece/
004334	151	166	145	.ASCII	/ive/
004337	144	000	000	P.ACE: .ASCII	/d/<00><00>
004342	045	116	045	.ASCII	/N# /
004345	101	052	052	.ASCII	/A**/
004350	040	117	160	.ASCII	/ Op/
004353	055	143	157	.ASCII	/-co/
004356	144	145	040	.ASCII	/de /
004361	045	117	063	.ASCII	/03/
004364	045	101	054	.ASCII	/A, /
004367	040	105	156	.ASCII	/ En/
004372	144	055	143	.ASCII	/d-c/
004375	157	144	145	.ASCII	/ode/
004400	040	045	117	.ASCII	/ #0/
004403	063	045	101	.ASCII	/3#A/
004406	040	146	157	.ASCII	/ fo/
004411	162	040	162	.ASCII	/r n/
004414	145	146	040	.ASCII	/ef /
004417	043	040	045	.ASCII	/q #/
004422	117	066	045	.ASCII	/06# /
004425	101	057	045	.ASCII	/A/<57>/# /
004430	117	066	045	.ASCII	/06# /
004433	101	040	050	.ASCII	/A /
004436	070	051	000	.ASCII	/R/<00>
004441	000			.ASCII	<00>
004442	045	116	045	P.ACE: .ASCII	/N# /
004445	101	052	052	.ASCII	/A**/
004450	040	103	155	.ASCII	/ Cm/

65

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11 Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1199 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0058
Page 58
(34)

004453	144	055	142	.ASCII	/d-b/
004456	143	040	045	.ASCII	/c #/
004461	117	066	045	.ASCII	/06#/
004464	101	057	045	.ASCII	/A/<57>/#/
004467	117	066	045	.ASCII	/06#/
004472	101	040	122	.ASCII	/A R/
004475	163	160	055	.ASCII	/sp /
004500	142	143	040	.ASCII	/bc /
004503	045	117	066	.ASCII	/#06/
004506	045	101	057	.ASCII	/#A/<57>
004511	045	117	066	.ASCII	/#06/
004514	045	101	040	.ASCII	/#A /
004517	146	157	162	.ASCII	/for/
004522	040	045	117	.ASCII	/ #0/
004525	066	045	101	.ASCII	/6#A/
004530	057	045	117	.ASCII	<57>/#0/
004533	066	045	101	.ASCII	/6#A/
004536	040	050	070	.ASCII	/ (8/
004541	051	000	000	.ASCII	/)/<00><00>
004544	045	116	045	P.ACG: .ASCII	/#N#/
004547	101	052	052	.ASCII	/A**/
004552	040	122	145	.ASCII	/ Re/
004555	163	166	157	.ASCII	/spo/
004560	156	163	145	.ASCII	/nse/
004563	040	141	154	.ASCII	/ al/
004566	162	145	141	.ASCII	/rea/
004571	144	171	040	.ASCII	/dy /
004574	162	145	143	.ASCII	/rec/
004577	145	151	166	.ASCII	/eiv/
004602	145	144	040	.ASCII	/ed /
004605	146	157	162	.ASCII	/for/
004610	040	143	155	.ASCII	/ cm/
004613	144	040	045	.ASCII	/d #/
004616	117	066	045	.ASCII	/06#/
004621	101	057	045	.ASCII	/A/<57>/#/
004624	117	066	045	.ASCII	/06#/
004627	101	040	050	.ASCII	/A (/
004632	070	051	000	.ASCII	/8)/<00>
004635	000			.ASCII	<00>
004636	045	116	045	P.ACH: .ASCII	/#N#/
004641	101	052	052	.ASCII	/A**/
004644	040	106	141	.ASCII	/ fa/
004647	151	154	165	.ASCII	/ilu/
004652	162	145	040	.ASCII	/re /
004655	164	157	040	.ASCII	/to /
004660	163	145	156	.ASCII	/sen/
004663	144	040	143	.ASCII	/d c/
004666	157	155	155	.ASCII	/omm/
004671	141	156	144	.ASCII	/and/
004674	040	141	146	.ASCII	/ af/
004677	164	145	162	.ASCII	/ter/
004702	040	043	040	.ASCII	/ # /
004705	045	117	066	.ASCII	/#06/

H5

ZRDAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Release 16 V4.0-579
DISK\$USER2:[POWERS]ZRDAD0.8L1;6SEQ 0059
Page 59
(34)

004710	045	101	057		.ASCII	/A/<57>
004713	045	117	060		.ASCII	/06/
004716	045	101	040		.ASCII	/A /
004721	050	070	051		.ASCII	/(8)/
004724	000	000			.ASCII	<00><00>
004726	045	116	045	P.ACI:	.ASCII	/N/
004731	101	125	116		.ASCII	/AUN/
004734	111	124	045		.ASCII	/IT/
004737	104	062	045		.ASCII	/D2/
004742	101	040	104		.ASCII	/A D/
004745	122	117	120		.ASCII	/ROP/
004750	120	105	104		.ASCII	/PED/
004753	040	055	040		.ASCII	/ /
004756	000	000			.ASCII	<00><00>
004760	045	101	125	P.ACK:	.ASCII	/AU/
004763	123	105	122		.ASCII	/SER/
004766	040	103	117		.ASCII	/ CO/
004771	115	115	101		.ASCII	/MMA/
004774	116	104	045		.ASCII	/ND/
004777	116	000	000		.ASCII	/N/<00><00>
005002	045	101	103	P.ACL:	.ASCII	/AC/
005005	117	116	106		.ASCII	/ONF/
005010	111	107	125		.ASCII	/IGU/
005013	122	101	124		.ASCII	/RAT/
005016	111	117	116		.ASCII	/ION/
005021	040	105	122		.ASCII	/ ER/
005024	122	117	122		.ASCII	/ROR/
005027	045	116	000		.ASCII	/N/<00>
005032	045	101	111	P.ACM:	.ASCII	/AI/
005035	116	111	124		.ASCII	/NIT/
005040	040	105	122		.ASCII	/ ER/
005043	122	117	122		.ASCII	/ROR/
005046	045	116	000		.ASCII	/N/<00>
005051	000				.ASCII	<00>
005052	045	101	124	P.ACN:	.ASCII	/AT/
005055	122	101	116		.ASCII	/RAN/
005060	123	106	105		.ASCII	/SFE/
005063	122	040	114		.ASCII	/R L/
005066	111	115	111		.ASCII	/IMI/
005071	124	040	122		.ASCII	/T R/
005074	105	101	103		.ASCII	/EAC/
005077	110	105	104		.ASCII	/HED/
005102	045	116	000		.ASCII	/N/<00>
005105	000				.ASCII	<00>
005106	045	101	105	P.ACO:	.ASCII	/AI/
005111	122	122	117		.ASCII	/RRO/
005114	122	040	114		.ASCII	/R L/
005117	111	115	111		.ASCII	/IMI/
005122	124	040	122		.ASCII	/T R/
005125	105	101	103		.ASCII	/EAC/
005130	110	105	104		.ASCII	/HED/
005133	045	116	000		.ASCII	/N/<00>
005136	045	101	125	P.ACP:	.ASCII	/AU/

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0060
Page 60
(34)

005141	116	122	105	.ASCII	/NRE/
005144	103	117	126	.ASCII	/COV/
005147	105	122	101	.ASCII	/ERA/
005152	102	114	105	.ASCII	/BLE/
005155	040	104	122	.ASCII	/ DR/
005160	111	126	105	.ASCII	/IVE/
005163	040	105	122	.ASCII	/ ER/
005166	122	117	122	.ASCII	/ROR/
005171	045	116	000	.ASCII	/N/<00>
005174	045	101	125	P.ACQ: .ASCII	/AU/
005177	116	122	105	.ASCII	/NRE/
005202	103	117	126	.ASCII	/COV/
005205	105	122	101	.ASCII	/ERA/
005210	102	114	105	.ASCII	/BLE/
005213	040	103	117	.ASCII	/ CC/
005216	116	124	122	.ASCII	/NTR/
005221	117	114	114	.ASCII	/OLL/
005224	105	122	040	.ASCII	/ER /
005227	105	122	122	.ASCII	/ERR/
005232	117	122	045	.ASCII	/OR*/
005235	116	000	000	.ASCII	/N/<00><00>
005240	045	101	106	P.ACR: .ASCII	/AF/
005243	101	111	114	.ASCII	/AIL/
005246	105	104	040	.ASCII	/ED /
005251	124	117	040	.ASCII	/TO /
005254	103	117	115	.ASCII	/COM/
005257	105	040	117	.ASCII	/E O/
005262	116	114	111	.ASCII	/NLI/
005265	116	105	045	.ASCII	/NE*/
005270	116	000		.ASCII	/N/<00>
005272	045	101	106	P.ACS: .ASCII	/AF/
005275	101	111	114	.ASCII	/AIL/
005300	105	104	040	.ASCII	/ED /
005303	124	117	040	.ASCII	/TO /
005306	101	103	103	.ASCII	/ACC/
005311	105	123	123	.ASCII	/ESS/
005314	040	105	111	.ASCII	/ EI/
005317	124	110	105	.ASCII	/THE/
005322	122	040	106	.ASCII	/R I/
005325	111	122	123	.ASCII	/IRS/
005330	124	040	117	.ASCII	/T O/
005333	122	040	114	.ASCII	/R I/
005336	101	123	124	.ASCII	/AGI/
005341	040	124	122	.ASCII	/ IR/
005344	101	103	113	.ASCII	/ACK/
005347	040	104	125	.ASCII	/ DU/
005352	122	111	116	.ASCII	/RIN/
005355	107	040	111	.ASCII	/G I/
005360	116	111	124	.ASCII	/NIT/
005363	045	116	000	.ASCII	/N/<00>
005366	045	101	104	P.ACT: .ASCII	/AD/
005371	111	123	113	.ASCII	/ISK/
005374	040	127	122	.ASCII	/ WR/

J5

ZRRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRRQADO.BL1;6SEQ 0061
Page 61
(34)

005377	111	124	105	.ASCII	/ITE/
005402	040	120	122	.ASCII	/PR/
005405	117	124	105	.ASCII	/OTE/
005410	103	124	105	.ASCII	/CTE/
005413	104	045	116	.ASCII	/D#N/
005416	000	000		.ASCII	<00><00>
005420	045	101	103	P.ACU:	.ASCII
005423	117	115	115		.ASCII
005426	101	116	104		.ASCII
005431	040	124	111		.ASCII
005434	115	105	040		.ASCII
005437	117	125	124		.ASCII
005442	045	116	000		.ASCII
005445	000				.ASCII
005446	004760			P.ACJ:	.WORD
005450	005002				.WORD
005452	005032				.WORD
005454	005052				.WORD
005456	005106				.WORD
005460	005136				.WORD
005462	005174				.WORD
005464	005240				.WORD
005466	005272				.WORD
005470	005366				.WORD
005472	005420				.WORD
005474	045	116	045	P.ACV:	.ASCII
005477	101	120	117		.ASCII
005502	127	105	122		.ASCII
005505	040	104	105		.ASCII
005510	114	101	131		.ASCII
005513	040	055	040		.ASCII
005516	127	101	111		.ASCII
005521	124	111	116		.ASCII
005524	107	000			.ASCII
005526	045	116	045	P.ACW:	.ASCII
005531	101	106	125		.ASCII
005534	116	103	124		.ASCII
005537	111	117	116		.ASCII
005542	101	114	040		.ASCII
005545	124	105	123		.ASCII
005550	124	040	123		.ASCII
005553	124	101	122		.ASCII
005556	124	105	104		.ASCII
005561	000				.ASCII
005562	045	116	045	P.ACX:	.ASCII
005565	116	045	101		.ASCII
005570	105	130	105		.ASCII
005573	123	103	111		.ASCII
005576	123	105	124		.ASCII
005601	040	123	124		.ASCII
005604	101	122	124		.ASCII
005607	105	104	045		.ASCII
005612	116	000			.ASCII

K⁵

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0052
Page 62
(34)

005614	045	116	045	P.ACY:	.ASCII	/N#
005617	116	045	101		.ASCII	/N#A/
005622	125	116	124		.ASCII	/UNT/
005625	040	104	123		.ASCII	/DS/
005630	113	045	123		.ASCII	/K#5/
005633	070	045	101		.ASCII	/8#A/
005636	043	040	117		.ASCII	/O/
005641	106	040	040		.ASCII	/F/
005644	040	043	040		.ASCII	/O/
005647	102	131	124		.ASCII	/BYT/
005652	105	123	040		.ASCII	/ES/
005655	040	040	043		.ASCII	/O/
005660	040	117	106		.ASCII	/OF/
005663	040	040	040		.ASCII	/O/
005666	040	043	040		.ASCII	/O/
005671	102	131	124		.ASCII	/BYT/
005674	105	123	000		.ASCII	/ES/<00>
005677	000				.ASCII	<00>
005700	045	101	040	P.ACZ:	.ASCII	/A/
005703	040	055	055		.ASCII	/O/
005706	110	101	122		.ASCII	/HAR/
005711	104	040	105		.ASCII	/DE/
005714	122	122	117		.ASCII	/RRO/
005717	122	123	055		.ASCII	/RS-/
005722	055	040	055		.ASCII	/O/
005725	055	123	117		.ASCII	/SO/
005730	106	124	040		.ASCII	/FT/
005733	105	122	122		.ASCII	/ERR/
005736	117	122	123		.ASCII	/ORS/
005741	055	055	000		.ASCII	/O/<00>
005744	045	116	045	P.ADA:	.ASCII	/N#
005747	101	040	043		.ASCII	/A O/
005752	040	040	040		.ASCII	/O/
005755	043	040	040		.ASCII	/O/
005760	124	131	120		.ASCII	/TYP/
005763	105	040	040		.ASCII	/F/
005766	122	105	101		.ASCII	/REA/
005771	104	123	040		.ASCII	/DS/
005774	040	040	040		.ASCII	/O/
005777	040	122	105		.ASCII	/RE/
006002	101	04	040		.ASCII	/AD/
006005	040	040	127		.ASCII	/W/
006010	122	111	124		.ASCII	/RIT/
006013	105	123	040		.ASCII	/ES/
006016	040	040	127		.ASCII	/W/
006021	122	111	124		.ASCII	/RIT/
006024	124	105	116		.ASCII	/TEN/
006027	000				.ASCII	<00>
006030	045	101	040	P.ADB:	.ASCII	/A/
006033	040	123	105		.ASCII	/SE/
006036	113	040	104		.ASCII	/K D/
006041	101	124	040		.ASCII	/AT/
006044	104	122	126		.ASCII	/DRV/

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Blist. 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0063
Page 63
(34)

006047	040	110	123	.ASCII	/HS/
006052	124	040	123	.ASCII	/TS/
006055	105	113	040	.ASCII	/EK/
006060	104	101	124	.ASCII	/DAT/
006063	040	104	122	.ASCII	/DR/
006066	126	040	110	.ASCII	/VH/
006071	123	124	000	.ASCII	/ST/<00>
006074	045	116	045	P.ADC:	.ASCII
006077	101	055	055		.ASCII
006102	055	040	055		.ASCII
006105	055	055	040		.ASCII
006110	055	055	055		.ASCII
006113	055	040	040		.ASCII
006116	055	055	055		.ASCII
006121	055	055	040		.ASCII
006124	040	055	055		.ASCII
006127	055	055	055		.ASCII
006132	055	055	055		.ASCII
006135	055	040	055		.ASCII
006140	055	055	055		.ASCII
006143	055	055	040		.ASCII
006146	040	055	055		.ASCII
006151	055	055	055		.ASCII
006154	055	055	055		.ASCII
006157	055	000	000	P.ADD:	.ASCII
006162	045	101	040		.ASCII
006165	055	055	055		.ASCII
006170	040	055	055		.ASCII
006173	055	040	055		.ASCII
006176	055	055	040		.ASCII
006201	055	055	055		.ASCII
006204	040	055	055		.ASCII
006207	055	040	055		.ASCII
006212	055	055	040		.ASCII
006215	055	055	055		.ASCII
006220	040	055	055		.ASCII
006223	055	000	000	P.ADE:	.ASCII
006226	045	116	045		.ASCII
006231	104	062	045		.ASCII
006234	104	064	045		.ASCII
006237	123	062	045		.ASCII
006242	124	000			.ASCII
006244	045	104	064	P.ADF:	.ASCII
006247	045	132	063		.ASCII
006252	045	104	063		.ASCII
006255	045	101	054		.ASCII
006260	045	132	063		.ASCII
006263	045	101	054		.ASCII
006266	045	132	063		.ASCII
006271	000				.ASCII
006272	045	104	064	P.ADG:	.ASCII
006275	045	104	064		.ASCII
006300	045	104	064		.ASCII

M5

SEQ 0064
Page 64
(34)ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

006303	045	104	064	.ASCII	/D4/
006306	045	104	064	.ASCII	/D4/
006311	045	104	064	.ASCII	/D4/
006314	045	104	064	.ASCII	/D4/
006317	045	104	064	.ASCII	/D4/
006322	000	000		.ASCII	<00><00>
006324	045	116	045	P.ADH: .ASCII	/N/
006327	101	040	056	.ASCII	/A /
006332	040	040	040	.ASCII	/ /
006335	056	040	040	.ASCII	/ /
006340	103	116	124	.ASCII	/CNT/
006343	122	040	040	.ASCII	/R /
006346	040	040	040	.ASCII	/ /
006351	040	056	040	.ASCII	/ /
006354	040	056	056	.ASCII	/ /
006357	056	056	056	.ASCII	/ /
006362	056	056	056	.ASCII	/ /
006365	056	040	040	.ASCII	/ /
006370	040	040	040	.ASCII	/ /
006373	040	056	040	.ASCII	/ /
006376	040	056	056	.ASCII	/ /
006401	056	056	056	.ASCII	/ /
006404	056	056	056	.ASCII	/ /
006407	056	000	000	P.ADI: .ASCII	/ /<00><00>
006412	045	101	040	.ASCII	/A /
006415	040	040	056	.ASCII	/ /
006420	040	040	040	.ASCII	/ /
006423	056	045	104	.ASCII	/D/
006426	064	045	101	.ASCII	/4A/
006431	040	040	040	.ASCII	/ /
006434	056	040	040	.ASCII	/ /
006437	040	056	040	.ASCII	/ /
006442	040	040	056	.ASCII	/ /
006445	045	104	064	.ASCII	/D4/
006450	045	101	040	.ASCII	/A /
006453	040	040	056	.ASCII	/ /
006456	000	000		.ASCII	<00><00>
006460	124	117	117	P.ADJ: .ASCII	/T00/
006463	040	115	101	.ASCII	/MA/
006466	116	131	040	.ASCII	/N/
006471	125	116	111	.ASCII	/UNI/
006474	124	123	000	.ASCII	/TS/<00>
006477	000			.ASCII	<00>
006500	116	117	124	P.ADK: .ASCII	/NOT/
006503	040	105	116	.ASCII	/EN/
006506	117	125	107	.ASCII	/OUG/
006511	110	040	106	.ASCII	/H F/
006514	122	105	105	.ASCII	/REE/
006517	040	115	105	.ASCII	/ME/
006522	115	117	122	.ASCII	/MOR/
006525	131	040	106	.ASCII	/Y F/
006530	117	122	040	.ASCII	/OR /
006533	101	114	114	.ASCII	/ALL/

N5

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6SEQ 0065
Page 65
(34)

006536	117	103	101	.ASCII	/OCA/
006541	124	111	116	.ASCII	/TIN/
006544	107	040	122	.ASCII	/G R/
006547	105	101	104	.ASCII	/EAD/
006552	057	127	122	.ASCII	<57>/WR/
006555	111	124	105	.ASCII	/ITE/
006560	040	102	125	.ASCII	/BU/
006563	106	106	105	.ASCII	/FFE/
006566	122	123	000	.ASCII	/RS/<00>
006571	000			.ASCII	<00>
006572	122	105	107	P.ADL:	.ASCII /REG/
006575	111	123	124		.ASCII /IS/
006600	105	122	040		.ASCII /ER /
006603	105	130	111		.ASCII /EXI/
006606	123	124	105		.ASCII /STE/
006611	116	103	105		.ASCII /NCE/
006614	040	124	105		.ASCII /TE/
006617	123	124	040		.ASCII /ST /
006622	106	101	111		.ASCII /FAI/
006625	114	105	104		.ASCII /LED/
006630	000	000			.ASCII <00><00>
006632	126	105	103	P.ADM:	.ASCII /VEC/
006635	124	117	122		.ASCII /TOR/
006640	040	124	105		.ASCII /TE/
006643	123	124	040		.ASCII /ST /
006646	106	101	111		.ASCII /FAI/
006651	114	105	104		.ASCII /LED/
006654	000	000			.ASCII <00><00>
006656	102	122	040	P.ADN:	.ASCII /BR /
006661	114	105	126		.ASCII /LEV/
006664	105	114	040		.ASCII /EL /
006667	124	105	123		.ASCII /TFS/
006672	124	040	106		.ASCII /T F/
006675	101	111	114		.ASCII /AIL/
006700	105	104	000		.ASCII /ED/<00>
006703	000				.ASCII <00>
006704	111	116	111	P.ADO:	.ASCII /INI/
006707	124	040	123		.ASCII /T S/
006712	105	121	125		.ASCII /EQU/
006715	105	116	103		.ASCII /ENC/
006720	105	040	106		.ASCII /E F/
006723	101	111	114		.ASCII /AIL/
006726	105	104	000		.ASCII /ED/<00>
006731	000				.ASCII <00>
006732	106	101	124	P.ADP:	.ASCII /EAT/
006735	101	114	040		.ASCII /AL /
006740	103	117	116		.ASCII /CON/
006743	124	122	117		.ASCII /TRO/
006746	114	114	105		.ASCII /LE /
006751	122	040	105		.ASCII /R F/
006754	122	122	117		.ASCII /RR0/
006757	122	000	000		.ASCII /R/<00><00>
006762	117	116	114	P.ADQ:	.ASCII /ONL/

135

ZRQAM1
VOL.6

RD BY EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1150 16 V4.0 5/9
DISK\$USER2:(POWER5)ZRQADO.BL1:6

SEQ 0066
Page 66
(34)

006765	111	116	105	.ASCII	/INE/	
006770	040	106	101	.ASCII	/FA/	
006773	111	114	105	.ASCII	/ILE/	
006776	104	000		.ASCII	/D/<00>	
007000	127	127	111	P.ADR:	.ASCII	/WRI/
007003	124	105	055	.ASCII	/TE/	
007006	120	127	117	.ASCII	/PRO/	
007011	124	105	103	.ASCII	/TEC/	
007014	124	040	103	.ASCII	/T C/	
007017	117	116	106	.ASCII	/ONE/	
007022	114	111	103	.ASCII	/LIC/	
007025	124	000	000	.ASCII	/T/<00><00>	
007030	101	103	103	P.ADS:	.ASCII	/ACC/
007033	105	123	123	.ASCII	/ESS/	
007036	040	106	101	.ASCII	/FA/	
007041	111	114	105	.ASCII	/ILE/	
007044	104	000		.ASCII	/D/<00>	
007046	106	101	124	P.ADT:	.ASCII	/EAT/
007051	101	114	040	.ASCII	/AL/	
007054	111	057	117	.ASCII	/I/<57>/D/	
007057	040	105	122	.ASCII	/ER/	
007062	127	117	122	.ASCII	/ROR/	
007065	000			.ASCII	<00>	
007066	104	111	123	P.ADU:	.ASCII	/DIS/
007071	113	040	124	.ASCII	/K T/	
007074	131	120	105	.ASCII	/YPE/	
007077	040	125	116	.ASCII	/UN/	
007107	113	116	117	.ASCII	/KNO/	
007105	127	116	040	.ASCII	/WN/	
007110	124	117	040	.ASCII	/TO/	
007113	105	130	105	.ASCII	/EXE/	
007116	122	103	111	.ASCII	/RCI/	
007121	123	105	122	.ASCII	/SER/	
007124	000	000		.ASCII	<00><00>	
007126	106	101	111	P.ADV:	.ASCII	/EAT/
007131	114	105	104	.ASCII	/LED/	
007134	040	124	117	.ASCII	/TO/	
007137	040	123	105	.ASCII	/SE/	
007142	116	104	040	.ASCII	/ND/	
007145	123	105	124	.ASCII	/SET/	
007150	055	103	117	.ASCII	/CO/	
007153	116	124	122	.ASCII	/NTR/	
007156	117	114	114	.ASCII	/OLE/	
007161	105	122	055	.ASCII	/ER/	
007164	103	110	101	.ASCII	/CHA/	
007167	122	101	103	.ASCII	/RAC/	
007172	124	105	122	.ASCII	/TER/	
007175	111	123	124	.ASCII	/IST/	
007200	111	103	123	.ASCII	/ICH/	
007203	040	103	117	.ASCII	/CO/	
007206	115	115	101	.ASCII	/MMA/	
007211	116	104	000	.ASCII	/ND<00>	
007214	123	105	124	P.ADW:	.ASCII	/SET/

CF

ZRQAM1
V01.6

RD-RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1199 16 V4.0 5/9
DISK\$USER2:[POWER5]ZRQADO.BL1:6

SEQ 0067
Page 67
(34)

007217	055	103	117	.ASCII	/CO/
007218	116	124	122	.ASCII	/NTR/
007219	117	114	114	.ASCII	/OLL/
007230	105	122	055	.ASCII	/ER /
007233	103	110	101	.ASCII	/CHA/
007236	122	101	103	.ASCII	/RAC/
007241	124	105	122	.ASCII	/TER/
007244	111	123	124	.ASCII	/ISI/
007247	111	103	123	.ASCII	/ICS/
007252	040	122	105	.ASCII	/RE/
007255	123	120	117	.ASCII	/SP0/
007260	116	123	105	.ASCII	/NSE/
007263	040	110	101	.ASCII	/HA/
007266	123	040	102	.ASCII	/S B/
007271	101	104	040	.ASCII	/AD /
007274	105	116	104	.ASCII	/END/
007277	103	117	104	.ASCII	/COD/
007302	105	040	117	.ASCII	/E O/
007305	122	040	106	.ASCII	/R F/
007310	114	101	107	.ASCII	/LAG/
007313	123	040	111	.ASCII	/S I/
007316	116	040	105	.ASCII	/N E/
007321	122	122	117	.ASCII	/RR0/
007324	122	000		.ASCII	/R/ <00>
007326	106	101	111	P,ADx: .ASCII	/FAI/
007331	114	105	104	.ASCII	/LED/
007334	040	124	117	.ASCII	/TO/
007337	040	123	105	.ASCII	/SE/
007342	116	104	040	.ASCII	/ND /
007345	117	116	055	.ASCII	/ON /
007350	114	111	116	.ASCII	/LIN/
007353	105	040	103	.ASCII	/E C/
007356	117	115	115	.ASCII	/OMM/
007361	101	116	104	.ASCII	/AND/
007364	000	000		.ASCII	<00> <00>
007366	117	116	055	P,ADy: .ASCII	/ON /
007371	114	111	116	.ASCII	/LIN/
007374	105	040	122	.ASCII	/E R/
007377	105	123	120	.ASCII	/ESP/
007402	117	116	123	.ASCII	/ONS/
007405	105	040	110	.ASCII	/E H/
007410	101	123	040	.ASCII	/AS /
007413	102	101	104	.ASCII	/BAD/
007416	040	105	116	.ASCII	/EN/
007421	104	103	117	.ASCII	/DC0/
007424	104	105	000	.ASCII	/DE <00>
007427	000			.ASCII	<00>
007430	117	116	055	P,ADz: .ASCII	/ON /
007433	114	111	116	.ASCII	/LIN/
007436	105	040	122	.ASCII	/E R/
007441	105	123	120	.ASCII	/ESP/
007444	117	116	123	.ASCII	/ONS/
007447	105	040	110	.ASCII	/E H/

106

ZRQAM1
VOL 6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B11:16 V4.0.5.79
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0068
Page 68
(34)

007452	101	123	040	.ASCII	/AS /
007455	125	116	113	.ASCII	/UNK/
007460	116	117	127	.ASCII	/NOW/
007463	116	040	104	.ASCII	/N D/
007466	105	126	111	.ASCII	/EVI/
007471	103	105	000	.ASCII	/CF/<00>
007474	111	057	117	P.AEA: .ASCII	/I/<57>/O/
007477	040	122	105	.ASCII	/ RE/
007502	121	125	105	.ASCII	/QUE/
007505	123	124	040	.ASCII	/ST /
007510	106	101	111	.ASCII	/FAI/
007513	114	105	104	.ASCII	/LFD/
007516	000	000		.ASCII	<00><00>
007520	045	101	115	P.AEB: .ASCII	/NAM/
007523	117	122	105	.ASCII	/ORE/
007526	040	124	110	.ASCII	/ TH/
007531	101	116	040	.ASCII	/AN /
007534	045	104	062	.ASCII	/SD2/
007537	045	101	040	.ASCII	/SA /
007542	125	116	111	.ASCII	/UNI/
007545	124	123	040	.ASCII	/TS /
007550	123	120	105	.ASCII	/SPE/
007553	103	111	106	.ASCII	/CIF/
007556	111	105	104	.ASCII	/IED/
007561	000			.ASCII	<00>
007562	045	101	052	P.AEC: .ASCII	/AA*/
007565	040	116	117	.ASCII	/ NO/
007570	040	122	105	.ASCII	/ RE/
007573	123	120	117	.ASCII	/SPO/
007576	116	123	105	.ASCII	/NSE/
007601	040	101	124	.ASCII	/ AT/
007604	040	101	104	.ASCII	/ AD/
007607	104	122	105	.ASCII	/DRE/
007612	123	123	040	.ASCII	/SS /
007615	045	117	066	.ASCII	/#06/
007620	000	000		.ASCII	<00><00>
007622	045	101	052	P.AED: .ASCII	/AA*/
007625	040	111	116	.ASCII	/ IN/
007630	103	117	122	.ASCII	/COR/
007633	122	105	103	.ASCII	/REC/
007636	124	040	102	.ASCII	/T B/
007641	122	040	114	.ASCII	/R L/
007644	105	126	105	.ASCII	/EVE/
007647	114	040	106	.ASCII	/L F/
007652	117	122	040	.ASCII	/OR /
007655	104	122	111	.ASCII	/DR1/
007660	126	105	040	.ASCII	/VE /
007663	045	117	066	.ASCII	/#06/
007666	000	000		.ASCII	<00><00>
007670	045	101	052	P.AEE: .ASCII	/AA*/
007673	040	123	124	.ASCII	/ SI/
007676	105	120	040	.ASCII	/EP /
007701	045	104	061	.ASCII	/#D1/

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0059
Page 69
(34)

007704	045	101	040	.ASCII	/A /	
007707	122	105	101	.ASCII	/REA/	
007710	104	040	105	.ASCII	/D E/	
007715	122	122	117	.ASCII	/RRO/	
007720	122	000		.ASCII	/R/<00>	
007722	045	101	052	P.AEF:	.ASCII	/AA/
007725	040	102	101	.ASCII	/BA/	
007730	104	040	123	.ASCII	/D S/	
007733	101	040	103	.ASCII	/A C/	
007736	117	104	105	.ASCII	/ODE/	
007741	040	106	122	.ASCII	/FR/	
007744	117	115	040	.ASCII	/OM /	
007747	104	122	111	.ASCII	/DRI/	
007752	126	105	040	.ASCII	/VE /	
007755	045	117	066	.ASCII	/06/	
007760	000	000		.ASCII	<00><00>	
007762	045	101	052	P.AEG:	.ASCII	/AA/
007765	040	104	111	.ASCII	/DI/	
007770	123	113	045	.ASCII	/SK/	
007773	104	062	045	.ASCII	/D2/	
007776	101	040	127	.ASCII	/A W/	
010001	105	116	124	.ASCII	/ENT/	
010004	040	117	106	.ASCII	/OF/	
010007	106	114	111	.ASCII	/FLI/	
010012	116	105	000	.ASCII	/NE/<00>	
010015	000			.ASCII	<00>	
010016	045	101	052	P.AEH:	.ASCII	/AA/
010021	040	104	122	.ASCII	/DR/	
010024	111	126	105	.ASCII	/IVE/	
010027	040	045	117	.ASCII	/O/	
010032	066	045	101	.ASCII	/6A/	
010035	040	116	117	.ASCII	/NO/	
010040	124	040	120	.ASCII	/I P/	
010043	122	117	103	.ASCII	/ROC/	
010046	105	123	123	.ASCII	/ESS/	
010051	111	116	107	.ASCII	/ING/	
010054	040	103	117	.ASCII	/CO/	
010057	115	115	101	.ASCII	/MMA/	
010062	116	104	040	.ASCII	/ND /	
010065	120	101	103	.ASCII	/PAC/	
010070	113	105	124	.ASCII	/KET/	
010073	123	000	000	.ASCII	/S/<00><00>	
010076	045	101	052	P.AEI:	.ASCII	/AA/
010101	040	104	111	.ASCII	/DI/	
010104	123	113	045	.ASCII	/SK/	
010107	104	062	045	.ASCII	/D2/	
010112	101	040	127	.ASCII	/A W/	
010115	105	116	124	.ASCII	/ENT/	
010120	040	124	117	.ASCII	/IO/	
010123	040	124	110	.ASCII	/TH/	
010126	105	040	042	.ASCII	/E /	
010131	101	126	101	.ASCII	/AVA/	
010134	111	114	101	.ASCII	/ILA/	

F6

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6SEQ 0070
Page 70
(34)

010137	102	114	105	.ASCII	/BLE/
010142	042	040	123	.ASCII	/ " S/
010145	124	101	124	.ASCII	/TAT/
010150	105	000		.ASCII	/E/<00>
010152	040	055	040	P.AEJ:	.ASCII / - /
010155	125	116	122		.ASCII /UNR/
010160	105	103	117		.ASCII /ECO/
010163	107	116	111		.ASCII /GNI/
010166	132	105	104		.ASCII /ZED/
010171	040	115	105		.ASCII / ME/
010174	123	123	101		.ASCII /SSA/
010177	107	105	040		.ASCII /GE /
010202	124	131	120		.ASCII /TYP/
010205	105	000	000		.ASCII /E/<00><00>
010210	040	055	040	P.AEK:	.ASCII / - /
010213	125	116	122		.ASCII /UNR/
010216	105	103	117		.ASCII /ECO/
010221	107	116	111		.ASCII /GNI/
010224	132	105	104		.ASCII /ZED/
010227	040	103	117		.ASCII / CO/
010232	116	116	105		.ASCII /NNE/
010235	103	124	111		.ASCII /CTI/
010240	117	116	040		.ASCII /ON /
010243	111	104	000		.ASCII /ID/<00>
010246	040	055	040	P.AEL:	.ASCII / - /
010251	125	116	122		.ASCII /UNR/
010254	105	103	117		.ASCII /ECO/
010257	107	116	111		.ASCII /GNI/
010262	132	105	104		.ASCII /ZED/
010265	040	122	105		.ASCII / RE/
010270	124	125	122		.ASCII /TUR/
010273	116	040	115		.ASCII /N M/
010276	105	123	123		.ASCII /ESS/
010301	101	107	105		.ASCII /AGE/
010304	000	000			.ASCII <00><00>
010306	040	055	040	P.AEM:	.ASCII / - /
010311	125	116	122		.ASCII /UNR/
010314	105	103	117		.ASCII /ECO/
010317	107	116	111		.ASCII /GNI/
010322	132	105	104		.ASCII /ZED/
010325	040	122	105		.ASCII / RE/
010330	124	125	122		.ASCII /TUR/
010333	116	040	120		.ASCII /N P/
010336	101	103	113		.ASCII /ACK/
010341	105	124	000		.ASCII /ET/<00>
010344	040	055	040	P.AEN:	.ASCII / - /
010347	125	116	122		.ASCII /UNR/
010352	105	103	117		.ASCII /ECO/
010355	107	116	111		.ASCII /GNI/
010360	132	105	104		.ASCII /ZED/
010363	040	103	122		.ASCII / CR/
010366	116	000			.ASCII /N/<00>
010370	040	055	040	P.AEO:	.ASCII / - /

G6

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0071
Page 71
(34)

010373	125	116	122	.ASCII	/UNR/
010376	105	103	117	.ASCII	/ECO/
010401	107	116	111	.ASCII	/GNI/
010404	132	105	104	.ASCII	/ZED/
010407	040	117	120	.ASCII	/ OP/
010412	103	117	104	.ASCII	/COI/
010415	105	000	000	.ASCII	/E/<00><00>
010420	040	055	040	P.AEP:	.ASCII / - /
010423	115	123	103	.ASCII	/MSC/
010426	120	040	123	.ASCII	/P S/
010431	124	101	124	.ASCII	/TAT/
010434	125	123	040	.ASCII	/US /
010437	103	117	104	.ASCII	/COD/
010442	105	040	105	.ASCII	/E E/
010445	122	122	000	.ASCII	/RR/<00>
010450	040	055	040	P.AEQ:	.ASCII / - /
010453	104	125	120	.ASCII	/DUP/
010456	040	123	124	.ASCII	/ ST/
010461	101	124	125	.ASCII	/ATU/
010464	123	040	103	.ASCII	/S C/
010467	117	104	105	.ASCII	/ODE/
010472	040	105	122	.ASCII	/ ER/
010475	122	000	000	.ASCII	/R/<00><00>
010500	040	055	040	P.AER:	.ASCII / - /
010503	125	116	122	.ASCII	/UNR/
010506	105	103	117	.ASCII	/ECO/
010511	107	116	111	.ASCII	/GNI/
010514	132	105	104	.ASCII	/ZED/
010517	040	123	124	.ASCII	/ ST/
010522	101	124	125	.ASCII	/ATU/
010525	123	040	103	.ASCII	/S C/
010530	117	104	105	.ASCII	/ODE/
010533	000			.ASCII	<00>
010534	040	055	040	P.AES:	.ASCII / - /
010537	114	102	116	.ASCII	/LBN/
010542	040	110	117	.ASCII	/ HO/
010545	123	124	040	.ASCII	/ST /
010550	103	117	115	.ASCII	/COM/
010553	120	101	122	.ASCII	/PAR/
010556	105	040	105	.ASCII	/E E/
010561	122	122	000	.ASCII	/RR/<00>
010564	040	055	040	P.AET:	.ASCII / - /
010567	104	102	116	.ASCII	/DBN/
010572	040	110	117	.ASCII	/ HO/
010575	123	124	040	.ASCII	/ST /
010600	103	117	115	.ASCII	/COM/
010603	120	101	122	.ASCII	/PAR/
010606	105	040	105	.ASCII	/E E/
010611	122	122	000	.ASCII	/RR/<00>
010614	040	055	040	P.AEU:	.ASCII / - /
010617	125	116	101	.ASCII	/UNA/
010622	102	114	105	.ASCII	/BLE/
010625	040	124	117	.ASCII	/ TU/

H6

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1:6SEQ 0072
Page 72
(34)

010630	040	114	117	.ASCII	/ LO/
010633	101	104	040	.ASCII	/AD /
010636	104	125	120	.ASCII	/DUP/
010641	040	115	105	.ASCII	/ ME/
010644	104	111	101	.ASCII	/DIA/
010647	000			.ASCII	<00>
010650	040	055	040	P.AEV:	.ASCII / - /
010653	105	122	122	.ASCII	/ERR/
010656	040	111	116	.ASCII	/ IN/
010661	040	104	125	.ASCII	/ DU/
010664	120	040	120	.ASCII	/P P/
010667	113	124	040	.ASCII	/KT /
010672	127	110	105	.ASCII	/WHE/
010675	116	040	125	.ASCII	/N U/
010700	123	111	116	.ASCII	/SIN/
010703	107	040	103	.ASCII	/G C/
010706	124	114	122	.ASCII	/TLR/
010711	040	114	103	.ASCII	/ LC/
010714	040	120	122	.ASCII	/ PR/
010717	117	107	000	.ASCII	/OG/<00>
010722	045	101	052	P.AEW:	.ASCII /A+/
010725	040	104	111	.ASCII	/ DI/
010730	123	113	045	.ASCII	/SK/
010733	104	062	000	.ASCII	/D2/<00>
010736	045	101	111	P.AEY:	.ASCII /AI/
010741	116	126	101	.ASCII	/NVA/
010744	114	111	104	.ASCII	/LID/
010747	040	103	117	.ASCII	/ CO/
010752	115	115	101	.ASCII	/MMA/
010755	116	104	000	.ASCII	/ND/<00>
010760	045	101	103	P.AEZ:	.ASCII /AC/
010763	117	115	115	.ASCII	/OMM/
010766	101	116	104	.ASCII	/AND/
010771	040	101	102	.ASCII	/ AB/
010774	117	122	124	.ASCII	/ORT/
010777	105	104	000	.ASCII	/ED/<00>
011002	045	101	125	P.AFA:	.ASCII /AU/
011005	116	111	124	.ASCII	/NIT/
011010	040	117	106	.ASCII	/ OF/
011013	106	114	111	.ASCII	/FLI/
011016	116	105	000	.ASCII	/NE/<00>
011021	000			.ASCII	<00>
011022	045	101	124	P.AFB:	.ASCII /AI/
011025	122	101	116	.ASCII	/RAN/
011030	123	111	124	.ASCII	/SII/
011033	111	117	116	.ASCII	/ION/
011036	040	124	117	.ASCII	/ TO/
011041	040	101	126	.ASCII	/ AV/
011044	101	111	114	.ASCII	/AIL/
011047	101	102	114	.ASCII	/ABI/
011052	105	040	123	.ASCII	/E S/
011055	124	101	124	.ASCII	/IAT/
011060	105	000		.ASCII	/F/<00>

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6SEQ 0073
Page 73
(34)

011062	045	101	115	P.AFC:	.ASCII	/AM/
011065	105	104	111		.ASCII	/EDI/
011070	101	040	106		.ASCII	/AF/
011073	117	122	115		.ASCII	/ORM/
011076	101	124	040		.ASCII	/AT /
011101	105	122	122		.ASCII	/ERR/
011104	117	122	000		.ASCII	/OR/<00>
011107	000				.ASCII	<00>
011110	045	101	127	P.AFD:	.ASCII	/AW/
011113	122	111	124		.ASCII	/RII/
011116	105	055	120		.ASCII	/E P/
011121	122	117	124		.ASCII	/ROT/
011124	105	103	124		.ASCII	/ECT/
011127	105	104	000		.ASCII	/ED/<00>
011130	045	101	104	P.AFE:	.ASCII	/AD/
011135	105	126	111		.ASCII	/EVI/
011140	103	105	040		.ASCII	/CE /
011143	103	117	115		.ASCII	/COM/
011146	120	101	122		.ASCII	/PAR/
011151	105	040	105		.ASCII	/E E/
011154	122	122	117		.ASCII	/RRO/
011157	122	000	000		.ASCII	/R/<00><00>
011162	045	101	104	P.AFF:	.ASCII	/AD/
011165	101	124	101		.ASCII	/ATA/
011170	040	105	122		.ASCII	/ ER/
011173	122	117	122		.ASCII	/ROR/
011176	000	000			.ASCII	<00><00>
011200	045	101	110	P.AFG:	.ASCII	/AH/
011203	117	123	124		.ASCII	/OST/
011206	040	102	125		.ASCII	/ BU/
011211	106	106	105		.ASCII	/FFE/
011214	122	040	101		.ASCII	/R A/
011217	103	103	105		.ASCII	/CCE/
011222	123	123	040		.ASCII	/SS /
011225	105	122	122		.ASCII	/ERR/
011230	117	122	000		.ASCII	/OR/<00>
011233	000				.ASCII	<00>
011234	045	101	103	P.AFH:	.ASCII	/AC/
011237	117	116	124		.ASCII	/ONT/
011242	122	117	114		.ASCII	/ROL/
011245	114	105	122		.ASCII	/LER/
011250	040	105	122		.ASCII	/ ER/
011253	122	117	122		.ASCII	/ROR/
011256	000	000			.ASCII	<00><00>
011260	045	101	104	P.AFI:	.ASCII	/AD/
011263	122	111	126		.ASCII	/RIV/
011266	105	040	105		.ASCII	/E E/
011271	122	122	117		.ASCII	/RRO/
011274	122	000			.ASCII	/R/<00>
011276	045	101	115	P.AFJ:	.ASCII	/AM/
011301	105	123	123		.ASCII	/ESS/
011304	101	107	105		.ASCII	/AGE/
011307	040	106	122		.ASCII	/ FR/

J6

ZRQAM1
VOL.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6SEQ 0074
Page 74
(34)

011312	117	115	040	.ASCII	/OM /
011315	111	116	124	.ASCII	/INT/
011320	105	122	116	.ASCII	/ERN/
011323	101	114	040	.ASCII	/AL /
011326	104	111	101	.ASCII	/DIA/
011331	107	116	117	.ASCII	/GNO/
011334	123	124	111	.ASCII	/STI/
011337	103	123	000	.ASCII	/CS/<00>
011342	045	101	110	P.AFK: .ASCII	/AH/
011345	117	123	124	.ASCII	/OST/
011350	040	103	117	.ASCII	/CO/
011353	115	120	101	.ASCII	/MPA/
011356	122	105	040	.ASCII	/RE /
011361	105	122	122	.ASCII	/ERR/
011364	117	122	000	.ASCII	/OR/<00>
011367	000			.ASCII	<00>
011370	045	101	103	P.AFL: .ASCII	/AC/
011373	117	115	115	.ASCII	/OMM/
011376	101	116	104	.ASCII	/AND/
011401	040	124	111	.ASCII	/II/
011404	115	105	117	.ASCII	/ME0/
011407	125	124	000	.ASCII	/UI/<00>
011412	010736'			P.AEX: .WORD	P.AEY
011414	010760'			.WORD	P.AEZ
011416	011002'			.WORD	P.AFA
011420	011022'			.WORD	P.AFB
011422	011062'			.WORD	P.AFC
011424	011110'			.WORD	P.AFD
011426	011132'			.WORD	P.AFE
011430	011162'			.WORD	P.AFF
011432	011200'			.WORD	P.AFG
011434	011234'			.WORD	P.AFH
011436	011260'			.WORD	P.AFI
011440	011276'			.WORD	P.AFJ
011442	011342'			.WORD	P.AFK
011444	011370'			.WORD	P.AFL
011446	045	101	105	P.AFM: .ASCII	/AE/
011451	122	122	117	.ASCII	/RR0/
011454	122	040	114	.ASCII	/RL/
011457	117	107	040	.ASCII	/OG /
011462	115	105	123	.ASCII	/MES/
011465	123	101	107	.ASCII	/SAG/
011470	105	040	122	.ASCII	/ER/
011473	105	103	105	.ASCII	/ECH/
011476	111	126	105	.ASCII	/IVE/
011501	104	072	045	.ASCII	/D;*/
011504	116	000		.ASCII	/N/<00>
011506	045	101	052	P.AFO: .ASCII	/A*/
011511	040	103	117	.ASCII	/CO/
011514	116	124	122	.ASCII	/NTR/
011517	117	114	114	.ASCII	/OLI/
011522	105	122	040	.ASCII	/ER /
011525	105	122	122	.ASCII	/ERR/

K6

 ZRQAM1
 V01.6

 RD/RX EXERCISER
 PROTECTION TABLE

 11-Apr-1984 11:56:01
 11-Apr-1984 11:45:02

 VAX-11 Bliss-16 V4.0-579
 DISK\$USER2:[POWERS]ZRQADO.BL1:6

011530	117	122	045		.ASCII	/OR#
011533	116	000	000		.ASCII	/N/<00><00>
011536	045	101	052	P.AFP:	.ASCII	/A#
011541	040	110	117		.ASCII	/HO/
011544	123	124	040		.ASCII	/ST /
011547	115	105	115		.ASCII	/MEM/
011552	117	122	131		.ASCII	/ORY/
011555	040	101	103		.ASCII	/AC/
011560	103	105	123		.ASCII	/CES/
011563	123	040	105		.ASCII	/S E/
011566	122	122	117		.ASCII	/RRO/
011571	122	045	116		.ASCII	/R#N/
011574	000	000			.ASCII	<00><00>
011576	045	101	052	P.AFQ:	.ASCII	/A#
011601	040	104	111		.ASCII	/DI/
011604	123	113	045		.ASCII	/SK#
011607	104	062	045		.ASCII	/D2#
011612	101	040	055		.ASCII	/A -/
011615	040	104	111		.ASCII	/DI/
011620	123	113	040		.ASCII	/SK /
011623	124	122	101		.ASCII	/TRA/
011626	116	123	106		.ASCII	/NSF/
011631	105	122	040		.ASCII	/ER /
011634	105	122	122		.ASCII	/ERR/
011637	117	122	045		.ASCII	/OR#
011642	116	000			.ASCII	/N/<00>
011644	045	101	052	P.AFR:	.ASCII	/A#
011647	040	104	111		.ASCII	/DI/
011652	123	113	045		.ASCII	/SK#
011655	107	062	045		.ASCII	/D2#
011660	101	040	055		.ASCII	/A -/
011663	040	042	123		.ASCII	/ "S/
011666	124	101	116		.ASCII	/TAN/
011671	104	101	122		.ASCII	/DAR/
011674	104	040	104		.ASCII	/D D/
011677	111	123	113		.ASCII	/ISK/
011702	040	111	116		.ASCII	/IN/
011705	124	105	122		.ASCII	/TER/
011710	103	117	116		.ASCII	/CON/
011713	116	105	103		.ASCII	/NEC/
011716	124	042	040		.ASCII	/T" /
011721	105	122	122		.ASCII	/ERR/
011724	117	122	045		.ASCII	/OR#
011727	116	000	000		.ASCII	/N/<00><00>
011732	045	101	052	P.AFS:	.ASCII	/A#
011735	040	104	111		.ASCII	/DI/
011740	123	113	045		.ASCII	/SK#
011743	104	062	045		.ASCII	/D2#
011746	101	040	055		.ASCII	/A -/
011751	040	042	123		.ASCII	/ "S/
011754	115	101	114		.ASCII	/MAL/
011757	114	040	104		.ASCII	/L D/
011762	111	123	113		.ASCII	/ISK/

ZRGAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0 579
DISK:USER2:[POWERS]ZRGAD0.BL1:6SEQ 0076
Page 76
(34)

011765	042	040	105		.ASCII	/ " E /
011770	122	122	117		.ASCII	/RR0/
011773	122	045	116		.ASCII	/R#N/
011776	000	000			.ASCII	<00><00>
012000	011506			P.AFN:	.WORD	P.AFO
012002	011536				.WORD	P.AFP
012004	011576				.WORD	P.AFQ
012006	011644				.WORD	P.AFR
012010	011732				.WORD	P.AFS
012012	045	116	045	P.AFT:	.ASCII	/#N# /
012015	101	052	040		.ASCII	/A* /
012020	123	101	072		.ASCII	/SA: /
012023	040	045	117		.ASCII	/ #0 /
012026	066	000			.ASCII	/6/ <00>
012030	045	116	045	P.AFU:	.ASCII	/#N# /
012033	101	052	040		.ASCII	/A* /
012036	123	124	101		.ASCII	/STA/
012041	124	125	123		.ASCII	/TUS/
012044	040	103	117		.ASCII	/ CO /
012047	104	105	072		.ASCII	/DE: /
012052	040	045	117		.ASCII	/ #0 /
012055	062	000	000		.ASCII	/2/ <00><00>
012060	045	117	064	P.AFV:	.ASCII	/#04/
012063	000				.ASCII	<00>
012064	045	116	045	P.AFW:	.ASCII	/#N# /
012067	101	052	040		.ASCII	/A* /
012072	123	125	102		.ASCII	/SUB/
012075	137	103	117		.ASCII	/ CO /
012100	104	105	072		.ASCII	/DE: /
012103	040	000	000		.ASCII	/ / <00><00>
012106	045	116	045	P.AFX:	.ASCII	/#N# /
012111	101	052	040		.ASCII	/A* /
012114	103	117	115		.ASCII	/COM/
012117	115	101	116		.ASCII	/MAN/
012122	104	072	040		.ASCII	/D: /
012125	000				.ASCII	<00>
012126	045	101	122	P.AFY:	.ASCII	/#AR/
012131	105	101	104		.ASCII	/EAD/
012134	000	000			.ASCII	<00><00>
012136	045	101	127	P.AFZ:	.ASCII	/#AW/
012141	122	111	124		.ASCII	/RIT/
012144	105	000			.ASCII	/E/ <00>
012146	045	101	055	P.AGA:	.ASCII	/#A /
012151	103	117	115		.ASCII	/COM/
012154	120	101	122		.ASCII	/PAR/
012157	105	000	000		.ASCII	/E/ <00><00>
012162	045	101	117	P.AGB:	.ASCII	/#AO/
012165	116	114	111		.ASCII	/NLI/
012170	116	105	000		.ASCII	/NE/ <00>
012173	000				.ASCII	<00>
012174	045	101	101	P.AGC:	.ASCII	/#AA/
012177	103	103	105		.ASCII	/CCE/
012202	123	123	000		.ASCII	/SS/ <00>

M6

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0 579
DISK\$USER2:[POWERS]ZRQAD0.BL1:6SEQ 0077
Page 77
(34)

012205	000				.ASCII	<00>
012206	045	117	063	P.AGD:	.ASCII	/03/
012211	000				.ASCII	<00>
012212	045	116	045	P.AGE:	.ASCII	/N/
012215	101	052	040		.ASCII	/A/
012220	102	101	104		.ASCII	/BAD/
012223	040	102	114		.ASCII	/BL/
012226	117	103	113		.ASCII	/OCK/
012231	040	050	110		.ASCII	/ (H/
012234	157	163	164		.ASCII	/ost/
012237	040	162	145		.ASCII	/ re/
012242	160	154	141		.ASCII	/pla/
012245	143	145	141		.ASCII	/cea/
012250	142	154	145		.ASCII	/ble/
012253	051	072	040		.ASCII	/): /
012256	045	104	065		.ASCII	/05/
012261	045	101	056		.ASCII	/A./
012264	040	050	117		.ASCII	/ (0/
012267	103	124	040		.ASCII	/CI /
012272	045	117	066		.ASCII	/06/
012275	045	101	051		.ASCII	/A)/
012300	000	000			.ASCII	<00><00>
012302	045	116	045	P.AGF:	.ASCII	/N/
012305	101	052	040		.ASCII	/A/
012310	061	163	164		.ASCII	/1st/
012313	040	102	101		.ASCII	/BA/
012316	104	040	102		.ASCII	/D B/
012321	114	117	103		.ASCII	/LOC/
012324	113	040	050		.ASCII	/K (/
012327	110	157	163		.ASCII	/Hos/
012332	164	040	162		.ASCII	/t r/
012335	145	160	154		.ASCII	/ep1/
012340	141	143	145		.ASCII	/ace/
012343	141	142	154		.ASCII	/abl/
012346	145	051	072		.ASCII	/e): /
012351	040	045	104		.ASCII	/ 0/
012354	065	045	101		.ASCII	/5A/
012357	056	040	050		.ASCII	/ (/
012362	117	103	124		.ASCII	/OCI/
012365	040	045	117		.ASCII	/ 0/
012370	066	045	101		.ASCII	/6A/
012373	051	000	000		.ASCII	/)/<00><00>
012376	045	116	045	P.AGG:	.ASCII	/N/
012401	101	052	040		.ASCII	/A/
012404	102	101	104		.ASCII	/BAD/
012407	040	102	114		.ASCII	/BL/
012412	117	103	113		.ASCII	/OCK/
012415	040	122	105		.ASCII	/ RE/
012420	120	117	122		.ASCII	/POR/
012423	124	105	104		.ASCII	/TED/
012426	040	050	122		.ASCII	/ (R/
012431	145	160	154		.ASCII	/ep1/
012434	141	143	145		.ASCII	/ace/

N6

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0078
Page 78
(34)

012437	144	051	072	.ASCII	/d):/
012442	040	045	104	.ASCII	/ #D/
012445	045	101	056	.ASCII	/ #A./
012450	040	050	117	.ASCII	/ (0/
012453	103	124	040	.ASCII	/CI /
012456	045	117	066	.ASCII	/ #06/
012461	045	101	051	.ASCII	/ #A)/
012464	000	000		.ASCII	<00><00>
012466	045	116	045	P.AGH: .ASCII	/ #N# /
012471	101	052	040	.ASCII	/A* /
012474	114	102	116	.ASCII	/LBN/
012477	072	040	045	.ASCII	/: #/
012502	104	065	045	.ASCII	/D5# /
012505	101	056	040	.ASCII	/A. /
012510	050	117	103	.ASCII	/ (OC/
012513	124	040	045	.ASCII	/T #/
012516	117	066	045	.ASCII	/06# /
012521	101	051	000	.ASCII	/A)/<00>
012524	045	116	045	P.AGI: .ASCII	/ #N# /
012527	101	052	040	.ASCII	/A* /
012532	120	102	116	.ASCII	/PBN/
012535	072	040	045	.ASCII	/: #/
012540	104	065	045	.ASCII	/D5# /
012543	101	056	040	.ASCII	/A. /
012546	050	117	103	.ASCII	/ (OC/
012551	124	040	045	.ASCII	/T #/
012554	117	066	045	.ASCII	/06# /
012557	101	051	000	.ASCII	/A)/<00>
012562	045	116	045	P.AGJ: .ASCII	/ #N# /
012565	101	052	040	.ASCII	/A* /
012570	114	102	116	.ASCII	/LBN/
012573	072	040	050	.ASCII	/: (/
012576	122	105	101	.ASCII	/REA/
012601	104	051	040	.ASCII	/D) /
012604	045	104	065	.ASCII	/ #D5/
012607	045	101	056	.ASCII	/ #A./
012612	040	050	117	.ASCII	/ (0/
012615	103	124	040	.ASCII	/CI /
012620	045	117	066	.ASCII	/ #06/
012623	045	101	051	.ASCII	/ #A)/
012626	000	000		.ASCII	<00><00>
012630	045	116	045	P.AGK: .ASCII	/ #N# /
012633	101	052	040	.ASCII	/A* /
012636	114	102	116	.ASCII	/LBN/
012641	072	040	050	.ASCII	/: (/
012644	127	122	111	.ASCII	/WR1/
012647	124	105	051	.ASCII	/TE)/
012652	040	045	104	.ASCII	/ #D/
012655	065	045	101	.ASCII	/5#A/
012660	056	040	050	.ASCII	/, (/
012663	117	103	124	.ASCII	/OCI/
012666	040	045	117	.ASCII	/ #0/
012671	066	045	101	.ASCII	/6#A/

137

ZRQAM1
VOL 1,6

RD RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1100 16 V4.0 5/9
DISK\$USER2:[POWER5]ZRQADO.BL1,6

SEQ 0079
Page 79
(34)

012674	051	000			.ASCII	/) / < 00 >
012676	045	116	045	P,AGL:	.ASCII	/ < N > /
012701	101	052	040		.ASCII	/ A * /
012704	122	105	120		.ASCII	/ REP /
012707	114	101	103		.ASCII	/ LAC /
012712	105	115	105		.ASCII	/ FME /
012715	116	124	040		.ASCII	/ NT /
012720	102	114	117		.ASCII	/ BLO /
012723	105	113	040		.ASCII	/ CK /
012726	116	117	056		.ASCII	/ NO /
012731	040	045	104		.ASCII	/ < D /
012734	065	045	101		.ASCII	/ 5 < A /
012737	056	040	050		.ASCII	/ , (/
012742	117	103	124		.ASCII	/ OCT /
012745	040	045	117		.ASCII	/ < O /
012750	066	045	101		.ASCII	/ 6 < P /
012753	051	000	000	P,AGM:	.ASCII	/) / < 00 > < 00 >
012756	045	116	045		.ASCII	/ < N > /
012761	101	052	040		.ASCII	/ A * /
012764	102	131	124		.ASCII	/ B * I /
012767	105	040	103		.ASCII	/ E C /
012772	117	125	116		.ASCII	/ OUN /
012775	124	040	111		.ASCII	/ T I /
013000	116	040	103		.ASCII	/ N C /
013003	117	115	115		.ASCII	/ OMH /
013006	101	116	104		.ASCII	/ AND /
013011	072	040	045		.ASCII	/ : < /
013014	104	065	045		.ASCII	/ D5 < /
013017	101	056	000	P,AGN:	.ASCII	/ A , / < 00 >
013022	045	116	045		.ASCII	/ < N > /
013025	101	052	040		.ASCII	/ A * /
013030	102	131	124		.ASCII	/ B * I /
013033	105	040	103		.ASCII	/ E C /
013036	117	125	116		.ASCII	/ OUN /
013041	124	040	111		.ASCII	/ T I /
013044	116	040	122		.ASCII	/ N R /
013047	105	101	104		.ASCII	/ EAD /
013052	040	103	117		.ASCII	/ CO /
013055	115	115	101		.ASCII	/ MMA /
013060	116	104	072		.ASCII	/ NO /
013063	040	045	104		.ASCII	/ < D /
013066	065	045	101		.ASCII	/ 5 < A /
013071	056	000	000	P,AGU:	.ASCII	/) / < 00 > < 00 >
013074	045	116	045		.ASCII	/ < N > /
013077	101	052	040		.ASCII	/ A * /
013102	102	131	124		.ASCII	/ B * I /
013105	105	040	103		.ASCII	/ E C /
013110	117	125	116		.ASCII	/ OUN /
013113	124	040	111		.ASCII	/ T I /
013116	116	040	122		.ASCII	/ N W /
013121	122	111	124		.ASCII	/ RIT /
013124	105	040	103		.ASCII	/ E C /
013127	117	115	115		.ASCII	/ OMH /

C /

ZRRQAH1
VOL.6RD RX EXERCISER
PROTECTION TABLE11 Apr 1984 11:56:01
11 Apr 1984 11:45:02VAX 11 B1159 16 V4.0 579
DISK\$USER2:[POWERS]ZRRQADO.BL1:6SEQ 0080
Page 80
(34)

013132	101	116	104	.ASCII	/AND/
013135	072	040	045	.ASCII	/: #/
013140	104	065	045	.ASCII	/D5#/
013143	101	056	000	.ASCII	/A./<00>
013146	045	116	045	P,AGP: .ASCII	/#N#/
013151	101	052	040	.ASCII	/A# /
013154	101	103	124	.ASCII	/ACI/
013157	125	101	114	.ASCII	/UAL/
013162	040	045	040	.ASCII	/ # /
013165	117	106	040	.ASCII	/OF /
013170	102	131	124	.ASCII	/BYT/
013173	105	123	040	.ASCII	/ES /
013176	124	122	101	.ASCII	/TRA/
013201	116	123	106	.ASCII	/NSF/
013204	105	122	122	.ASCII	/ERR/
013207	105	104	072	.ASCII	/ED:/
013212	040	045	104	.ASCII	/ #D/
013215	065	045	101	.ASCII	/5#A/
013220	056	000		.ASCII	/./<00>
013222	045	116	045	P,AGQ: .ASCII	/#N#/
013225	101	052	040	.ASCII	/A# /
013230	111	057	117	.ASCII	/I./57>/0/
013233	040	102	125	.ASCII	/ BU/
013236	106	106	105	.ASCII	/FFE/
013241	122	040	101	.ASCII	/R A/
013244	104	104	122	.ASCII	/DDR/
013247	105	123	123	.ASCII	/ESS/
013252	040	050	063	.ASCII	/ (3/
013255	062	040	142	.ASCII	/2 b/
013260	151	164	163	.ASCII	/its/
013263	051	072	040	.ASCII	/): /
013266	045	117	066	.ASCII	/#06/
013271	045	101	040	.ASCII	/#A /
013274	045	117	066	.ASCII	/#06/
013277	000			.ASCII	<00>
013300	045	116	045	P,AGR: .ASCII	/#N#/
013303	101	052	040	.ASCII	/A# /
013306	111	057	117	.ASCII	/I./57>/0/
013311	040	102	125	.ASCII	/ BU/
013314	106	106	105	.ASCII	/FFE/
013317	122	040	101	.ASCII	/R A/
013322	104	104	122	.ASCII	/DDR/
013325	105	123	123	.ASCII	/ESS/
013330	040	106	117	.ASCII	/ FO/
013333	122	040	122	.ASCII	/R R/
013336	105	101	104	.ASCII	/EAD/
013341	040	050	063	.ASCII	/ (3/
013344	062	040	142	.ASCII	/2 b/
013347	151	164	163	.ASCII	/its/
013352	051	072	040	.ASCII	/): /
013355	045	117	066	.ASCII	/#06/
013360	045	101	040	.ASCII	/#A /
013363	045	117	066	.ASCII	/#06/

107

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11:16 V4.0 579
DISK:USER2:[POWERS]ZRQADO.BL1;6

SEQ 0081
Page 81
(34)

013366	000	000		P.AGS:	.ASCII	<00><00>
013370	045	116	045		.ASCII	/N#
013373	101	052	040		.ASCII	/A#
013376	111	057	117		.ASCII	/I/<57>/O/
013401	040	102	125		.ASCII	/BU/
013404	106	106	105		.ASCII	/FFE/
013407	122	040	101		.ASCII	/R A/
013412	104	104	122		.ASCII	/DDR/
013415	105	123	123		.ASCII	/ESS/
013420	040	106	117		.ASCII	/FO/
013423	122	040	127		.ASCII	/R W/
013426	122	111	124		.ASCII	/RI/
013431	105	040	050		.ASCII	/E (/
013434	063	062	040		.ASCII	/32 /
013437	142	151	164		.ASCII	/bit/
013442	163	051	072		.ASCII	/s):/
013445	040	045	117		.ASCII	/ #0/
013450	066	045	101		.ASCII	/6#A/
013453	040	045	117		.ASCII	/ #0/
013456	066	000			.ASCII	/6/<00>
013460	045	116	045	P.AGT:	.ASCII	/N#
013463	101	103	117		.ASCII	/ACO/
013466	116	124	105		.ASCII	/NTE/
013471	116	124	123		.ASCII	/NTS/
013474	040	117	106		.ASCII	/ OF/
013477	040	103	117		.ASCII	/ CO/
013502	115	115	101		.ASCII	/MMA/
013505	116	104	057		.ASCII	/ND/<57>
013510	122	105	123		.ASCII	/RES/
013513	120	117	116		.ASCII	/PON/
013516	123	105	040		.ASCII	/SE /
013521	120	101	103		.ASCII	/PAC/
013524	113	105	124		.ASCII	/KET/
013527	040	123	101		.ASCII	/ SA/
013532	126	105	040		.ASCII	/VE /
013535	101	122	105		.ASCII	/ARE/
013540	101	072	045		.ASCII	/A: #/
013543	116	000	000		.ASCII	/N/<00><00>
013546	045	101	040	P.AGU:	.ASCII	/ #A /
013551	045	117	066		.ASCII	/ #06/
013554	000	000			.ASCII	<00><00>
013556	045	116	045	P.AGV:	.ASCII	/N#
013561	101	052	040		.ASCII	/A#
013564	124	111	115		.ASCII	/TIM/
013567	105	072	040		.ASCII	/E: /
013572	045	132	062		.ASCII	/ #22/
013575	045	101	072		.ASCII	/ #A: /
013600	045	132	062		.ASCII	/ #22/
013603	045	101	040		.ASCII	/ #A /
013606	110	117	125		.ASCII	/HOU/
013611	122	123	045		.ASCII	/R5# /
013614	116	000			.ASCII	/N/<00>
013616	045	116	045	P.AGW:	.ASCII	/N#

1 /

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11as-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0082
Page 82
(34)

013621	101	040	052	.ASCII	/A +/
013624	040	104	111	.ASCII	/DI/
013627	123	113	040	.ASCII	/SK /
013632	072	040	045	.ASCII	/: #/
013635	104	062	000	.ASCII	/D2/<00>
013640	045	116	045	P.AGX: .ASCII	/N#/
013643	101	104	102	.ASCII	/ADB/
013646	116	072	040	.ASCII	/N: /
013651	045	104	065	.ASCII	/D5/
013654	045	101	056	.ASCII	/A./
013657	040	050	117	.ASCII	/ (0/
013662	103	124	040	.ASCII	/CT /
013665	045	117	066	.ASCII	/06/
013670	045	101	051	.ASCII	/A)/
013673	000			.ASCII	<00>
013674	045	116	045	P.AGY: .ASCII	/N#/
013677	101	102	131	.ASCII	/ABY/
013702	124	105	040	.ASCII	/TE /
013705	116	125	115	.ASCII	/NUM/
013710	102	105	122	.ASCII	/BER/
013713	072	040	045	.ASCII	/: #/
013716	104	063	000	.ASCII	/D3/<00>
013721	000			.ASCII	<00>
013722	045	116	045	P.AGZ: .ASCII	/N#/
013725	101	122	101	.ASCII	/ARA/
013730	116	104	117	.ASCII	/NDO/
013733	115	040	127	.ASCII	/M W/
013736	122	111	124	.ASCII	/RIT/
013741	124	105	116	.ASCII	/TEN/
013744	040	127	117	.ASCII	/WO/
013747	122	104	040	.ASCII	/RD /
013752	072	045	102	.ASCII	/: #B/
013755	061	066	000	.ASCII	/16/<00>
013760	045	116	045	P.AHA: .ASCII	/N#/
013763	101	122	101	.ASCII	/ARA/
013766	116	104	117	.ASCII	/NDO/
013771	115	040	122	.ASCII	/M R/
013774	105	101	104	.ASCII	/EAD/
013777	040	127	117	.ASCII	/WO/
014002	122	104	040	.ASCII	/RD /
014005	142	151	156	.ASCII	/bin/
014010	072	045	102	.ASCII	/: #B/
014013	061	066	045	.ASCII	/16#/
014016	101	040	157	.ASCII	/A o/
014021	143	164	072	.ASCII	/ct:/
014024	045	117	066	.ASCII	/06/
014027	000			.ASCII	<00>
014030	045	116	045	P.AHB: .ASCII	/N#/
014033	101	104	125	.ASCII	/ADU/
014036	120	114	111	.ASCII	/PI I/
014041	103	101	124	.ASCII	/CAT/
014044	105	040	125	.ASCII	/E U/
014047	116	111	124	.ASCII	/NIT/

F7

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6SEQ 0083
Page 83
(34)

014052	072	045	104		.ASCII	/: #D/
014055	062	045	101		.ASCII	/2 #A/
014060	040	101	124		.ASCII	/ AT/
014063	040	111	120		.ASCII	/ IP/
014066	072	040	045		.ASCII	/: #/
014071	117	066	000		.ASCII	/06/ <00>
014074	045	116	045	P.AHC:	.ASCII	/ #N #/
014077	101	115	117		.ASCII	/AM0/
014102	122	105	040		.ASCII	/RE /
014105	124	110	101		.ASCII	/THA/
014110	116	040	045		.ASCII	/N #/
014113	104	061	045		.ASCII	/D1 #/
014116	101	040	104		.ASCII	/A D/
014121	111	106	106		.ASCII	/IFF/
014124	105	122	105		.ASCII	/ERE/
014127	116	124	040		.ASCII	/NT /
014132	111	120	040		.ASCII	/IP /
014135	101	104	104		.ASCII	/ADD/
014140	122	105	123		.ASCII	/RES/
014143	123	105	123		.ASCII	/SES/
014146	000	000			.ASCII	<00> <00>
014150	045	101	123	P.AHD:	.ASCII	/ #AS/
014153	120	111	116		.ASCII	/PIN/
014156	055	104	117		.ASCII	/-DO/
014161	127	116	040		.ASCII	/WN /
014164	111	107	116		.ASCII	/IGN/
014167	117	122	105		.ASCII	/ORE/
014172	104	000			.ASCII	/D/ <00>
014174	045	101	123	P.AHE:	.ASCII	/ #AS/
014177	124	111	114		.ASCII	/TIL/
014202	114	040	103		.ASCII	/L C/
014205	117	116	116		.ASCII	/ONN/
014210	105	103	124		.ASCII	/ECT/
014213	105	104	000		.ASCII	/ED/ <00>
014216	045	101	104	P.AHF:	.ASCII	/ #AD/
014221	125	120	114		.ASCII	/UPL/
014224	111	103	101		.ASCII	/ICA/
014227	124	105	040		.ASCII	/TE /
014232	125	116	111		.ASCII	/UNI/
014235	124	040	116		.ASCII	/T N/
014240	125	115	102		.ASCII	/UMB/
014243	105	122	000		.ASCII	/ER/ <00>
014246	045	101	101	P.AHG:	.ASCII	/ #AA/
014251	114	122	105		.ASCII	/LRE/
014254	101	104	131		.ASCII	/AD1/
014257	040	117	116		.ASCII	/ ON/
014262	114	111	116		.ASCII	/LIN/
014265	105	000	000		.ASCII	/E/ <00> <00>
014270	045	101	123	P.AHH:	.ASCII	/ #AS/
014273	124	111	114		.ASCII	/TIL/
014276	114	040	117		.ASCII	/L O/
014301	116	114	111		.ASCII	/NLI/
014304	116	105	000		.ASCII	/NE/ <00>

67

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0084
Page 84
(34)

014307	000				.ASCII	<00>
014310	045	101	125	P,AHI:	.ASCII	/AU/
014313	116	111	124		.ASCII	/NIT/
014316	040	125	116		.ASCII	/UN/
014321	113	116	117		.ASCII	/KNO/
014324	127	116	040		.ASCII	/WN /
014327	117	122	040		.ASCII	/OR /
014332	117	116	114		.ASCII	/ONL/
014335	111	116	105		.ASCII	/INE/
014340	040	124	117		.ASCII	/TO/
014343	040	101	116		.ASCII	/AN/
014346	117	124	110		.ASCII	/OTH/
014351	105	122	040		.ASCII	/ER /
014354	103	117	116		.ASCII	/CON/
014357	124	122	117		.ASCII	/TRO/
014362	114	114	105		.ASCII	/LLE/
014365	122	000	000		.ASCII	/R/<00><00>
014370	045	101	116	P,AHJ:	.ASCII	/AN/
014373	117	040	126		.ASCII	/O V/
014376	117	114	125		.ASCII	/OLU/
014401	115	105	040		.ASCII	/ME /
014404	115	117	125		.ASCII	/MOU/
014407	116	124	105		.ASCII	/NTE/
014412	104	040	117		.ASCII	/D O/
014415	122	040	104		.ASCII	/R D/
014420	122	111	126		.ASCII	/RIV/
014423	105	040	104		.ASCII	/E D/
014426	111	123	101		.ASCII	/ISA/
014431	102	114	105		.ASCII	/BLE/
014434	104	040	102		.ASCII	/D B/
014437	131	040	123		.ASCII	/Y S/
014442	127	111	124		.ASCII	/WIT/
014445	103	110	000		.ASCII	/CH/<00>
014450	045	101	125	P,AHK:	.ASCII	/AU/
014453	116	111	124		.ASCII	/NIT/
014456	040	111	116		.ASCII	/IN/
014461	117	120	105		.ASCII	/OPE/
014464	122	101	124		.ASCII	/RAT/
014467	111	126	105		.ASCII	/IVE/
014472	040	050	122		.ASCII	/ (R/
014475	104	065	061		.ASCII	/D51/
014500	057	065	062		.ASCII	<57>/52/
014503	040	167	162		.ASCII	/ wr/
014506	151	164	145		.ASCII	/ite/
014511	040	146	141		.ASCII	/ fa/
014514	165	154	164		.ASCII	/ult/
014517	051	000	000		.ASCII	/)/<00><00>
014522	045	101	125	P,AHL:	.ASCII	/AU/
014525	116	111	124		.ASCII	/NIT/
014530	040	104	111		.ASCII	/ DI/
014533	123	101	102		.ASCII	/SAH/
014536	114	105	104		.ASCII	/LED/
014541	040	102	131		.ASCII	/ BY/

H/

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0085
Page 85
(34)

014544	040	106	111	.ASCII	/FI/	
014547	105	114	104	.ASCII	/ELD/	
014552	040	123	105	.ASCII	/SE/	
014555	122	126	111	.ASCII	/RVI/	
014560	103	105	040	.ASCII	/CE /	
014563	117	122	040	.ASCII	/OR /	
014566	111	116	124	.ASCII	/INT/	
014571	105	122	116	.ASCII	/ERN/	
014574	101	114	040	.ASCII	/AL /	
014577	104	111	101	.ASCII	/DIA/	
014602	107	116	117	.ASCII	/GNO/	
014605	123	124	111	.ASCII	/STI/	
014610	103	123	000	.ASCII	/CS/<00>	
014613	000			.ASCII	<00>	
014614	045	101	042	P.AHM;	.ASCII	/A"/
014617	106	117	122	.ASCII	/FOR/	
014622	103	105	104	.ASCII	/CED/	
014625	040	105	122	.ASCII	/ER/	
014630	122	117	122	.ASCII	/ROR/	
014633	042	040	104	.ASCII	/ " D/	
014636	105	124	105	.ASCII	/ETE/	
014641	103	124	105	.ASCII	/CTE/	
014644	104	040	127	.ASCII	/D W/	
014647	110	111	114	.ASCII	/HIL/	
014652	105	040	101	.ASCII	/E A/	
014655	103	103	105	.ASCII	/CCE/	
014660	123	123	111	.ASCII	/SSI/	
014663	116	107	040	.ASCII	/NG /	
014666	106	103	124	.ASCII	/FCT/	
014671	040	117	122	.ASCII	/ OR/	
014674	040	122	103	.ASCII	/ RC/	
014677	124	000	000	.ASCII	/T/<00><00>	
014702	045	101	123	P.AHN;	.ASCII	/AS/
014705	105	103	124	.ASCII	/ECT/	
014710	117	122	040	.ASCII	/OR /	
014713	110	101	104	.ASCII	/HAD/	
014716	040	102	105	.ASCII	/ BE/	
014721	105	116	040	.ASCII	/EN /	
014724	127	122	111	.ASCII	/WRI/	
014727	124	124	105	.ASCII	/TTE/	
014732	116	040	127	.ASCII	/N W/	
014735	111	124	110	.ASCII	/ITH/	
014740	040	042	106	.ASCII	/ " F/	
014743	117	122	103	.ASCII	/ORC/	
014746	105	104	040	.ASCII	/ED /	
014751	105	122	122	.ASCII	/ERR/	
014754	117	122	042	.ASCII	/OR"/	
014757	040	115	117	.ASCII	/ MU/	
014762	104	111	106	.ASCII	/DIE/	
014765	111	105	122	.ASCII	/IER/	
014770	000	000		.ASCII	<00><00>	
014772	045	101	106	P.AHO;	.ASCII	/AF/
014775	103	124	040	.ASCII	/CT /	

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6SEQ 0086
Page 86
(34)

015000	117	122	040	.ASCII	/OR /
015003	122	103	124	.ASCII	/RCT/
015006	040	125	116	.ASCII	/ UN/
015011	122	105	101	.ASCII	/REA/
015014	104	101	102	.ASCII	/DAB/
015017	114	105	040	.ASCII	/LE /
015022	055	040	111	.ASCII	/- I/
015025	116	126	101	.ASCII	/NVA/
015030	114	111	104	.ASCII	/LID/
015033	040	123	105	.ASCII	/ SE/
015036	103	124	117	.ASCII	/CTO/
015041	122	040	110	.ASCII	/R H/
015044	105	101	104	.ASCII	/EAD/
015047	105	122	000	.ASCII	/ER/<00>
015052	045	101	110	P.AHP: .ASCII	/AH/
015055	105	101	104	.ASCII	/EAD/
015060	105	122	040	.ASCII	/ER /
015063	103	117	115	.ASCII	/COM/
015066	120	101	122	.ASCII	/PAR/
015071	105	040	105	.ASCII	/E E/
015074	122	122	117	.ASCII	/RR0/
015077	122	040	050	.ASCII	/R (/
015102	126	141	154	.ASCII	/Val/
015105	151	144	040	.ASCII	/id /
015110	150	145	141	.ASCII	/hea/
015113	144	145	162	.ASCII	/der/
015116	040	156	157	.ASCII	/ no/
015121	164	040	146	.ASCII	/t f/
015124	157	165	156	.ASCII	/oun/
015127	144	051	000	.ASCII	/d)/<00>
015132	045	101	106	P.AHQ: .ASCII	/AF/
015135	103	124	040	.ASCII	/CT /
015140	117	122	040	.ASCII	/OR /
015143	122	103	124	.ASCII	/RCT/
015146	040	125	116	.ASCII	/ UN/
015151	122	105	101	.ASCII	/REA/
015154	104	101	102	.ASCII	/DAB/
015157	114	105	040	.ASCII	/LE /
015162	055	040	104	.ASCII	/ D/
015165	101	124	101	.ASCII	/ATA/
015170	040	123	131	.ASCII	/ SY/
015173	116	103	040	.ASCII	/NC /
015176	124	111	115	.ASCII	/TIM/
015201	105	117	125	.ASCII	/EQU/
015204	124	000		.ASCII	/I/<00>
015206	045	101	104	P.AHR: .ASCII	/AD/
015211	101	124	101	.ASCII	/ATA/
015214	040	123	131	.ASCII	/ SY/
015217	116	103	040	.ASCII	/NC /
015222	116	117	124	.ASCII	/NOT/
015225	040	106	117	.ASCII	/ FU/
015230	125	116	104	.ASCII	/UND/
015233	040	050	104	.ASCII	/ (D/

37

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B11ss-16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BU1:6

SEQ 0087
Page 87
(34)

015236	141	164	141	.ASCII	/ata/
015241	040	163	171	.ASCII	/sy/
015244	156	143	040	.ASCII	/nc/
015247	164	151	155	.ASCII	/tim/
015252	145	157	165	.ASCII	/eou/
015255	164	051	000	.ASCII	/t)/<00>
015260	045	101	106	P.AHS:	.ASCII
015263	103	124	040	.ASCII	/CT/
015266	117	122	040	.ASCII	/OR/
015271	122	103	124	.ASCII	/RCT/
015274	040	125	116	.ASCII	/UN/
015277	122	105	101	.ASCII	/REA/
015302	104	101	102	.ASCII	/DAB/
015305	114	105	040	.ASCII	/LE/
015310	055	040	125	.ASCII	/U/
015313	116	103	117	.ASCII	/NCO/
015316	122	122	105	.ASCII	/RRE/
015321	103	124	101	.ASCII	/CIA/
015324	102	114	105	.ASCII	/BLE/
015327	040	105	103	.ASCII	/EC/
015332	103	040	105	.ASCII	/CE/
015335	122	122	117	.ASCII	/RRO/
015340	122	000		.ASCII	/R/<00>
015342	045	101	125	P.AHT:	.ASCII
015345	116	103	117	.ASCII	/AU/
015350	122	122	105	.ASCII	/NCO/
015353	103	124	101	.ASCII	/RRE/
015356	102	114	105	.ASCII	/CIA/
015361	040	105	103	.ASCII	/BLE/
015364	103	040	105	.ASCII	/EC/
015367	122	122	117	.ASCII	/CE/
015372	122	000		.ASCII	/RRO/
015374	045	101	122	P.AHU:	.ASCII
015377	103	124	040	.ASCII	/AR/
015402	103	117	122	.ASCII	/CT/
015405	122	125	120	.ASCII	/COR/
015410	124	105	104	.ASCII	/RUP/
015413	000			.ASCII	/TED/
015414	045	101	116	P.AHV:	.ASCII
015417	117	040	122	.ASCII	/AN/
015422	105	120	114	.ASCII	/OR/
015425	101	103	105	.ASCII	/EPL/
015430	115	105	116	.ASCII	/ACE/
015433	124	040	102	.ASCII	/MEN/
015436	114	117	103	.ASCII	/TB/
015441	113	040	101	.ASCII	/OC/
015444	126	101	111	.ASCII	/K A/
015447	114	101	102	.ASCII	/VAT/
015452	114	105	040	.ASCII	/LAB/
015455	050	122	103	.ASCII	/LE/
015460	124	040	146	.ASCII	/RC/
015463	165	154	154	.ASCII	/I f/
015466	051	000		.ASCII	/G11/
				.ASCII	/)/<00>

ZRQAM1
V01.6

RD-RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0088
Page 88
(34)

015430	045	101	104	P.AHW:	.ASCII	/AD/
015433	111	123	113		.ASCII	/ISK/
015436	040	116	117		.ASCII	/NO/
015501	124	040	106		.ASCII	/IF/
015504	117	122	115		.ASCII	/ORM/
015507	101	124	124		.ASCII	/ATT/
015512	105	104	040		.ASCII	/ED/
015515	122	111	124		.ASCII	/WIT/
015520	110	040	065		.ASCII	/H 5/
015523	061	062	040		.ASCII	/12/
015526	102	131	124		.ASCII	/BIT/
015531	105	040	123		.ASCII	/E 5/
015534	105	103	124		.ASCII	/ECT/
015537	117	122	123		.ASCII	/ORS/
015542	000	000			.ASCII	<00><00>
015544	045	101	104	P.AHX:	.ASCII	/AD/
015547	111	123	113		.ASCII	/ISK/
015552	040	116	117		.ASCII	/NO/
015555	124	040	106		.ASCII	/IF/
015560	117	122	115		.ASCII	/ORM/
015563	101	124	124		.ASCII	/ATT/
015566	105	104	040		.ASCII	/ED/
015571	117	122	040		.ASCII	/OR/
015574	106	103	124		.ASCII	/ECT/
015577	040	103	117		.ASCII	/CO/
015602	122	122	125		.ASCII	/RRU/
015605	120	124	105		.ASCII	/PTE/
015610	104	000			.ASCII	/D/<00>
015612	045	101	117	P.AHY:	.ASCII	/AD/
015615	116	105	040		.ASCII	/NE/
015620	123	131	115		.ASCII	/SYM/
015623	102	117	114		.ASCII	/BOL/
015626	040	105	103		.ASCII	/EC/
015631	103	040	105		.ASCII	/CE/
015634	122	122	117		.ASCII	/RR0/
015637	122	000	000		.ASCII	/R/<00><00>
015642	045	101	124	P.AHZ:	.ASCII	/AD/
015645	122	117	040		.ASCII	/WU/
015650	123	131	115		.ASCII	/SYM/
015653	102	117	114		.ASCII	/BOL/
015656	040	105	103		.ASCII	/EC/
015661	103	040	105		.ASCII	/CE/
015664	122	122	117		.ASCII	/RR0/
015667	122	000	000		.ASCII	/R/<00><00>
015672	045	101	124	P.AIA:	.ASCII	/AD/
015675	110	122	105		.ASCII	/HRE/
015700	105	040	123		.ASCII	/E 5/
015703	131	115	102		.ASCII	/IMB/
015706	117	114	040		.ASCII	/OL/
015711	105	103	103		.ASCII	/ECC/
015714	040	105	122		.ASCII	/ER/
015717	122	117	122		.ASCII	/ROR/
015722	000	000			.ASCII	<00><00>

ZRQAM1
V01.6RD, RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0089
Page 89
(34)

015724	045	101	106
015727	117	125	122
015732	040	123	131
015735	115	102	117
015740	114	040	105
015743	103	103	040
015746	105	122	122
015751	117	122	000
015754	045	101	106
015757	111	126	105
015762	040	123	131
015765	115	102	117
015770	114	040	105
015773	103	103	040
015776	105	122	122
016001	117	122	000
016004	045	101	123
016007	111	130	040
016012	123	131	115
016015	102	117	114
016020	040	105	103
016023	103	040	105
016026	122	122	117
016031	122	000	000
016034	045	101	123
016037	105	126	105
016042	116	040	123
016045	131	115	102
016050	117	114	040
016053	105	103	103
016056	040	105	122
016061	122	117	122
016064	000	000	
016066	045	101	105
016071	111	107	110
016074	124	040	123
016077	131	115	102
016102	117	114	040
016105	105	103	103
016110	040	105	122
016113	122	117	122
016116	000	000	
016120	045	101	103
016123	117	122	122
016126	105	103	124
016131	101	102	114
016134	105	040	105
016137	122	122	117
016142	122	040	111
016145	116	040	105
016150	103	103	040
016153	106	111	105
016156	114	104	000

P.AIB:	.ASCII	/AF/
	.ASCII	/OUR/
	.ASCII	/SY/
	.ASCII	/MBO/
	.ASCII	/L E/
	.ASCII	/CC /
	.ASCII	/ERR/
	.ASCII	/OR/<00>
P.AIC:	.ASCII	/AF/
	.ASCII	/IVE/
	.ASCII	/SY/
	.ASCII	/MBO/
	.ASCII	/L E/
	.ASCII	/CC /
	.ASCII	/ERR/
	.ASCII	/OR/<00>
P.AID:	.ASCII	/AS/
	.ASCII	/IX /
	.ASCII	/SYM/
	.ASCII	/BOL/
	.ASCII	/EC/
	.ASCII	/C E/
	.ASCII	/RRO/
	.ASCII	/R/<00><00>
P.AIE:	.ASCII	/AS/
	.ASCII	/EVE/
	.ASCII	/N S/
	.ASCII	/YMB/
	.ASCII	/OL /
	.ASCII	/ECC/
	.ASCII	/ER/
	.ASCII	/ROR/
	.ASCII	<00><00>
P.AIF:	.ASCII	/AE/
	.ASCII	/IGH/
	.ASCII	/T S/
	.ASCII	/YMB/
	.ASCII	/OL /
	.ASCII	/ECC/
	.ASCII	/ER/
	.ASCII	/ROR/
	.ASCII	<00><00>
P.AIG:	.ASCII	/AC/
	.ASCII	/ORR/
	.ASCII	/CTI/
	.ASCII	/ABL/
	.ASCII	/E E/
	.ASCII	/RRO/
	.ASCII	/R I/
	.ASCII	/N E/
	.ASCII	/CC /
	.ASCII	/IE/
	.ASCII	/LD/<00>

M7

ZRGAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO,BL1;6SEQ 0090
Page 90
(34)

016161	000				.ASCII	<00>
016162	045	101	125	P.AIH:	.ASCII	/AU/
016165	116	111	124		.ASCII	/NIT/
016170	040	123	117		.ASCII	/SG/
016173	106	124	127		.ASCII	/FTW/
016176	101	122	105		.ASCII	/ARE/
016201	040	127	122		.ASCII	/WR/
016204	111	124	105		.ASCII	/ITE/
016207	040	120	122		.ASCII	/PR/
016212	117	124	105		.ASCII	/OTE/
016215	103	124	105		.ASCII	/CTE/
016220	104	000			.ASCII	/D/<00>
016223	045	101	125	P.AII:	.ASCII	/AU/
016225	116	111	124		.ASCII	/NIT/
016230	040	110	101		.ASCII	/HA/
016233	122	104	127		.ASCII	/RDW/
016236	101	122	105		.ASCII	/ARE/
016241	040	127	122		.ASCII	/WR/
016244	111	124	105		.ASCII	/ITE/
016247	040	120	122		.ASCII	/PR/
016252	117	124	105		.ASCII	/OTE/
016255	103	124	105		.ASCII	/CTE/
016260	104	000			.ASCII	/D/<00>
016262	045	101	117	P.AIJ:	.ASCII	/AU/
016265	104	104	040		.ASCII	/DD /
016270	124	122	101		.ASCII	/TRA/
016273	116	123	106		.ASCII	/NSF/
016276	105	122	040		.ASCII	/ER /
016301	101	104	104		.ASCII	/ADD/
016304	122	105	123		.ASCII	/RES/
016307	123	000	000		.ASCII	/S/<00><00>
016312	045	101	117	P.AIK:	.ASCII	/AU/
016315	104	104	040		.ASCII	/DD /
016320	102	131	124		.ASCII	/BYT/
016323	105	040	103		.ASCII	/C /
016326	117	125	116		.ASCII	/CN/
016331	124	000	000		.ASCII	/T/<00><00>
016334	045	101	116	P.AIL:	.ASCII	/AN/
016337	117	116	055		.ASCII	/ON /
016342	105	130	111		.ASCII	/EXI/
016345	123	124	105		.ASCII	/STF/
016350	116	124	040		.ASCII	/NI /
016353	110	117	123		.ASCII	/HOS/
016356	124	040	115		.ASCII	/T M/
016361	105	115	117		.ASCII	/EMO/
016364	122	131	000		.ASCII	/RY/<00>
016367	000				.ASCII	<00>
016370	045	101	110	P.AIM:	.ASCII	/AH/
016373	117	123	124		.ASCII	/OSI/
016376	040	115	105		.ASCII	/ME/
016401	115	117	122		.ASCII	/MOR/
016404	131	040	120		.ASCII	/Y P/
016407	101	122	111		.ASCII	/ARI/

N7

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0091
Page 91
(34)

016412	124	131	040		.ASCII	/TY /
016415	105	122	122		.ASCII	/ERR/
016420	117	122	000		.ASCII	/OR/<00>
016423	000				.ASCII	<00>
016424	045	101	103	P.AIN:	.ASCII	/MAC/
016427	117	115	115		.ASCII	/OMM/
016432	101	116	104		.ASCII	/AND/
016435	040	124	111		.ASCII	/ TI/
016440	115	117	125		.ASCII	/MOU/
016443	124	040	117		.ASCII	/T O/
016446	122	040	122		.ASCII	/R R/
016451	105	124	122		.ASCII	/ETR/
016454	131	040	114		.ASCII	/Y L/
016457	111	115	111		.ASCII	/IMI/
016462	124	040	105		.ASCII	/T E/
016465	130	103	105		.ASCII	/XCE/
016470	105	104	105		.ASCII	/EDE/
016473	104	000	000		.ASCII	/D/<00><00>
016476	045	101	123	P.AIO:	.ASCII	/MS/
016501	105	122	111		.ASCII	/ERI/
016504	101	114	111		.ASCII	/ALI/
016507	132	105	122		.ASCII	/ZER/
016512	057	104	105		.ASCII	<57>/DE/
016515	123	105	122		.ASCII	/SER/
016520	111	101	114		.ASCII	/IAL/
016523	111	132	105		.ASCII	/IZE/
016526	122	040	117		.ASCII	/R O/
016531	126	105	122		.ASCII	/VER/
016534	122	125	116		.ASCII	/RUN/
016537	040	117	122		.ASCII	/ OR/
016542	040	125	116		.ASCII	/ UN/
016545	104	105	122		.ASCII	/DER/
016550	122	125	116		.ASCII	/RUN/
016553	000				.ASCII	<00>
016554	045	101	042	P.AIP:	.ASCII	/MA"/
016557	105	122	122		.ASCII	/ERR/
016562	117	122	040		.ASCII	/OR /
016565	104	105	124		.ASCII	/DE I/
016570	105	103	124		.ASCII	/ECI/
016573	111	117	116		.ASCII	/ION/
016576	040	103	117		.ASCII	/ CO/
016601	104	105	042		.ASCII	/DE"/
016604	040	105	122		.ASCII	/ ER/
016607	122	117	122		.ASCII	/ROR/
016612	000	000			.ASCII	<00><00>
016614	045	101	111	P.AIQ:	.ASCII	/MAI/
016617	116	105	117		.ASCII	/NCO/
016622	116	123	111		.ASCII	/NSI/
016625	123	124	105		.ASCII	/STE/
016630	116	124	040		.ASCII	/NI /
016633	111	116	124		.ASCII	/INI/
016636	105	122	116		.ASCII	/ERN/
016641	101	114	040		.ASCII	/AL /

ZRQAD0
V01.16RD RX EXERCISER
PROTECTION TABLE11 Apr 1984 11:56:01
11 Apr 1984 11:45:02VAX 11 B11: 16 V4.0 5/79
DISK\$USER2:[POWERS]ZRQAD0.BL1:6SEQ 0092
Page 92
(34)

016644	104	101	124	.ASCII	/DAT/
016647	101	040	125	.ASCII	/A S/
016652	124	122	125	.ASCII	/TRU/
016655	103	124	125	.ASCII	/CIU/
016660	122	105	000	.ASCII	/RE/ <00>
016665	000			.ASCII	<00>
016664	045	101	104	P, AIR: .ASCII	/A0/
016667	122	111	126	.ASCII	/RIV/
016672	105	040	103	.ASCII	/E C/
016675	117	115	115	.ASCII	/OMM/
016700	101	116	104	.ASCII	/AM/
016703	040	124	111	.ASCII	/TI/
016707	115	105	117	.ASCII	/MEU/
016711	125	124	040	.ASCII	/UT/
016714	050	116	157	.ASCII	/No/
016717	040	162	145	.ASCII	/re/
016722	163	160	157	.ASCII	/spo/
016725	156	163	145	.ASCII	/nse/
016730	040	157	162	.ASCII	/or/
016733	040	163	145	.ASCII	/se/
016736	145	153	040	.ASCII	/ek/
016741	151	156	143	.ASCII	/inc/
016744	157	155	160	.ASCII	/omp/
016747	154	145	164	.ASCII	/let/
016752	145	051	000	.ASCII	/e/ <00>
016755	000			.ASCII	<00>
016756	045	101	103	P, AIS: .ASCII	/AC/
016761	117	116	124	.ASCII	/ONT/
016764	122	117	114	.ASCII	/ROL/
016767	114	105	122	.ASCII	/LER/
016772	040	104	105	.ASCII	/DE/
016775	124	105	103	.ASCII	/TEC/
017000	124	105	104	.ASCII	/TED/
017003	040	124	122	.ASCII	/TR/
017006	101	116	123	.ASCII	/ANS/
017011	115	111	123	.ASCII	/HIS/
017014	123	111	117	.ASCII	/SIO/
017017	116	040	117	.ASCII	/N D/
017022	122	040	120	.ASCII	/R P/
017025	122	117	124	.ASCII	/RUT/
017030	117	103	117	.ASCII	/OCU/
017033	114	040	105	.ASCII	/L E/
017036	122	122	117	.ASCII	/RRU/
017041	122	000	000	.ASCII	/R/ <00> <00>
017044	045	101	120	P, AIT: .ASCII	/AP/
017047	117	123	111	.ASCII	/OSI/
017052	124	111	117	.ASCII	/TIO/
017055	116	040	105	.ASCII	/N E/
017060	122	122	117	.ASCII	/RRU/
017063	122	040	050	.ASCII	/R C/
017066	115	151	163	.ASCII	/MIS/
017071	055	163	145	.ASCII	/se/
017074	145	153	051	.ASCII	/ek/

08

ZRQAM1
VOL.6

RD-RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1100 16 V4.0 579
DISK\$USER2:[POWER5]ZRQADO.BL116

SEQ 0093
Page 93
(34)

017077	000			P, AIU:	.ASCII <00>
017100	045	101	114		.ASCII /MAL/
017103	117	123	124		.ASCII /OST/
017106	040	122	105		.ASCII /RE/
017111	101	104	057		.ASCII /AD/<57>
017114	127	122	111		.ASCII /WRI/
017117	124	105	040		.ASCII /TL/
017122	122	105	101		.ASCII /REA/
017125	104	131	040		.ASCII /DI/
017130	104	125	122		.ASCII /DUR/
017133	111	116	107		.ASCII /ING/
017136	057	102	105		.ASCII <57>/BE/
017141	124	127	105		.ASCII /TWE/
017144	105	116	040		.ASCII /EN/
017147	124	122	101		.ASCII /TRA/
017152	116	123	106		.ASCII /NSF/
017155	105	122	123		.ASCII /ERS/
017160	000	000		P, AIV:	.ASCII <00><00>
017162	045	101	104		.ASCII /RAD/
017165	122	111	126		.ASCII /RIV/
017170	105	040	103		.ASCII /E C/
017173	114	117	103		.ASCII /LOC/
017176	113	040	104		.ASCII /K D/
017201	122	117	120		.ASCII /ROP/
017204	117	125	124		.ASCII /OUT/
017207	000			P, AIW:	.ASCII <00>
017210	045	101	114		.ASCII /MAL/
017213	117	123	124		.ASCII /OST/
017216	040	122	105		.ASCII /RE/
017221	103	105	111		.ASCII /CEI/
017224	126	105	122		.ASCII /VER/
017227	040	122	105		.ASCII /RE/
017232	101	104	131		.ASCII /AD/
017235	040	102	105		.ASCII /BE/
017240	124	127	105		.ASCII /TWE/
017243	105	116	040		.ASCII /EN/
017246	123	105	103		.ASCII /SEC/
017251	124	117	122		.ASCII /TOR/
017254	123	000		P, AIX:	.ASCII /57<00>
017256	045	101	104		.ASCII /RAD/
017261	122	111	126		.ASCII /RIV/
017264	105	040	104		.ASCII /E D/
017267	105	124	105		.ASCII /E TE/
017272	103	124	105		.ASCII /CTE/
017275	104	040	105		.ASCII /D E/
017300	122	122	117		.ASCII /RRR/
017303	122	000	000	P, AIT:	.ASCII /R/<00><00>
017306	045	101	103		.ASCII /MAC/
017311	117	116	124		.ASCII /ONT/
017314	127	117	114		.ASCII /ROI/
017317	114	105	122		.ASCII /LER/
017321	040	104	105		.ASCII /DEF/
017325	124	105	103		.ASCII /TEC/

[D8

ZRQAM1
VOL.6

RD-RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11sg 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0094
Page 94
(34)

017330	124	105	104	.ASCII	/TED/	
017333	040	120	125	.ASCII	/PU/	
017336	114	123	105	.ASCII	/LSE/	
017341	040	117	122	.ASCII	/OR/	
017344	040	123	124	.ASCII	/ST/	
017347	101	124	105	.ASCII	/ATE/	
017352	040	120	101	.ASCII	/PA/	
017355	122	111	124	.ASCII	/RII/	
017360	131	040	105	.ASCII	/Y E/	
017363	122	122	117	.ASCII	/RRO/	
017366	122	000		.ASCII	/R/<00>	
017370	045	101	103	P.AJA:	.ASCII	/MAC/
017373	117	116	124	.ASCII	/ONT/	
017376	122	117	114	.ASCII	/ROL/	
017401	114	105	122	.ASCII	/LER/	
017404	040	124	111	.ASCII	/TI/	
017407	115	105	117	.ASCII	/MEO/	
017412	125	124	000	.ASCII	/UT/<00>	
017415	000			.ASCII	<00>	
017416	045	101	105	P.AJB:	.ASCII	/MAE/
017421	116	126	105	.ASCII	/NVE/	
017424	114	117	120	.ASCII	/LOP/	
017427	105	057	120	.ASCII	/E/<57>/P/	
017432	101	103	113	.ASCII	/ACK/	
017435	105	124	040	.ASCII	/ET /	
017440	122	105	101	.ASCII	/RFA/	
017443	104	040	105	.ASCII	/D E/	
017446	122	122	117	.ASCII	/RRO/	
017451	122	040	050	.ASCII	/R (/	
017454	120	141	162	.ASCII	/Par/	
017457	151	164	171	.ASCII	/ity/	
017462	040	157	162	.ASCII	/ or/	
017465	040	164	151	.ASCII	/ti/	
017470	155	145	157	.ASCII	/meo/	
017473	165	164	051	.ASCII	/ut)/	
017476	000	000		.ASCII	<00><00>	
017500	045	101	105	P.AJC:	.ASCII	/MAE/
017503	116	126	105	.ASCII	/NVE/	
017506	114	117	120	.ASCII	/LOP/	
017511	105	057	120	.ASCII	/E/<57>/P/	
017514	101	103	113	.ASCII	/ACK/	
017517	105	124	040	.ASCII	/ET /	
017522	127	122	111	.ASCII	/WRI/	
017525	124	105	040	.ASCII	/TE /	
017530	105	122	122	.ASCII	/ERR/	
017533	117	122	040	.ASCII	/OR /	
017536	050	120	141	.ASCII	/(Pa/	
017541	162	151	164	.ASCII	/rit/	
017544	171	040	157	.ASCII	/y o/	
017547	162	040	164	.ASCII	/r t/	
017552	151	155	145	.ASCII	/ime/	
017555	157	165	164	.ASCII	/out/	
017560	051	000		.ASCII	/)/<00>	

F-8

ZRQAM1 VOL.6	RD/RX EXERCISER PROTECTION TABLE			11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 Bliss-16 V4.0-579 DISK\$USER2:[POWERS]ZRQADO.BL1;6	SEQ 0095 Page 95 (34)
017562	045	101	103	P.AJD:	.ASCII /AC/	
017565	117	116	124		.ASCII /ONT/	
017570	122	117	114		.ASCII /ROL/	
017573	114	105	122		.ASCII /LER/	
017576	040	122	117		.ASCII / RO/	
017601	115	040	101		.ASCII /M A/	
017604	116	104	040		.ASCII /ND /	
017607	122	101	115		.ASCII /RAM/	
017612	040	120	101		.ASCII / PA/	
017615	122	111	124		.ASCII /RIT/	
017620	131	040	105		.ASCII /Y E/	
017623	122	122	117		.ASCII /RRO/	
017626	122	000			.ASCII /R/<00>	
017630	045	101	103	P.AJE:	.ASCII /AC/	
017633	117	116	124		.ASCII /ONT/	
017636	122	117	114		.ASCII /ROL/	
017641	114	105	122		.ASCII /LER/	
017644	040	122	101		.ASCII / RA/	
017647	115	040	120		.ASCII /M P/	
017652	101	122	111		.ASCII /ARI/	
017655	124	131	040		.ASCII /TY /	
017660	105	122	122		.ASCII /ERR/	
017663	117	122	000		.ASCII /OR/<00>	
017666	045	101	103	P.AJF:	.ASCII /AC/	
017671	117	116	124		.ASCII /ONT/	
017674	122	117	114		.ASCII /ROL/	
017677	114	105	122		.ASCII /LER/	
017702	040	122	117		.ASCII / RO/	
017705	115	040	120		.ASCII /M P/	
017710	101	122	111		.ASCII /ARI/	
017713	124	131	040		.ASCII /TY /	
017716	105	122	122		.ASCII /ERR/	
017721	117	122	000		.ASCII /OR/<00>	
017724	045	101	122	P.AJG:	.ASCII /AR/	
017727	111	116	107		.ASCII /ING/	
017732	040	122	105		.ASCII / RE/	
017735	101	104	040		.ASCII /AD /	
017740	105	122	122		.ASCII /ERR/	
017743	117	122	040		.ASCII /OR /	
017746	050	120	141		.ASCII / (P/	
017751	162	151	164		.ASCII /rit/	
017754	171	040	157		.ASCII /y o/	
017757	162	040	164		.ASCII /r t/	
017762	151	155	145		.ASCII /ime/	
017765	157	165	164		.ASCII /out/	
017770	051	000			.ASCII /)/<00>	
017772	045	101	122	P.AJH:	.ASCII /AR/	
017775	111	116	107		.ASCII /ING/	
020000	040	127	122		.ASCII / WR/	
020003	111	124	105		.ASCII /II /	
020006	040	105	122		.ASCII / FR/	
020011	122	117	122		.ASCII /ROR/	
020014	040	050	120		.ASCII / (P/	

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1:gs-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0096
Page 96
(34)

020017	141	162	151	.ASCII	/ari/
020022	164	171	040	.ASCII	/ty /
020025	157	162	040	.ASCII	/or /
020030	164	151	155	.ASCII	/tim/
020033	145	157	165	.ASCII	/eou/
020036	164	051	000	.ASCII	/t)/<00>
020041	000			.ASCII	<00>
020042	111	116	124	P.AJI:	.ASCII /INT/
020045	105	122	122		.ASCII /ERR/
020050	125	120	124		.ASCII /UPT/
020053	040	115	101		.ASCII / MA/
020056	123	124	105		.ASCII /STE/
020061	122	040	106		.ASCII /R F/
020064	101	111	114		.ASCII /AIL/
020067	125	122	105		.ASCII /URE/
020072	000	000			.ASCII <00><00>
020074	045	101	110	P.AJJ:	.ASCII /AH/
020077	117	123	124		.ASCII /OST/
020102	040	101	103		.ASCII / AC/
020105	103	105	123		.ASCII /CES/
020110	123	040	124		.ASCII /S T/
020113	111	115	105		.ASCII /IME/
020116	117	125	124		.ASCII /OUT/
020121	040	050	110		.ASCII / (H/
020124	151	147	150		.ASCII /igh/
020127	145	162	040		.ASCII /er /
020132	154	145	166		.ASCII /lev/
020135	145	154	040		.ASCII /el /
020140	160	162	157		.ASCII /pro/
020143	164	157	143		.ASCII /toc/
020146	157	154	040		.ASCII /ol /
020151	144	145	160		.ASCII /dep/
020154	145	156	144		.ASCII /end/
020157	145	156	164		.ASCII /ent/
020162	051	000			.ASCII /)/<00>
020164	045	101	103	P.AJK:	.ASCII /AC/
020167	122	105	104		.ASCII /RED/
020172	111	124	040		.ASCII /IT /
020175	114	111	115		.ASCII /LIM/
020200	111	124	040		.ASCII /IT /
020203	105	130	103		.ASCII /EXC/
020206	105	105	104		.ASCII /EED/
020211	105	104	000		.ASCII /ED/<00>
020214	045	101	121	P.AJL:	.ASCII /AQ/
020217	055	102	125		.ASCII / BU/
020222	123	040	115		.ASCII /S H/
020225	101	123	124		.ASCII /AST/
020230	105	122	040		.ASCII /ER /
020233	105	122	122		.ASCII /ERR/
020236	117	122	000		.ASCII /OR/<00>
020241	000				.ASCII <00>
020242	045	101	103	P.AJM:	.ASCII /AC/
020245	117	116	124		.ASCII /ONT/

(38)

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6SEQ 0097
Page 97
(34)

020250	122	117	114	.ASCII	/ROL/	
020253	114	105	122	.ASCII	/LER/	
020256	040	106	101	.ASCII	/FA/	
020261	124	101	114	.ASCII	/TAL/	
020264	040	105	122	.ASCII	/ER/	
020267	122	117	122	.ASCII	/ROR/	
020272	000	000		.ASCII	<00><00>	
020274	045	101	111	P.AJN:	.ASCII	/AI/
020277	116	123	124	.ASCII	/NST/	
020302	122	125	103	.ASCII	/RUC/	
020305	124	111	117	.ASCII	/TIO/	
020310	116	040	114	.ASCII	/N L/	
020313	117	117	120	.ASCII	/OOP/	
020316	040	124	111	.ASCII	/TI/	
020321	115	105	117	.ASCII	/MEO/	
020324	125	124	000	.ASCII	/UT/<00>	
020327	000			.ASCII	<00>	
020330	045	101	111	P.AJO:	.ASCII	/AI/
020333	114	114	105	.ASCII	/LLE/	
020336	107	101	114	.ASCII	/GAL/	
020341	040	126	111	.ASCII	/VI/	
020344	122	124	125	.ASCII	/RTU/	
020347	101	114	040	.ASCII	/AL/	
020352	103	111	122	.ASCII	/CIR/	
020355	103	125	111	.ASCII	/CUI/	
020360	124	040	111	.ASCII	/T I/	
020363	104	000	000	.ASCII	/D/<00><00>	
020366	045	101	111	P.AJP:	.ASCII	/AI/
020371	116	124	105	.ASCII	/NTE/	
020374	122	122	125	.ASCII	/RRU/	
020377	120	124	040	.ASCII	/PT/	
020402	126	105	103	.ASCII	/VEC/	
020405	124	117	122	.ASCII	/TOR/	
020410	040	111	114	.ASCII	/IL/	
020413	114	105	107	.ASCII	/LEG/	
020416	101	114	000	.ASCII	/AL/<00>	
020421	000			.ASCII	<00>	
020422	045	101	115	P.AJQ:	.ASCII	/AM/
020425	101	111	116	.ASCII	/AIN/	
020430	124	105	116	.ASCII	/TEN/	
020433	101	116	103	.ASCII	/ANC/	
020436	105	040	122	.ASCII	/E R/	
020441	105	101	104	.ASCII	/EAD/	
020444	057	127	122	.ASCII	<57>/WR/	
020447	111	124	105	.ASCII	/TTE/	
020452	040	111	116	.ASCII	/IN/	
020455	126	101	114	.ASCII	/VAL/	
020460	111	104	040	.ASCII	/ID/	
020463	122	105	107	.ASCII	/REG/	
020466	111	117	116	.ASCII	/ION/	
020471	040	111	104	.ASCII	/ID/	
020474	105	116	124	.ASCII	/ENI/	
020477	111	106	111	.ASCII	/IFI/	

H8

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0098
Page 98
(34)

020502	105	122	000	.ASCII	/ER/<00>
020505	000			.ASCII	<00>
020506	045	101	115	P.AJR:	.ASCII /MAM/
020511	101	111	116		.ASCII /ATN/
020514	124	105	116		.ASCII /TEN/
020517	101	116	103		.ASCII /ANC/
020522	105	040	127		.ASCII /E W/
020525	122	111	124		.ASCII /RIT/
020530	105	040	114		.ASCII /E L/
020533	117	101	104		.ASCII /QAD/
020536	040	124	117		.ASCII / IO/
020541	040	116	117		.ASCII / NO/
020544	116	055	114		.ASCII /N-L/
020547	117	101	104		.ASCII /QAD/
020552	101	102	114		.ASCII /ABL/
020555	105	040	103		.ASCII /E C/
020560	117	116	124		.ASCII /ONT/
020563	122	117	114		.ASCII /ROL/
020566	114	105	122		.ASCII /LER/
020571	000				.ASCII <00>
020572	045	101	103	P.AJS:	.ASCII /MAC/
020575	117	116	124		.ASCII /ONT/
020600	122	117	114		.ASCII /ROL/
020603	114	105	122		.ASCII /LER/
020606	040	122	101		.ASCII / RA/
020611	115	040	105		.ASCII /M E/
020614	122	122	117		.ASCII /RRO/
020617	122	040	050		.ASCII /R (/
020622	116	157	156		.ASCII /Non/
020625	055	160	141		.ASCII /-pa/
020630	162	151	164		.ASCII /nit/
020633	171	051	000		.ASCII /y)/<00>
020636	045	101	111	P.AJT:	.ASCII /MAI/
020641	116	111	124		.ASCII /NIT/
020644	040	123	105		.ASCII / SE/
020647	121	125	105		.ASCII /QUE/
020652	116	103	105		.ASCII /NCE/
020655	040	105	122		.ASCII / ER/
020660	122	117	122		.ASCII /ROR/
020663	000				.ASCII <00>
020664	045	101	110	P.AJU:	.ASCII /MAH/
020667	111	107	110		.ASCII /IGH/
020672	105	122	040		.ASCII /ER /
020675	114	105	126		.ASCII /LEV/
020700	105	114	040		.ASCII /EL /
020703	120	122	117		.ASCII /PRO/
020706	124	117	103		.ASCII /TUC/
020711	117	114	040		.ASCII /OL /
020714	111	116	103		.ASCII /INC/
020717	117	115	120		.ASCII /OMP/
020722	101	124	111		.ASCII /ATI/
020725	102	111	114		.ASCII /BII/
020730	111	124	131		.ASCII /IIV/

ZRQAM1 V01.6	RD/RX EXERCISER PROTECTION TABLE			11 Apr-1984 11:56:01	VAX-11 B11ss 16 V4.0-579	SEQ 0099
				11 Apr-1984 11:45:02	DISK\$USER2:(POWERS)ZRQADO.BL1;6	Page 99 (34)

020733	040	105	122		.ASCII	/ ER/
020736	122	117	122		.ASCII	/ROR/
020741	000				.ASCII	<00>
020742	045	101	120	P.AJV:	.ASCII	/MAP/
020745	125	122	107		.ASCII	/URG/
020750	105	057	120		.ASCII	/E/<57>/P/
020753	117	114	114		.ASCII	/OLL/
020756	040	110	101		.ASCII	/ HA/
020761	122	104	127		.ASCII	/RDW/
020764	101	122	105		.ASCII	/ARE/
020767	040	106	101		.ASCII	/ FA/
020772	111	114	125		.ASCII	/ILU/
020775	122	105	000		.ASCII	/RE/<00>
021000	045	101	115	P.AJW:	.ASCII	/MAM/
021003	101	120	120		.ASCII	/APP/
021006	111	116	107		.ASCII	/ING/
021011	040	122	105		.ASCII	/ RE/
021014	107	111	123		.ASCII	/GIS/
021017	124	105	122		.ASCII	/TER/
021022	040	122	105		.ASCII	/ RE/
021025	101	104	040		.ASCII	/AD /
021030	106	101	111		.ASCII	/FAI/
021033	114	125	122		.ASCII	/LUR/
021036	105	040	050		.ASCII	/E (/
021041	120	141	162		.ASCII	/Par/
021044	151	164	171		.ASCII	/ity/
021047	040	157	162		.ASCII	/ or/
021052	040	164	151		.ASCII	/ ti/
021055	155	145	157		.ASCII	/meo/
021060	165	164	051		.ASCII	/ut)/
021063	000				.ASCII	<00>
021064	017370			P.AIZ:	.WORD	P.AJA
021066	017416				.WORD	P.AJB
021070	017500				.WORD	P.AJC
021072	017562				.WORD	P.AJD
021074	017630				.WORD	P.AJE
021076	017666				.WORD	P.AJF
021100	017724				.WORD	P.AJG
021102	017772				.WORD	P.AJH
021104	020042				.WORD	P.AJI
021106	020074				.WORD	P.AJJ
021110	020164				.WORD	P.AJK
021112	020214				.WORD	P.AJL
021114	020242				.WORD	P.AJM
021116	020274				.WORD	P.AJN
021120	020330				.WORD	P.AJO
021122	020366				.WORD	P.AJP
021124	020422				.WORD	P.AJQ
021126	020506				.WORD	P.AJR
021130	020572				.WORD	P.AJS
021132	020636				.WORD	P.AJT
021134	020664				.WORD	P.AJU
021136	020742				.WORD	P.AJV

ZRGADM
VOL.6RD RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss 16 V4.0 579
DISK\$USER2:[POWERS]ZRGADO.BL.1;6SEQ 0100
Page 100
(34)

021140	021000				
021142	045	101	124	P.AJY:	.WORD P.AJW
021145	061	061	040		.ASCII /MAT/
021150	103	120	125		.ASCII /11 /
021153	040	106	101		.ASCII /CPU/
021156	111	114	125		.ASCII /FA/
021161	122	105	000		.ASCII /ILU/
021164	045	101	116	P.AJZ:	.ASCII /RE/<00>
021167	117	116	055		.ASCII /MAN/
021172	120	101	122		.ASCII /ON-/
021175	111	124	131		.ASCII /PAR/
021200	040	122	101		.ASCII /IT1/
021203	115	040	105		.ASCII /RA/
021206	122	122	117		.ASCII /M E/
021211	122	000	000		.ASCII /RRO/
021214	045	101	123	P.AKA:	.ASCII /R/<00><00>
021217	124	101	124		.ASCII /MAS/
021222	105	040	115		.ASCII /TAT/
021225	101	103	110		.ASCII /E M/
021230	111	116	105		.ASCII /ACH/
021233	040	106	101		.ASCII /INE/
021236	111	114	125		.ASCII /FA/
021241	122	105	040		.ASCII /ILU/
021244	055	040	124		.ASCII /RE /
021247	061	061	040		.ASCII /- T/
021252	101	104	104		.ASCII /11 /
021255	122	105	123		.ASCII /ADD/
021260	123	040	122		.ASCII /RES/
021263	105	107	111		.ASCII /S R/
021266	123	124	105		.ASCII /EGI/
021271	122	000	000		.ASCII /STE/
021274	045	101	123	P.AKB:	.ASCII /R/<00><00>
021277	124	101	124		.ASCII /MAS/
021302	105	040	115		.ASCII /TAT/
021305	101	103	110		.ASCII /E M/
021310	111	116	105		.ASCII /ACH/
021313	040	106	101		.ASCII /INE/
021316	111	114	125		.ASCII /FA/
021321	122	105	040		.ASCII /ILU/
021324	055	040	121		.ASCII /RE /
021327	055	102	125		.ASCII /- Q/
021332	123	040	101		.ASCII /BU/
021335	104	104	122		.ASCII /S A/
021340	105	123	123		.ASCII /DDR/
021343	040	122	105		.ASCII /ESS/
021346	107	111	123		.ASCII /RE /
021351	124	105	122		.ASCII /GIS/
021354	000	000			.ASCII /TER/
021356	045	101	123	P.AKC:	.ASCII <00><00>
021361	124	101	124		.ASCII /MAS/
021364	105	040	115		.ASCII /TAT/
021367	101	103	110		.ASCII /E M/
021372	111	116	105		.ASCII /ACH/
					.ASCII /INE/

K8

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0101
Page 101
(34)

021375	040	106	101
021400	111	114	125
021403	122	105	040
021406	055	040	103
021411	122	103	040
021414	122	105	107
021417	111	123	124
021422	105	122	000
021425	000		
021426	045	101	123
021431	124	101	124
021434	105	040	115
021437	101	103	110
021442	111	116	105
021445	040	106	101
021450	111	114	125
021453	122	105	040
021456	055	040	123
021461	105	122	111
021464	101	114	111
021467	132	105	122
021472	057	104	105
021475	123	105	122
021500	111	101	114
021503	111	132	105
021506	122	040	122
021511	105	107	111
021514	123	124	105
021517	122	000	000
021522	045	101	123
021525	124	101	124
021530	105	040	115
021533	101	103	110
021536	111	116	105
021541	040	106	101
021544	111	114	125
021547	122	105	040
021552	055	040	127
021555	122	117	116
021560	107	040	110
021563	101	122	104
021566	127	101	122
021571	105	040	126
021574	105	122	123
021577	111	117	116
021602	000	000	
021604	021142'		
021606	021164'		
021610	021214'		
021612	021274'		
021614	021356'		
021616	021426'		
021620	021522'		

P.AKD:

P.AKE:

P.AJX:

.ASCII	/FA/
.ASCII	/ILU/
.ASCII	/RE /
.ASCII	/C/
.ASCII	/RC /
.ASCII	/REG/
.ASCII	/ISI/
.ASCII	/ER/<00>
.ASCII	<00>
.ASCII	/AS/
.ASCII	/TAT/
.ASCII	/E M/
.ASCII	/ACH/
.ASCII	/INE/
.ASCII	/FA/
.ASCII	/ILU/
.ASCII	/RE /
.ASCII	/S/
.ASCII	/ERI/
.ASCII	/ALI/
.ASCII	/ZER/
.ASCII	<57>/DE/
.ASCII	/SER/
.ASCII	/IAL/
.ASCII	/IZE/
.ASCII	/R R/
.ASCII	/EGI/
.ASCII	/SIE/
.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/
.ASCII	<00><00>
.WORD	P.AJY
.WORD	P.AJZ
.WORD	P.AKA
.WORD	P.AKB
.WORD	P.AKC
.WORD	P.AKD
.WORD	P.AKE

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0,BL1;6SEQ 0102
Page 102
(34)

021622	045	116	045	P,AKF:	.ASCII	/N#
021625	101	132	122		.ASCII	/AZR/
021630	121	101	040		.ASCII	/QA /
021633	104	105	126		.ASCII	/DEV/
021636	040	106	124		.ASCII	/ FT/
021641	114	040	040		.ASCII	/L /
021644	045	132	065		.ASCII	/Z5/
021647	045	101	040		.ASCII	/A /
021652	117	116	040		.ASCII	/ON /
021655	125	116	111		.ASCII	/UNI/
021660	124	040	045		.ASCII	/T #/
021663	132	062	045		.ASCII	/Z2#/
021666	101	040	124		.ASCII	/A T/
021671	123	124	040		.ASCII	/ST /
021674	060	060	061		.ASCII	/001/
021677	040	123	125		.ASCII	/ SU/
021702	102	040	060		.ASCII	/B 0/
021705	060	060	040		.ASCII	/00 /
021710	120	103	072		.ASCII	/PC:/
021713	040	045	117		.ASCII	/ #0/
021716	066	000			.ASCII	/6/<00>
021720	045	116	045	P,AKG:	.ASCII	/N#
021723	101	132	122		.ASCII	/AZR/
021726	121	101	040		.ASCII	/QA /
021731	110	122	104		.ASCII	/HRD/
021734	040	105	122		.ASCII	/ ER/
021737	122	040	040		.ASCII	/R /
021742	045	132	065		.ASCII	/Z5/
021745	045	101	040		.ASCII	/A /
021750	117	116	040		.ASCII	/ON /
021753	125	116	111		.ASCII	/UNI/
021756	124	040	045		.ASCII	/T #/
021761	132	062	045		.ASCII	/Z2#/
021764	101	040	124		.ASCII	/A T/
021767	123	124	040		.ASCII	/ST /
021772	060	060	061		.ASCII	/001/
021775	040	123	125		.ASCII	/ SU/
022000	102	040	060		.ASCII	/B 0/
022003	060	060	040		.ASCII	/00 /
022006	120	103	072		.ASCII	/PC:/
022011	040	045	117		.ASCII	/ #0/
022014	066	000			.ASCII	/6/<00>
022016	045	116	045	P,AKH:	.ASCII	/N#
022021	101	132	122		.ASCII	/AZR/
022024	121	101	040		.ASCII	/QA /
022027	123	106	124		.ASCII	/SET/
022032	040	105	122		.ASCII	/ ER/
022035	122	040	040		.ASCII	/R /
022040	045	132	065		.ASCII	/Z5/
022043	045	101	040		.ASCII	/A /
022046	117	116	040		.ASCII	/ON /
022051	125	116	111		.ASCII	/UNI/
022054	124	040	045		.ASCII	/T #/

M8

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0103
Page 103
(34)

022057	132	062	045	.ASCII	/Z2#
022062	101	040	124	.ASCII	/A T/
022065	123	124	040	.ASCII	/ST /
022070	060	060	061	.ASCII	/001/
022073	040	123	125	.ASCII	/ SU/
022076	102	040	060	.ASCII	/B O/
022101	060	060	040	.ASCII	/00 /
022104	120	103	072	.ASCII	/PC:/
022107	040	045	117	.ASCII	/ #0/
022112	066	045	116	.ASCII	/6#N/
022115	000			.ASCII	<00>
022116	045	116	045	P.AKI: .ASCII	/#N#
022121	101	111	057	.ASCII	/AI/<57>
022124	117	040	122	.ASCII	/O R/
022127	105	121	125	.ASCII	/EQU/
022132	105	123	124	.ASCII	/EST/
022135	040	106	101	.ASCII	/ FA/
022140	111	114	105	.ASCII	/ILE/
022143	104	045	116	.ASCII	/D#N/
022146	000	000		.ASCII	<00><00>
022150	045	123	064	P.AKJ: .ASCII	/#S4/
022153	000			.ASCII	<00>
022154	045	116	000	P.AKK: .ASCII	/#N/<00>
022157	000			.ASCII	<00>
022160	045	101	040	P.AKI: .ASCII	/#A /
022163	055	040	000	.ASCII	/- /<00>
022166	045	101	052	P.AKM: .ASCII	/#A*/
022171	040	000	000	.ASCII	/ /<00><00>
022174	000000C			L.\$HWLEN: .WORD	<<L\$NDHW-L\$HWLEN>/2>
022176	172150			HWPT.IP.ADDR: .WORD	-5630
022200	000154			HWPT.VECTOR: .WORD	154
022202	000004			HWPT.BR.LEVEL: .WORD	4
022204	100340			HWPT.DISK: .WORD	-77440
022206	000000			HWPTS0.LBN: .WORD	0
022210	000000			HWPTS1.LBN: .WORD	0
022212	177777			HWPTF0.LBN: .WORD	-1
022214	000000			HWPTF1.LBN: .WORD	0
022216				L\$NDHW: .BLKW	1
022220	000000C			L\$SWLEN: .WORD	<<L\$NDSW-L\$SWLEN>/2>
022222	000040			SWP.ERROR: .WORD	40
022224	000000			SWP.XFER: .WORD	0

N8

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1:55-16 V4.0-579
DTSK:USER2:[POWERS]ZRQAD0.BL1:6

SEQ 0104
Page 104
(34)

022226 054046
022230 000000
022232 000143
022234 000000
022236 000013
022240 000020
022242
022302
022304 000000
022306 177777
022310 000006

SWP.FLAGS::
.WORD 54046
SWP.DPAT::
.WORD 0
SWP.RAT::
.WORD 143
SWP.TIME::
.WORD 0
DUPROUND::
.WORD 13
SWP.UCNT::
.WORD 20
SWP.UDPAT::
.BLKW 20
L\$NDSW::.BLKW 1
L\$PROT::.WORD 0
.WORD 1
.WORD 6

000000
000000
000126

000130
000152

000154
000156

000160
000200
000530
000532
001534
001534 001
001535 001
001536 001
001537 001
001540 000020

.PSECT \$FFF\$, D , GBL
CST::.BLKW 53
CST.ADDR::
.BLKW 1
DCT::.BLKW 11
DCT.ADDR::
.BLKW 1
RDRX.ADDR::
.BLKW 1
IRDRX.ADDR::
.BLKW 1
BST::.BLKW 10
TALLY::.BLKW 154
T.ADDR::.BLKW 1
DUPPKT::.BLKW 401
TRK.SGN::
.BYTE 1
.BYTE 1
.BYTE 1
.BYTE 1
RDM.CNT::
.WORD 20
RANDOM::.BLKW 20
C.FRR.TBL::
.BLKW 1
MSCP.PKT::
.BLKW 644
IPKT.ADDR::
.BLKW 1
PKT.USE::
.BLKW 6
RETPKT::.BLKW 260

001542
001602

001604

003314

003316

003332

{ 1 }

ZRQAM1
VOL 1.6

RD RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11-B1155-16 V4.0 579
DISK\$USERP:[POWERS]ZRQADO.BL1.6

SEQ 0105
Page 105
(34)

004012	RP,USE::	.BLKW	4
004107	RP,INDX::		
		.BLKW	1
004104	RP,ADDR::		
		.BLKW	1
004106	FLAG,PKT::		
		.BLKW	655
005640	BUFF,ADDR::		
		.BLKW	10
005660	BUFF,OWN::		
		.BLKW	4
005670	IODQ::	.BLKW	4
005700	IODQ,IN::		
		.BLKW	1
005707	IODQ,OUT::		
		.BLKW	1
005704	ENTRY,REASON::		
		.BLKB	1
005705	FOP,FLAG::		
		.BLKB	1
005706	DUP,FLAGS::		
		.BLKW	1
005710	CTLR::	.BLKW	1
005712	CDISK::	.BLKW	1
005714	CUOFF::	.BLKW	1
005716	CTLR,CNT::		
		.BLKW	1
005720	DUR::	.BLKW	2
005724	QIO::	.BLKB	1
		.EVEN	
005726	FREE,MEM,ADDR::		
		.BLKW	1
005730	BYTES,PER,QIO::		
		.BLKW	1
005732	ST,CODE::		
		.BLKW	1
005734	SB,CODE::		
		.BLKW	1
005736	STEP::	.BLKW	1
005740	OF,RC::	.BLKW	1
005742	SA,REG::	.BLKW	1
005744	CMD,TIME::		
		.BLKW	1
005746	NEX::	.BLKW	1
005750	CRN,LOW::		
		.BLKW	1
005752	CRN,HIGH::		
		.BLKW	1
005754	CREDIT,BAL::		
		.BLKW	1
005756	NEXT,PRT,USE::		
		.BLKB	1
005757	HOURS::	.BLKB	1

[9]

PROGRAM
V01.6 RD RX EXERCISER
PROTECTION TABLE

11 Apr-1984 11:56:01
11 Apr-1984 11:45:02

VAX 11 B1184 16 V4.0 5/9
DISK\$USER2:[POWER5]ZRGADO.BL1;6

SEQ 0106
Page 106
(34)

005760		MINUTES::	
		.BLKB	1
005762		.EVEN	
		CLK,TICKS::	
005764		.BLKW	1
		CLK,PRESENT::	
005765		.BLKB	1
		HOF,FLAG::	
005766		.BLKB	1
		S,PATTERN::	
005770		.BLKW	1
		S,DUPPKT::	
005772		.BLKW	1
		P,INDEX::	
005774		.BLKW	1
		FORCED.ERROR::	
		.BLKB	1
		.EVEN	
005776		FER,LBN::	
		.BLKW	1
006000		FER,BC::	1
006002		INIT.OCCURED::	
006002	000	.BYTE	0
006003		ADDR,VECT.OK::	
006003	000	.BYTE	0

.GLOBL L\$SOFT, L\$PTHV, L\$RPT, L\$INIT
.GLOBL L\$CLEAN, L\$LAST, L\$HARD, L\$QVTP
.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
000100	BIT6--	100
000040	BIT5--	40

[]

ZRQAM1 V01.6	RD/RX EXERCISER PROTECTION TABLE	11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 B1199 16 V4.0 5/9 DISK\$USER2:[POWERS]ZRQAD0.BL1;6	SEQ 0107 Page 107 (34)
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		
022222	L\$SW--	L\$SWLEN*2		
022176	L\$HW--	L\$HWLEN*2		
000011	L\$DEPO--	L\$REV*1		
000136	HWQ1--	P,AAA		
000152	HWQ2--	P,AAH		
000162	HWQ3--	P,AAC		
000174	HWQ4--	P,AAD		
000212	HWQ5--	P,AAE		
000262	HWQ6A--	P,AAI		
000334	HWQ6B--	P,AAJ		
000410	HWQ7A--	P,AAK		
000460	HWQ7B--	P,AAL		
000530	HWQ8--	P,AAM		
000602	HWQ9--	P,AAN		
000702	HWQ10--	P,AAO		
000732	HWQ11--	P,AAQ		
000764	SWQ1--	P,AAQ		
001006	SWQ2--	P,AAQ		
001070	SWQ4--	P,AAQ		
001112	SWQ7--	P,AAQ		

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0108
Page 108
(34)

001164'	SWQ9**	P.AAR
001240'	SWQ10**	P.AAS
001304'	SWQ11**	P.AAT
001336'	SWQ12**	P.AAU
001434'	SWQ13**	P.AAV
001512'	SWQ14**	P.AAW
001564'	SWQ15**	P.AAX
001634'	SWQ17**	P.AAY
001732'	SWQ19**	P.AAZ
002022'	SWQ20**	P.ABA
002120'	SWQ21**	P.ABB
002222'	SWQ22**	P.ABC
002300'	SWQ23**	P.ABD
002372'	SWQ24**	P.ABE
002470'	SWQ25**	P.ABF
002570'	SWM1**	P.ABG
002660'	NULL**	P.ABH
002662'	DBM5**	P.ABI
002710'	DBM12**	P.ABJ
002764'	DBM15**	P.ABK
003014'	DBM18**	P.ABL
003066'	DBM19**	P.ABM
003152'	DBM20**	P.ABN
003226'	DBM21**	P.ABO
003310'	DBM22**	P.ABP
003354'	DBM23**	P.ABQ
003412'	DBM25**	P.ABR
003460'	DBM26**	P.ABS
003512'	DBM27**	P.AET
003564'	DBM28A**	P.ABU
003624'	DBM28B**	P.ABV
003664'	DBM29**	P.ABW
003732'	DBM32**	P.ABX
004006'	DBM101**	P.ABY
004034'	DBM104**	P.ABZ
004076'	DBM105**	P.ACA
004134'	DBM107**	P.ACB
004172'	DBM108**	P.ACC
004262'	DBM109**	P.ACD
004342'	DBM111**	P.ACE
004442'	DBM112**	P.ACF
004544'	DBM120**	P.ACG
004636'	DBM121**	P.ACH
004726'	DU.MSG**	P.ACI
005446'	DU.RSN**	P.ACJ
005474'	MSG.01**	P.ACV
005526'	MSG.02**	P.ACW
005562'	MSG.03**	P.ACX
005614'	RPT1**	P.ACY
005700'	RPT2**	P.ACZ
005744'	RPT3**	P.ADA
006030'	RPT4**	P.ADB
006074'	RPT5**	P.ADC

ZRQAM1
V01.6

RD/RX EXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0109
Page 109
(34)

006162'	RPT6**	P.ADD
006226'	RPT7**	P.ADE
006244'	RPT8**	P.ADF
006272'	RPT9**	P.ADG
006324'	RPT10**	P.ADH
006412'	RPT11**	P.ADI
006460'	EGS.01**	P.ADJ
006500'	EGS.02**	P.ADK
006572'	EGD.10**	P.ADL
006632'	EGD.11**	P.ADM
006656'	EGD.12**	P.ADN
006704'	EGD.13**	P.ADO
006732'	EGD.14**	P.ADP
006762'	EGD.15**	P.ADQ
007000'	EGD.16**	P.ADR
007030'	EGD.17**	P.ADS
007046'	EGD.18**	P.ADT
007066'	EGD.19**	P.ADU
007126'	EGD.20**	P.ADV
007214'	EGD.21**	P.ADW
007326'	EGD.22**	P.ADX
007366'	EGD.23**	P.ADY
007430'	EGD.24**	P.ADZ
007474'	EGH.30**	P.AEA
007520'	EBS.01**	P.AEB
007562'	EBD.10**	P.AEC
007622'	EBD.12**	P.AED
007670'	EBD.13**	P.AEE
007722'	EBD.14**	P.AEF
007762'	EBD.18**	P.AEG
010016'	EBD.19**	P.AEH
010076'	EBD.24**	P.AEI
010152'	EH.0**	P.AEJ
010210'	EH.1**	P.AEK
010246'	EH.2**	P.AEL
010306'	EH.3**	P.AEM
010344'	EH.4**	P.AEN
010370'	EH.5**	P.AEO
010420'	EH.6**	P.AEP
010450'	EH.7**	P.AEQ
010500'	EH.8**	P.AER
010534'	EH.9**	P.AES
010564'	EH.10**	P.AET
010614'	EH.12**	P.AEU
010650'	EH.13**	P.AEV
010722'	ERR.00**	P.AEW
011412'	ERR.COD**	P.AEX
011446'	ELG.00**	P.AFM
012000'	ELG.FMT**	P.AFN
012012'	EX.SA**	P.AFT
012030'	EX.SC**	P.AFU
012060'	EX.SBU**	P.AFV
012064'	EX.SB**	P.AFW

ZRQAM1
V01.6RD/RX EXERCISER
PROTECTION TABLE11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0110
Page 110
(34)

012106'	EX.CMD**	P.AFX
012126'	EX.RD**	P.AFY
012136'	EX.WRT**	P.AFZ
012146'	EX.CMP**	P.AGA
012162'	EX.ONL**	P.AGB
012174'	EX.ACC**	P.AGC
012206'	EX.OP**	P.AGD
012212'	EX.BB**	P.AGE
012302'	EX.BB1**	P.AGF
012376'	EX.BBU**	P.AGG
012466'	EX.LBN**	P.AGH
012524'	EX.PBN**	P.AGI
012562'	EX.LBR**	P.AGJ
012630'	EX.LBW**	P.AGK
012676'	EX.RBN**	P.AGL
012756'	EX.CBC**	P.AGM
013022'	EX.CBR**	P.AGN
013074'	EX.CBW**	P.AGO
013146'	EX.BC**	P.AGP
013222'	EX.BD**	P.AGQ
013300'	EX.BDR**	P.AGR
013370'	EX.BDW**	P.AGS
013460'	EX.RP**	P.AGT
013546'	EX.WRD**	P.AGU
013556'	EX.TIM**	P.AGV
013616'	XX13**	P.AGW
013640'	XX23**	P.AGX
013674'	XX32**	P.AGY
013722'	XX33**	P.AGZ
013750'	XX34**	P.AHA
014030'	CER.01**	P.AHB
014074'	CER.02**	P.AHC
014150'	SC.SDI**	P.AHD
014174'	SC.CON**	P.AHE
014216'	SC.DUP**	P.AHF
014246'	SC.ONL**	P.AHG
014270'	SC.SON**	P.AHH
014310'	SC.UNK**	P.AHI
014370'	SC.VOL**	P.AHJ
014450'	SC.IOP**	P.AHK
014522'	SC.DIS**	P.AHL
014614'	SC.FER**	P.AHM
014702'	SC.FE2**	P.AHN
014772'	SC.ISH**	P.AHO
015052'	SC.IS2**	P.AHP
015132'	SC.DST**	P.AHQ
015206'	SC.DS2**	P.AHR
015260'	SC.ECC**	P.AHS
015342'	SC.ECD**	P.AHT
015374'	SC.RCT**	P.AHU
015414'	SC.FUL**	P.AHV
015470'	SC.S76**	P.AHW
015544'	SC.FCT**	P.AHX

[-9]

ZRQAM1
V01.6

RD/RX LEXERCISER
PROTECTION TABLE

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0111
Page 111
(34)

015612	SC.EC1**	P.AHY
015642	SC.EC2**	P.AHZ
015672	SC.EC3**	P.AIA
015724	SC.EC4**	P.AIB
015754	SC.EC5**	P.AIC
016004	SC.EC6**	P.AID
016034	SC.EC7**	P.AIE
016066	SC.EC8**	P.AIF
016120	SC.EC9**	P.AIG
016162	SC.SWP**	P.AIH
016222	SC.HWP**	P.AII
016262	SC.ODA**	P.AIJ
016312	SC.ODB**	P.AIK
016334	SC.NXM**	P.AIL
016370	SC.PAR**	P.AIM
016424	SC.CTO**	P.AIN
016476	SC.SDS**	P.AIO
016554	SC.EDC**	P.AIP
016614	SC.IDS**	P.AIQ
016664	SC.SRI**	P.AIR
016756	SC.SRI**	P.AIS
017044	SC.POE**	P.AIT
017100	SC.RDY**	P.AIU
017162	SC.CLK**	P.AIV
017210	SC.RSP**	P.AIW
017256	SC.SUR**	P.AIX
017306	SC.PSP**	P.AIY
021064	CNTR.ERR**	P.AIZ
021604	RDRX.ERR**	P.AJX
021622	DF.MSG**	P.AKF
021720	HRD.MSG**	P.AKG
022016	SFT.MSG**	P.AKH
022116	HRD.SUB**	P.AKI
022150	SPACE4**	P.AKJ
022154	CRLF**	P.AKK
022160	DASH**	P.AKL
022166	ASTERISK**	P.AKM
022176	DFPTBL**	L\$HWLEN+2
022222	SFPTBL**	L\$SWLEN+2

PSECT SUMMARY

Psect Name	Words	Attributes
\$CODE\$	4709	RO , ! , LCL , REL , CON
\$FFF\$	1538	RW , D , GBL , REL , CON

Library Statistics

ZRQAM1	RD/RX EXERCISER	11-Apr-1984 11:56:01	VAX-11 Bliss-16 V4.0 579	SEQ 0112
V01.6	PROTECTION TABLE	11 Apr-1984 11:45:02	DISK\$USER2:[POWERS]ZRQADO.BL1:6	Page 112 (34)

File	Symbols		Pages	Processing
	Total	Loaded		
DISK\$USER2:[POWERS]ZRQADO.L16:3	404	181	21	00:00.1

```

:
:          COMMAND QUALIFIER:
:
:  BLISS/PDP11 ZRQADO.BL1/LIST=ZRQADO.LS1/OBJECT=ZRQADO.OB1/SOURCE=PAGE:53

```


(J)

ZRQAM, RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 Bliss-16 V4.0-579 SEQ 0113
 PROTECTION TABLE 11-Apr-1984 11:45:02 DISK\$USER2:[POWERS]ZRQADO.BL1;6 Page 113
 (35)

```

: 3214 0 module ZRQAM2 (
: 3215 0
: 3216 0 *title 'RD/RX EXERCISER'
: 3217 0         ident = 'V01.6',
: 3218 0         addressing_mode (absolute),
: 3219 0         environment (noeis)
: 3220 0     ) =
: 3221 0
: 3222 1 begin
: 3223 1
: 3224 1 *sbttl 'DECLARATIONS'
: 3225 1
: 3226 1 library 'ZRQADO.L16';           ! RDRX EXERCISER GLOBAL LIBRARY
: 3227 1
: 3228 1 require 'BLSMAC.REQ';         ! DIAGNOSTIC SUPERVISOR LIBRARY
: 4719 1
: 4720 1 forward routine
: 4721 1     NEX_TRAP : L$ISR novalue,
: 4722 1     EMS_01 : novalue,
: 4723 1     EMS_TIM : novalue,
: 4724 1     EMS_DBN : NOVALUE,
: 4725 1     EMS_BLK : NOVALUE,
: 4726 1     SET_CPAR : novalue,
: 4727 1     SET_UPAR : novalue;
: 4728 1
: 4729 1 external
: 4730 1     CST : blockvector [MAX_CTLR, CST_LEN, word] field (CST_FIELDS),
: 4731 1         ! RUN-TIME CONTROLLER STATUS TABLES
: 4732 1     CST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 4733 1         ! CONTROLLER STATUS TABLE ADDRESS OF "CURRENT" CONTROLLER
: 4734 1     DCT : blockvector [MAX_CTLR, DCT_LEN, word] field (DCT_FIELDS),
: 4735 1         ! DRIVER CONTROLLER TABLES
: 4736 1     DCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 4737 1         ! ADDRESS OF "CURRENT" DRIVER CONTROLLER TABLE
: 4738 1     RDRX_ADDR : ref ndrx field (RC_REG),
: 4739 1         ! DEVICE ADDRESS OF "CURRENT" CONTROLLER
: 4740 1     IRDRX_ADDR : ref ndrx field (RC_REG),
: 4741 1         ! DEVICE ADDRESS OF INTERRUPTING CONTROLLER
: 4742 1     BST : BLOCKVECTOR [MAX_UNITS, 2, WORD],
: 4743 1         !CONTAINS LBNS (HI + LO FIELDS) FOR SEQUENTIAL
: 4744 1         !I/O TRANSFER FOR EACH UNIT.
: 4745 1     TALLY : vector [MAX_UNITS * TALLY_LEN, word] field (T_FIELDS),
: 4746 1         ! STATISTICS TABLES
: 4747 1     T_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 4748 1         ! ADDRESS OF STATISTICS TABLE (TALLY) FOR CURRENT UNIT
: 4749 1     DUPPKT : BLOCK [257, WORD] FIELD (DP_FIELDS),
: 4750 1         !INFO FROM RECEIVE AND SEND CMDS
: 4751 1     TRK_SGN : VECTOR [MAX_UNITS, BYTE, SIGNED], !CURRENT TRACK DIRECTION
: 4752 1     RDM_CNT : WORD,
: 4753 1     RANDOM : VECTOR [RDM_LEN, WORD],
: 4754 1     C_ERR_TBL : blockvector [MAX_CTLR, C_ERR_LEN, word] field (C_ERR_FIELDS),
: 4755 1         ! STATISTICS TABLE FOR CONTROLLER ERRORS
: 4756 1     MSCP_PKT : blockvector [PKT_CNT, PKT_LEN, word] field (PKT_FIELDS),

```

ZRRAMP
VOL.6RD/RX EXERCISER
DECLARATIONS11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRRADO.BL1;6SEQ 0114
Page 114
(35)

```

: 4757 1      ! MSCP PACKET POOL
: 4758 1      IPKT_ADDR : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 4759 1      ! ADDRESS OF AN MSCP PACKET (INTERRUPT PROCESSING)
: 4760 1      PKT_USE : vector [PKT_CNT, byte, signed],
: 4761 1      ! MSCP PACKET POOL ALLOCATION TABLE
: 4762 1      RETPKT : blockvector [RP_CNT, RP_LEN, word] field (RP_FIELDS),
: 4763 1      ! RETURN PACKET POOL
: 4764 1      RP_USE : vector [RP_CNT, byte, signed],
: 4765 1      ! RETURN PACKET POOL ALLOCATION TABLE
: 4766 1      RP_INDX : word,      ! CURRENT RETURN PACKET INDEX
: 4767 1      RP_ADDR : ref block [RP_LEN, word] field (RP_FIELDS),
: 4768 1      ! CURRENT RETURN PACKET ADDRESS
: 4769 1      ELOG_PKT : blockvector [EP_CNT + 1, EP_LEN, word] field (EP_FIELDS),
: 4770 1      ! ERROR-LOG PACKET SAVE AREA
: 4771 1      BUFF_ADDR : vector [MAX_BUF_CNT],      ! TABLE OF I/O BUFFER DESCRIPTORS
: 4772 1      BUFF_OWN : vector [MAX_BUF_CNT, byte, signed],      ! I/O BUFFER OWNERSHIP (CONTROLLER NUMBER)
: 4773 1      IODQ : vector [IODQ_LEN, byte],      ! I/O DONE QUEUE - CIRCULAR QUEUE OF RETPKT INDECS
: 4774 1      IODQ_IN : word,      ! I/O DONE QUEUE IN POINTER
: 4775 1      IODQ_OUT : word,      ! I/O DONE QUEUE OUT POINTER
: 4776 1      ENTRY_REASON : byte,      ! CURRENT OPERATOR COMMAND
: 4777 1      EOP_FLAG : byte,      ! END-OF-PASS FLAG
: 4778 1      DUP_FLAGS : word,      ! DUP FLAGS      ZZZ
: 4779 1      CCTLR : word,      ! NUMBER OF "CURRENT" CONTROLLER
: 4780 1      CDISK : word,      ! CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 4781 1      CUOFF : word,      ! CURRENT UNIT CST OFFSET
: 4782 1      CTLR_CNT : word,      ! TOTAL NUMBER OF CONFIGURED CONTROLLERS
: 4783 1      DUR : vector [MAX_UNITS, byte],      ! DROP UNIT REASON
: 4784 1      QIO : vector [MAX_CTLR, byte],      ! NUMBER OF OUTSTANDING QIOs PER CONTROLLER
: 4785 1      FREE_MEM_ADDR,      ! START OF FREE MEMORY
: 4786 1      BYTS_PER_QIO : word,      ! SIZE (BYTES) OF AN I/O BUFFER
: 4787 1      ST_CODE : word,      ! CURRENT STATUS CODE
: 4788 1      SB_CODE : word,      ! CURRENT SUB-CODE
: 4789 1      STEP : word,      ! CURRENT STEP IN HARD INIT
: 4790 1      OF_RC : signed word,      ! OFFSET (0 OR 2) TO READ IP OR SA
: 4791 1      SA_REG : word,      ! STORAGE FOR SA REGISTER READS AND WRITES
: 4792 1      CMD_TIME : word,      ! COMMAND TIMEOUT VALUE (IN SECONDS)
: 4793 1      NEX : word,      ! NON-EXISTENT MEMORY TRAP INDICATOR
: 4794 1      CRN_LOW : word,      ! COMMAND REF NUMBER OF LAST COMMAND SENT
: 4795 1      CRN_HIGH : word,      ! COMMAND REF NUMBER (HI ORDER)
: 4796 1      CREDIT_BAL : word,      ! CREDIT BALANCE
: 4797 1      NEXT_PKT_USE : byte,      ! POINTER TO NEXT ENTRY IN PKT_USE TABLE
: 4798 1      HOURS : byte,      ! TIME OF DAY (HOURS)
: 4799 1      MINUTES : byte,      ! TIME OF DAY (MINUTES)
: 4800 1      CLK_TICKS : word,      ! TIME OF DAY (LINE-CLOCK TICKS)
: 4801 1      CLK_PRESENT : byte,      ! FLAG INDICATES IF LINE-CLOCK PRESENT
: 4802 1      HOE_FLAG : byte,      ! FLAG INDICATES IF "HALT ON ERROR" FLAG SET
: 4803 1      FORCED_ERROR : byte,      ! "FORCED ERROR" DETECTED IN LAST READ
: 4804 1      FER_LBN : word,      ! LBN OF THE "FORCED ERROR" BLOCK
: 4805 1      FER_BC : word,      ! BYTE COUNT OF THE "FORCED ERROR" BLOCK
: 4806 1      INIT_OCCURED : byte,      ! EXERCISER INITIALIZATION COMPLETE
: 4807 1      ADDR_VECT_OK : byte,      ! FLAG INDICATES IF ADDRESS/VECTOR TEST PASSED
: 4808 1      DBMS,
: 4809 1      P_INDEX : SIGNED WORD,      ! CURRENT MESSAGE PACKET INDEX      ZZZ

```

ZRQAM2
V01.6RD/RX EXERCISER
DECLARATIONS11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0115
Page 115
(35)

```

: 4810 1      S_PATTERN : WORD,
: 4811 1      S_DUPPKT : WORD,
: 4812 1      DDM107,
: 4813 1      DU_MSG,
: 4814 1      DU_RSN : vector [11],
: 4815 1      RPT1,
: 4816 1      RPT2,
: 4817 1      RPT3,
: 4818 1      RPT4,
: 4819 1      RPT5,
: 4820 1      RPT6,
: 4821 1      RPT7,
: 4822 1      RPT8,
: 4823 1      RPT9,
: 4824 1      RPT10,
: 4825 1      RPT11,
: 4826 1      MSG_01,
: 4827 1      EGS_01,
: 4828 1      EBS_01,
: 4829 1      EBD_10,
: 4830 1      EBD_12,
: 4831 1      EBD_13,
: 4832 1      EBD_14,
: 4833 1      EBD_18,
: 4834 1      EBD_19,
: 4835 1      EBD_24,
: 4836 1      ERR_00,
: 4837 1      ERR_COD : vector [14],
: 4838 1      ELG_00,
: 4839 1      ELG_FMT : vector [5],
: 4840 1      EX_TIM,
: 4841 1      XX13,
: 4842 1      XX23,
: 4843 1      XX32,
: 4844 1      XX33,
: 4845 1      XX34,
: 4846 1      EX_SA,
: 4847 1      EX_SC,
: 4848 1      EX_SBO,
: 4849 1      EX_SB,
: 4850 1      EX_RP,
: 4851 1      EX_WRD,
: 4852 1      EX_CMD,
: 4853 1      EX_RD,
: 4854 1      EX_WRT,
: 4855 1      EX_CMP,
: 4856 1      EX_ONL,
: 4857 1      EX_ACC,
: 4858 1      EX_OP,
: 4859 1      EX_BB,
: 4860 1      EX_BB1,
: 4861 1      EX_BB0,
: 4862 1      EX_LBN,

```

```

!PATTERN FOR DUP WRITES
!DBN BYTE COUNTER

```

```

ZZZ
ZZZ

```

```

!ZZZ
!ZZZ
!ZZZ
!ZZZ
!ZZZ

```

M9

ZRQAM2
VO1.6

RD/RX EXERCISER
DECLARATIONS

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0116
Page 116
(35)

:	4863	1	EX_PBN,
:	4864	1	EX_LBR,
:	4865	1	EX_LBW,
:	4866	1	EX_RBN,
:	4867	1	EX_CBC,
:	4868	1	EX_CBR,
:	4869	1	EX_CBW,
:	4870	1	EX_BC,
:	4871	1	EX_BD,
:	4872	1	EX_BDR,
:	4873	1	EX_BDW,
:	4874	1	SC_SDI,
:	4875	1	SC_CON,
:	4876	1	SC_DUP,
:	4877	1	SC_ONL,
:	4878	1	SC_SON,
:	4879	1	SC_UNK,
:	4880	1	SC_VOL,
:	4881	1	SC_IOP,
:	4882	1	SC_DIS,
:	4883	1	SC_FER,
:	4884	1	SC_FE2,
:	4885	1	SC_ISH,
:	4886	1	SC_IS2,
:	4887	1	SC_DST,
:	4888	1	SC_DS2,
:	4889	1	SC_ECC,
:	4890	1	SC_ECD,
:	4891	1	SC_RCT,
:	4892	1	SC_FUL,
:	4893	1	SC_576,
:	4894	1	SC_FCT,
:	4895	1	SC_SWP,
:	4896	1	SC_HWP,
:	4897	1	SC_EC1,
:	4898	1	SC_EC2,
:	4899	1	SC_EC3,
:	4900	1	SC_EC4,
:	4901	1	SC_EC5,
:	4902	1	SC_EC6,
:	4903	1	SC_EC7,
:	4904	1	SC_EC8,
:	4905	1	SC_EC9,
:	4906	1	SC_ODA,
:	4907	1	SC_ODB,
:	4908	1	SC_NXM,
:	4909	1	SC_PAR,
:	4910	1	SC_CTO,
:	4911	1	SC_SDS,
:	4912	1	SC_EDC,
:	4913	1	SC_IDS,
:	4914	1	SC_SRI,
:	4915	1	SC_SR1,

N9

ZRQAMP
V01.6

RD/RX EXERCISER
DECLARATIONS

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 011/
Page 117
(35)

```

: 4916 1 SC_POE,
: 4917 1 SC_RD1,
: 4918 1 SC_CLK,
: 4919 1 SC_RSP,
: 4920 1 SC_SUR,
: 4921 1 SC_PSP,
: 4922 1 CER_01,
: 4923 1 CER_02,
: 4924 1 CNTR_ERR : vector [23],
: 4925 1 RDRX_ERR : vector [7],
: 4926 1 SPACE4,
: 4927 1 CRLF,
: 4928 1 DASH,
: 4929 1 ASTERISK,
: 4930 1 HWQ1,
: 4931 1 HWQ2,
: 4932 1 HWQ3,
: 4933 1 HWQ4,
: 4934 1 HWQ5,
: 4935 1 HWQ6A,
: 4936 1 HWQ6B,
: 4937 1 HWQ7A,
: 4938 1 HWQ7B,
: 4939 1 HWQ8,
: 4940 1 HWQ9,
: 4941 1 HWQ10,
: 4942 1 HWQ11,
: 4943 1 SWQ1,
: 4944 1 SWQ2,
: 4945 1 SWQ4,
: 4946 1 SWQ7,
: 4947 1 SWQ9,
: 4948 1 SWQ10,
: 4949 1 SWQ11,
: 4950 1 SWQ12,
: 4951 1 SWQ13,
: 4952 1 SWQ14,
: 4953 1 SWQ15,
: 4954 1 SWQ17,
: 4955 1 SWQ19,
: 4956 1 SWQ20,
: 4957 1 SWQ21,
: 4958 1 SWQ22,
: 4959 1 SWQ23,
: 4960 1 SWQ24,
: 4961 1 SWQ25,
: 4962 1 EH_0,
: 4963 1 EH_1,
: 4964 1 EH_2,
: 4965 1 EH_3,
: 4966 1 EH_4,
: 4967 1 EH_5,
: 4968 1 EH_6,

```

!ZZZ
!ZZZ

!ZZZ
!ZZZ
!ZZZ
!ZZZ
!ZZZ
!ZZZ
!ZZZ

BIO

ZRQAM,
VOL.6

RD RX EXERCISER
DECLARATIONS

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX-11 B11:16 V4.0 5/9
DISK:USER2:[POWERS]ZRQADO.BI 1:6

SEQ 0118
Page 118
(35)

```

: 4969 1      EH 7,
: 4970 1      EH 8,
: 4971 1      EH 9,
: 4972 1      EH 10,
: 4973 1     EH 12,
: 4974 1     EH 13,
: 4975 1     SWM1,
: 4976 1     NULL,
: 4977 1     SWP_FLAGS : word,
: 4978 1     L3MEM,
: 4979 1     L3LIM,
: 4980 1     L3UNIT,
: 4981 1     ! O BRK;
: 4982 1
: 4983 1
: 4984 1     own
: 4985 1     TBL_SUC : vector [17] initial (NULL, SC_SDI, SC_CON, NULL, SC_DUP, NULL, NULL,
: 4986 1     NULL, SC_ONL, NULL, NULL, NULL, NULL, NULL, NULL, SC_SON),
: 4987 1     TBL_OFI : vector [9] initial (SC_UNK, SC_VOL, SC_IOP, NULL, SC_DUP, NULL, NULL,
: 4988 1     NULL, SC_DIS),
: 4989 1     TBL_MFE : vector [11] initial (SC_FER, NULL, SC_ISH, SC_DST, SC_EC9, SC_E76,
: 4990 1     SC_FCI, SC_ECC, SC_RCI, SC_FUL, SC_EC1),
: 4991 1     TBL_WPT : vector [5] initial (NULL, SC_SWP, SC_HWP),
: 4992 1     TBL_DFI : vector [16] initial (SC_FER, NULL, SC_IS2, SC_DS2, SC_EC9, NULL, NULL,
: 4993 1     SC_ECD, SC_EC1, SC_EC2, SC_EC3, SC_EC4, SC_EC5, SC_EC6, SC_EC7, SC_EC8),
: 4994 1     TBL_HST : vector [5] initial (NULL, SC_ODA, SC_ODB, SC_NXM, SC_PAB),
: 4995 1     TBL_CNT : vector [4] initial (SC_CIO, SC_SDS, SC_EDQ, SC_IDS),
: 4996 1     TBL_DRV : vector [9] initial (NULL, SC_SRT, SC_SRI, SC_POE, SC_RDY, SC_CLK, SC_RSP,
:         SC_SUR, SC_PSP);

```


C10

ZRQADO	RD/RX EXERCISER	11 Apr 1984 11:56:01	VAX 11 B1159 16 V4.0 5/9	SEQ 0119
V01.6	TYPE AND DESCRIPTION	11 Apr 1984 11:45:02	DISK\$USER2:[POWERS]ZRQADO.BL1:6	Page 119
:	4997 1			(36)
:	4998 1			
:	4999 1			
:	5000 1			
:	5001 1			
:	5002 1			

*sbt1 'TYPE AND DESCRIPTION'

EQUALS;

DEVTYP (*asciz'RD/RX EXERCISER');

! TEST DESCRIPTION

! NAME OF DEVICE SUPPORTED BY PROGRAM

D10

ZRQAMP
V01.6

RD/RX EXERCISER
HARDWARE PARAMETER CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAY-11 B1199 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0120
Page 120
(37)

```

: 5003 1 *sbttl 'HARDWARE PARAMETER CODING SECTION'
: 5004 1
: 5005 1 !
: 5006 1 ! THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
: 5007 1 ! THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: 5008 1 ! MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: 5009 1 ! INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: 5010 1 ! MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: 5011 1 ! WITH THE OPERATOR.
: 5012 1 !
: 5013 1
: 5014 1 BGNHRD;
: 5015 1
: 5016 1 GPRMA (HWQ1, 0, 0, %o'160000', %o'177777', YES, 1);
: 5017 1 GPRMA (HWQ2, 2, 0, %o'4', %o'774', YES, 1);
: 5018 1 GPRMD (HWQ3, 4, 0, %o'377', %o'0', %o'7', YES, 1);
: 5019 1 GPRMD (HWQ4, 6, 0, %o'17', %decimal'0', %decimal'15', YES, 1);
: 5020 1 GPRML (HWQ10, 6, %o'000040', YES, 1);
: 5021 1 XFERF (NODU);
: 5022 1 GPRML (HWQ11, 6, %o'000100', YES, 1);
: 5023 1 $L (NODU);
: 5024 1 GPRML (HWQ5, 6, %o'000200', YES, 1);
: 5025 1 XFERT (TOQ8);
: 5026 1 GPRMD (HWQ6A, 8, 0, %o'177777', %decimal'0', %o'177777', YES, 1);
: 5027 1 GPRMD (HWQ6B, 10, 0, %o'177777', %decimal'0', %o'177777', YES, 1);
: 5028 1 GPRMD (HWQ7A, 12, 0, %o'177777', GP$ATLO (8), %o'177777', YES, 1);
: 5029 1 GPRMD (HWQ7B, 14, 0, %o'177777', %decimal'0', %o'177777', YES, 1);
: 5030 1 $L (TOQ8);
: 5031 1 GPRML (HWQ8, 6, %o'100000', NO, 1);
: 5032 1 XFERF (HWDONE);
: 5033 1 GPRML (HWQ9, 6, %o'100000', NO, 1);
: 5034 1 $L (HWDONE);
: 5035 1
: 5036 1 ENDRD;

```

```

! IP ADDRESS
! VECTOR
! BR LEVEL
! RDRX DRIVE NUMBER
! ALSO RUN DUP EXERCISER
!
! WRITE DIAG AREA
!
! TEST ENTIRE CUSTOMER AREA?
! BR IF YES
! STARTING LBN LO
! STARTING LBN HI
! ENDING LBN LO
! ENDING LBN HI
!
! EXER ON CUST DATA AREA
! NO DONE
! ** WARNING / CONFIRM

```



```

: 5037 1 *sbttl 'SOFTWARE PARAMETER CODING SECTION'
: 5038 1
: 5039 1 !
: 5040 1 ! THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
: 5041 1 ! THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: 5042 1 ! MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: 5043 1 ! INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: 5044 1 ! MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: 5045 1 ! WITH THE OPERATOR.
: 5046 1 !
: 5047 1
: 5048 1 BGNSFT;
: 5049 1
: 5050 1 !GPRML (SWQ16, 4, SWF_TRC, YES, 1);
: 5051 1 GPRMD (SWQ24, 10, D, #o'177777', 0, 2359, YES, 1);
: 5052 1 GPRMD (SWQ1, 0, D, #o'177777', 0, 65535, YES, 1);
: 5053 1 GPRMD (SWQ2, 2, D, #o'177777', 0, 99, YES, 1);
: 5054 1 GPRMD (SWQ17, 8, D, #o'177777', 0, 100, YES, 1);
: 5055 1 GPRML (SWQ15, 4, SWF_CST, YES, 1);
: 5056 1 GPRML (SWQ20, 4, SWF_FER, YES, 1);
: 5057 1 GPRML (SWQ23, 4, SWF_BLK, YES, 1);
: 5058 1 GPRML (SWQ21, 4, SWF_HRD, YES, 1);
: 5059 1 GPRML (SWQ22, 4, SWF_SFT, YES, 1);
: 5060 1 GPRML (SWQ25, 4, SWF_TRY, YES, 1);
: 5061 1 GPRML (SWQ4, 4, SWF_RDM, YES, 1);
: 5062 1 XFERF (SW1);
: 5063 1 XFER (SW2);
: 5064 1 $L (SW1);
: 5065 1 GPRML (SWQ19, 4, SWF_SEQ, YES, 1);
: 5066 1 $L (SW2);
: 5067 1 GPRML (SWQ7, 4, SWF_CRC, YES, 1);
: 5068 1 DISPLAY (SWM1);
: 5069 1 GPRML (SWQ9, 4, SWF_CWC, YES, 1);
: 5070 1 XFERF (SW3);
: 5071 1 XFER (SW4);
: 5072 1 $L (SW3);
: 5073 1 GPRML (SWQ10, 4, SWF_HWC, YES, 1);
: 5074 1 $L (SW4);
: 5075 1 GPRML (SWQ11, 4, SWF_UDP, YES, 1);
: 5076 1 XFERF (SW5);
: 5077 1 XFER (SW6);
: 5078 1 $L (SW5);
: 5079 1 GPRMD (SWQ12, 6, D, #o'177777', 0, DP CNT, YES, 1);
: 5080 1 XFER (SW7);
: 5081 1 $L (SW6);
: 5082 1 GPRMD (SWQ13, 12, D, #o'177777', 1, MAX_UDP_CNT, YES, 1);
: 5083 1 GPRMD (SWQ14, 14, 0, #o'177777', 0, #o'177777', NO, 12);
: 5084 1 $L (SW7);
: 5085 1
: 5086 1 ENDSFT;

```

```

! ENABLE DIAGNOSTIC TRACE
! START TIME
! ERROR LIMIT
! TRANSFER LIMIT
! PERCENT OF RD OPERATIONS
! CLEAR STATISTICAL TABLES ?
! REWRITE BLOCKS WHEN "FORCED ERROR" BIT SET?
! HALT ON BAD-BLOCK TYPE ERRORS WITH 'HOE' FLAG?
! HALT ON HARD ERRORS WITH 'HOE' FLAG SET?
! HALT ON SOFT ERRORS WITH 'HOE' FLAG SET?
! COUNT EACH RETRY AS ANOTHER SOFT-ERROR?
! RANDOM SEEK MODE ?
! IF NO, DO NEXT QUESTION

!
! RANDOM OR SEQUENTIAL SELECTION OF DRIVES
!
! READ-COMPARES AT CONTROLLER ?
! REMAINING QUESTIONS ONLY APPLY ...
! WRITE-COMPARES AT CONTROLLER ?
! IF NO, DO NEXT QUESTION

!
! CHECK WRITES AT HOST BY READING ?
!
! USER-DEFINED DATA PATTERN ?
! IF NO, DO NEXT QUESTION

!
! SELECT PRE-DEFINED DATA PATTERN
! DONE
!
! NO. OF WORDS IN USER DATA PATTERN
! PATTERN VALUES

```

ZRQAM2
V01.6RD/RX EXERCISER
SOFTWARE PARAMETER CODING SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0122
Page 122
(39)

```

: 5087 1
: 5088 1
: 5089 1 *sbttl 'REPORT CODING SECTION'
: 5090 1
: 5091 1
: 5092 1
: 5093 1 !
: 5094 1 ! THE REPORT CODING SECTION CONTAINS THE
: 5095 1 ! "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
: 5096 1 !-
: 5097 1
: 5098 2 BGNRPT;
: 5099 2
: 5100 2 local
: 5101 2     CUR_PRIORITY : word;
: 5102 2
: 5103 2 GETPRI (CUR_PRIORITY);
: 5104 2 SETPRI (PRI04);
: 5105 2
: 5106 2 PRINTS (RPT1);
: 5107 2 PRINTS (RPT2);
: 5108 2 PRINTS (RPT3);
: 5109 2 PRINTS (RPT4);
: 5110 2 PRINTS (RPT5);
: 5111 2 PRINTS (RPT6);
: 5112 2
: 5113 2 incr CTLR from 0 to MAX_CTLR - 1 do
: 5114 2
: 5115 3     begin
: 5116 3     SET_CPAR (.CTLR);
: 5117 3
: 5118 3     incr DISK from (0 * OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE * OF_UN) by UNIT_SIZE do
: 5119 3
: 5120 4         begin
: 5121 4         SET_UPAR (.DISK);
: 5122 4
: 5123 4
: 5124 4         if .CST_ADDR [.DISK * OF_DATA, D_PRES] eq1 PRESENT
: 5125 4         then
: 5126 4
: 5127 5             begin
: 5128 5             PRINTS (RPT7,
: 5129 5             .L$LUN, .CST_ADDR [.DISK * OF_DATA, D_DISK_NUM], CST [.CTLR, .DISK * OF_NAME_0, D_NAME_0]);
: 5130 5             PRINTS (RPT8,
: 5131 5             .T_ADDR [TOT_READS_HI], .T_ADDR [TOT_READS_LO],
: 5132 5             .T_ADDR [MTOT_BYT_RED], .T_ADDR [TOT_BYT_RED_HI], .T_ADDR [TOT_BYT_RED_LO]);
: 5133 5             PRINTS (RPT8,
: 5134 5             .T_ADDR [TOT_WRITES_HI], .T_ADDR [TOT_WRITES_LO],
: 5135 5             .T_ADDR [MTOT_BYT_WRT], .T_ADDR [TOT_BYT_WRT_HI], .T_ADDR [TOT_BYT_WRT_LO]);
: 5136 5             PRINTS (RPT9,
: 5137 5             .T_ADDR [ERR_HRD_SEK], .T_ADDR [ERR_HRD_DAT], .T_ADDR [ERR_HRD_DRV], .T_ADDR [ERR_HRD_HST],
: 5138 5             .T_ADDR [ERR_SFT_SEK], .T_ADDR [ERR_SFT_DAT], .T_ADDR [ERR_SFT_DRV], .T_ADDR [ERR_SFT_HST]);
: 5139 4             end;

```

G10

ZRQAM2 RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 Bliss-16 V4.0-579 SEQ 0123
 V01.6 REPORT CODING SECTION 11-Apr-1984 11:45:02 DISK\$USER2:[POWERS]ZRQADO.BL1;6 Page 123
 (39)

```

; 5140 3      end;
; 5141 3
; 5142 3
; 5143 3      if .CST [.CTRL, STATE] eq1 PRESENT
; 5144 3      then
; 5145 3
; 5146 4      begin
; 5147 4      PRINTS (RPT10);
; 5148 4      PRINTS (RPT11, .C_ERR_TBL [.CTRL, C_ERR_HRD], .C_ERR_TBL [.CTRL, C_ERR_SFT]);
; 5149 3      end;
; 5150 3
; 5151 3      PRINTS (CRLF);
; 5152 3
; 5153 2      end;
; 5154 2
; 5155 2      SETPRI (.CUR_PRIORITY);
; 5156 2
; 5157 1      ENDRPT;

```

.TITLE ZRQAM2 RD/RX EXERCISER
 .IDENT /V01.6/
 .ENABL AMA

000000				.PSECT	\$CODE\$, RO
000000	122	121	104	L\$DVTYP::	
				.ASCII	/RQD/
000003	130	061	040	.ASCII	/X1 /
000006	157	162	040	.ASCII	/or /
000011	122	125	130	.ASCII	/RU)/
000014	065	060	000	.ASCII	/50/<00>
000017	000			.ASCII	<00>
000020	122	104	057	L\$DESC::	.ASCII /RD/<57>
000023	122	130	040	.ASCII	/RX /
000026	105	130	105	.ASCII	/EXE/
000031	122	103	111	.ASCII	/RCI/
000034	123	105	122	.ASCII	/SER/
000037	000			.ASCII	<00>
000040				.BLKB	2
000042	000000C			L\$HRDLN::	
				.WORD	<<<L\$NDHRD-L\$HRDLN>/2>-1>
000044	000031			GP\$1::	.WORD 31
000046	000000G			.WORD	HWQ1
000050	160000			.WORD	-20000
000052	177777			.WORD	1
000054	001031			GP\$2::	.WORD 1031
000056	000000G			.WORD	HWQ2
000060	000004			.WORD	4
000062	000774			.WORD	774
000064	002032			GP\$3::	.WORD 2032
000066	000000G			.WORD	HWQ3
000070	000377			.WORD	377

H10

ZRQAM?
V01.6RD/RX EXERCISER
REPORT CODING SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0124
Page 124
(39)

000072	000000		.WORD	0
000074	000007		.WORD	7
000076	003052	GP\$4::	.WORD	3052
000100	000000G		.WORD	HWQ4
000102	000017		.WORD	17
000104	000000		.WORD	0
000106	000017		.WORD	17
000110	003130	GP\$5::	.WORD	3130
000112	000000G		.WORD	HWQ10
000114	000040		.WORD	40
000116	000000C	\$NODU:	.WORD	<<<<\$LNODU-\$NODU>*400>+4>+40>
000120	003130	GP\$6::	.WORD	3130
000122	000000G		.WORD	HWQ11
000124	000100		.WORD	100
000126	001004	\$LNODU:	.WORD	1004
000130	003130	GP\$7::	.WORD	3130
000132	000000G		.WORD	HWQ5
000134	000200		.WORD	200
000136	000000C	\$TOQ8:	.WORD	<<<<\$LTOQ8-\$TOQ8>*400>+4>+20>
000140	004032	GP\$8::	.WORD	4032
000142	000000G		.WORD	HWQ6A
000144	177777		.WORD	-1
000146	000000		.WORD	0
000150	177777		.WORD	-1
000152	005032	GP\$9::	.WORD	5032
000154	000000G		.WORD	HWQ6B
000156	177777		.WORD	-1
000160	000000		.WORD	0
000162	177777		.WORD	-1
000164	006432	GP\$10::	.WORD	6432
000166	000000G		.WORD	HWQ7A
000170	177777		.WORD	-1
000172	000004		.WORD	4
000174	177777		.WORD	-1
000176	000001		.WORD	1
000200	007032	GP\$11::	.WORD	7032
000202	000000G		.WORD	HWQ7B
000204	177777		.WORD	-1
000206	000000		.WORD	0
000210	177777		.WORD	-1
000212	001004	\$LTOQ8:	.WORD	1004
000214	003120	GP\$12::	.WORD	3120
000216	000000G		.WORD	HWQ8
000220	100000		.WORD	100000
000222	000000C	\$HWDONE:	.WORD	<<<<\$LHWDONE-\$HWDONE>*400>+4>+40>
000224	003120	GP\$13::	.WORD	3120
000226	000000G		.WORD	HWQ9
000230	100000		.WORD	-100000
000232	001004	\$LHWDONE:		
			.WORD	1004
000234		L\$NDHRD::		
			.BLKW	1
000236	000000C	L\$SFTLN::		

ZRQADO
V01.6RD-RX EXERCISER
REPORT CODING SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6SEQ 0125
Page 125
(39)

000240	005052	GP\$14::	.WORD	<<<L\$NDSFT-L\$SFTLN>/2>-1>
000242	000000G		.WORD	5052
000244	177777		.WORD	SWQ24
000246	000000		.WORD	-1
000250	004467		.WORD	0
000252	000052	GP\$15::	.WORD	4467
000254	000000G		.WORD	52
000256	177777		.WORD	SWQ1
000260	000000		.WORD	-1
000262	177777		.WORD	0
000264	001052	GP\$16::	.WORD	-1
000266	000000G		.WORD	1052
000270	177777		.WORD	SWQ2
000272	000000		.WORD	-1
000274	000143		.WORD	0
000276	004052	GP\$17::	.WORD	143
000300	000000G		.WORD	4052
000302	177777		.WORD	SWQ17
000304	000000		.WORD	-1
000306	000144		.WORD	0
000310	002130	GP\$18::	.WORD	144
000312	000000G		.WORD	2130
000314	000200		.WORD	SWQ15
000316	002130	GP\$19::	.WORD	200
000320	000000G		.WORD	2130
000322	004000		.WORD	SWQ20
000324	002130	GP\$20::	.WORD	4000
000326	000000G		.WORD	2130
000330	040000		.WORD	SWQ23
000332	002130	GP\$21::	.WORD	40000
000334	000000G		.WORD	2130
000336	010000		.WORD	SWQ21
000340	002130	GP\$22::	.WORD	10000
000342	000000G		.WORD	2130
000344	020000		.WORD	SWQ22
000346	002130	GP\$23::	.WORD	20000
000350	000000G		.WORD	2130
000352	100000		.WORD	SWQ25
000354	002130	GP\$24::	.WORD	100000
000356	000000G		.WORD	2130
000360	000002		.WORD	SWQ4
000362	000000C		.WORD	2
000364	000000C	\$SW1:	.WORD	<<<<\$LSW1-\$SW1>*400>.4>.40>
000366	001004	\$SW2:	.WORD	<<<\$LSW2-\$SW2>*400>.4>
000370	002130	\$LSW1:	.WORD	1004
000372	000000G	GP\$25::	.WORD	2130
000374	001000		.WORD	SWQ19
000376	001004		.WORD	1000
000400	002130	\$LSW2:	.WORD	1004
000402	000000G	GP\$26::	.WORD	2130
000404	000004		.WORD	SWQ7
000406	000003		.WORD	4
		GP\$DISP::		

J10

ZRQAM2
V01.6

RD RX EXERCISER
REPORT CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0126
Page 126
(39)

000410	000000G		.WORD	3
000412	002130		.WORD	SWM1
000414	000000G	GP\$27::	.WORD	2130
000416	000020		.WORD	SWQ9
000420	000000C		.WORD	20
000422	000000C	\$SW3:	.WORD	<<<<\$LSW3-\$SW3>*400>+4>+40>
000424	001004	\$SW4:	.WORD	<<<\$LSW4-\$SW4>*400>+4>
000426	002130	\$LSW3:	.WORD	1004
000430	000000G	GP\$28::	.WORD	2130
000432	000040		.WORD	SWQ10
000434	001004		.WORD	40
000436	002130	\$LSW4:	.WORD	1004
000440	000000G	GP\$29::	.WORD	2130
000442	000100		.WORD	SWQ11
000444	000000C		.WORD	100
000446	000000C	\$SW5:	.WORD	<<<<\$LSW5-\$SW5>*400>+4>+40>
000450	001004	\$SW6:	.WORD	<<<\$LSW6-\$SW6>*400>+4>
000452	003052	\$LSW5:	.WORD	1004
000454	000000G	GP\$30::	.WORD	3052
000456	177777		.WORD	SWQ12
000460	000000		.WORD	-1
000462	000025		.WORD	0
000464	000000C		.WORD	25
000466	001004	\$SW7:	.WORD	<<<\$LSW7-\$SW7>*400>+4>
000470	006052	\$LSW6:	.WORD	1004
000472	000000G	GP\$31::	.WORD	6052
000474	177777		.WORD	SWQ13
000476	000001		.WORD	-1
000500	000020		.WORD	1
000502	007222		.WORD	20
000504	000000G	GP\$32::	.WORD	7222
000506	177777		.WORD	SWQ14
000510	000000		.WORD	-1
000512	177777		.WORD	0
000514	000006		.WORD	-1
000516	001004	\$LSW7:	.WORD	6
000520		L\$NDSFT:	.WORD	1004
			.BLKW	1

000000			.PSECT	\$OWN\$, 0
000000	000000G	TBL,SUC:	.WORD	NULL
000002	000000G		.WORD	SC,SDI
000004	000000G		.WORD	SC,CON
000006	000000G		.WORD	NULL
000010	000000G		.WORD	SC,DUP
000012	000000G		.WORD	NULL
000014	000000G		.WORD	NULL
000016	000000G		.WORD	NULL
000020	000000G		.WORD	SC,ONL
000022	000000G		.WORD	NULL

K10

ZRQAM2
V01.6

RD/RX EXERCISER
REPORT CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0127
Page 127
(39)

000024	000000G		.WORD	NULL
000026	000000G		.WORD	NULL
000030	000000G		.WORD	NULL
000032	000000G		.WORD	NULL
000034	000000G		.WORD	NULL
000036	000000G		.WORD	NULL
000040	000000G		.WORD	SC.SON
000042	000000G	TBL.GFL:	.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.EC0
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.CTC
000174	000000G		.WORD	SC.SDS

L10

ZRQAM2
V01.6

RD/RX EXERCISER
REPORT CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0128
Page 128
(39)

000176 000000G
000200 000000G
000202 000000G
000204 000000G
000206 000000G
000210 000000G
000212 000000G
000214 000000G
000216 000000G
000220 000000G
000222 000000G

TBL.DRV: .WORD SC.EDC
.WORD SC.IDS
.WORD NULL
.WORD SC.SRT
.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, BST, TALLY, T.ADDR
.GLOBL DUPPKT, TRK.SGN, RDM.CNT, RANDOM
.GLOBL C.ERR.TBL, MSCP.PKT, IPKT.ADDR
.GLOBL PKT.USE, RETPKT, RP.USE, RP.INDX
.GLOBL RP.ADDR, ELOG.PKT, BUFF.ADDR, BUFF.OWN
.GLOBL IODQ, IODQ.IN, IODQ.OUT, ENTRY.REASON
.GLOBL EOP.FLAG, DUP.FLAGS, CCTL, CDISK
.GLOBL CUOFF, CCTL.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, CREDIT.BAL
.GLOBL NEXT.PKT.USE, HOURS, MINUTES, CLK.TICKS
.GLOBL CLK.PRESENT, HOE.FLAG, FORCED.ERROR
.GLOBL FER.LBN, FER.BC, INIT.OCCURED
.GLOBL ADDR.VECT.OK, DBM5, P.INDEX, S.PATTERN
.GLOBL S.DUPPKT, DBM107, DU.MSG, DU.RSN
.GLOBL RPT1, RPT2, RPT3, RPT4, RPT5, RPT6
.GLOBL RPT7, RPT8, RPT9, RPT10, RPT11
.GLOBL MSG.01, EGS.01, EBS.01, EBD.10
.GLOBL EBD.12, EBD.13, EBD.14, EBD.18
.GLOBL EBD.19, EBD.24, ERR.00, ERR.COD
.GLOBL ELG.00, ELG.FMT, EX.TIM, XX13
.GLOBL XX23, XX32, XX33, XX34, EX.SA
.GLOBL EX.SC, EX.SB0, EX.SB, EX.RP, EX.WRD
.GLOBL EX.CMD, EX.RD, EX.WRT, EX.CMP
.GLOBL EX.ONL, EX.ACC, EX.OP, EX.BB, EX.BB1
.GLOBL EX.BBU, EX.LBN, EX.PBN, EX.LBR
.GLOBL EX.LBW, EX.RBN, EX.CBC, EX.CBR
.GLOBL EX.CBW, EX.BC, EX.BD, EX.BDR, EX.BDW
.GLOBL SC.SDI, SC.CON, SC.DUP, SC.ONL
.GLOBL SC.SON, SC.UNK, SC.VOL, SC.IOP
.GLOBL SC.DIS, SC.FER, SC.FE2, SC.ISH
.GLOBL SC.IS2, SC.DST, SC.DS2, SC.ECC
.GLOBL SC.ECD, SC.RCT, SC.FUL, SC.S76
.GLOBL SC.FCT, SC.SWP, SC.HWP, SC.EC1
.GLOBL SC.EC2, SC.EC3, SC.EC4, SC.EC5
.GLOBL SC.EC6, SC.EC7, SC.EC8, SC.EC9
.GLOBL SC.ODA, SC.ODB, SC.NXM, SC.PAR
.GLOBL SC.CTO, SC.SDS, SC.EDC, SC.IDS

M10

ZRQAM2
V01.6

RD/RX EXERCISER
REPORT CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0129
Page 129
(39)

```
.GLOBL SC.SRT, SC.SRI, SC.POE, SC.RDY
.GLOBL SC.CLK, SC.RSP, SC.SUR, SC.PSP
.GLOBL CER.01, CER.02, CNTR.ERR, RDRX.ERR
.GLOBL SPACE4, CRLF, DASH, ASTERISK, HWQ1
.GLOBL HWQ2, HWQ3, HWQ4, HWQ5, HWQ6A
.GLOBL HWQ6B, HWQ7A, HWQ7B, HWQ8, HWQ9
.GLOBL HWQ10, HWQ11, SWQ1, SWQ2, SWQ4
.GLOBL SWQ7, SWQ9, SWQ10, SWQ11, SWQ12
.GLOBL SWQ13, SWQ14, SWQ15, SWQ17, SWQ19
.GLOBL SWQ20, SWQ21, SWQ22, SWQ23, SWQ24
.GLOBL SWQ25, EH.0, EH.1, EH.2, EH.3
.GLOBL EH.4, EH.5, EH.6, EH.7, EH.8, EH.9
.GLOBL EH.10, EH.12, EH.13, SWM1, NULL
.GLOBL SWP.FLAGS, L$HMEM, L$LUN, L$UNIT
```

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

N10

ZRQAM2
V01.6

RD/RX EXERCISER
REPORT CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0130
Page 130
(39)

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	PNI==	1000
002000	PRI==	2000
004000	IXE==	4000
010000	IBE==	10000
020000	IER==	20000
040000	LOE==	40000
100000	HOE==	-100000
000044'	L\$HARD==	L\$HRDLN+2
000240'	L\$SOFT==	L\$SFTLN+2

000522

.SBTTL LRPT REPORT CODING SECTION
.PSECT \$CODE\$, R0

000000	004137	000000G	LRPT:	JSR	R1,\$SAVE4	:	5086
000004	104440			TRAP	40	:	5103
000006	010004			MOV	R0,R4	: *.CUR.PRIORITY	
000010	012700	000200		MOV	\$200,R0	:	5104
000014	104441			TRAP	41	:	
000016	012746	000000G		MOV	\$RPT1, (SP)	:	5106
000022	012746	000001		MOV	\$1, (SP)	:	
000026	010600			MOV	SP,R0	: SP, *	
000030	104416			TRAP	16	:	
000032	012716	000000G		MOV	\$RPT2, (SP)	:	5107
000036	012746	000001		MOV	\$1, (SP)	:	
000042	010600			MOV	SP,R0	: SP, *	
000044	104416			TRAP	16	:	
000046	012716	000000G		MOV	\$RPT3, (SP)	:	5108
000052	012746	000001		MOV	\$1, (SP)	:	
000056	010600			MOV	SP,R0	: SP, *	
000060	104416			TRAP	16	:	
000062	012716	000000G		MOV	\$RPT4, (SP)	:	5109
000066	012746	000001		MOV	\$1, (SP)	:	
000072	010600			MOV	SP,R0	: SP, *	
000074	104416			TRAP	16	:	
000076	012716	000000G		MOV	\$RPT5, (SP)	:	5110
000102	012746	000001		MOV	\$1, (SP)	:	
000106	010600			MOV	SP,R0	: SP, *	
000110	104416			TRAP	16	:	
000112	012716	000000G		MOV	\$RPT6, (SP)	:	5111
000116	012746	000001		MOV	\$1, (SP)	:	
000122	010600			MOV	SP,R0	: SP, *	
000124	104416			TRAP	16	:	
000126	005003			CLR	R3	: CLR	5113

[]

ZRQAM, V01.6	RD-RX EXERCISER REPORT CODING SECTION		11 Apr 1984 11:56:01 11 Apr 1984 11:45:02	VAX 11 B11aa 16 V4.0 5/79 DISK#USER2:[POWERS]ZRQADO.B1 1:6	SEQ 0131 Page 151 (39)
000130	010316		13: MOV R3,(SP)	1 CTR,*	5116
000132	004737	000000V	JSR PC,SET,CPAR		
000136	012702	000003	MOV #3,R2	1 *,DISK	5118
000142	010316		23: MOV R2,(SP)	1 DISK,*	5121
000144	004737	000000V	JSR PC,SET,UPAR		
000150	010201		MOV R2,R1	1 DISK,*	5124
000152	006301		ASL R1		
000154	063701	000000G	ADD CST,ADDR,R1		
000160	032711	040000	BIT #40000,(R1)		
000164	001535		BFO 31		
000166	010316		MOV R3,(SP)	1 CTR,*	5129
000170	012746	000053	MOV #3,(SP)		
000174	004737	000000G	JSR PC,BL,MUX		
000200	060200		ADD R2,RO	1 DISK,*	
000202	006300		ASL RO		
000204	062700	000000G	ADD #CST,RO		
000210	010016		MOV RO,(SP)		
000212	062716	000012	ADD #12,(SP)		
000216	111146		MOVH (R1),(SP)		
000220	042716	177760	BIC #177760,(SP)		
000224	013746	000000G	MOV LILUN,(SP)		
000230	012746	000000G	MOV #RPT7,(SP)		
000234	012746	000004	MOV #4,(SP)		
000240	010600		MOV SP,RO	1 SP,*	
000242	104416		TRAP 16		
000244	013700	000000G	MOV T,ADDR,RO		
000250	016016	000032	MOV 32(RO),(SP)		5132
000254	016046	000034	MOV 34(RO),(SP)		
000260	016046	000036	MOV 36(RO),(SP)		
000264	016046	000016	MOV 16(RO),(SP)		
000270	016046	000020	MOV 20(RO),(SP)		
000274	012746	000000G	MOV #RPT8,(SP)		
000300	012746	000006	MOV #6,(SP)		
000304	010600		MOV SP,RO	1 SP,*	
000306	104416		TRAP 16		
000310	013700	000000G	MOV T,ADDR,RO		
000314	016016	000040	MOV 40(RO),(SP)		5135
000320	016046	000042	MOV 42(RO),(SP)		
000324	016046	000044	MOV 44(RO),(SP)		
000330	016046	000024	MOV 24(RO),(SP)		
000334	016046	000026	MOV 26(RO),(SP)		
000340	012746	000000G	MOV #RPT8,(SP)		
000344	012746	000006	MOV #6,(SP)		
000350	010600		MOV SP,RO	1 SP,*	
000352	104416		TRAP 16		
000354	013700	000000G	MOV T,ADDR,RO		
000360	005016		CLR (SP)		5138
000362	116016	000055	MOVH 55(RO),(SP)		
000366	005046		CLR (SP)		
000370	116016	000054	MOVH 54(RO),(SP)		
000374	005046		CLR (SP)		
000376	116016	000053	MOVH 53(RO),(SP)		
000402	005046		CLR (SP)		

C11

ZRGADM
VOL.6RD RX EXERCISER
REPORT CODING SECTION11 Apr 1984 11:56:01
11 Apr 1984 11:45:02VAX 11 B1100 16 V4.0 5/79
DISK\$USER2:[POWER5]ZRGADO.BL1;6SEQ 0132
Page 132
(39)

000404	116016	000052		MOVB	52(R0),(SP)		
000410	005046			CLR	(SP)		
000412	116016	000051		MOVB	51(R0),(SP)		
000416	005046			CLR	(SP)		
000420	116016	000050		MOVB	50(R0),(SP)		
000424	005046			CLR	(SP)		
000426	116016	000047		MOVB	47(R0),(SP)		
000432	005046			CLR	(SP)		
000434	116016	000046		MOVB	46(R0),(SP)		
000440	012746	000000G		MOV	0RPT9,(SP)		
000444	012746	000011		MOV	011,(SP)		
000450	010600			MOV	SP,R0		: SP,*
000452	104416			TRAP	16		
000454	062706	000064		ADD	064,SP		
000460	062702	000017	3:	ADD	012,R2		: *,DISK
000464	020227	000041		CMP	R2,041		: DISK,*
000470	003674			BUE	28		
000472	010316			MOV	R3,(SP)		: CLR,*
000474	012746	000126		MOV	0126,(SP)		
000500	004757	000000G		JGR	PC,BL\$MUL		
000504	005726			TST	(SP)		
000506	005760	000002G		TST	CST 2(R0)		
000512	100026			BPL	48		
000514	012716	000000G		MOV	0RPT10,(SP)		
000520	012746	000001		MOV	01,(SP)		: SP,*
000524	010600			MOV	SP,R0		: CLR,*
000526	104416			TRAP	16		
000530	010300			MOV	R3,R0		
000532	006300			ASL	R0		
000534	005016			CLR	(SP)		
000536	116016	000001G		MOVB	C.ERR,TBL+1(R0),(SP)		
000542	005046			CLR	(SP)		
000544	116016	000000G		MOVB	C.ERR,TBL(R0),(SP)		
000550	012746	000000G		MOV	0RPT11,(SP)		
000554	012746	000003		MOV	03,(SP)		
000560	010600			MOV	SP,R0		: SP,*
000562	104416			TRAP	16		
000564	062706	000010		ADD	010,SP		
000570	012716	000000G	4:	MOV	0CRIF,(SP)		
000574	012746	000001		MOV	01,(SP)		
000600	010600			MOV	SP,R0		: SP,*
000602	104416			TRAP	16		
000604	005726			TST	(SP)		
000606	005203			INC	R4		: CLR
000610	000243			WORD	CLVCLC		
000612	003002			BGT	53		
000614	000137	000652	5:	JMP	13		
000620	010300			MOV	R4,R0		: CUR,PRIORITY,*
000622	104441			TRAP	41		
000624	062706	000016		ADD	016,SP		
000630	000207			RTS	PC		

011

ZRQAM2 RD/RX EXERCISER
V01.6 REPORT CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11sa 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0133
Page 133
(39)

; Routine Size: 205 words, Routine Base: \$CODE\$ + 0522
; Maximum stack depth per invocation: 40 words

000000	004737	000522	,SBITL	L\$RPT REPORT CODING SECTION	
000004	104425		L\$RPT:: JSR	PC,L\$RPT	
000006	000207		TRAP	25	
			RTS	PC	

5155

; Routine Size: 4 words, Routine Base: \$CODE\$ + 1354
; Maximum stack depth per invocation: 2 words

; 5158 1

E11

ZRQAM,
V01.6

RD/RX EXERCISER
INITIALIZE SECTION

11 Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0134
Page 134
(40)

```

: 5159 1 *sbttl 'INITIALIZE SECTION'
: 5160 1
: 5161 2 BGNINIT;
: 5162 2
: 5163 2 local
: 5164 2     DELAY_MULT : word,
: 5165 2     FLAG : byte,
: 5166 2     TEMP : word,
: 5167 2     HWPT_REF : ref block [HWPT_LEN, word] field (HWP_FIELDS),
: 5168 2     CLEAR_TABLES : byte,
: 5169 2     SMALLEST_DRIVE : byte,
: 5170 2     HWPT_ADDRESS : vector [MAX_UNITS, word];
: 5171 2
: 5172 2 SETPRI (PRI07);
: 5173 2
: 5174 2 if READEP (EF_NEW)
: 5175 2 then
: 5176 3     begin
: 5177 3         ENTRY_REASON = NEW_PASS;
: 5178 3
: 5179 4         if not BIT_TST (SWP_FLAGS, SWF_CST)
: 5180 3         then
: 5181 3             CLEAR_TABLES = FALSE;
: 5182 3         else
: 5183 3             CLEAR_TABLES = TRUE;
: 5184 3
: 5185 2         end;
: 5186 2
: 5187 2 if READEP (EF_START)
: 5188 2 then
: 5189 3     begin
: 5190 3         BRESET;
: 5191 3         ENTRY_REASON = START;
: 5192 3         CLEAR_TABLES = TRUE;
: 5193 3         ADDR_VECT_OK = FALSE;
: 5194 3         INIT_OCCURED = FALSE;
: 5195 2     end;
: 5196 2
: 5197 2 if READEP (EF_RESTART)
: 5198 2 then
: 5199 3     begin
: 5200 3         ENTRY_REASON = RESTART;
: 5201 3         CLEAR_TABLES = TRUE;
: 5202 2     end;
: 5203 2
: 5204 2 if READEP (EF_CONTINUE)
: 5205 2 then
: 5206 3     begin
: 5207 3         ENTRY_REASON = CONT;
: 5208 3
: 5209 4         if not BIT_TST (SWP_FLAGS, SWF_CST)
: 5210 3         then
: 5211 3             CLEAR_TABLES = FALSE

```

! NO INTERRUPTS ALLOWED DURING INIT

! IS THIS A NEW PASS?

! IS THIS A START?

! IS THIS A RESTART?

! IS THIS A CONTINUE?

F11

ZRQAM2
V01.6

RD/RX EXERCISER
INITIALIZE SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0135
Page 135
(40)

```

: 5212 3      else
: 5213 3      CLEAR_TABLES = TRUE;
: 5214 3
: 5215 2      end;
: 5216 2
: 5217 2      if READEP (EF_PWR)
: 5218 2      then
: 5219 3          begin
: 5220 3              ENTRY_REASON = PWR_FAIL;
: 5221 3              ADDR_VECT_OK = FALSE;
: 5222 3              INIT_OCCURED = FALSE;
: 5223 3              CLEAR_TABLES = TRUE;
: 5224 3              PRINTF (MSG_01);
: 5225 3
: 5226 3              incr COUNT from 0 to 60 do
: 5227 4                  begin
: 5228 4                      DELAY_MULT = 333;
: 5229 4                      DELAY (.DELAY_MULT);
: 5230 4                      BREAK;
: 5231 3                  end;
: 5232 3
: 5233 2          end;
: 5234 2      !SETVEC (O_TVEC, O_BRK, PRI07);
: 5235 2
: 5236 2      !
: 5237 2      !
: 5238 2      !     MAKE SURE THAT NOT MORE THAN MAX_UNITS HAVE BEEN SPECIFIED.
: 5239 2      !     IF THERE ARE TOO MANY, NOTIFY USER AND RETURN TO SUPERVISOR.
: 5240 2      !     (DIAGNOSTIC IS ABORTED).
: 5241 2      !
: 5242 2
: 5243 2      if .L$UNIT gtru MAX_UNITS
: 5244 2      then
: 5245 3          begin
: 5246 3              ERRSF (1, EGS_01, EMS_01);
: 5247 3              DOCLN;
: 5248 2          end;
: 5249 2
: 5250 2      !
: 5251 2      !     THE FOLLOWING CODE IS EXECUTED FOR ALL ENTRY REASONS EXCEPT NEW_PASS.
: 5252 2      !     ALL RUN-TIME CONTROLLER STATUS TABLES (CSTs) ARE CLEARED TO 0. THEN
: 5253 2      !     LOADED WITH CONFIGURATION DATA FROM THE HARDWARE P-TABLES.
: 5254 2      !
: 5255 2
: 5256 2      if .ENTRY_REASON neq NEW_PASS
: 5257 2      then
: 5258 3          begin
: 5259 3              SMALLEST_DRIVE = 255;
: 5260 3
: 5261 3              incr COUNT from 0 to ((MAX_CTLR * CST_LEN * 2) - 2) by 2 do
: 5262 3                  (CST * .COUNT) = 0;
: 5263 3
: 5264 3              incr UNIT from 0 to (.L$UNIT - 1) do

```

! ARE WE HERE BECAUSE OF POWER FAIL

! "POWER DELAY - WAITING"

! WAIT APPROX. 60 SECONDS

! BREAK FOR ACT

! SET JDT TRAP VECTOR

! LARGEST DISK NO. ALLOWED BY MSCP

! LOOP THROUGH ALL UNITS

611

```

ZRQAM2      RD/RX EXERCISER      11-Apr-1984 11:56:01      VAX-11 Bliss-16 V4.0-579      SEQ 0136
VOL.6       INITIALIZE SECTION  11-Apr-1984 11:45:02      DISK$USER2:[POWERS]ZRQADO.BL1:6      Page 136
                                                (40)

```

<pre> : 5265 3 : 5266 3 : 5267 3 : 5268 3 : 5269 3 : 5270 3 : 5271 3 : 5272 3 : 5273 3 : 5274 3 : 5275 3 : 5276 3 : 5277 4 : 5278 4 : 5279 4 : 5280 4 : 5281 4 : 5282 4 : 5283 4 : 5284 4 : 5285 4 : 5286 4 : 5287 4 : 5288 4 : 5289 5 : 5290 5 : 5291 5 : 5292 5 : 5293 5 : 5294 5 : 5295 5 : 5296 5 : 5297 5 : 5298 5 : 5299 5 : 5300 5 : 5301 5 : 5302 5 : 5303 5 : 5304 5 : 5305 5 : 5306 5 : 5307 5 : 5308 5 : 5309 5 : 5310 5 : 5311 5 : 5312 5 : 5313 5 : 5314 5 : 5315 5 : 5316 4 : 5317 5 </pre>	<pre> if (HWPT_ADDRESS [.UNIT] * GPBARD (.UNIT, HWPT_REF)) neqa 0 then if .HWPT_REF [HWP_DISK_NUM] lssu .SMALLEST_DRIVE then SMALLEST_DRIVE = .HWPT_REF [HWP_DISK_NUM]; incr UNIT from 0 to (.L\$UNIT - 1) do if .HWPT_ADDRESS [.UNIT] neqa 0 then begin FLAG = NOT_FOUND; HWPT_REF = .HWPT_ADDRESS [.UNIT]; incr CTLR from 0 to (MAX_CTLR - 1) do if .CST [.CTLR, IP_ADDR] eqa .HWPT_REF [HWP_IP_ADDR] then if .CST [.CTLR, (.HWPT_REF [HWP_DISK_NUM] * .SMALLEST_DRIVE) * UNIT_SIZE + OF_UN + OF_DATA, D_PRES] eq1 NOT_PRESENT then begin TEMP = (.HWPT_REF [HWP_DISK_NUM] * .SMALLEST_DRIVE) * UNIT_SIZE + OF_UN; CST [.CTLR, .TEMP + OF_DATA, D_ALL] = .HWPT_REF [HWP_DISK]; CST [.CTLR, .TEMP + OF_DATA, D_UNIT] = .UNIT; CST [.CTLR, .TEMP + OF_DATA, D_FATAL] = FALSE; CST [.CTLR, .TEMP + OF_DATA, D_PRES] = PRESENT; CST [.CTLR, .TEMP + OF_BEG, D_BEG0] = .HWPT_REF [HWP_BEG_TRK]; CST [.CTLR, .TEMP + OF_BEG1, D_BEG1] = .HWPT_REF [HWP_BEG_TRK1]; CST [.CTLR, .TEMP + OF_END, D_END0] = .HWPT_REF [HWP_END_TRK]; CST [.CTLR, .TEMP + OF_END1, D_END1] = .HWPT_REF [HWP_END_TRK1]; CST [.CTLR, .TEMP + OF_DUPFLAGS, D_DBN] = 0; CST [.CTLR, .TEMP + OF_DUPFLAGS, D_DUPMEDIA] = NOT (.HWPT_REF [HWP_DISK_DUPX]); CST [.CTLR, .TEMP + OF_DUPFLAGS, D_DUPWRITE] = (.HWPT_REF [HWP_DISK_DUPWT]); CST [.CTLR, .TEMP + OF_COUNT, D_COUNT] = 0; FLAG = FOUND; exitloop; end else begin </pre>	<pre> ! IF HWP TABLE FOUND ! FIND OUT THE SMALLEST DISK NUMBER ! LOOP THROUGH ALL UNITS ! IF HWP TABLE FOUND ! LOOP THROUGH ALL CSTs ! IF EMPTY SLOT FOUND ! COPY DISK ADDR AND PROT BIT ! DUPLICATE UNIT </pre>

H11

ZRQAM,
VOL.6

RD/RX EXERCISER
INITIALIZE SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0137
Page 137
(40)

```

5318 5          PRINTF (CER_01, .HWPT_REF [HWP_DISK_NUM], .HWPT_REF [HWP_IP_ADDR]);
5319 5
5320 5          DUR [.UNIT] = DU_CONF;          ! "DUPLICATE UNIT; X AT IP: XXXXXX"
5321 5          DODU (.UNIT);                    ! CONFIGURATION ERROR
5322 5          FLAG = FOUND;                    ! DROP UNIT
5323 5          exitloop;
5324 4          end;
5325 4
5326 4      if .FLAG eql NOT_FOUND                ! IF NO IP MATCH TO EXISTING CST
5327 4      then
5328 5          begin
5329 5
5330 5          incr CTLR from 0 to (MAX_CTLR - 1) do      ! LOOP THROUGH EACH CST
5331 5
5332 5          if .CST [.CTLR, IP_ADDR] eql 0            ! IF EMPTY CST FOUND
5333 5          then
5334 6              begin
5335 6                  CST [.CTLR, IP_ADDR] = .HWPT_REF [HWP_IP_ADDR];
5336 6                  CST [.CTLR, VEC_ADDR] = .HWPT_REF [HWP_VECTOR];
5337 6                  CST [.CTLR, BR_LEV] = .HWPT_REF [HWP_BR_LEVEL];
5338 6                  TEMP = (.HWPT_REF [HWP_DISK_NUM] - .SMALLEST_DRIVE) * UNIT_SIZE + OF_UN;
5339 6                  CST [.CTLR, .TEMP + OF_DATA, D_ALL] = .HWPT_REF [HWP_DISK];
5340 6                  ! COPY DISK ADDR AND PROT BIT
5341 6                  CST [.CTLR, .TEMP + OF_DATA, D_UNIT] = .UNIT;
5342 6                  CST [.CTLR, .TEMP + OF_DATA, D_FATAL] = FALSE;
5343 6                  CST [.CTLR, .TEMP + OF_DATA, D_PRES] = PRESENT;
5344 6
5345 6                  IF .HWPT_REF [HWP_ENTIRE] EQL TRUE      !??? IF DEFAULT TEST RANGE,
5346 6                  THEN HWPT_REF [HWP_END_TRK] = ALL_ONES; !??? MAKE HI ADDR ALL ONES
5347 6
5348 6
5349 6                  CST [.CTLR, .TEMP + OF_BEG, D_BEG0] =      !???
5350 6                      .HWPT_REF [HWP_BEG_TRK];              !???
5351 6                  CST [.CTLR, .TEMP + OF_BEG1, D_BEG1] =      !???
5352 6                      .HWPT_REF [HWP_BEG_TRK1];              !???
5353 6                  CST [.CTLR, .TEMP + OF_END, D_END0] =      !???
5354 6                      .HWPT_REF [HWP_END_TRK];              !???
5355 6                  CST [.CTLR, .TEMP + OF_END1, D_END1] =      !???
5356 6                      .HWPT_REF [HWP_END_TRK1];              !???
5357 6
5358 6                  CST [.CTLR, .TEMP + OF_DUPFLAGS, D_DBN] = 0; !???
5359 6                  CST [.CTLR, .TEMP + OF_DUPFLAGS, NODUPMEDIA] = !???
5360 6                      NOT (.HWPT_REF [HWP_DISK_DUPX]);        !???
5361 6                  CST [.CTLR, .TEMP + OF_DUPFLAGS, DUPWRITE] = !???
5362 6                      (.HWPT_REF [HWP_DISK_DUPWT]);            !???
5363 6                  CST [.CTLR, .TEMP + OF_COUNT, D_COUNT] = 0; !???
5364 6                  FLAG = FOUND;
5365 6                  exitloop;
5366 5                  end;
5367 5
5368 5      if .FLAG eql NOT_FOUND                ! IF NO EMPTY CST FOUND
5369 5      then
5370 6          begin

```

ZRRQM,
VOL.6RD/RX EXERCISER
INITIALIZE SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRRQADO.BL1;6SEQ 0138
Page 138
(40)

```

: 5371 6          PRINTF (CER 02, MAX_CTLR);
: 5372 6          DUR (.UNIT) = DU_CONF;
: 5373 6          DODU (.UNIT);
: 5374 5          end;
: 5375 5
: 5376 4          end;
: 5377 4
: 5378 3          end;
: 5379 3
: 5380 3          ! CONFIGURATON CHECK FOR LEGAL RDRX UNIT MIX BECAUSE WE HAVE DIFFERENT
: 5381 3          ! DRIVES : THE RD51, RD52, AND RX50.
: 5382 3          ! (NEEDED?)
: 5383 3          !
: 5384 2          end;
: 5385 2
: 5386 2          if .ENTRY_REASON eq1 NEW_PASS
: 5387 2          then
: 5388 3              begin
: 5389 3
: 5390 3                  incr UNIT from 0 to (.L$UNIT - 1) do
: 5391 3                      GPHARD (.UNIT, HWPT_REF);
: 5392 3
: 5393 3                  incr CTLR from 0 to (MAX_CTLR - 1) do
: 5394 4                      begin
: 5395 4                          CST [.CTLR, U_CNT] = 0;
: 5396 4
: 5397 4                          incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5398 4                              CST [.CTLR, .OFFSET + OF_DATA, D_STAT] = OFFLINE;
: 5399 4
: 5400 3                      end;
: 5401 3
: 5402 2                  end;
: 5403 2
: 5404 2          if .ENTRY_REASON eq1 START
: 5405 2          then
: 5406 3              begin
: 5407 3                  CTLR_CNT = 0;
: 5408 3
: 5409 3                  incr CTLR from 0 to (MAX_CTLR - 1) do
: 5410 3
: 5411 3                      if .CST [.CTLR, IP_ADDR] neq 0
: 5412 3                      then
: 5413 3                          CTLR_CNT = .CTLR_CNT + 1;
: 5414 3
: 5415 3                  MEMORY (FREE_MEM_ADDR);
: 5416 3
: 5417 2              end;
: 5418 2
: 5419 2          !
: 5420 2          ! CLEAR STATISTICS TABLES
: 5421 2          !
: 5422 2
: 5423 2          incr UNITS from 0 to MAX_UNITS - 1 do

```

! "MORE THAN X IP ADDRESSES."
! CONFIGURATION ERROR
! DROP UNIT

! IF NO IP ADDR MATCH IN CST
! IF GPHARD RETURNS A HWP TABLE

! END OF "NON NEW_PASS" INIT

! DUMMY GPHARDS FOR NEW PASS

! REINITIALIZE UNIT COUNT
! START EACH UNIT AS OFFLINE

! NUMBER OF CONFIGURED CONTROLLERS

! IF CONTROLLER IS PRESENT
! INCREMENT CONTROLLER COUNT
! GET START OF FREE MEMORY
! END OF "START" INITIALIZATION

! CLEAR CURRENT STATISTICS

J11

```

: 5424 2      incr COUNT from 0 to TALLY_CLEAR - 1 do
: 5425 2      TALLY [.UNITS * TALLY_LEN + .COUNT] = 0;
: 5426 2
: 5427 2      if .CLEAR_TABLES
: 5428 2      then
: 5429 2      incr UNITS from 0 to MAX_UNITS - 1 do
: 5430 2      incr COUNT from TALLY_CLEAR to TALLY_LEN - 1 do
: 5431 2      TALLY [.UNITS * TALLY_LEN + .COUNT] = 0;
: 5432 2
: 5433 2      if .CLEAR_TABLES
: 5434 2      then
: 5435 2      incr CTRL from 0 to MAX_CTRL - 1 do
: 5436 2      begin
: 5437 2      C_ERR_TBL [.CTRL, C_ERR_HRD] = 0;
: 5438 2      C_ERR_TBL [.CTRL, C_ERR_SFT] = 0;
: 5439 2      end;
: 5440 2
: 5441 2      !+
: 5442 2      ! MISCELLANEOUS INITIALIZATION
: 5443 2      !-
: 5444 2
: 5445 2      incr CTRL from 0 to (MAX_CTRL - 1) do
: 5446 2      QIO [.CTRL] = 0;
: 5447 2
: 5448 2      incr COUNT from 0 to (RP_CNT - 1) do
: 5449 2      RP_USE [.COUNT] = -1;
: 5450 2
: 5451 2      if .CLK_PRESENT
: 5452 2      then
: 5453 2      LINE_CLOCK = 0;
: 5454 2
: 5455 2      IODQ_IN * IODQ_OUT = 0;
: 5456 2      CRN_LOW * CRN_HIGH = 0;
: 5457 2      SETPRI (PRI00);
: 5458 2
: 5459 1      ENDINIT;

```

! IF CLEAR TABLES ON EVERY PASS

! INITIALIZE TOTALS

! INITIALIZE CONTROLLER ERRORS

! INIT NO. OF OUTSTANDING QIOs

! INITIALIZE RETURN PACKET POOL

! STOP CLOCK IF PRESENT

! INIT I/O DONE QUEUE POINTERS

! INIT COMMAND REFERENCE NUMBER

! SET PROGRAM PRIORITY TO 0

.GLOBL L\$DLY

000000	004137	000000G	.SBTTL	LINIT INITIALIZE SECTION		
000004	162706	000030	LINIT:	JSR	R1,\$SAVE5	5157
000010	012700	000340		SUB	#50,SP	
000014	104441			MOV	#340,R0	5172
000016	012700	000035		TRAP	41	
000022	104447			MOV	#35,R0	5174
000024	103014			TRAP	47	
000026	112737	000005 000000G		BHIS	2\$	
000034	105737	000000G		MOVB	#5,ENTRY,REASON	5177
000040	100403			TSTR	SWP,FLAGS	5179
000042	105066	000010		BMI	1\$	
				CLRB	10(SP)	5181
					: CLEAR, TABLES	

```
1-Apr-1984 11:56:01
1-Apr-1984 11:45:02
```

SEQ 0140
Page 140
(40)

ZRQAM2 V01.6	RD/RX EXERCISER INITIALIZE SECTION	11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 Bliss-16 V4.0-579 DISK\$USER2:([POWERS])ZRQADO.BL1;6	SEQ 0140 Page 140 (40)
000046	000403		BR 2\$	
000050	112766	000001 000010	1\$: MOV8 #1,10(SP)	5179
000056	012700	000040	2\$: MOV #40,R0	5183
000062	104447		TRAP 47	5187
000064	103013		BHIS 3\$	
000066	104433		TRAP 33	
000070	112737	000001 000000G	MOV8 #1,ENTRY,REASON	5189
000076	112766	000001 000010	MOV8 #1,10(SP)	5191
000104	105037	000000G	CLRB ADDR,VECT,OK	5192
000110	105037	000000G	CLRB INIT.OCCURED	5193
000114	012700	000037	3\$: MOV #37,R0	5194
000120	104447		TRAP 47	5197
000122	103006		BHIS 4\$	
000124	112737	000002 000000G	MOV8 #2,ENTRY,REASON	
000132	112766	000001 000010	MOV8 #1,10(SP)	5200
000140	012700	000036	4\$: MOV #36,R0	5201
000144	104447		TRAP 47	5204
000146	103014		BHIS 6\$	
000150	112737	000003 000000G	MOV8 #3,ENTRY,REASON	5207
000156	105737	000000G	TSTB SWP,FLAGS	5209
000162	100403		BMI 5\$	
000164	105066	000010	CLRB 10(SP)	
000170	000403		BR 6\$	5211
000172	112766	000001 000010	5\$: MOV8 #1,10(SP)	5209
000200	012700	000034	6\$: MOV #34,R0	5213
000204	104447		TRAP 47	5217
000206	103043		BHIS 12\$	
000210	112737	000004 000000G	MOV8 #4,ENTRY,REASON	
000216	105037	000000G	CLRB ADDR,VECT,OK	5220
000222	105037	000000G	CLRB INIT.OCCURED	5221
000226	112766	000001 000010	MOV8 #1,10(SP)	5222
000234	012746	000000G	MOV #MSG,01,-(SP)	5223
000240	012746	000001	MOV #1,-(SP)	5224
000244	010600		MOV SP,R0	
000246	104417		TRAP 17	
000250	012702	000075	MOV #75,R2	
000254	012703	000515	7\$: MOV #515,R3	5226
000260	010301		MOV R3,R1	5228
000262	001411		8\$: BEQ 11\$	5229
000264	013700	000000G	MOV 1\$DLY,R0	
000270	001404		BEQ 10\$	
000272	005066	000022	9\$: CLR 22(SP)	
000276	005300		DEC R0	
000300	001374		BNE 9\$	
000302	005301		10\$: DEC R1	
000304	000766		BR 8\$	
000306	104422		11\$: TRAP 22	
000310	005302		DEC R2	
000312	001360		BNE 7\$	
000314	022626		CMP (SP)+,(SP)+	
000316	023727	000000G 000004	12\$: CMP 1\$UNIT,#4	5234
000324	101405		BLOS 13\$	5243
000326	104454		TRAP 54	5246

1.1.1

ZRQAM2
V01.6

RD/RX EXERCISER
INITIALIZE SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.0L1;6

SEQ 0141
Page 141
(40)

000330	000001			.WORD	1		
000332	000000G			.WORD	EGS.01		
000334	000000V			.WORD	EMS.01		
000336	104444			TRAP	44		
000340	123727	000000G 000005	13\$:	CMPB	ENTRY.REASON,#5	:	5256
000346	001002			BNE	14\$		
000350	000137	003362'		JMP	41\$		
000354	112766	000377 000006	14\$:	MOVB	#377,6(SP)	: *,SMALLEST.DRIVE	5259
000362	005000			CLR	R0	: COUNT	5261
000364	005060	000000G	15\$:	CLR	CST(R0)	: *(COUNT)	5262
000370	062700	000002		ADD	#2,R0	: *,COUNT	5261
000374	020027	000124		CMP	R0,#124	: COUNT, *	
000400	003771			BLE	15\$		
000402	013704	000000G		MOV	L\$UNIT,R4	:	5264
000406	005003			CLR	R3	: UNIT	
000410	000435			BR	18\$		
000412	010302		16\$:	MOV	R3,R2	: UNIT, *	5266
000414	006302			ASL	R2		
000416	012700	000020		MOV	#20,R0		
000422	060600			ADD	SP,R0	: HWPT.ADDRESS, *	
000424	060002			ADD	R0,R2		
000426	010300			MOV	R3,R0	: UNIT, *	
000430	104442			TRAP	42		
000432	010001			MOV	R0,R1	: *,HWPT.REF	
000434	010112			MOV	R1,(R2)	: HWPT.REF, *	
000436	001421			BEQ	17\$		
000440	005002			CLR	R2		5269
000442	156602	000006		BISB	6(SP),R2	: SMALLEST.DRIVE, *	
000446	116100	000006		MOVB	6(R1),R0	: *(HWPT.REF), *	
000452	042700	177760		BIC	#177760,R0		
000456	020002			CMP	R0,R2		
000460	103010			BHIS	17\$		
000462	116100	000006		MOVB	6(R1),R0	: *(HWPT.REF), *	5271
000466	042700	177760		BIC	#177760,R0		
000472	105066	000006		CLRB	6(SP)	: SMALLEST.DRIVE	
000476	050066	000006		BIS	R0,6(SP)	: *,SMALLEST.DRIVE	
000502	005203		17\$:	INC	R3	: UNIT	5264
000504	020304		18\$:	CMP	R3,R4	: UNIT, *	
000506	002741			BLT	16\$		
000510	013766	000000G 000014		MOV	L\$UNIT,14(SP)	:	5273
000516	005004			CLR	R4	: UNIT	
000520	000137	003340'		JMP	39\$		
000524	010400		19\$:	MOV	R4,R0	: UNIT, *	5275
000526	006300			ASL	R0		
000530	012703	000020		MOV	#20,R3		
000534	060603			ADD	SP,R3	: HWPT.ADDRESS, *	
000536	060300			ADD	R3,R0		
000540	005710			IST	(R0)		
000542	001002			BNE	20\$		
000544	000137	003336'		JMP	38\$		
000550	105066	000004	20\$:	CLRB	4(SP)	: FLAG	5278
000554	011001			MOV	(R0),R1	: *,HWPT.REF	5279
000556	005016			CLR	(SP)	: CLR	5281

M1.1

ZRQAM2
V01.6RD/RX EXERCISER
INITIALIZE SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6SEQ 0142
Page 142
(40)

000560	011646		21\$:	MOV	(SP), -(SP)			
000562	012746	000126		MOV	#126, -(SP)		; CTRL, *	5283
000566	004737	000000G		JSR	PC, BL\$MUL			
000572	022626			CMP	(SP)+, (SP)+			
000574	026011	000000G		CMP	CST(R0), (R1)		; *, HWPT.REF	
000600	001402			BEQ	22\$			
000602	000137	002612'		JMP	27\$			
000606	012766	000001 000012	22\$:	MOV	#1, 12(SP)			5313
000614	112766	000001 000004		MOVB	#1, 4(SP)		; *, FLAG	
000622	012705	000006		MOV	#6, R5			5286
000626	060105			ADD	R1, R5		; HWPT.REF, *	
000630	111546			MOVB	(R5), -(SP)			
000632	042716	177760		BIC	#177760, (SP)			
000636	005000			CLR	R0			
000640	156600	000010		BISB	10(SP), R0		; SMALLEST.DRIVE, *	
000644	160016			SUB	R0, (SP)			
000646	012746	000012		MOV	#12, -(SP)			
000652	004737	000000G		JSR	PC, BL\$MUL			
000656	010066	000006		MOV	R0, 6(SP)			
000662	022626			CMP	(SP)+, (SP)+			
000664	011646			MOV	(SP), -(SP)		; CTRL, *	5287
000666	012746	000053		MOV	#53, -(SP)			
000672	004737	000000G		JSR	PC, BL\$MUL			
000676	010003			MOV	R0, R3			
000700	022626			CMP	(SP)+, (SP)+			
000702	066600	000002		ADD	2(SP), R0			
000706	006300			ASL	R0			
000710	032760	040000 000006G		BIT	#40000, CST+6(R0)			
000716	001121			BNE	26\$			
000720	016602	000002		MOV	2(SP), R2		; *, TEMP	5290
000724	062702	000003		ADD	#3, R2		; *, TEMP	
000730	010300			MOV	R3, R0			5291
000732	060200			ADD	R2, R0		; TEMP, *	
000734	006300			ASL	R0			
000736	062700	000000G		ADD	#CST, R0			
000742	011510			MOV	(R5), (R0)			
000744	010446			MOV	R4, -(SP)		; UNIT, *	5293
000746	000316			SWAP	(SP)			
000750	042716	170377		BIC	#170377, (SP)			
000754	042710	007400		BIC	#7400, (R0)			
000760	052610			BIS	(SP)+, (R0)			
000762	042710	010000		BIC	#10000, (R0)			5294
000766	052710	040000		BIS	#40000, (R0)			5295
000772	010300			MOV	R3, R0			5296
000774	060200			ADD	R2, R0		; TEMP, *	
000776	006300			ASL	R0			
001000	016160	000010 000002G		MOV	10(R1), CST+2(R0)		; *(HWPT.REF), *	
001006	010300			MOV	R3, R0			5299
001010	060200			ADD	R2, R0		; TEMP, *	
001012	006300			ASL	R0			
001014	016160	000012 000004G		MOV	12(R1), CST+4(R0)		; *(HWPT.REF), *	
001022	010300			MOV	R3, R0			5301
001024	060200			ADD	R2, R0		; TEMP, *	

N11

ZRQAM2
V01.6

RD/RX EXERCISER
INITIALIZE SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11gs-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0143
Page 143
(40)

001026	006300		ASL	R0			
001030	016160	000014 000006G	MOV	14(R1),CST+6(R0)	; *(HWPT.REF),*		
001036	010300		MOV	R3,R0	;		5303
001040	060200		ADD	R2,R0	; TEMP,*		
001042	006300		ASL	R0			
001044	016160	000016 000010G	MOV	16(R1),CST+10(R0)	; *(HWPT.REF),*		
001052	010300		MOV	R3,R0	;		5306
001054	060200		ADD	R2,R0	; TEMP,*		
001056	006300		ASL	R0			
001060	062700	000020G	ADD	0CST+20,R0			
001064	105010		CLR	(R0)			
001066	111546		MOV	(R5),-(SP)	;		5308
001070	005046		MOV	-(SP)			
001072	032766	000040 000002	BIT	040,2(SP)			
001100	001401		BEQ	23\$			
001102	005216		INC	(SP)			
001104	005116	23\$:	COM	(SP)			
001106	011646		MOV	(SP),-(SP)			
001110	042710	100000	BIC	0100000,(R0)			
001114	006026		ROR	(SP),*			
001116	103002		BCC	24\$			
001120	052710	100000	BIS	0100000,(R0)			
001124	005726	24\$:	TST	(SP),*			
001126	111516		MOVB	(R5),(SP)	;		5310
001130	042710	010000	BIC	010000,(R0)			
001134	032726	000100	BIT	0100,(SP),*			
001140	001402		BEQ	25\$			
001142	052710	010000	BIS	010000,(R0)			
001146	010300	25\$:	MOV	R3,R0	;		5312
001150	060200		ADD	R2,R0	; TEMP,*		
001152	006300		ASL	R0			
001154	005060	000022G	CLR	CST+22(R0)			
001160	000427		BR	28\$	;		5318
001162	011146	26\$:	MOV	(R1),-(SP)	; HWPT.REF,*		
001164	111546		MOVB	(R5),-(SP)			
001166	042716	177760	BIC	0177760,(SP)			
001172	012746	000000G	MOV	0CER.01, -(SP)			
001176	012746	000003	MOV	03, -(SP)			
001202	010600		MOV	SP,R0	; SP,*		
001204	104417		TRAP	17			
001206	062706	000010	ADD	010,SP			
001212	112764	000001 000000G	MOVB	01.DUR(R4)	; *,*(UNIT)		5320
001220	010400		MOV	R4,R0	; UNIT,*		5321
001222	104451		TRAP	51			
001224	000405		BR	28\$	;		5317
001226	005216	27\$:	INC	(SP)	; CLR		5281
001230	000243		.WORD	CLV:CLC			
001232	003002		BGT	28\$			
001234	000137	002144'	JMP	21\$			
001240	105766	000004	IS	4(SP)	; FLAG		5326
001244	001402		BE	29\$			
001246	000137	003336'	JMP	38\$			

1311

PROGRAM VOL 1,6	RD RX EXERCISER INITIALIZE SECTION		11 Apr 1984 11:45:01 11 Apr 1984 11:45:02	VAX 11 B1145 16 V4.0 5/9 DISK105FR2:[POWER5]ZQADO,BI 1,6	SEQ 0144 Page 144 (40)
001252 005066 000012		294:	CLR	12(SP)	1 CLR
001256 016646 000012		301:	MOV	12(SP), (SP)	1 CLR,*
001262 012746 000126			MOV	0126, (SP)	
001266 004737 000000G			JSR	PC,BL 1MUL	
001272 022626			CMP	(SP), (SP)	
001274 005760 000000G			TST	CST(R0)	
001300 001174			BNE	354	
001302 011160 000000G			MOV	(R1), CST(R0)	1 HWPT,REF,*
001306 016103 000002			MOV	2(R1), R3	1 *(HWPT,REF),*
001312 042703 177000			BIC	0177000, R3	
001316 042760 000777 000000,4			BIC	0777, CST, 2(R0)	
001324 050360 000000,4			BIC	R3, CST, 2(R0)	
001330 116160 000004 000004G			MOVB	4(R1), CST, 4(R0)	1 *(HWPT,REF),*
001336 012705 000006			MOV	06, R5	1
001342 060105			ADD	R1, R5	1 HWPT,REF,*
001344 111546			MOVB	(R5), (SP)	
001346 042716 177760			BIC	0177760, (SP)	
001352 005000			CLR	R0	
001354 156600 000010			BISB	10(SP), R0	1 SMALLEST, DRIVE,*
001360 160016			SUB	R0, (SP)	
001362 012746 000012			MOV	012, (SP)	
001366 004737 000000G			JSR	PC,BL 1MUL	
001372 005726			TST	(SP),	
001374 010002			MOV	R0, R2	1 *,TEMP
001376 062702 000003			ADD	03, R2	1 *,TEMP
001402 016616 000014			MOV	14(SP), (SP)	1 CLR,*
001406 012746 000053			MOV	053, (SP)	
001412 004737 000000G			JSR	PC,BL 1MUL	
001416 010003			MOV	R0, R4	
001420 005726			TST	(SP),	
001422 060200			ADD	R2, R0	1 TEMP,*
001424 006300			ASH	R0	
001426 062700 000000G			ADD	0CST, R0	
001432 011510			MOV	(R5), (R0)	
001434 010416			MOV	R4, (SP)	1 UNIT,*
001436 000316			SWAB	(SP)	
001440 042716 170377			BIC	0170377, (SP)	
001444 042710 007400			BIC	07400, (R0)	
001450 052610			BIS	(SP), (R0)	
001452 042710 010000			BIC	01000, (R0)	
001456 052710 040000			BIS	040000, (R0)	
001462 105715			TSTB	(R5)	
001464 100003			BPL	311	
001466 012761 177777 000014			MOV	0 1, 14(R1)	1 *,*(HWPT,REF)
001474 010300		311:	MOV	R5, R0	1
001476 060200			ADD	R1, R0	1 TEMP,*
001500 006300			ASH	R0	
001502 016160 000010 000002G			MOV	10(R1), CST, 2(R0)	1 *(HWPT,REF),*
001510 010300			MOV	R5, R0	
001512 060200			ADD	R2, R0	1 TEMP,*
001514 006300			ASH	R0	
001516 016160 000012 000004G			MOV	12(R1), CST, 4(R0)	1 *(HWPT,REF),*
001524 010300			MOV	R3, R0	1

C12

ZRGAND		RD RX EXERCISE		11 Apr 1984 11:56:01		VAX 11 B1150 16 V4.0 5/9		SEQ 0145
V01.6		INITIALIZE SECTION		11 Apr 1984 11:45:02		DISK USER2:[POWER5]ZRGADO.BL 1:5		Page 145 (40)
001526	060200			ADD	R2,R0	:	TEMP,*	
001530	006300			ASL	R0	:		
001532	016160	000014	000006G	MOV	14(R1),CST+6(R0)	:	*(HWPT,REF),*	
001540	010300			MOV	R3,R0	:		5.355
001542	060200			ADD	R2,R0	:	TEMP,*	
001544	006300			ASL	R0	:		
001546	016160	000016	000010G	MOV	16(R1),CST+10(R0)	:	*(HWPT,REF),*	
001554	010300			MOV	R3,R0	:		5.358
001556	060200			ADD	R2,R0	:	TEMP,*	
001560	006300			ASL	R0	:		
001562	062700	000020G		ADD	0CST+20,R0	:		
001566	105010			CLRB	(R0)	:		
001570	111546			MOVH	(R5),(SP)	:		5.360
001572	005046			CLR	(SP)	:		
001574	032766	000040	000002	BIT	040,2(SP)	:		
001602	001401			BEQ	52:	:		
001604	005216			INC	(SP)	:		
001606	005116			COM	(SP)	:		
001610	011646			MOV	(SP),(SP)	:		
001612	042710	100000		BIC	0100000,(R0)	:		
001616	006026			ROR	(SP),*	:		
001620	103002			BCC	33:	:		
001622	052710	100000		BIS	0100000,(R0)	:		
001626	005226			TST	(SP),*	:		
001630	111516			MOVH	(R5),(SP)	:		5.361
001632	042710	010000		BIC	010000,(R0)	:		
001636	052726	000100		BIT	0100,(SP),*	:		
001642	001402			BEQ	54:	:		
001644	052710	010000		BIS	010000,(R0)	:		
001650	010300			MOV	R3,R0	:		5.363
001652	060200			ADD	R2,R0	:	TEMP,*	
001654	006300			ASL	R0	:		
001656	005060	000022G		CLR	CST+22(R0)	:		
001662	112766	000001	000004	MOVH	01,4(SP)	:	*,FLAG	5.364
001670	000410			BR	57:	:		5.364
001672	005266	000012		INC	12(SP)	:	CLR	5.360
001676	000243			WORD	CLV:CLC	:		
001700	003002			BGT	36:	:		
001702	000137	00264,1		JMP	30:	:		
001706	105766	000004		TSTB	4(SP)	:	FLAG	5.363
001712	001017			BNE	58:	:		
001714	012746	000001		MOV	01,(SP)	:		5.361
001720	012746	000000X,		MOV	0CER,02,(SP)	:		
001724	012746	000002		MOV	02,(SP)	:		
001730	010600			MOV	SP,R0	:	SP,*	
001732	104417			TRAP	1	:		
001734	112764	000001	000000G	MOVH	01,DUR(R4)	:	*,=UNIT)	5.361
001742	010400			MOV	R4,R0	:	UNIT,*	5.362
001744	104451			TRAP	51	:		
001746	062706	000006		ADD	06,SP	:		5.360
001752	005264			INC	R4	:	UNIT	5.363
001754	020466	000014		CMF	R4,14(SP)	:	UNIT,*	

DLP

ZRQAMP V01.6	RD-RX EXERCISER INITIALIZE SECTION	11-Apr-1984 11:56:01 11 Apr-1984 11:45:02	VAX 11 B1199-16 V4.0 579 DISK\$USER2:[POWERS]ZRQADO,BL1;6	SEQ 0145 Page 146 (40)
001760 002002		BGE 40\$		
001762 000137	0021107	JMP 19\$		
001766 123727	000000G 000005	40\$: CMPB ENTRY,REASON,05	:	5386
001774 001051		BNE 46\$		
001776 013704	000000G	41\$: MOV 1\$UNIT,R4	:	5390
002002 005003		CLR R3	: UNIT	
002004 000404		BR 43\$		
002006 010300		42\$: MOV R3,R0	: UNIT, *	5391
002010 104442		TRAP 42		
002012 010001		MOV R0,R1	: *,HWPT,REF	
002014 005203		INC R3	: UNIT	5390
002016 020304		43\$: CMP R3,R4	: UNIT, *	
002020 002772		BLT 42\$		
002022 005003		CLR R3	: CLR	5393
002024 010346		44\$: MOV R3,-(SP)	: CLR, *	5395
002026 012746	000126	MOV 0126,-(SP)		
002032 004737	000000G	JSR PC,BL\$MUL		
002036 105060	000005G	CLRB CST+5(R0)		
002042 010316		MOV R3,(SP)	: CLR, *	5398
002044 012746	000053	MOV 053,-(SP)		
002050 004737	000000G	JSR PC,BL\$MUL		
002054 012701	000003	MOV 03,R1	: *,OFFSET	5397
002060 010002		45\$: MOV R0,R2	: *	5398
002062 060102		ADD R1,R2	: OFFSET, *	
002064 006302		ASL R2		
002066 042762	020000 000000G	BIC 020000,CST(R2)		
002074 062701	000012	ADD 012,R1	: *,OFFSET	5399
002100 020127	000041	CMP R1,041	: OFFSET, *	
002104 003765		BLE 45\$		
002106 062706	000006	ADD 06,SP		5394
002112 005203		INC R3	: CLR	5393
002114 000243		.WORD CLV:CLC		
002116 003742		BLE 44\$		
002120 123727	000000G 000001	46\$: CMPB ENTRY,REASON,01	:	5404
002126 001017		BNE 49\$		
002130 005037	000000G	CLR CTLR,CNT		5407
002134 005000		CLR R0	: CLR	5409
002136 005760	000000G	47\$: TST CST(R0)	: *(CLR)	5411
002142 001402		BEQ 48\$		
002144 005237	000000G	INC CTLR,CNT	:	5413
002150 062700	000126	48\$: ADD 0126,R0	: *,CLR	5409
002154 000243		.WORD CLV:CLC		
002156 003767		BLE 47\$		
002160 104431		TRAP 31		5415
002162 010037	000000G	MOV R0,FREE,MEM,ADDR	:	
002166 005001		49\$: CLR R1	: UNITS	5415
002170 005003		50\$: CLR R3	: COUNT	5414
002172 010300		51\$: MOV R3,R0	: COUNT, *	5415
002174 060100		ADD R1,R0	: UNITS, *	
002176 006300		ASL R0		
002200 005060	000000G	CLR TALL+(R0)		

E12

ZRQAM, V01.6	RD/RX EXERCISER INITIALIZE SECTION	11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 B11:gs 16 V4.0-579 DISK\$USER2:([POWERS])ZRQADO.BL1;6	SEQ 0147 Page 147 (40)
002204 005203		INC R3	: COUNT	5424
002206 020327 000006		CMP R3,#6	: COUNT,*	
002212 003767		BLE 51\$		
002214 062701 000033		ADD #33,R1	: *,UNITS	5423
002220 020127 000121		CMP R1,#121	: UNITS,*	
002224 003761		BLE 50\$		
002226 032766 000001 000010		BIT #1,10(SP)	: *,CLEAR, TABLES	5427
002234 001436		BEQ 55\$		
002236 005001		CLR R1	: UNITS	5429
002240 012703 000007	52\$:	MOV #7,R3	: *,COUNT	5430
002244 010300	53\$:	MOV R3,R0	: COUNT,*	5431
002246 060100		ADD R1,R0	: UNITS,*	
002250 006300		ASL R0		
002252 005060 000000G		CLR TALLY(R0)		
002256 005203		INC R3	: COUNT	5430
002260 020327 000032		CMP R3,#32	: COUNT,*	
002264 003767		BLE 53\$		
002266 062701 000033		ADD #33,R1	: *,UNITS	5429
002272 020127 000121		CMP R1,#121	: UNITS,*	
002276 003760		BLE 52\$		
002300 032766 000001 000010		BIT #1,10(SP)	: *,CLEAR, TABLES	5433
002306 001411		BEQ 55\$		
002310 005000		CLR R0	: CTRL	5435
002312 105060 000000G	54\$:	CLRB C.ERR.TBL(R0)	: *(CTRL)	5437
002316 105060 000001G		CLRB C.ERR.TBL+1(R0)	: *(CTRL)	5438
002322 062700 000002		ADD #2,R0	: *,CTRL	5435
002326 000243		.WORD CLV!CLC		
002330 003770		BLE 54\$		
002332 005000	55\$:	CLR R0	: CTRL	5445
002334 105060 000000G	56\$:	CLRB QIO(R0)	: *(CTRL)	5446
002340 005200		INC R0	: CTRL	5445
002342 000243		.WORD CLV!CLC		
002344 003773		BLE 56\$		
002346 005000		CLR R0	: COUNT	5448
002350 112760 000377 000000G	57\$:	MOVB #377,RP.USE(R0)	: *,*(COUNT)	5449
002356 005200		INC R0	: COUNT	5448
002360 020077 000007		CMP R0,#7	: COUNT,*	
002364 003771		BLE 57\$		
002366 132737 000001 000000G		BITB #1,CLK.PRESENT		5451
002374 001402		BEQ 58\$		
002376 005037 177546		CLR #0177546		5453
002402 005037 000000G	58\$:	CLR IOOQ.OUT		5455
002406 005037 000000G		CLR IOOQ.IN		
002412 005037 000000G		CLR CRN.HIGH		5456
002416 005037 000000G		CLR CRN.LOW		
002422 005000		CLR R0		5457
002424 104441		TRAP 41		
002426 062706 000030		ADD #30,SP		5157
002432 000207		RTS PC		

; Routine Size: 654 words, Routine Base: \$CODE\$ + 1364

F12

ZRQAMD RD/RX EXERCISER
V01.6 INITIALIZE SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0148
Page 148
(40)

; Maximum stack depth per invocation: 24 words

000000	004737	001364'	.SBTTL	L\$INIT INITIALIZE SECTION	
000004	104411		L\$INIT::JSR	PC,LINIT	
000006	000207		TRAP	11	
			RTS	PC	

5457

; Routine Size: 4 words, Routine Base: \$CODE\$ + 4020
; Maximum stack depth per invocation: 2 words

612

ZRQAM, RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 Bliss-16 V4.0-579 SEQ 0149
 V01.6 AUTODROP SECTION 11-Apr-1984 11:45:02 DISK\$USER2:[POWERS]ZRQADO.BL1;6 Page 149
 (41)

```

; 5460 1 *sbttl 'AUTODROP SECTION
; 5461 1
; 5462 1 !
; 5463 1 ! THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
; 5464 1 ! THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
; 5465 1 ! SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
; 5466 1 ! DROPPED FROM TESTING.
; 5467 1 !
; 5468 1
; 5469 2 BGNAUTO;
; 5470 2
; 5471 2 !IF BIT_TST (SWF_FLAGS, SWF_TRC)
; 5472 2 !then
; 5473 2 ! PRINTF (DBM3);
; 5474 2
; 5475 2 return;
; 5476 2
; 5477 1 ENDAUTO;
  
```

```

000000 000207          LAUTO: .SBTTL LAUTO AUTODROP SECTION
                                RTS    PC
;
; Routine Size: 1 word,      Routine Base: $CODE$ + 4030
; Maximum stack depth per invocation: 0 words
                                5459
  
```

```

000000 004737 004030'      L$AUTO: .SBTTL L$AUTO AUTODROP SECTION
000004 104461              JSR    PC,L$AUTO
000006 000207              TRAP   61
                                RTS    PC
;
; Routine Size: 4 words,      Routine Base: $CODE$ + 4032
; Maximum stack depth per invocation: 2 words
                                5475
  
```

H12

ZRRQM,
V01.6RD/RX EXERCISER
CLEANUP CODING SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0 579
DISK\$USER2:[POWERS]ZRRQADO,BL1;6SEQ 0150
Page 150
(42)

```

: 5478 1 *sbttl 'CLEANUP CODING SECTION'
: 5479 1
: 5480 1 !
: 5481 1 ! THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
: 5482 1 ! AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
: 5483 1 !
: 5484 1
: 5485 2 BGNCLN;
: 5486 2
: 5487 2 DORPT;
: 5488 2
: 5489 2 !CLRVEC (0_TVEC); ! RETURN ODT TRAP TO DIAGNOSTIC SUPERVISER
: 5490 2
: 5491 2 if .CLK_PRESENT
: 5492 2 then
: 5493 3 begin
: 5494 3 LINE_CLOCK = 0; ! STOP THE LINE-CLOCK
: 5495 3 ! CLRVEC (%0'100'); ! RETURN LINE-CLOCK'S VECTOR TO SUPERVISOR
: 5496 3 end;
: 5497 2
: 5498 2 incr CTLR from 0 to (MAX_CTLR - 1) do ! FOR EACH CONTROLLER
: 5499 2
: 5500 2 if (RDRX_ADDR = .CST [.CTLR, IP_ADDR]) neq 0 ! IF CONTROLLER EXISTS
: 5501 2 then
: 5502 3 begin
: 5503 3
: 5504 3 if .ADDR_VECT_OK
: 5505 3 then
: 5506 4 begin
: 5507 4
: 5508 4 if .OCT [.CTLR, STAT] eq 1 ONLINE ! IF CONTROLLER ALIVE
: 5509 4 then
: 5510 4
: 5511 4 incr COUNT from 1 to 10000 do
: 5512 5 begin
: 5513 5 DELAY (1);
: 5514 5
: 5515 5 if .OCT [.CTLR, CRING_CNT] eq 0 ! WAIT TILL OUTSTANDING COMMANDS FINISHED
: 5516 5 then
: 5517 5 exitloop;
: 5518 5
: 5519 4 end;
: 5520 4
: 5521 4 WRT_RDRX (RCIP, RC_ALL, ALL_ONES); ! WRITE IP TO STOP DEVICE
: 5522 4 end;
: 5523 3
: 5524 3 CLRVEC (.CST [.CTLR, VEC_ADDR]); ! RETURN CONTROLLER'S TRAP VECTOR TO SUPERVISOR
: 5525 3 end;
: 5526 2
: 5527 1 ENDCLN;

```

.SBTTL LCLEAN CLEANUP CODING SECTION

112

ZRQAM, V01.6	RD/RX EXERCISER CLEANUP CODING SECTION	11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 Bliss-16 V4.0-579 DISK\$USER2:[POWERS]ZRQADO.BL1;6	SEQ 0151 Page 151 (42)
000000 004137 000000G	LCLEAN: JSR	R1,\$SAVE5	:	5477
000004 005746	TST	-(SP)	:	
000006 104424	TRAP	24	:	5485
000010 132737 000001 000000G	BITB	#1,CLK.PRESENT	:	5491
000016 001402	BEQ	1\$	:	
000020 005037 177546	CLR	#0177546	:	5494
000024 005005	1\$: CLR	R5	: CTRL	5498
000026 010546	2\$: MOV	R5, -(SP)	: CTRL,*	5500
000030 012746 000126	MOV	#126, -(SP)		
000034 004737 000000G	JSR	PC,BL\$MUL		
000040 010001	MOV	R0,R1		
000042 022626	CMP	(SP)*, (SP)*		
000044 016137 000000G 000000G	MOV	CST(R1),RDRX,ADDR		
000052 001450	BEQ	10\$		
000054 132737 000001 000000G	BITB	#1,ADDR.VECT,OK	:	5504
000062 001437	BEQ	9\$	:	
000064 010546	MOV	R5, -(SP)	: CTRL,*	5508
000066 012746 000022	MOV	#22, -(SP)		
000072 004737 000000G	JSR	PC,BL\$MUL		
000076 022626	CMP	(SP)*, (SP)*		
000100 005760 000000G	TST	DCT(R0)		
000104 100022	BPL	8\$		
000106 012704 023420	MOV	#23420,R4	: *,COUNT	5511
000112 012703 000001	3\$: MOV	#1,R3	: *,\$\$TMP2	5513
000116 001410	4\$: BEQ	7\$		
000120 013702 000000G	MOV	L\$DLY,R2	: *,\$\$TMP1	
000124 001403	BEQ	6\$		
000126 005016	5\$: CLR	(SP)	: \$\$TMP	
000130 005302	DEC	R2	: \$\$TMP1	
000132 001375	BNE	5\$		
000134 005303	6\$: DEC	R3	: \$\$TMP2	
000136 000767	BR	4\$		
000140 105760 000000G	7\$: TSTB	DCT(R0)	:	5515
000144 001402	BEQ	8\$	:	5517
000146 005304	DEC	R4	: COUNT	5511
000150 001360	BNE	3\$		
000152 012700 177777	8\$: MOV	#-1,R0	: *,RC,REG	5521
000156 010077 000000G	MOV	R0,RDRX,ADDR	: RC,REG,*	
000162 016100 000002G	9\$: MOV	CST+2(R1),RC	:	5524
000166 042700 177000	BIC	#177000,R0		
000172 104436	TRAP	36		
000174 005205	10\$: INC	R5	: CTRL	5498
000176 000243	.WORD	CLV!CLC		
000200 003712	BLF	2\$		
000202 005726	TST	(SP)*	:	5477
000204 000207	RTS	PC		

; Routine Size: 67 words, Routine Base: \$CODE\$ + 4042
; Maximum stack depth per invocation: 10 words

J12

ZRQAM2
V01.6

RD/RX EXERCISER
CLEANUP CODING SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0152
Page 152
(42)

000000 004737 004042' .SBTTL L\$CLEAN CLEANUP CODING SECTION
L\$CLEAN;;
000004 104412 JSR PC,L\$CLEAN
000006 000207 TRAP 12
RTS PC

5525

; Routine Size: 4 words, Routine Base: \$CODE\$ + 4250
; Maximum stack depth per invocation: 2 words

K12

ZRQAM2
V01.6

RD/RX EXERCISER
DROP UNIT SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0153
Page 153
(43)

```

: 5528 1 *sbttl 'DROP UNIT SECTION'
: 5529 1
: 5530 1 !
: 5531 1 ! THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
: 5532 1 ! TO NO LONGER BE TESTED.
: 5533 1 !-
: 5534 1
: 5535 2 BGNDU;
: 5536 2
: 5537 2 local
: 5538 2     UNIT : word,
: 5539 2     PRINT : byte initial (byte (FALSE));
: 5540 2
: 5541 2 label
: 5542 2     SEARCH;
: 5543 2
: 5544 3 begin
: 5545 3
: 5546 3 register
: 5547 3     INPUT = 0;
: 5548 3
: 5549 3 UNIT = .INPUT;
: 5550 2 end;
: 5551 2
: 5552 3 if BITLIST (SWP_FLAGS, SWF_TRC)
: 5553 2 then
: 5554 2     PRINTF (DBM5, .UNIT);
: 5555 2
: 5556 2 SEARCH :
: 5557 3 begin
: 5558 3
: 5559 3 incr CTLR from 0 to (MAX_CTLR - 1) do
: 5560 3
: 5561 3     incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5562 3
: 5563 3         if (.CST [.CTLR, .OFFSET + OF_DATA, D_UNIT] eq1 .UNIT) and
: 5564 4             (.CST [.CTLR, .OFFSET + OF_DATA, D_PRES] eq1 PRESENT)
: 5565 3         then
: 5566 4             begin
: 5567 4
: 5568 4                 if (.CST [.CTLR, .OFFSET + OF_DATA, D_STAT] eq1 ONLINE) or
: 5569 4                     (.DUR [.UNIT] eq1 DU_ONLINE) or
: 5570 5                     (.DUR [.UNIT] eq1 DU_PROTECT)
: 5571 4                 then
: 5572 5                     begin
: 5573 5                         PRINT = TRUE;
: 5574 5
: 5575 5                         if (.CST [.CTLR, U_CNT] gtru 0) and
: 5576 6                             (.CST [.CTLR, .OFFSET + OF_DATA, D_STAT] eq1 ONLINE)
: 5577 5                         then
: 5578 5                             CST [.CTLR, U_CNT] = .CST [.CTLR, U_CNT] - 1;
: 5579 5
: 5580 5                         if (.CST [.CTLR, U_CNT] eq1 0) and

```

! UNIT NUMBER
! NO PRINTING

! RO = UNIT NO.
! GET UNIT NUMBER
! UNDECLARE RO

! SEARCH BLOCK

! FOR EACH CNTR
! FOR EACH UNIT
! IF UNIT MATCHES

! IF UNIT ALIVE

! O.K. TO PRINT

! DECREMENT COUNT

112

ZRQAM2
V01.6

RD/RX EXERCISER
DROP UNIT SECTION

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0154
Page 154
(43)

```

: 5581 6      (.CST [.CTRL, .OFFSET + OF_DATA, D_STAT] eq1 ONLINE)
: 5582 5      then
: 5583 5          EOP_FLAG = TRUE;
: 5584 5
: 5585 5      CST [.CTRL, .OFFSET + OF_DATA, D_STAT] = OFFLINE;
: 5586 4      end;
: 5587 4
: 5588 4      leave SEARCH;
: 5589 3      end;
: 5590 3
: 5591 2      end;
: 5592 2
: 5593 2      if .PRINT or
: 5594 2          (.DUR [.UNIT] eq1 DU_CONF) or
: 5595 2          (.DUR [.UNIT] eq1 DU_INIT) or
: 5596 2          (.DUR [.UNIT] eq1 DU_ONLINE) or
: 5597 3          (.DUR [.UNIT] eq1 DU_PROTECT)
: 5598 2      then
: 5599 3          begin
: 5600 3              PRINTF (DU_MSG, .UNIT);
: 5601 3              PRINTF (.DU_RSN [.DUR [.UNIT]]);
: 5602 2          end;
: 5603 2
: 5604 1      ENDDU;

```

! ALL UNITS OFFLINE
! MARK UNIT OFFLINE
! IF UNIT ALIVE
! EXIT SEARCH BLOCK
! IF UNIT FOUND
! IF OK TO PRINT
! "UNIT XX DROPPED"
! REASON

000000	004137	000000G	LDU:	.SBITL	LDU DROP UNIT SECTION		
000004	005746			JSR	R1, \$SAVE5	:	5527
000006	105046			IST	-(SP)		
000010	010001			CLRB	-(SP)	:	PRINT
000012	032737	000001 000000G		MOV	R0, R1	:	INPUT, UNIT
000020	001411			BIT	#1, SWP_FLAGS	:	5549
000022	010146			BEQ	1\$		5552
000024	012746	000000G		MOV	R1, (SP)	:	UNIT, *
000030	012746	000002		MOV	#DBM5, -(SP)		5554
000034	010600			MOV	#2, (SP)		
000036	104417			MOV	SP, R0	:	SP, *
000040	062706	000006		TRAP	17		
000044	005005			ADD	#6, SP		
000046	010546		1\$:	CLR	R5	:	CTRL
000050	012746	000053	2\$:	MOV	R5, -(SP)	:	CTRL, *
000054	004737	000000G		MOV	#53, (SP)		5559
000060	010066	000006		JSR	PC, BL \$MUL		5563
000064	012703	000003		MOV	R0, 6(SP)		
000070	010300			MOV	#3, R3	:	*.OFFSET
000072	066600	000006	3\$:	MOV	R3, R0	:	OFFSET, *
000076	006300			ADD	6(SP), R0		5561
000100	012702	000000G		ASL	R0		5563
000104	060002			MOV	#C51, R2		
000106	010104			ADD	R0, R2		
000110	011200			MOV	R1, R4	:	UNIT, *
000112	000300			MOV	(R2), R0		
				SWAB	R0		

M12

ZRQAM2
V01.6RD/RX EXERCISER
DROP UNIT SECTION11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX 11 B11ss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6SEQ 0155
Page 155
(43)

000114	042700	177760		BIC	#177760,R0		
000120	020004			CMP	R0,R4		
000122	001055			BNE	9\$		
000124	032712	040000		BIT	#40000,(R2)	:	5564
000130	001452			BEQ	9\$		
000132	005004			CLR	R4	:	5568
000134	032712	020000		RIT	#20000,(R2)		
000140	001402			BEQ	4\$		
000142	005204			INC	R4		
000144	000410			BR	5\$		
000146	126127	000000G 000007	4\$:	CMPB	DUR(R1),#7	: *(UNIT),*	5569
000154	001404			BEQ	5\$		
000156	126127	000000G 000011		CMPB	DUR(R1),#11	: *(UNIT),*	5570
000164	001032			BNF	8\$		
000166	112766	000001 000004	5\$:	MOVB	#1,4(SP)	: *,PRINT	5573
000174	010516			MOV	R5,(SP)	: CTRL,*	5575
000176	012746	000126		MOV	#126,-(SP)		
000202	004737	000000G		JSR	PC,BL\$MUL		
000206	005726			TST	(SP)+		
000210	062700	000004G		ADD	#CST+4,R0		
000214	105760	000001		TSTB	1(R0)		
000220	001404			BEQ	6\$		
000222	006004			ROR	R4	:	5576
000224	105660	000001		SBCB	1(R0)	:	5578
000230	001006			BNE	7\$	:	5580
000232	032712	020000	6\$:	BIT	#20000,(R2)	:	5581
000236	001403			BEQ	7\$		
000240	112737	000001 000000G		MOVB	#1,EOP.FLAG	:	5583
000246	042712	020000		BIC	#20000,(R2)	:	5585
000252	022626			CMP	(SP)+,(SP)+	:	5586
000254	000411			BR	10\$		
000256	062703	000012	9\$:	ADD	#12,R3	: *,OFF SET	5561
000262	020327	000041		CMP	R3,#41	: OFFSET,*	
000266	003700			BLE	3\$		
000270	022626			CMP	(SP)+,(SP)+		
000272	005205			INC	R5	: CTRL	5559
000274	000243			.WORD	CLV:CLC		
000276	003663			BLE	2\$		
000300	032716	000001	10\$:	BIT	#1,(SP)	: *,PRINT	5593
000304	001020			BNE	11\$		
000306	126127	000000G 000001		CMPB	DUR(R1),#1	: *(UNIT),*	5594
000314	001414			BEQ	11\$		
000316	126127	000000G 000002		CMPB	DUR(R1),#2	: *(UNIT),*	5595
000324	001410			BEQ	11\$		
000326	126127	000000G 000007		CMPB	DUR(R1),#7	: *(UNIT),*	5596
000334	001404			BEQ	11\$		
000336	126127	000000G 000011		CMPB	DUR(R1),#11	: *(UNIT),*	5597
000344	001024			BNF	12\$		
000346	010146		11\$:	MOV	R1,-(SP)	: UNIT,*	5600
000350	012746	000000G		MOV	#DU.MSG,-(SP)		
000354	012746	000002		MOV	#2,(SP)		
000360	010600			MOV	SP,R0	: SP,*	

N12

ZRQAM, V01.6	RD/RX EXERCISER DROP UNIT SECTION	11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 Bliss-16 V4.0.579 DISK\$USER2:[POWERS]ZRQADO.BL1;6	SEQ 0156 Page 156 (43)
000362 104417		TRAP 17		
000364 116101	000000G	MOVB DUR(R1),R1	; *(UNIT),*	5601
000370 042701	177400	BIC #177400,R1		
000374 006301		ASL R1		
000376 016116	000000G	MOV DU,RSN(R1),(SP)		
000402 012746	000001	MOV #1, -(SP)		
000406 010600		MOV SP,R0	; SP, *	
000410 104417		TRAP 17		
000412 062706	000010	ADD #10, SP		5599
000416 022626		12\$: CMP (SP)+, (SP)+		5527
000420 000207		RTS PC		

; Routine Size: 137 words, Routine Base: \$CODE\$ + 4260
; Maximum stack depth per invocation: 14 words

000000 004737	004260'	.SBTTL L\$DU DROP UNIT SECTION		
000004 104453		L\$DU:: JSP PC, LDU		5602
000006 000207		TRAP 53		
		RTS PC		

; Routine Size: 4 words, Routine Base: \$CODE\$ + 4702
; Maximum stack depth per invocation: 2 words

1114

```

: 5605 1 *4bttl 'ADD UNIT SECTION
: 5606 1
: 5607 1
: 5608 1
: 5609 1 ! THE ADD UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
: 5610 1 ! TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
: 5611 1 ! TO THE TEST CYCLE.
: 5612 1
: 5613 1
: 5614 1
: 5615 1 local
: 5616 1     STINDEX : word,
: 5617 1     ENDOX : word;
: 5618 1
: 5619 1 register
: 5620 1     UNIT = 0;
: 5621 1
: 5622 1 IF BIT 15 (SWP_FLAGS, SWP_CST)
: 5623 1 then
: 5624 1     begin
: 5625 1         STINDEX = UNIT + TALLY_LEN;
: 5626 1         ENDOX = STINDEX + TALLY_LEN - 1;
: 5627 1
: 5628 1         incr COUNT from STINDEX to ENDOX do
: 5629 1             TALLY [COUNT] = 0;
: 5630 1
: 5631 1     end;
: 5632 1
: 5633 1 ENDAU;

```

000000	004137	0000000	LAU;	4BTTT	LAU ADD UNIT SECTION		
000004	105757	0000000		JSR	R1, \$SAVE		5604
000010	100025			TSB	SWP_FLAGS		5605
000012	010046			BPL	\$1		
000014	012746	000035		MOV	RO, (SP)	! UNIT, *	5606
000016	004757	0000000		MOV	\$55, (SP)		
000018	010002			JSR	PC, B1MCM		
000024	062702	000032		MOV	RO, R2	! STINDEX, ENDOX	5607
000026	010001			ADD	\$55, R2	! *, ENDOX	
000032	005301			MOV	RO, R1	! STINDEX, COUNT	5608
000034	000404			DEC	R1	! COUNT	
000036	010100		15:	BR	\$1		
000040	006300			MOV	R1, RO	! COUNT, *	5609
000044	005060	0000000		ASL	R2		
000050	005201		25:	CLL	TALLY (RO)		
000052	020102			INC	R1	! COUNT	5610
000054	005221			CMF	R1, R2	! COUNT, ENDOX	
000056	022726			BLE	\$1		
000060	000207		35:	CMF	(SP), (SP)		5611
				RTI	PC		5612

! Routine Size: 25 words, Routine Base: 1CODE1 + 4712

C13

ZRQADO
VOL.6

RD RX EXERCISER
ADD UNIT SECTION

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX 11 B1154 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0158
Page 158
(44)

; Maximum stack depth per invocation: 6 words

000000	004757	004712	LSAU::	LSBTL	LSAU ADD UNIT SECTION		
000004	104452			JSR	PC,LSAU	1	5631
000008	000707			TRAP	%2		
				RTS	PC		

; Routine Size: 4 words, Routine Base: \$CODE\$ + 4774
; Maximum stack depth per invocation: 2 words

[0] 3

ZRQAM,
V01.6

RDRX EXERCISER
NON-EXISTENT MEMORY TRAP HANDLER

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1199 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL 116

SEQ 0159
Page 159
(45)

```
: 5631 1 *sbttl 'NON-EXISTENT MEMORY TRAP HANDLER'
: 5635 1
: 5636 1 !
: 5637 1 ! THIS TRAP HANDLER IS VECTORED FROM LOCATION 4 FOR ALL UNIBUS TIMEOUT
: 5638 1 ! ERRORS, INDICATING THAT AN ATTEMPT WAS MADE TO REFERENCE A NON-EXISTENT
: 5639 1 ! MEMORY LOCATION. ITS MAIN PURPOSE IS TO SET A FLAG FOR THE RDRX
: 5640 1 ! REGISTER EXISTENCE TEST, INDICATING THE ABSENCE OF A DEVICE REGISTER.
: 5641 1 !
: 5642 1 !
: 5643 2 BUNSRV (NEX TRAP);
: 5644 2
: 5645 2 NEX = TRUE; ! NEX TRAP OCCURRED
: 5646 2
: 5647 1 ENDSRV;
```

```
000000 012737 000001 000000G .SBTTL NEX.TRAP NON-EXISTENT MEMORY TRAP HANDLER
                                NEX.TRAP::
000006 000002                MOV    01,NEX
                                RTI
```

5645
5643

; Routine Size: 4 words, Routine Base: \$CODE\$ + 5004
; Maximum stack depth per invocation: 0 words

115

ZRQAM2 RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 B1195 16 V4.0 579 SEQ 0160
V01.6 TIME OF DAY 11-Apr-1984 11:45:02 DISK1USER2:[POWERS]ZRQADO.BL1;6 Page 160
(46)

```

: 5648 1 *sbttl 'TIME OF DAY'
: 5649 1
: 5650 1 !
: 5651 1 ! THIS INTERRUPT SERVICE ROUTINE KEEPS TRACK OF THE TIME-OF-DAY
: 5652 1 !
: 5653 1
: 5654 2 BGNSSRV (TIME);
: 5655 2
: 5656 2 CLK_TICKS = .CLK_TICKS + 1; ! INCREMENT CLOCK TICKS
: 5657 2
: 5658 2 IF .CLK_TICKS gequ 3600
: 5659 2 then
: 5660 3 begin
: 5661 3 MINUTES = .MINUTES + 1; ! UPDATE MINUTE COUNT
: 5662 3 CLK_TICKS = 0;
: 5663 3 end;
: 5664 2
: 5665 2 IF .MINUTES gequ 60
: 5666 2 then
: 5667 3 begin
: 5668 3 HOURS = .HOURS + 1; ! UPDATE HOUR COUNT
: 5669 3 MINUTES = 0;
: 5670 3 end;
: 5671 2
: 5672 2 IF .HOURS gequ 24
: 5673 2 then
: 5674 2 HOURS = 0; ! RATIONALIZE HOURS
: 5675 2
: 5676 1 ENDSRV;

```

000000	005237	000000G	TIME::	.SBTTL	TIME TIME OF DAY	
000004	023727	000000G 007020		INC	CLK_TICKS	5656
000012	103404			CMP	CLK_TICKS, #7020	5658
000014	105237	000000G		BLO	1	
000020	005037	000000G		INCB	MINUTES	5661
000024	123727	000000G 000074	1\$:	CLR	CLK_TICKS	5662
000032	103404			CMPB	MINUTES, #74	5663
000034	105237	000000G		BLO	2	
000040	105037	000000G		INCB	HOURS	5664
000044	123727	000000G 000030	2\$:	CLRB	MINUTES	5665
000052	103402			CMPB	HOURS, #30	5666
000054	105037	000000G		BLO	3	
000060	000002		3\$:	CLRB	HOURS	5667
				RTI		5654

; Routine Size: 25 words, Routine Base: \$CODE\$ + 5014
; Maximum stack depth per invocation: 0 words

ZRGAMP
V01.6RD/RX EXERCISER
GLOBAL ROUTINES11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B1155 16 V4.0 579
DISK\$USER2:([POWERS])ZRGADO.BL1;6SEQ 0161
Page 161
(47)

```

: 5677 1 *sbttl 'GLOBAL ROUTINES'
: 5678 1
: 5679 1 global routine SET_CPAR (CTLR) : novalue *
: 5680 1
: 5681 1 !
: 5682 1 ! THIS ROUTINE SETS UP THE COMMONLY-USED CONTROLLER-RELATED DATA ITEMS
: 5683 1 ! FOR THE GIVEN CONTROLLER NUMBER.
: 5684 1 !
: 5685 1 ! INPUTS:
: 5686 1 !     CTLR - CONTROLLER NUMBER
: 5687 1 !
: 5688 1 ! IMPLICIT OUTPUTS:
: 5689 1 !     CCTLR - CURRENT CONTROLLER NUMBER
: 5690 1 !     CST_ADDR - ADDRESS OF CONTROLLER'S STATUS TABLE
: 5691 1 !     DCT_ADDR - ADDRESS OF CONTROLLER'S DRIVER TABLE
: 5692 1 !     RDRX_ADDR - ADDRESS OF CONTROLLER'S IP REGISTER
: 5693 1 !
: 5694 1 !
: 5695 2 begin
: 5696 2 CCTLR = .CTLR; ! SET CURRENT CONTROLLER NUMBER
: 5697 2 CST_ADDR = CST + (.CTLR * CST_LEN * 2); ! CALCULATE ADDRESS OF CONTROLLER'S CST
: 5698 2 DCT_ADDR = DCT + (.CTLR * DCT_LEN * 2); ! CALCULATE ADDRESS OF CONTROLLER'S DCT
: 5699 2 RDRX_ADDR = .CST_ADDR [IP_ADDR]; ! GET CONTROLLER'S DEVICE ADDRESS
: 5700 1 end;

```

000600 010146		.SBTTL SET_CPAR GLOBAL ROUTINES	
000002	016601	000004	MOV R1, -(SP)
000006	010137	000000G	MOV 4(SP), R1
000012	010146		MOV R1, CCTLR
000014	012746	000126	MOV R1, -(SP)
000020	004737	000000G	MOV 126, (SP)
000024	062700	000000G	JSR PC, BL\$MUL
000030	010037	000000G	ADD 0CST, R0
000034	010116		MOV R0, CST_ADDR
000036	012746	000022	MOV R1, (SP)
000042	004737	000000G	MOV 22, -(SP)
000046	062700	000000G	JSR PC, BL\$MUL
000052	010037	000000G	ADD 0DCT, R0
000056	017737	000000G 000000G	MOV R0, DCT_ADDR
000064	062706	000006	MOV 0CST_ADDR, RDRX_ADDR
000070	012601		ADD 6, SP
000072	000207		MOV (SP), R1
			RTS PC

```

: Routine Size: 30 words, Routine Base: $CODE$ + 5076
: Maximum stack depth per invocation: 5 words

```

613

ZRQAM,
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11gs-16 V4.0 579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0162
Page 162
(48)

```

: 5701 1 global routine SET_UPAR (OFFSET) : novalue *
: 5702 1 !
: 5703 1 ! THIS ROUTINE SETS UP THE COMMONLY-USED UNIT-RELATED DATA ITEMS FOR
: 5704 1 ! THE CURRENT CONTROLLER AND GIVEN CST OFFSET.
: 5705 1 !
: 5706 1 ! INPUTS:
: 5707 1 !     OFFSET - WORD OFFSET INTO CURRENT CONTROLLER'S CST WHICH
: 5708 1 !             DESCRIBES A UNIT
: 5709 1 !
: 5710 1 ! IMPLICIT INPUTS:
: 5711 1 !     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 5712 1 !
: 5713 1 ! IMPLICIT OUTPUTS:
: 5714 1 !     CUOFF - CURRENT UNIT'S CST OFFSET
: 5715 1 !     CDISK - CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 5716 1 !     L$LUN - CURRENT UNIT NUMBER (DRS UNIT NUMBER)
: 5717 1 !     T_ADDR - ADDRESS OF CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 5718 1 !
: 5719 2 begin
: 5720 2 CUOFF = .OFFSET;
: 5721 2 CDISK = .CST_ADDR [.OFFSET + OF_DATA, D_DISK_NUM];
: 5722 2 L$LUN = .CST_ADDR [.OFFSET + OF_DATA, D_UNIT];
: 5723 2 T_ADDR = TALLY + (.L$LUN * TALLY_LEN * 2);
: 5724 1 end;

```

000000 010146		.SBTTL SET_UPAR GLOBAL ROUTINES			
		SET_UPAR::			
000002	016637	000004	000000G	MOV R1, -(SP)	5701
000010	016600	000004		MOV 4(SP), CUOFF	5702
000014	006300			MOV 4(SP), RO	5703
000016	063700	000000G		ASL RO	
000022	111037	000000G		ADD CST_ADDR, RO	
000026	042737	177760	000000G	MOVB (RO), CDISK	
000034	011001			BIC #177760, CDISK	
000036	000301			MOV (RO), R1	5722
000040	042701	177760		SWAB R1	
000044	010137	000000G		BIC #177760, R1	
000050	010146			MOV R1, L\$LUN	
000052	012746	000066		MOV R1, -(SP)	5723
000056	004737	000000G		MOV #66, -(SP)	
000062	062700	000000G		JSR PC, BL\$MUL	
000066	010037	000000G		ADD #TALLY, RO	
000072	022626			MOV RO, T_ADDR	
000074	012601			CMP (SP), (SP)	5701
000076	000207			MOV (SP), R1	
				RTS PC	

; Routine size: 32 words, Routine Base: \$CODE\$ + 517;
; Maximum stack depth per invocation: 4 words

H13

ZRQAM,
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1159 16 V4.0 579
DISK\$USER2:([POWERS]ZRQADO.BL1;6

SEQ 0163
Page 163
(49)

```

: 5725 1
: 5726 1
: 5727 1
: 5728 1
: 5729 1
: 5730 1
: 5731 1
: 5732 1
: 5733 1
: 5734 1
: 5735 1
: 5736 1
: 5737 1
: 5738 2
: 5739 2
: 5740 2
: 5741 2
: 5742 2
: 5743 2
: 5744 2
: 5745 2
: 5746 2
: 5747 2
: 5748 2
: 5749 2
: 5750 3
: 5751 3
: 5752 3
: 5753 3
: 5754 3
: 5755 3
: 5756 4
: 5757 4
: 5758 4
: 5759 4
: 5760 4
: 5761 4
: 5762 5
: 5763 5
: 5764 4
: 5765 5
: 5766 5
: 5767 5
: 5768 5
: 5769 4
: 5770 4
: 5771 4
: 5772 4
: 5773 4
: 5774 4
: 5775 5
: 5776 5
: 5777 5

global routine GET_PKT (CTLR) =
!
! THIS ROUTINE SEARCHES THE MSCP PACKET POOL ALLOCATION TABLE (PKT_USE)
! FOR A FREE MSCP PACKET TO ALLOCATE TO THE GIVEN CONTROLLER. IF ONE IS
! FOUND, THE PACKET IS ZEROED OUT, AND THE PACKET INDEX IS RETURNED
! TO THE CALLER. OTHERWISE, A -1 IS RETURNED INDICATING NONE AVAILABLE.
!
! INPUTS:
! CTLR - CONTROLLER NUMBER REQUESTING ALLOCATION
!
begin
local
index : signed word initial (-1),
RING_ADDR : word,
PACKET_OWNED : byte,
NEXT_PACKET : byte;

NEXT_PACKET = .NEXT_PKT_USE;           ! NEXT PACKET TO TRY

incr COUNT from 0 to (PKT_CNT - 1) do  ! FOR EACH ENTRY IN ALLOCATION TABLE
begin
PACKET_OWNED = FALSE;

if .PKT_USE [.NEXT_PACKET] lss 0      ! IF ENTRY INDICATES FREE PACKET
then
begin
RING_ADDR = .DCT_ADDR [RR_BEG];      ! FIRST RESPONSE PACKET'S ADDRESS

incr I from 1 to (RRING_LEN + CRING_LEN) do ! FOR EACH PACKET ADDRESS

if (.RING_ADDR eqa .MSCP_PKT [.NEXT_PACKET, PKT_LD]) and
(((.RING_ADDR + 2) and ED_OWN) eq1 ED_OWN)

then
begin
PACKET_OWNED = TRUE;                ! CHECK ADDRESS AND OWNERSHIP
exitloop;                          ! PACKET OWNED BY CONTROLLER
end
else
RING_ADDR = .RING_ADDR + 4;          ! ADDRESS OF NEXT PACKET IN RING

if not .PACKET_OWNED                ! IF NOT ALREADY USED
then
begin
PKT_USE [.NEXT_PACKET] = .CTLR;      ! ALLOCATE PACKET TO CONTROLLER
index = .NEXT_PACKET;

```

ZRQAMP RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 B1139-16 V4.0-579 SEQ 0164
 V01.6 GLOBAL ROUTINES 11-Apr-1984 11:45:02 DISK\$USER2:[POWERS]ZRQADO.BL1;6 Page 164
 (49)

```

: 5778 5
: 5779 5      incr J from 2 to (PKT_LEN - 1) do      ! ZERO OUT PACKET
: 5780 5      MSCP_PKT [NEXT_PACKET, J, 0, 16, 0] = 0;
: 5781 5
: 5782 5      exitloop;                                ! DONE
: 5783 5
: 5784 4      end;
: 5785 4
: 5786 3      end;
: 5787 3
: 5788 3      NEXT_PACKET = NEXT_PACKET + 1;          ! TRY NEXT PACKET IN RING
: 5789 3
: 5790 3      if NEXT_PACKET geq PKT_CNT
: 5791 3
: 5792 3      then
: 5793 3          NEXT_PACKET = 0;                    ! IF BEYOND ALL PACKETS, START AT THE TOP
: 5794 3
: 5795 2      end;
: 5796 2
: 5797 2      if (.index geq 0) and                    ! IF PACKET FOUND
: 5798 3          (.PKT_USE [.index] geq 0)
: 5799 2      then
: 5800 2
: 5801 3          begin
: 5802 3              MSCP_PKT [.index, MSGLEN] = SZ_GEN;    ! PACKET SIZE - ONLY ONLINE AND SCC CHANGE IT
: 5803 3              MSCP_PKT [.index, CREDITS] = 1;        ! CREDIT SIZE
: 5804 3              NEXT_PKT_USE = NEXT_PACKET + 1;        ! NEXT PACKET TO ALLOCATE
: 5805 3
: 5806 3              if NEXT_PKT_USE geq PKT_CNT
: 5807 3              then
: 5808 3                  NEXT_PKT_USE = 0;                ! CYCLE BACK TO BEGINNING IF AT END
: 5809 3
: 5810 2              end;
: 5811 2
: 5812 2      return .index;
: 5813 2
: 5814 1      end;

```

000000	004137	000000G	.SBTTL	GET.PKT GLOBAL ROUTINES		
			GET.PKT::			
000004	162706	000006	JSR	R1, \$SAVE5	:	5778
000010	012704	177777	SUB	#6, SP	:	5779
000014	113766	000000G 000004	MOV	#-1, R4	:	5780
000022	012766	000014 000002	MOVB	NEXT_PKT_USE - 1(SP)	:	5781
000030	105016		MOV	#14, 2(SP)	:	5782
000032	005001		1\$: CLR B	(SP)	:	5783
000034	156601	000004	CLR	R1	:	5784
000040	105761	000000G	BISB	4(SP), R1	:	5785
000044	002072		TSIB	PKT_USE(R1)	:	5786
000046	013700	000000G	BGE	7\$	:	5787
000052	016005	000004	MOV	DCT_ADDR, R0	:	5788
			MOV	4(R0), R5	:	5789
					:	5790
					:	5791
					:	5792
					:	5793
					:	5794
					:	5795
					:	5796
					:	5797
					:	5798
					:	5799
					:	5800
					:	5801
					:	5802
					:	5803
					:	5804
					:	5805
					:	5806
					:	5807
					:	5808
					:	5809
					:	5810
					:	5811
					:	5812
					:	5813
					:	5814

J13

ZRQAM2 V01.6	RD/RX EXERCISER GLOBAL ROUTINES		11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 Bliss 16 V4.0-579 DISK\$USER2:[POWERS]ZRQADO.BL1;6	SEQ 0165 Page 165 (49)
000056	010146		MOV R1, -(SP)	:	5761
000060	012746	000106	MOV #106, -(SP)	:	
000064	004737	000000G	JSR PC, BL\$MUL	:	
000070	012702	000010	MOV #10, R2	: *, I	5759
000074	021560	000000G	2\$: CMP (R5), MSCP, PKT(R0)	: RING, ADDR, *	5761
000100	001014		BNE 3\$	:	
000102	012703	000002	MOV #2, R3	: RING, ADDR, *	5762
000106	060503		ADD R5, R3	:	
000110	042703	077777	BIC #77777, R3	:	
000114	020327	100000	CMP R3, #-100000	:	
000120	001004		BNE 3\$	:	
000122	112766	000001 000004	MOVB #1, 4(SP)	: *, PACKET, OWNED	5766
000130	000404		BR 4\$	:	5765
000132	062705	000004	3\$: ADD #4, R5	: *, RING, ADDR	5770
000136	005302		DEC R2	: I	5759
000140	001355		BNE 2\$	:	
000142	032766	000001 000004	4\$: BIT #1, 4(SP)	: *, PACKET, OWNED	5772
000150	001027		BNE 6\$	:	
000152	116661	000030 000000G	MOVB 30(SP), PKT, USE(R1)	: CLR, *	5776
000160	010104		MOV R1, R4	: *, INDEX	5777
000162	010116		MOV R1, (SP)	:	5780
000164	012746	000043	MOV #43, -(SP)	:	
000170	004737	000000G	JSR PC, BL\$MUL	:	
000174	005726		TST (SP),	:	
000176	012702	000002	MOV #2, R2	: *, J	5779
000202	010003		5\$: MOV R0, R3	:	5780
000204	060203		ADD R2, R3	: J, *	
000206	006303		ASL R3	:	
000210	005063	000000G	CLR MSCP, PKT(R3)	:	
000214	005202		INC R2	: J	5779
000216	020227	000042	CMP R2, #42	: J, *	
000222	003767		BLE 5\$	:	
000224	022626		CMP (SP), (SP),	:	5775
000226	000414		BR 9\$	:	
000230	022626		6\$: CMP (SP), (SP),	:	5756
000232	105266	000004	7\$: INCB 4(SP)	: NEXT, PACKET	5788
000236	126627	000004 000014	CMPB 4(SP), #14	: NEXT, PACKET, *	5790
000244	103402		BLO 8\$	:	
000246	105066	000004	CLRB 4(SP)	: NEXT, PACKET	5793
000252	005366	000002	8\$: DEC 2(SP)	: COUNT	5749
000256	001264		BNE 1\$	:	
000260	005703		9\$: TST R4	: INDEX	5797
000262	002435		BUT 11\$	:	
000264	105764	000000G	TSTB PKT, USE(R4)	: *(INDEX)	5798
000270	002432		BUT 11\$	:	
000272	010446		MOV R4, (SP)	: INDEX, *	5802
000274	012746	000106	MOV #106, -(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)	:	5805
000320	152760	000001 000010G	BISB #1, MSCP, PKT+10(R0)	:	
000326	005000		CLR R0	:	5804
000330	156600	000010	BISB 10(SP), R0	: NEXT, PACKET, *	

K13

ZRQAM2
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1154 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0166
Page 166
(49)

000334	005200		INC	R0		
000336	110037	000000G	MOV8	R0, NEXT.PKT.USE		
000342	120027	000014	CMPI	R0, #14	; NEXT.PKT.USE, *	5806
000346	103402		BLO	10\$		
000350	105037	000000G	CLRB	NEXT.PKT.USE		5808
000354	022626	10\$:	CMP	(SP)+, (SP)+		5801
000356	010400	11\$:	MOV	R4, R0	; INDEX, *	5758
000360	062706	000006	ADD	#6, SP		5726
000364	000207		RTS	PC		

; Routine Size: 123 words, Routine Base: \$CODE\$ + 5272
; Maximum stack depth per invocation: 13 words

; 5815 1
; 5816 1

1.13

ZRQAMP
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0167
Page 167
(50)

```

: 5817 1
: 5818 1
: 5819 1 global routine PUT_PKT (index) : novalue *
: 5820 1
: 5821 1 !*
: 5822 1 ! THE MSCP PACKET DESIGNATED BY "INDEX" IS RETURNED TO THE POOL BY THIS
: 5823 1 ! ROUTINE.
: 5824 1 !-
: 5825 1
: 5826 1
: 5827 2 begin
: 5828 2
: 5829 2
: 5830 2 local
: 5831 2 RING_ADDR : word,
: 5832 2 OWNER : word;
: 5833 2
: 5834 2 RING_ADDR = .DCT_ADDR [RR_BEG];
: 5835 2 ! ADDRESS IN FIRST RESPONSE RING
: 5836 2 incr COUNT from 1 to (RRING_LEN + CRING_LEN) do
: 5837 3 ! FOR EACH ADDRESS IN THE RINGS
: 5838 3 begin
: 5839 3 if .MSCP_PKT [.index, PKT_LO] eqia ..RING_ADDR
: 5840 3 ! IF ADDRESS MATCHES
: 5841 3 then
: 5842 4 begin
: 5843 4 OWNER = .RING_ADDR + 2;
: 5844 4 ! ADDRESS OF OWNERSHIP WORD
: 5845 4 ! GIVE OWNERSHIP TO HOST
: 5846 4 ! OWNER = ..OWNER and (not (ED_OWN)) and (not (ED_FLAG));
: 5847 4 end;
: 5848 3 RING_ADDR = .RING_ADDR + 4;
: 5849 3 ! LOOK AT NEXT PACKET ADDRESS IN RING
: 5850 3 end;
: 5851 2
: 5852 2 PKT_USE [.index] = -1;
: 5853 2
: 5854 1 end;

```

000000	004137	000000G	PUT_PKT GLOBAL ROUTINES		
			PUT_PKT::		
000004	013700	000000G	JSR	R1, \$SAVE4	5817
000010	016001	000004	MOV	DCT_ADDR, R0	5818
000014	016602	000014	MOV	4(R0), R1	5819
000020	010246		MOV	14(SP), R2	5820
000024	012746	000106	MOV	R2, (SP)	5821
000026	004737	000000G	MOV	#106, (SP)	5822
000030	012704	000010	JSR	PC, BL \$MUI	5823
000036	026011	000000G	MOV	#10, R4	5824
000040	001005		1\$: CMP	MSCP_PKT(R0), (R1)	5825
000044	012703	000002	BNE	2\$	5826
			MOV	#2, R3	5827
				! *, COUNT	5828
				! *, RING_ADDR	5829
				! *, OWNER	5830

M13

ZRQAMP
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B116: 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0168
Page 168
(50)

000050	060103		ADD	R1,R3	; RING, ADDR, OWNER	
000052	042713	140000	BIC	#140000,(R3)	; *, OWNER	5844
000056	062701	000004	ADD	#4,R1	; *, RING, ADDR	5848
000062	005304		DEC	R4	; COUNT	5836
000064	001364		BNE	1\$		
000066	112762	000377 000000G	MOV8	#377,PKT,USE(R2)		5852
000074	022626		CMP	(SP)+,(SP)+		5827
000076	000207		RTS	PC		5819

; Routine Size: 32 words, Routine Base: \$CODE\$ + 5660
; Maximum stack depth per invocation: 8 words

; 5855 1
; 5856 1

N13

ZRQAM.2 RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 Bliss-16 V4.0-579 SEQ 0169
V01.6 GLOBAL ROUTINES 11-Apr-1984 11:45:02 DISK\$USER2:[POWERS]ZRQAD0.BL1;6 Page 169
(51)

```

: 5857 1 routine PUTA_PKT (CTLR) : novalue =
: 5858 1
: 5859 1 !,
: 5860 1 ! THIS ROUTINE DEALLOCATES ALL MSCP PACKETS WHICH HAVE BEEN ALLOCATED
: 5861 1 ! TO A PARTICULAR CONTROLLER.
: 5862 1 !
: 5863 1 ! INPUTS:
: 5864 1 ! CTLR - CONTROLLER NUMBER
: 5865 1 !-
: 5866 1
: 5867 1 incr COUNT from 0 to (PKT_CNT - 1) do ! FOR EACH ENTRY IN ALLOCATION TABLE
: 5868 1
: 5869 1 if .PKT_USE [.COUNT] =ql .CTLR ! IF PACKET IS ALLOCATED TO GIVEN CONTROLLER
: 5870 1 then
: 5871 1 PKT_USE [.COUNT] = -1; ! DEALLOCATE IT

```

000000	010146		PUTA_PKT:	MOV	R1, -(SP)	:	5857
000002	005000			CLR	R0	:	5857
000004	116001	000000G	1\$:	MOVB	PKT_USE(R0), R1	:	5859
000010	020166	000004		CMP	R1, 4(SP)	:	
000014	001003			BNE	2\$	:	
000016	112760	000377 000000G		MOVB	#377, PKT_USE(R0)	:	5871
000024	005200		2\$:	INC	R0	:	5867
000026	020027	000013		CMP	R0, #13	:	
000032	003764			BLI	1\$	:	
000034	012601			MOV	(SP)+, R1	:	5857
000036	000207			RTS	PC	:	

; Routine Size: 16 words, Routine Base: \$CODE\$ + 5760
; Maximum stack depth per invocation: 2 words

11/14

```

5872 1 global routine GET RETPKT (CTLR) *
5873 1
5874 1 ..
5875 1 THIS ROUTINE SEARCHES THE RETURN PACKET POOL ALLOCATION TABLE (RP USE)
5876 1 FOR A FREE RETURN PACKET TO ALLOCATE TO THE GIVEN CONTROLLER. IF ONE IS
5877 1 FOUND, THE PACKET IS ZEROED OUT, AND THE PACKET INDEX IS RETURNED TO
5878 1 THE CALLER. OTHERWISE, A -1 IS RETURNED INDICATING NONE AVAILABLE.
5879 1
5880 1 INPUTS:
5881 1 CTLR = CONTROLLER NUMBER REQUESTING ALLOCATION
5882 1
5883 1
5884 1
5885 1 begin
5886 1
5887 1 local
5888 1 index : signed word initial (-1) : ASSUME NONE AVAILABLE
5889 1
5890 1 incr COUNT from 0 to (RP CNT - 1) do : FOR EACH ENTRY IN TABLE
5891 1
5892 1 IF (RP USE [.COUNT]) AND 0 : IF FREE RETPKT IS FOUND
5893 1 then
5894 1 begin
5895 1 RP USE [.COUNT] = .CTLR : ALLOCATE RETURN PACKET TO CONTROLLER
5896 1 index = .COUNT :
5897 1
5898 1 incr J from 0 to (RP LEN - 1) do : ZERO OUT RETPKT
5899 1 RETPKT [.COUNT, J, 0, 16, 0] = 0 :
5900 1
5901 1 exit loop : DONE
5902 1
5903 1 return .index : RETURN PACKET INDEX FOR -1 TO CALLER
5904 1 end

```

000000	004157	000000	15872	15873	15874	15875	15876	15877	15878	15879	15880	15881	15882	15883	15884	15885	15886	15887	15888	15889	15890	15891	15892	15893	15894	15895	15896	15897	15898	15899	15900	15901	15902	15903	15904	
000004	012704	177777	15904	15905	15906	15907	15908	15909	15910	15911	15912	15913	15914	15915	15916	15917	15918	15919	15920	15921	15922	15923	15924	15925	15926	15927	15928	15929	15930	15931	15932	15933	15934	15935		
000010	005001		15936	15937	15938	15939	15940	15941	15942	15943	15944	15945	15946	15947	15948	15949	15950	15951	15952	15953	15954	15955	15956	15957	15958	15959	15960	15961	15962	15963	15964	15965	15966	15967	15968	
000012	105761	000000	15969	15970	15971	15972	15973	15974	15975	15976	15977	15978	15979	15980	15981	15982	15983	15984	15985	15986	15987	15988	15989	15990	15991	15992	15993	15994	15995	15996	15997	15998	15999	16000	16001	
000016	002025		16002	16003	16004	16005	16006	16007	16008	16009	16010	16011	16012	16013	16014	16015	16016	16017	16018	16019	16020	16021	16022	16023	16024	16025	16026	16027	16028	16029	16030	16031	16032	16033	16034	16035
000020	116661	000014 000000	16036	16037	16038	16039	16040	16041	16042	16043	16044	16045	16046	16047	16048	16049	16050	16051	16052	16053	16054	16055	16056	16057	16058	16059	16060	16061	16062	16063	16064	16065	16066	16067	16068	16069
000024	010103		16070	16071	16072	16073	16074	16075	16076	16077	16078	16079	16080	16081	16082	16083	16084	16085	16086	16087	16088	16089	16090	16091	16092	16093	16094	16095	16096	16097	16098	16099	16100	16101	16102	16103
000028	010146		16104	16105	16106	16107	16108	16109	16110	16111	16112	16113	16114	16115	16116	16117	16118	16119	16120	16121	16122	16123	16124	16125	16126	16127	16128	16129	16130	16131	16132	16133	16134	16135	16136	16137
000032	012746	000024	16138	16139	16140	16141	16142	16143	16144	16145	16146	16147	16148	16149	16150	16151	16152	16153	16154	16155	16156	16157	16158	16159	16160	16161	16162	16163	16164	16165	16166	16167	16168	16169	16170	16171
000036	004737	000000	16172	16173	16174	16175	16176	16177	16178	16179	16180	16181	16182	16183	16184	16185	16186	16187	16188	16189	16190	16191	16192	16193	16194	16195	16196	16197	16198	16199	16200	16201	16202	16203	16204	16205
000040	022626		16206	16207	16208	16209	16210	16211	16212	16213	16214	16215	16216	16217	16218	16219	16220	16221	16222	16223	16224	16225	16226	16227	16228	16229	16230	16231	16232	16233	16234	16235	16236	16237	16238	16239
000044	005007		16240	16241	16242	16243	16244	16245	16246	16247	16248	16249	16250	16251	16252	16253	16254	16255	16256	16257	16258	16259	16260	16261	16262	16263	16264	16265	16266	16267	16268	16269	16270	16271	16272	16273
000048	010004		16274	16275	16276	16277	16278	16279	16280	16281	16282	16283	16284	16285	16286	16287	16288	16289	16290	16291	16292	16293	16294	16295	16296	16297	16298	16299	16300	16301	16302	16303	16304	16305	16306	16307
000052	060204		16308	16309	16310	16311	16312	16313	16314	16315	16316	16317	16318	16319	16320	16321	16322	16323	16324	16325	16326	16327	16328	16329	16330	16331	16332	16333	16334	16335	16336	16337	16338	16339	16340	16341
000056	060504		16342	16343	16344	16345	16346	16347	16348	16349	16350	16351	16352	16353	16354	16355	16356	16357	16358	16359	16360	16361	16362	16363	16364	16365	16366	16367	16368	16369	16370	16371	16372	16373	16374	16375
000060	005064	000000	16376	16377	16378	16379	16380	16381	16382	16383	16384	16385	16386	16387	16388	16389	16390	16391	16392	16393	16394	16395	16396	16397	16398	16399	16400	16401	16402	16403	16404	16405	16406	16407	16408	16409

C14

PROGRAM	RD RX EXERCISER	11 Apr 1984 11:56:01	VAX 11 B1154 16 V4,0 5/79	SEQ 01/1
VOL 1,6	GLOBAL ROUTINES	11 Apr 1984 11:45:02	DISKUSER2:[POWERS]ZRGADO.BL 1,6	Page 1/1
000060 005202		INC R1		
000062 020227	000025	CMP R2,025		5897
000066 003267		BLE 25		
000070 000404		BR 45		
000072 005201	55:	INC R1		5895
000074 020127	000007	CMP R1,07		5889
000100 003244		BLE 15		
000102 010300	45:	MOV R3,R0		5884
000104 000207		RTN PC		5872
; Routine size: 51 words, Routine Base: \$CODE\$ + 6020				
; Maximum stack depth per invocation: 8 words				

[] 14

ZRQ-12
VOL.1

RD RX EXERCISER
GLOBAL ROUTINES

11-Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX 11-B1144 16 V4.0 579
DISK1USER2:[POWERS]ZRQADO.BL1;6

SEQ 01/2
Page 1/2
(53)

```

: 5905 1 global routine PUT RETPKT (index) : novalue =
: 5906 1
: 5907 1
: 5908 1 THE RETURN PACKET DESIGNATED BY "INDEX" IS RETURNED TO THE POOL BY THIS
: 5909 1 ROUTINE.
: 5910 1
: 5911 1
: 5912 1 RP_USE [,index] = 1;

```

```

000000 016600 000000 PUT, RETPKT GLOBAL ROUTINES
000004 11760 000377 0000006 MOV PC, SP, RO
000012 000007 MOVH 0377, RP, USE (RO)
RTS PC

```

```

: INDEX, *
:

```

```

5905
5905

```

```

: Routine size: n words, Routine Base: $CODE$ + 6126
: Maximum stack depth per invocation: 0 words

```

{ | 7 }

PROGRAM
V01.6

RD-RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1155 16 V4.0 579
DISK\$USER2:[POWER5]ZRDQAD0.BL1:6

SEQ 0173
Page 173
(54)

```

: 5913 1
: 5914 1
: 5915 1 global routine GET_IO_BUFF (ADDR) : novalue =
: 5916 1
: 5917 1
: 5918 1
: 5919 1
: 5920 1 THIS ROUTINE HANDLES THE ALLOCATION OF AN I/O BUFFER FROM THE BUFFER
: 5921 1 POOL.
: 5922 1
: 5923 1 INPUTS:
: 5924 1 ADDR ADDRESS TO STORE THE 2 WORD BUFFER DESCRIPTOR
: 5925 1
: 5926 1 IMPLICIT INPUTS:
: 5927 1 CCTLN - CURRENT CONTROLLER NUMBER
: 5928 1
: 5929 1 OUTPUTS:
: 5930 1 THE ALLOCATED BUFFER'S DESCRIPTOR IS LOADED INTO THE TWO
: 5931 1 WORDS AT "ADDR" AND "ADDR + 2". OTHERWISE, A ZERO IS RETURNED
: 5932 1 AT "ADDR" IF NO BUFFERS ARE AVAILABLE.
: 5933 1
: 5934 1
: 5935 1
: 5936 2 begin
: 5937 2 ADDR = 0;
: 5938 2
: 5939 2 incr COUNT from 0 to (QIO PER CILR * MAX_CILR - 1) do
: 5940 2
: 5941 2 IF .BUFF_OWN [.COUNT] lss 0
: 5942 2
: 5943 2 then
: 5944 2
: 5945 2 begin
: 5946 2 BUFF_OWN [.COUNT] = .CCTLN;
: 5947 2 ADDR = .BUFF_ADDR [.COUNT];
: 5948 2 exit loop;
: 5949 2 end;
: 5950 2
: 5951 2
: 5952 1 end;

```

000000	010146		GET_IO_BUFF ::	GET_IO_BUFF GLOBAL ROUTINES	
000002	005076	000004	MOV	R1, (SP)	
000006	005001		CLR	Q4(SP)	
000010	105761	000000G	CLR	R1	
000014	002011		TEST	BUFF_OWN(R1)	
000018	113761	000000G 000000G	BGE	25	
000022	010100		MOVH	CCTLN, BUFF_OWN(R1)	
000026	005700		MOV	R1, R0	
000030	016076	000000G 000004	MOV	BUFF_ADDR(R0), Q4(SP)	

ZRQAND
V01.6

RD RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11sa 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 01/4
Page 1/4
(54)

000036	000404		BR	3:
000040	005201	2:	INC	R1
000042	020127	000007	CMP	R1,#7
000046	003760		BIF	1:
000050	012601	3:	MOV	(SP)+,R1
000052	000207		RTS	PC

:	5945
:	5939
:	5915

; Routine Size: 22 words, Routine Base: \$CODE\$ + 6142
; Maximum stack depth per invocation: 2 words

; 5954 1
; 5954 1


```

; 5955 1 global routine PUT_IO_BUFF (ADDR) : novalue =
; 5956 1
; 5957 1
; 5958 1
; 5959 1
; 5960 1
; 5961 1
; 5962 1
; 5963 1
; 5964 1
; 5965 1
; 5966 1
; 5967 1
; 5968 1
; 5969 1
; 5970 2
; 5971 2
; 5972 2
; 5973 1

THIS ROUTINE HANDLES THE DEALLOCATION OF AN I/O BUFFER, RETURNING IT
TO THE BUFFER POOL.

INPUTS:
    ADDR - ADDRESS OF THE 2 WORD BUFFER DESCRIPTOR TO BE
           DEALLOCATED

incr COUNT from 0 to (QIO_PER_CTLR * MAX_CTLR - 1) do      ! FOR EACH ENTRY IN BUFFER TABLE
    IF .BUFF_ADDR [.COUNT] eqia ..ADDR                      ! IF THIS IS THE BUFFER'S ENTRY
    then
        begin
            .BUFF_OWN [.COUNT] = -1;                          ! DEALLOCATE BUFFER
        exitloop;                                              ! DONE
        end;

```

		.SBTIL PUT_IO_BUFF GLOBAL ROUTINES		
000000	C10146	PUT_IO_BUFF::		
000002	005001	MOV	R1, -(SP)	5955
000004	010100	CLR	R1	5956
000006	006300	1\$: MOV	R1, R0	5957
000010	026076	ASL	R0	
000016	001004	CMP	BUFF_ADDR(R0), 04(SP)	
000020	112761	BNE	2\$	
000026	000404	MOVB	0577, BUFF_OWN(R1)	5961
000030	005201	BR	3\$	5962
000032	020127	2\$: INC	R1	5963
000036	003762	CMP	R1, 07	
000040	012601	BLE	1\$	
000042	000207	3\$: MOV	(SP), R1	5955
		RTS	PC	

; Routine size: 18 words, Routine Base: \$CODE\$ + 6016
; Maximum stack depth per invocation: 2 words

H14

ZRQAM2
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B11ss 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0176
Page 176
(56)

```

: 5974 1 global routine PUTA_BUFF : novalue
: 5975 1
: 5976 1 !
: 5977 1 ! THIS ROUTINE DEALLOCATES ALL I/O BUFFERS WHICH HAVE BEEN ALLOCATED TO
: 5978 1 ! THE CURRENT CONTROLLER (CCTL).
: 5979 1 !
: 5980 1
: 5981 1      Incr COUNT from 0 to (QIO_PER_CTLR * MAX_CTLR - 1) do      ! FOR EACH ENTRY IN BUFFER TABLE
: 5982 1
: 5983 1      if .BUFF_OWN [.COUNT] eq1 .CCTLR      ! IF THIS BUFFER ALLOCATED TO CURRENT CONTROLLER
: 5984 1      then
: 5985 1          BUFF_OWN [.COUNT] = -1;      ! DEALLOCATE IT

```

		.SRTL	PUTA_BUFF GLOBAL ROUTINES	
000000	010146	PUTA_BUFF::	MOV R1, -(SP)	5974
000002	005000		CLR R0	5981
000004	116001	1\$:	MOVB BUFF_OWN(R0), R1	5983
000010	020137		CMP R1, CCTLR	
000014	001003		BNE 2\$	
000016	112760	000377 000000G	MOVB 0377, BUFF_OWN(R0)	5985
000024	005200	2\$:	INC R0	5981
000026	020027	000007	CMP R0, #7	
000032	003764		BLE 1\$	
000034	012601		MOV (SP)+, R1	5974
000036	000207		RTS PC	

: Routine Size: 16 words, Routine Base: \$CODE\$ + 6262
: Maximum stack depth per invocation: 2 words

ZRGAMP
V01.6RD/RX EXERCISER
GLOBAL ROUTINES11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX 11 Bliss 16 V4.0-5/9
DISK\$USER2:[POWERS]ZRGADO.BL1;6SEQ 0177
Page 177
(57)

```

: 5986 1 global routine OUT_IODQ
: 5987 1
: 5988 1 !+
: 5989 1 ! THIS ROUTINE RETURNS TO THE CALLER THE NEXT RETPKT INDEX TO BE
: 5990 1 ! PROCESSED FROM THE I/O DONE QUEUE (IODQ). THE "OUT" POINTER TO THE
: 5991 1 ! QUEUE IS ALSO UPDATED.
: 5992 1 !
: 5993 1 ! INPUTS:
: 5994 1 ! NONE
: 5995 1 !
: 5996 1 ! OUTPUTS:
: 5997 1 ! THE INDEX OF THE NEXT RETPKT TO BE PROCESSED.
: 5998 1 !-
: 5999 1
: 6000 2 begin
: 6001 2
: 6002 2 local
: 6003 2 index : word;
: 6004 2
: 6005 2 index = .IODQ [.IODQ_OUT]; ! GET NEXT RETPKT INDEX
: 6006 2 IODQ_OUT = .IODQ_OUT + 1; ! ADVANCE "OUT" POINTER
: 6007 2
: 6008 2 if .IODQ_OUT gequ IODQ_LEN ! IF BEYOND END OF QUEUE
: 6009 2 then !
: 6010 2 IODQ_OUT = 0; ! SET POINTER TO BEGINNING OF QUEUE
: 6011 2
: 6012 2 return .index; ! RETURN INDEX TO CALLER
: 6013 1 end;

```

			SBTL	OUT_IODQ GLOBAL ROUTINES		
000000	013700	000000G	OUT_IODQ::			
			MOV	IODQ_OUT,RO	:	6005
000004	116000	000000G	MOVH	IODQ(RO),RO	:	
000010	042700	177400	BIC	#177400,RO	:	
000014	005237	000000G	INC	IODQ_OUT	:	6006
000020	023727	000000G 000010	CMP	IODQ_OUT,#10	:	6008
000026	103402		BLN	11	:	
000030	005037	000000G	CLR	IODQ_OUT	:	6010
000034	000207		RTS	PC	:	5986

```

: Routine Size: 15 words, Routine Base: $CODE$ + 6322
: Maximum stack depth per invocation: 0 words

```

014

ZRQADM
V01.6

RDRX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0178
Page 178
(58)

```

: 6014 1 global routine IN IODQ (index) : novalue
: 6015 1
: 6016 1 !
: 6017 1 ! THIS ROUTINE INSERTS A RETURN PACKET INDEX INTO THE I/O DONE QUEUE, AND
: 6018 1 ! UPDATES THE IODQ_IN POINTER.
: 6019 1 !
: 6020 1
: 6021 1 IF ((.IODQ_IN + 1) eq1 .IODQ_OUT) or
: 6022 2 (.IODQ_IN - (IODQ_LEN - 1) eq1 .IODQ_OUT)
: 6023 1 then
: 6024 1 return
: 6025 1 else
: 6026 2 begin
: 6027 2 IODQ [.IODQ_IN] = .index; ! LOAD INDEX INTO QUEUE
: 6028 2 IODQ_IN = .IODQ_IN + 1; ! ADVANCE "IN" POINTER
: 6029 2
: 6030 2 IF .IODQ_IN geq1 IODQ_LEN ! IF BEYOND END OF QUEUE
: 6031 2 then
: 6032 2 IODQ_IN = 0; ! CYCLE BACK TO BEGINNING OF QUEUE
: 6033 2
: 6034 1 end; ! IF IODQ IS NOT FULL

```

		.SBTIL	IN.IODQ GLOBAL ROUTINES		
000000	010146		IN.IODQ::		
000002	013701	000000G	MOV R1, (SP)	:	6014
000006	010100		MOV IODQ_IN, R1	:	6021
000010	005700		MOV R1, R0		
000012	020037	000000G	INC R0		
000016	001421		CMP R0, IODQ_OUT		
000020	010100		BEQ 1\$		
000022	162700	0000007	MOV R1, R0	:	6022
000026	020037	000000G	SUB #7, R0		
000032	001413		CMP R0, IODQ_OUT		
000034	116661	0000004 000000G	BEQ 1\$	:	6024
000042	005237	000000G	MOVH 4(SP), IODQ(R1)	:	INDEX, *
000046	023727	000000G 000010	INC IODQ_IN	:	6027
000054	103402		CMP IODQ_IN, #10	:	6028
000056	005037	000000G	BLO 1\$	:	6030
000062	012601		CLR IODQ_IN	:	6031
000064	000207		1\$: MOV (SP)+, R1	:	6014
			RTS PC	:	

; Routine Size: 27 words, Routine Base: \$CODE\$ + 6360
; Maximum stack depth per invocation: 2 words

K14

ZRQAM2
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0 5/79
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0179
Page 179
(59)

```

: 6035 1
: 6036 1
: 6037 1
: 6038 1
: 6039 1
: 6040 1
: 6041 1
: 6042 1
: 6043 1
: 6044 1
: 6045 1
: 6046 1
: 6047 1
: 6048 1
: 6049 2
: 6050 2
: 6051 2
: 6052 2
: 6053 2
: 6054 2
: 6055 2
: 6056 2
: 6057 2
: 6058 3
: 6059 3
: 6060 3
: 6061 3
: 6062 3
: 6063 2
: 6064 1

global routine DROP_CTLR (CTLR, REASON) : novalue =

!
! THIS ROUTINE DROPS ALL UNITS ASSOCIATED WITH THE CONTROLLER DESIGNATED
! BY "CTLR". THE REASON FOR DROPPING THE DEVICE IS LOADED INTO THE DUR
! VECTOR FOR EACH ATTACHED UNIT. THIS DATA IS THEN USED BY THE DROP UNIT
! SECTION.
!
begin
  local
    UNIT;
  incr N from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
    ! FOR EACH UNIT
    IF (.CST [CTLR, N + OF_DATA, D PRES] eq1 PRESENT
      ! IF CONFIGURED
      begin
        UNIT = .CST [CTLR, N + OF_DATA, D_UNIT];
        ! DRS UNIT NUMBER
        DUR [UNIT] = .REASON;
        ! DROP REASON
        DODU (.UNIT);
        ! DROP UNIT
      end;
    end;
end;

```

			SBITL	DROP_CTLR GLOBAL ROUTINES		
000000	004137	000000G		DROP_CTLR:		
				JSR	R1, \$SAVE 5	
000004	016646	000014		MOV	14(SP), (SP)	: CTLR, *
000010	012746	000053		MOV	#53, (SP)	
000014	004757	000000G		JSR	PC, BL \$MUL	
000020	010003			MOV	R0, R2	
000022	012702	000003		MOV	#5, R2	: *, N
000026	010300		1\$:	MOV	R3, R0	
000030	060200			ADD	R2, R0	: N, *
000032	006300			ASL	R0	
000034	032760	040000 000000G		BIT	#40000, CST(R0)	
000042	001412			BEQ	2\$	
000044	016001	000000G		MOV	CST(R0), R1	: *, UNIT
000050	000301			SWAB	R1	: UNIT
000052	042701	177760		BIC	#177760, R1	: *, UNIT
000056	116661	000016 000000G		MOVH	16(SP), DUR(R1)	: REASON, *(UNIT)
000064	010100			MOV	R1, R0	: UNIT, *
000066	104451			TRAP	51	
000070	062702	000012	2\$:	ADD	#12, R2	: *, N
000074	020227	000041		CMP	R2, #41	: N, *

1 14

ZRQAND
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1155 10 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0180
Page 180
(59)

000100 003752
000102 022626
000104 000207

BLF 1\$
CMP (SP)+,(SP)+
RTS PC

;
;

6049
6037

; Routine Size: 35 words, Routine Base: \$CODE\$ + 6446
; Maximum stack depth per invocation: 8 words

; 6065 1
; 6066 1

M14

ZRQAMP
V01.6

RD/RX EXERCISER
GLOBAL ROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0181
Page 181
(60)

```

: 6067 1 global routine DRV_CTLERR (CTLR) : novalue =
: 6068 1
: 6069 1 !
: 6070 1 ! THIS ROUTINE IS CALLED BY DRV_TIMCHK AND FATAL_ERROR WHENEVER AN
: 6071 1 ! UNRECOVERABLE CONTROLLER ERROR HAS BEEN DETECTED. ITS PURPOSE IS TO
: 6072 1 ! CLEAN UP ALL CONTROLLER-RELATED DATA IN THE "DRIVER" PORTION OF THE
: 6073 1 ! PROGRAM. THIS INCLUDES MARKING THE CONTROLLER OFFLINE, CLEARING THE
: 6074 1 ! C-RING COUNT, AND DEALLOCATING MSCP PACKETS DESCRIBED IN THE RESPONSE
: 6075 1 ! RING.
: 6076 1 !
: 6077 1 ! INPUTS:
: 6078 1 ! CTLR - DYING CONTROLLER NUMBER
: 6079 1 !-
: 6080 1
: 6081 2 begin
: 6082 2
: 6083 2 local
: 6084 2 D_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS); ! CONTROLLER'S DCT ADDRESS
: 6085 2
: 6086 2 D_ADDR = DCT + (.CTLR * DCT_LEN * 2); ! GET CONTROLLER'S DCT ADDR
: 6087 2 D_ADDR [WORD0] = OFFLINE; ! MARK DCT OFFLINE AND CLEAR CRING_CNT
: 6088 2 PUTA_PKT (.CTLR); ! RELEASE ALL PACKETS ALLOCATED TO CONTROLLER
: 6089 2 DROP_CTLR (.CTLR, DU_CFATAL); ! DROP ALL UNITS ON THE CONTROLLER
: 6090 1 end; ! ROUTINE DRV_CTLERR

```

		.SBTTL	DRV_CTLERR GLOBAL ROUTINES		
000000	010146	DRV_CTLERR:			
000002	016601	000004	MOV R1, -(SP)	:	6067
000006	010146		MOV 4(SP), R1	:	6086
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)	:	6087
000030	004737	005760'	MOV R1, (SP)	:	6088
000034	010116		JSR PC, PUTA_PKT		
000036	012746	000006	MOV R1, (SP)	:	6089
000042	004737	006446'	MOV #6, (SP)		
000046	062706	000006	JSR PC, DROP_CTLR		
000052	012601		ADD #6, SP	:	6081
000054	000207		MOV (SP)+, R1	:	606
			RTS PC		

; Routine Size: 25 words, Routine Base: \$CODE\$ + 6554
; Maximum stack depth per invocation: 5 words

N14

PRUAM,
VOL. 6

RD/RX EXERCISER
GLOBAL ROUTINES

```
11-Apr-1984 11:56:01
11-Apr-1984 11:45:02
```

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQAU0.BL1;6

SEQ 0132
Page 182
(61)

```

6091 1      global routine SEND (index) =
6092 1
6093 1      !+
6094 1      !
6095 1      !     IF THE CURRENT RDRX IS ONLINE AND ITS CRING IS NOT FULL, THEN THIS
6096 1      !     ROUTINE "SENDS" A COMMAND TO THE RDRX BY LOADING THE PACKET
6097 1      !     DESCRIPTOR OF AN MSCP PACKET INTO THE COMMAND RING AND READING THE
6098 1      !     DEVICE'S IP REGISTER. IF THE
6099 1      !     CURRENT RDRX IS NOT ONLINE, THEN A FAILURE INDICATION IS RETURNED TO
6100 1      !     THE CALLER, AND NO ACTION IS TAKEN.
6101 1      !
6102 1      !     INPUTS:
6103 1      !           INDEX - INDEX OF MSCP PACKET CONTAINING THE COMMAND TO
6104 1      !           BE SENT
6105 1      !
6106 1      !     IMPLICIT INPUTS:
6107 1      !           CCTRL - CURRENT CONTROLLER NUMBER
6108 1      !           DCT_ADDR - ADDRESS OF CURRENT CONTROLLER'S DCT
6109 1      !-
6110 2      begin
6111 2
6112 2      local
6113 2          SLOT_ADDR,
6114 2          TEMP : word,
6115 2          CUR_PRIORITY : word;
6116 2
6117 2      if ((DCT_ADDR [CRING_CNT] lssu CRING_LEN) and
6118 3          ((DCT_ADDR [STAT] eql ONLINE) or
6119 3          (.MSCP_PKT [.index, OPCODE] eql OP_SCC))
6120 2      then
6121 2
6122 4          if (not ((.MSCP_PKT [.index, OPCODE] eql OP_ACC) or (.MSCP_PKT [.index, OPCODE] eql OP_UNL) or
6123 4          (.MSCP_PKT [.index, OPCODE] eql OP_RD) or (.MSCP_PKT [.index, OPCODE] eql OP_SCC) or
6124 4          (.MSCP_PKT [.INDEX, OPCODE] EQL OP_SDD) OR
6125 4          (.MSCP_PKT [.INDEX, OPCODE] EQL OP_RCD) OR
6126 4          (.MSCP_PKT [.INDEX, OPCODE] EQL OP_GDS) OR
6127 4          (.MSCP_PKT [.INDEX, OPCODE] EQL OP_FLP) OR
6128 4          (.MSCP_PKT [.INDEX, OPCODE] EQL OP_ABT) OR
6129 4          (.MSCP_PKT [.INDEX, OPCODE] EQL OP_LSP) OR
6130 3          (.MSCP_PKT [.index, OPCODE] eql OP_WRT)))
6131 2      then
6132 3          begin
6133 3              PRINTF (DBM107, .MSCP_PKT [.index, OPCODE]);
6134 3              return FAILURE;
6135 3          end
6136 2      else
6137 3          begin
6138 3
6139 3              do
6140 4                  BREAK
6141 4              until ((.MSCP_PKT [.index, CMD_TYPE] eql IMM_CMD) and
6142 3                  (.CREDIT_BAL gequ 1)) or
6143 3                  (.CREDIT_BAL gtru 1);

```


1111

ZRQAMP
VOL. 1

RD RX EXERCISE
GLOBAL ROUTINES

11 Apr-1984 11:56:01
11 Apr-1984 11:45:02

VAX 11 B1144 16 V4.0 579
DISKUSERP: [POWER5]ZRQADO.B1 116

SEQ 0184
Page 184
(61)

```

: 6144 3
: 6145 3
: 6146 3
: 6147 3
: 6148 3
: 6149 3
: 6150 3
: 6151 3
: 6152 3
: 6153 3
: 6154 3
: 6155 3
: 6156 3
: 6157 3
: 6158 3
: 6159 3
: 6160 3
: 6161 3
: 6162 3
: 6163 3
: 6164 3
: 6165 3
: 6166 3
: 6167 3
: 6168 3
: 6169 3
: 6170 3
: 6171 3
: 6172 3
: 6173 3
: 6174 4
: 6175 3
: 6176 3
: 6177 3
: 6178 3
: 6179 3
: 6180 3
: 6181 3
: 6182 3
: 6183 3
: 6184 1

MSCP PKT [.index, CRN LO] = (CRN LOW + .CRN LOW + 1);
IF .CRN LOW eq 0
then
    CRN HIGH = .CRN HIGH + 1;
MSCP PKT [.index, CRN HI] = .CRN HIGH;
SLOT ADDR = .DCT ADDR [CR NEXT];
do
    BREAK
until ((.SLOT ADDR + 2) and ED OWN) eq 0;
GETPRI (CR PRIORITY);
GETPRI (PRIO4);
.SLOT ADDR = .MSCP PKT [.index, PKT LO];
SLOT ADDR = .SLOT ADDR + 2;
.SLOT ADDR = .MSCP PKT [.index, PKT HI];
.SLOT ADDR = .SLOT ADDR and (not (ED FLAG));
.SLOT ADDR = .SLOT ADDR or ED OWN;
SLOT ADDR = .SLOT ADDR + 2;
IF .SLOT ADDR gtr .DCT ADDR [CR END]
then
    SLOT ADDR = .DCT ADDR [CR BEG];
DCT ADDR [CR NEXT] = .SLOT ADDR;
DCT ADDR [CRING CNT] = .DCT ADDR [CRING CNT] + 1;
IF (.MSCP PKT [.index, CONVID] EQ CID MSCP)
then (CREDIT BAL = .CREDIT BAL - 1);
TEMP = .RD RX ADDR [RCIP, RC ALL];
GETPRI (CR PRIORITY);
return SUCCESS;
end
else
    return FAILURE;
end;

```

```

: ASSIGN CMD REF NUM
: CMD REF NUM (HIGH ORDER)
: ADDR OF NEXT COMMAND SLOT
: WAIT TILL NEXT SLOT HOST OWNED
: NO INTERRUPTS WHILE POINTERS UPDATED
: LOAD BUFF DESC (LO) INTO COMMAND
: ADVANCE TO NEXT WORD
: LOAD BUFF DESC (HI) INTO COMMAND SLOT
: CLEAR INTERRUPT FLAG IN CASE SET
: GIVE OWNERSHIP TO CONTROLLER
: ADVANCE TO NEXT COMMAND SLOT
: IF BEYOND END OF CRING
: CYCLE BACK TO BEGINNING
: RESTORE CR NEXT POINTER IN DCT
: INCR # OF COMMANDS IN CRING
: IF MSCP COMMAND
: DECR CREDIT BALANCE
: HEAD IP TO FORCE PORT TO POLL
: LOWER PRIORITY
: IF DEVICE IS NOT ONLINE
: ROUTINE SEND

```

```

000000 004147 00000000
000004 005746
000006 127727 00000000 000004
000014 105106
000016 005727 00000000
000020 100415
000024 016746 000014
000028 017746 000106
000034 004757 00000000

SEND:
LBITL SEND GLOBAL ROUTINES
JBR R1,SAVE
LST
CMPI DCT,ADDR,24
BNEZ
LST DCT,ADDR
JBR 11
MOV 14,SP,1,DP
MOV 2108,1,DP
JBR PC,BL 1M0

```

INDEX *

C | 5

PROGRAM
VOL 1.6

RD RX EXERCISER
GLOBAL ROUTINES

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX 11 B1146 16 V4.0 5/79
DISKUSER2:[POWER5]ZRQADO.B1116

SEQ 0184
Page 184
(61)

000040	022626			CMP	(SP), (SP)		
000042	126027	0000226	000004	CMPB	MSCP, PKT+22(R), 04		
000050	001167			BNE	103		
000052	016646	000014		MOV	14(SP), (SP)	:	INDEX, *
000056	012746	000106		MOV	0106, (SP)		
000062	004757	0000006		JSR	PC, BL \$MUR		
000066	010002			MOV	RO, R2		
000070	022626			CMP	(SP), (SP)		
000072	005000			CLR	RO		
000074	156200	0000226		BISH	MSCP, PKT+22(R), RO		
000100	020027	000020		CMP	RO, 020		
000104	001445			BEQ	33		
000106	020027	000011		CMP	RO, 011		
000112	001442			BEQ	33		
000114	020027	000041		CMP	RO, 041	:	6123
000120	001437			BEQ	33		
000122	020027	000004		CMP	RO, 04		
000126	001434			BEQ	33		
000130	020027	000005		CMP	RO, 05	:	6125
000134	001431			BEQ	33		
000136	020027	000001		CMP	RO, 01	:	6126
000142	001426			BEQ	33	:	6127
000144	020027	000005		CMP	RO, 05	:	6128
000150	001423			BEQ	33	:	6129
000152	020027	000006		CMP	RO, 06	:	6130
000156	001420			BEQ	33	:	6131
000160	020027	000002		CMP	RO, 02	:	6132
000164	001415			BEQ	33	:	6133
000166	020027	000042		CMP	RO, 042	:	6134
000172	001412			BEQ	33	:	6135
000174	010046			MOV	RO, (SP)	:	6136
000176	012746	0000006		MOV	00B107, (SP)		
000202	012746	000002		MOV	02, (SP)		
000206	010600			MOV	SP, RO	:	INDEX, *
000210	104417			TRAP	17		
000212	022706	000006		ADD	06, SP	:	6137
000216	000504			BR	103	:	6138
000220	104422			TRAP	02	:	6139
000222	105762	0000046		TSTB	MSCP, PKT+4(R), 0	:	6140
000226	001003			BNE	43	:	6141
000230	005757	0000006		TST	CREDIT, BAL	:	6142
000234	001004			BNE	53	:	6143
000236	023727	0000006	000001	CMP	CREDIT, BAL, 01	:	6144
000244	101765			BLON	33	:	6145
000246	015700	0000006		MOV	CRN, LOW, RO	:	6146
000252	005200			INC	RO		
000254	010037	0000006		MOV	RO, CRN, LOW		
000260	010062	0000126		MOV	RO, MSCP, PKT+12(R), 0		
000264	001002			BNE	63	:	6147
000266	005237	0000006		INC	CRN, HIGH	:	6148
000272	015762	0000006	0000146	MOV	CRN, HIGH, MSCP, PKT+14(R), 0	:	6149
000276	015700	0000006		MOV	DET, ADDR, RO	:	6150
000304	016001	000020		MOV	20(R), R1	:	6151
						:	6152

[D] 5

PROGRAM	RD RX EXERCISER	11 Apr 1984 11:56:01	VAX 11 B119% 1% V4.0 5/9	SEQ 0185
VO1.6	GLOBAL ROUTINES	11 Apr 1984 11:45:02	DISK1USER2:[POWERS]ZRGADD.BU 1;6	Page 185
				(61)

000310	104422		75:	TRAP	22			
000312	052761	100000 000002		BTF	0 100000,2(R1)			6154
000320	001573			BNE	73		*,*(SLOT,ADDR)	6156
000332	104440			TRAP	40			6158
000334	010003			MOV	R0,R3		*,CUR,PRIORITY	
000336	012700	000200		MOV	0200,R0			6159
000338	104441			TRAP	41			
000334	016221	0000006		MOV	MSCP,PKT(R2),(R1)+		*,SLOT,ADDR	6161
000340	016211	0000026		MOV	MSCP,PKT,2(R2),(R1)		*,SLOT,ADDR	6162
000344	042711	040000		RSC	040000,(R1)		*,SLOT,ADDR	6164
000350	052721	100000		RIS	0100000,(R1)+		*,SLOT,ADDR	6165
000354	013700	0000006		MOV	DCT,ADDR,R0			6168
000360	020160	00001		CMR	R1,12(R0)		SLOT,ADDR,*	
000364	101402			BLOS	83			
000366	016001	000010		MOV	10(R0),R1		*,SLOT,ADDR	6170
000372	010160	000020	83:	MOV	R1,20(R0)		SLOT,ADDR,*	6171
000376	105210			INCH	(R0)			6173
000400	105762	0000116		TEST	MSCP,PKT+11(R2)			6174
000404	001002			BNE	93			
000406	005337	0000006		DEC	CREDIT,B3			6175
000412	017716	0000006	93:	MOV	0RDRX,ADDR,(SP)		*,RC,REF	6176
000416	010300			MOV	R3,R0		CUR,PRIORITY,*	6177
000420	104441			TRAP	41			
000422	012700	000001		MOV	01,R0			6178
000426	000401			BR	113			6179
000430	005000		103:	CLR	R0			
000432	005726		113:	TEST	(SP),			
000434	00020			RTS	PC			6091

: Routine Size: 143 words, Routine Base: 3CODES + 6632
: Maximum stack depth per invocation: 10 words

111)

BRAND,
VOL. 6

ROAD REX EXERCISER GLOBAL ROUTINES

11 Apr-1984 11:56:01
11 Apr-1984 11:45:02

VAX 11 B1160 16 V4.6 579
DISK\$USER2: (POWERS)ZRRQADO.B1.116

SEQ 0186
Page 186
(62)

```

: 6185 1      global routine WAIT : novalue -
: 6186 1
: 6187 1      !,
: 6188 1      !
: 6189 1      !      THE PURPOSE OF THIS ROUTINE IS TO KILL TIME UNTIL AN RDR: INTERRUPT
: 6190 1      !      RESULTS IN A RETURN PACKET INDEX BEING DEPOSITED INTO THE I/O DONE
: 6191 1      !      QUEUE (IODQ).
: 6192 1      !
: 6193 1
: 6194 1      do
: 6195 1          BREAK                                ! BREAK FOR ACT
: 6196 1      until .IODQ IN neq .IODQ OUT;

```

WAIT::		.SRITI WAIT GLOBAL ROUTINES	
000000	104420	10:	TRAP 20
000000			
000000	003757 000000G 000000G	CMF	10DQ.IN,10DQ.OUT
000010	001723	BEQ	10
000010	000007	BIS	PC

```

; Routine Size: 6 words,      Routine Base: $CODE$ + 7270
; Maximum stack depth per invocation: 2 words

```

i 6196 1

1 Apr 1984 11:56:01
1 Apr 1984 11:45:02

VAX-11 B1155 16 V4.0 579
DISK1USER2:[POWER5]ZRGADO.BL116

SEG 0187
Page 187
(65)

ROCKY EXERCISER GLOBAL ROUTINES

```

6197 1
6198 1 GLOBAL ROUTINE MODULAS (LO LIMIT, HI LIMIT) *
6199 1
6200 1 !* THE PURPOSE OF THIS ROUTINE IS TO GET A RANDOM NUMBER BETWEEN
6201 1 ! THE LOW AND HIGH LIMITS. THIS SHOULD WORK FOR A 16 BIT WORD.
6202 1 ! THE "MOD" FUNC ONLY WORKS ON 15 BITS.
6203 1
6204 2 BEGIN
6205 2 OWN X : WORD; !VARIABLE FOR RANDOM WD TABLE
6206 2 LOCAL ANSWER : UNSIGNED WORD; !FINAL ANSWER
6207 2 SAVESZ : UNSIGNED WORD; !SAVES SIZE OF WINDOW
6208 2 SIZE : UNSIGNED WORD; !SIZE OF WINDOW
6209 2
6210 2
6211 2 X = .X + 1;
6212 2 IF .X GEQ RDM_LEN
6213 2 THEN X = 0; !KEEP ROTATING RANDOM NUMBERS USED
6214 2
6215 2
6216 2 SIZE = .HI LIMIT - .LO LIMIT;
6217 2 SAVESZ = .HI LIMIT - .LO LIMIT;
6218 3 IF (.SIZE LEQU #0'0777777) !IF BIT 15 NOT SET
6219 3 THEN ANSWER = ((.RANDOM [.X] AND #0'0777777) MOD (.SIZE + 1))
6220 3 !ONLY 15 BIT WD, SO TAKE RANDOM SAMPLE
6221 3 ELSE !16 BIT WD
6222 3 BEGIN
6223 3 SIZE = .SIZE + 1; !MAKES SIZE A 15 BIT LENGTH, OR DIV BY 2
6224 3 ANSWER = (.RANDOM [.X] AND #0'0777777) MOD (.SIZE + 1);
6225 3 !GIVES 15 BIT RANDOM NUMBER
6226 3 ANSWER = .ANSWER + 1; !BUILD UP TO REGULAR SIZE
6227 3 ANSWER = .ANSWER + (.RANDOM [.X + 1] AND 1);
6228 3 !RANDOMLY FILL BIT 0
6229 3 IF (.ANSWER GTU SAVESZ) !ITS POSSIBLE TO BE 1 LARGER THAN SIZE
6230 3 THEN ANSWER = .SAVESZ; !SO CHECK.
6231 3 END;
6232 2 RETURN .ANSWER;
6233 1 END; !END MODULAS ROUTINE

```

000304 X; BLACK 1

MODULAS:			MODULAS:	GLOBAL ROUTINES
000000	004157	000000G		
000004	005746			
000006	005257	007504		
000012	005777	007504	000020	
000020	005402			
000024	005037	007504		
000026	016500	000012		
000032	016500	000014		
000036	010001			
000040	010016			

7-1-25

111

11

1955

P 44

620

615

ZRRQAM2		RD/RX EXERCISER		11-Apr-1984 11:56:01		VAX 11-81196 16 V4.0 5/9		SEQ 0188
V01.6		GLOBAL ROUTINES		11-Apr-1984 11:45:02		DISK\$USER2:[POWERS]ZRRQADO.B1 1:6		Page 188
								(63)
000042	013700	007304		MOV	X,R0	:		6218
000044	006300			ASL	R0	:		
000050	020127	077777		CMP	R1,077777	:	SIZE,*	6217
000054	101011			BHI	2\$	:		
000056	016046	000000G		MOV	RANDOM(R0),-(SP)	:		6218
000062	042716	100000		BIC	0100000,(SP)	:		
000066	010146			MOV	R1, -(SP)	:	SIZE,*	
000070	005216			INC	(SP)	:		
000072	004737	000000G		JSR	PC,BL\$MOD	:		
000076	000431			BR	3\$	:		
000100	006201		2\$:	ASR	R1	:	SIZE	6217
000102	016046	000000G		MOV	RANDOM(R0),-(SP)	:		6217
000106	042716	100000		BIC	0100000,(SP)	:		6213
000112	010146			MOV	R1, -(SP)	:	SIZE,*	
000114	005216			INC	(SP)	:		
000116	004737	000000G		JSR	PC,BL\$MOD	:		
000122	006300			ASL	R0	:	ANSWER	6215
000124	013701	007304		MOV	X,R1	:		6226
000130	006301			ASL	R1	:		
000132	116102	000002G		MOV8	RANDOM(R1),R2	:		
000136	042702	177776		BIC	0177776,R2	:		
000142	060200			ADD	R2,R0	:	*,ANSWER	
000144	012701	000004		MOV	04,R1	:		6228
000150	060601			ADD	SP,R1	:	SAVE\$2,*	
000152	070001			CMP	R0,R1	:	ANSWER,*	
000154	101402			BLOS	3\$	:		
000156	016600	000004		MOV	4(SP),R0	:	SAVE\$2,ANSWER	6219
000162	062706	000006	3\$:	ADD	01,SP	:		
000166	000207			RTS	PC	:		6198

; Routine size: 60 words, Routine Base: \$CODE\$ + 7306
; Maximum stack depth per invocation: 7 words

FILE

ZRQAM,
V01.6

RDRX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B1153 16 V4.0 5/79
DISK\$USER2:[POWER5]ZRQADO.BL1;6

SEQ 0189
Page 189
(64)

```

: 6233 1 *sbttl 'ERROR MESSAGE SUBROUTINES'
: 6234 1
: 6235 1 routine EMS_SA : novalue
: 6236 1
: 6237 1 !
: 6238 1 ! THIS ROUTINE PRINTS (EXTENDED) THE GLOBAL DATUM "SA_REG" WHICH CONTAINS
: 6239 1 ! THE CONTENTS OF THE SA REGISTER.
: 6240 1 !
: 6241 1
: 6242 1 begin
: 6243 1
: 6244 1 if (.SA_REG .eq1 %0'17777') ! IF CONTROLLER TIME-OUT
: 6245 1 then
: 6246 1 begin
: 6247 1 PRINTX (CRF);
: 6248 1 PRINTX (ASTERISK);
: 6249 1 PRINTX (.CNTR_ERR (0));
: 6250 1 end
: 6251 1 else
: 6252 1
: 6253 1 if ((.SA_REG .and %0'003777') .leq1 22) ! IF GENERIC CONTROLLER ERROR
: 6254 1 then
: 6255 1 begin
: 6256 1 PRINTX (CRF);
: 6257 1 PRINTX (ASTERISK);
: 6258 1 PRINTX (.CNTR_ERR (.SA_REG .and %0'003777'));
: 6259 1 end
: 6260 1 else
: 6261 1
: 6262 1 if ((.SA_REG .and %0'003777') - 400) .leq1 6 ! IF RDRX SPECIFIC CONTROLLER ERROR
: 6263 1 then
: 6264 1 begin
: 6265 1 PRINTX (CRF);
: 6266 1 PRINTX (ASTERISK);
: 6267 1 PRINTX (.RDRX_ERR ((.SA_REG .and %0'003777') - 400));
: 6268 1 end
: 6269 1 else
: 6270 1 PRINTX (EX_SA, .SA_REG); ! JUST PRINT CONTENTS OF SA
: 6271 1
: 6272 1 EMS_TIM (); ! TIME
: 6273 1 end;

```

000000 010146
000001 013701 0000006
000002 020127 177777
000013 001023
000014 012746 0000006
000020 012746 0000001
000024 010600
000027 104415
000030 012716 0000006

```

*SBTTL EMS_SA ERROR MESSAGE SUBROUTINES
EMS_SA: MOV R1, (SP)
MOV SA_REG, R1
CMP R1, 0
BNE 13
MOV @CRF, (SP)
MOV @1, (SP)
MOV SP, R0
TRAP 15
MOV @ASTERISK, (SP)

```

000000
000001
000002
000013
000014
000020
000024
000027
000030

PROGRAM		RD/RX EXERCISER		11 Apr-1984 11:56:01		VAX 11 B115s 16 V4.0 579		SEQ 0190		PROGRAM	
VC1.6		ERROR MESSAGE SUBROUTINES		11-Apr-1984 11:45:02		DISK\$USER2:[POWERS]ZRGADO.B11;6		Page 190		VC1.6	
								(64)			
000034	012746	000001	MOV	R1, (SP)							
000040	010600		MOV	SP, R0							
000042	104415		TRAP	15							
000044	015716	000000G	MOV	CNTR, ERR, (SP)							
000050	012746	000001	MOV	R1, (SP)						6249	
000054	010600		MOV	SP, R0							
000056	104415		TRAP	15							
000060	000475		BR	35							
000062	010100		MOV	R1, R0						6249	
000064	042700	174000	BIC	R174000, R0						6253	
000070	020027	000076	CMP	R0, R6							
000074	101030		BHI	25							
000076	012746	000000G	MOV	OCRLF, (SP)						6256	
000102	012746	000001	MOV	R1, (SP)							
000106	010600		MOV	SP, R0							
000110	104415		TRAP	15							
000112	012716	000000G	MOV	ASTERISK, (SP)							
000116	012746	000001	MOV	R1, (SP)						6257	
000122	010600		MOV	SP, R0							
000124	104415		TRAP	15							
000126	015700	000000G	MOV	SA, REG, R0							
000132	042700	174000	BIC	R174000, R0						6258	
000136	006500		ASL	R0							
000140	016016	000000G	MOV	CNTR, ERR(R0), (SP)							
000144	012746	000001	MOV	R1, (SP)							
000150	010600		MOV	SP, R0							
000152	104415		TRAP	15							
000154	000437		BR	35							
000156	010100		MOV	R1, R0						6259	
000160	042700	174000	BIC	R174000, R0						6260	
000164	162700	000620	SOB	R620, R0							
000170	020027	000006	CMP	R0, R6							
000174	101031		BHI	45							
000176	012746	000000G	MOV	OCRLF, (SP)						6265	
000202	012746	000001	MOV	R1, (SP)							
000206	010600		MOV	SP, R0							
000210	104415		TRAP	15							
000212	012716	000000G	MOV	ASTERISK, (SP)							
000216	012746	000001	MOV	R1, (SP)						6266	
000222	010600		MOV	SP, R0							
000224	104415		TRAP	15							
000226	015700	000000G	MOV	SA, REG, R0							
000232	042700	174000	BIC	R174000, R0						6267	
000236	006500		ASL	R0							
000240	016016	176340G	MOV	RDRX, ERR 1440(R0), (SP)							
000244	012746	000001	MOV	R1, (SP)							
000250	010600		MOV	SP, R0							
000252	104415		TRAP	15							
000254	005726		INT	(SP),						6268	
000256	000407		BR	55						6269	
000260	010146		MOV	R1, (SP)						6270	
000262	012746	000000G	MOV	DEL, SA, (SP)							
000266	012746	000002	MOV	R2, (SP)							

115

PROGRAM		RD-RX EXERCISER		11-Apr-1984 11:56:01		VAX 11 B11ss 16 V4.0 5/9	
VOL.6		ERROR MESSAGE SUBROUTINES		11-Apr-1984 11:45:02		DISK\$USER2:[POWERS]ZRGADO.BL1:6	
000272	010600			MOV	SP,RO		
000274	104415			TRAP	15		: SP,*
000276	004737	000000V	5\$:	JSR	PC,EMS,TIM		
000302	062706	000006		ADD	#6,SP		:
000306	012601			MOV	(SP)+,R1		:
000310	000207			RTS	PC		:

```

; Routine Size: 101 words,      Routine Base: $CODE$ + 7A76
; Maximum stack depth per invocation: 7 words

```

SEQ 0191
Page 191
(64)

200
VO

[illegible]

10

K11

RDQAM,
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B11ss-16 V4.0 5/9
DISK\$USER2:[POWERS]ZRDQAD0.BL1;6

SEQ 0192
Page 192
(65)

```

: 6274 1 routine EMS_SBC : novalue "
: 6275 1
: 6276 1
: 6277 1
: 6278 1
: 6279 1
: 6280 1
: 6281 1
: 6282 1
: 6283 2
: 6284 2
: 6285 2
: 6286 2
: 6287 3
: 6288 3
: 6289 3
: 6290 3
: 6291 3
: 6292 3
: 6293 3
: 6294 3
: 6295 3
: 6296 3
: 6297 3
: 6298 3
: 6299 3
: 6300 3
: 6301 3
: 6302 3
: 6303 3
: 6304 3
: 6305 3
: 6306 3
: 6307 3
: 6308 3
: 6309 3
: 6310 3
: 6311 3
: 6312 3
: 6313 3
: 6314 3
: 6315 3
: 6316 3
: 6317 3
: 6318 3
: 6319 3
: 6320 3
: 6321 3
: 6322 3
: 6323 3
: 6324 3
: 6325 3
: 6326 3

THIS ROUTINE PRINTS (EXTENDED) THE GLOBAL DATUM "SB_CODE" (SUB CODE) IF
EITHER THE STATUS CODE (ST_CODE) OR THE SUB_CODE IS NON ZERO. (A
NON-ZERO SUB_CODE ALWAYS HAS SIGNIFICANCE, WHEREAS A ZERO SUB_CODE ONLY
HAS MEANING WITH A NON-ZERO STATUS CODE).

begin
  if (.ST_CODE or .SB_CODE) neq 0
  then
    begin
      PRINTX (EX_SBC);
      case .ST_CODE from ST_SUC to ST_DRV of
        set
          [ST_SUC]:
            if .SB_CODE lequ 16
            then
              PRINTX (,TBL_SUC [ ,SB_CODE]);
            [ST_CMD]:
              PRINTX (EX_SBC, .SB_CODE / 8);
            [ST_ABO]:
              ;
            [ST_OFL]:
              if .SB_CODE lequ 8
              then
                PRINTX (,TBL_OFL [ ,SB_CODE]);
            [ST_AVL]:
              ;
            [ST_MFE]:
              if .SB_CODE lequ 10
              then
                PRINTX (,TBL_MFE [ ,SB_CODE]);
            [ST_WPT]:
              if (.SB_CODE / 128) lequ 2
              then
                PRINTX (,TBL_WPT [( ,SB_CODE / 128)));
            [ST_CMP]:
              ;
            [ST_DAT]:
              if .SB_CODE lequ 16
              then
                PRINTX (,TBL_DAT [ ,SB_CODE]);
            [ST_HST]:
              if .SB_CODE lequ 4
              then
                PRINTX (,TBL_HST [ ,SB_CODE]);
            [ST_CNT]:
              if .SB_CODE lequ 4
              then

```


ZRDAMP
V01.6RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss 16 V4.0 579
DISK\$USER2:[POWERS]ZRDADO.BL1:6SEQ 0193
Page 193
(65)

```

: 6327 3          PRINTX (,TBL_CNT (,SB_CODE));
: 6328 3
: 6329 3          [ST_DRV]:      If ,SB_CODE lequ 8          ! DRIVE ERROR
: 6330 3          then
: 6331 3              PRINTX (,TBL_DRV (,SB_CODE));
: 6332 3
: 6333 3          [outrange]:  PRINTX (EX_SBO, ,SB_CODE);      ! JUST PRINT SUB-CODE IF NO MATCH
: 6334 3          tes;
: 6335 3
: 6336 2          end;
: 6337 2
: 6338 1          end;

```

```

000000 013700 000000G      .SBTTL  EMS.SBC ERROR MESSAGE SUBROUTINES
000004 053700 000000G      EMS.SBC: MOV      ST.CODE,RO          ;
000010 001001          BIS      SB.CODE,RO          ;
000012 000207          BNE      1$          ;
000014 012746 000000G      RTS      PC          ;
000020 012746 000001      1$:  MOV      @EX.SB, -(SP)          ;
000024 010600          MOV      @1, (SP)          ;
000026 104415          MOV      SP,RO          ; SP,+
000030 013700 000000G      TRAP      15          ;
000034 020027 000013      MOV      ST.CODE,RO          ;
000040 101003          CMP      RO,@13          ;
000042 006300          BHI      3$          ;
000044 066007 000000G      ASL      RO          ;
000050 013716 000000G      ADD      P,AAA(RO),PC          ; Case dispatch
000054 012746 000000G      3$:  MOV      SB.CODE,(SP)          ;
000060 012746 000002      MOV      @EX.SBO, -(SP)          ;
000064 010600          MOV      @2, (SP)          ;
000066 104415          MOV      SP,RO          ; SP,+
000070 022626          TRAP      15          ;
000072 000435          CMP      (SP)+,(SP)+          ;
000074 023727 000000G 000020 4$:  BR      6$          ;
000102 101165          CMP      SB.CODE,@20          ;
000104 013700 000000G      BHI      14$          ;
000110 006300          MOV      SB.CODE,RO          ;
000112 016016 000000G      ASL      RO          ;
000116 012746 000001      MOV      TBL,SUC(RO),(SP)          ;
000122 010600          MOV      @1, (SP)          ;
000124 104415          MOV      SP,RO          ; SP,+
000126 000565          TRAP      15          ;
000130 013716 000000G      BR      15$          ;
000134 012746 000010      5$:  MOV      SB.CODE,(SP)          ;
000140 004737 000000G      MOV      @10, (SP)          ;
000144 010016          JSR      PC,BL$DTV          ;
000146 012746 000000G      MOV      RO,(SP)          ;
000152 012746 000002      MOV      @EX.SBO, -(SP)          ;
000156 010600          MOV      @2, (SP)          ;
000160 104415          MOV      SP,RO          ; SP,+
000162 062706 000006      TRAP      15          ;
                        ADD      @6,SP

```

M15

ZRQAM2
V01.6RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6SEQ 0194
Page 194
(65)

000166	000546		6\$:	BR	16\$	:	
000170	023727	000000G 000010	7\$:	CMP	SB.CODE, #10	:	6290
000176	101142			BHI	16\$	:	6301
000200	013700	000000G		MOV	SB.CODE, R0	:	
000204	006300			ASL	R0		6303
000206	016016	000042'		MOV	TBL.OFL(R0), (SP)		
000212	012746	000001		MOV	#1, (SP)		
000216	010600			MOV	SP, R0	: SP, +	
000220	104415			TRAP	15		
000222	000527			BR	15\$		
000224	023727	000000G 000012	8\$:	CMP	SB.CODE, #12	:	6307
000232	101124			BHI	16\$		
000234	013700	000000G		MOV	SB.CODE, R0	:	6309
000240	006300			ASL	R0		
000242	016016	000064'		MOV	TBL.MFE(R0), (SP)		
000246	012746	000001		MOV	#1, (SP)		
000252	010600			MOV	SP, R0	: SP, +	
000254	104415			TRAP	15		
000256	000511			BR	15\$		
000260	013716	000000G	9\$:	MOV	SB.CODE, (SP)	:	6311
000264	012746	000200		MOV	#200, (SP)		
000270	004737	000000G		JSR	PC.BL\$DIV		
000274	005726			TSI	(SP)+		
000276	020027	000002		CMP	R0, #2		
000302	101100			BHI	16\$		
000304	006300			ASL	R0	:	6313
000306	016016	000112'		MOV	TBL.WPT(R0), (SP)		
000312	012746	000001		MOV	#1, (SP)		
000316	010600			MOV	SP, R0	: SP, +	
000320	104415			TRAP	15		
000322	000467			BR	15\$		
000324	023727	000000G 000017	10\$:	CMP	SB.CODE, #17	:	6317
000332	101064			BHI	16\$		
000334	013700	000000G		MOV	SB.CODE, R0	:	6319
000340	006300			ASL	R0		
000342	016016	000120'		MOV	TBL.DAT(R0), (SP)		
000346	012746	000001		MOV	#1, (SP)		
000352	010600			MOV	SP, R0	: SP, +	
000354	104415			TRAP	15		
000356	000451			BR	15\$		
000360	023727	000000G 000004	11\$:	CMP	SB.CODE, #4	:	6321
000366	101046			BHI	16\$		
000370	013700	000000G		MOV	SB.CODE, R0	:	6323
000374	006300			ASL	R0		
000376	016016	000160'		MOV	TBL.HST(R0), (SP)		
000402	012746	000001		MOV	#1, (SP)		
000406	010600			MOV	SP, R0	: SP, +	
000410	104415			TRAP	15		
000412	000433			BR	15\$		
000414	023727	000000G 000003	12\$:	CMP	SB.CODE, #3	:	6325
000422	101030			BHI	16\$		
000424	013700	000000G		MOV	SB.CODE, R0	:	6327
000430	006300			ASL	R0		

N15

ZRQAM2 RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 Bliss-16 V4.0-579 SEQ 0195
 V01.6 ERROR MESSAGE SUBROUTINES 11-Apr-1984 11:45:02 DISK\$USER2:[POWERS]ZRGADO.BL1;6 Page 195
 (65)

000432	016016	000172'	MOV	TBL.CNT(R0),(SP)		
000436	012746	000001	MOV	#1,(SP)		
000442	010600		MOV	SP,R0	; SP,*	
000444	104415		TRAP	15		
000446	000415		BR	15\$		
000450	023727	000000G 000010	13\$:	CMP	SB.CODE,#10	
000456	101012		14\$:	BHI	16\$	6329
000460	013700	000000G		MOV	SB.CODE,R0	
000464	006300			ASL	R0	6331
000466	016016	000202'		MOV	TBL.DRV(R0),(SP)	
000472	012746	000001		MOV	#1,(SP)	
000476	010600			MOV	SP,R0	; SP,*
000500	104415			TRAP	15	
000502	005726		15\$:	TST	(SP)+	
000504	022626		16\$:	CMP	(SP)+,(SP)+	
000506	000207			RTS	PC	6287 6274

; Routine Size: 164 words, Routine Base: \$CODE\$ + 10010
 ; Maximum stack depth per invocation: 7 words

000000			.PSECT	\$PLIT\$, R0, D		
		P.AAA:				
000000	000024	2\$:	.WORD	24	; CASE Table for EMS.SBC+0044	6290
000002	000060		.WORD	60	; [4\$]	
000004	000434		.WORD	434	; [5\$]	
000006	000120		.WORD	120	; [16\$]	
000010	000434		.WORD	434	; [7\$]	
000012	000154		.WORD	154	; [16\$]	
000014	000210		.WORD	210	; [8\$]	
000016	000434		.WORD	434	; [9\$]	
000020	000254		.WORD	254	; [16\$]	
000022	000310		.WORD	310	; [10\$]	
000024	000344		.WORD	344	; [11\$]	
000026	000400		.WORD	400	; [12\$]	
					; [13\$]	

316

PROGRAM
VOL. 6

RD RX EXERCISER
ERROR MESSAGE SUBROUTINES

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX 11 B1159 16 V4.0 5/79
DISK4USER2:[POWER5]ZRGADO.BU 1:6

SEQ 0196
Page 196
(66)

```

: 6.539 1 routine EMS_CMD : novalue -
: 6.540 1
: 6.541 1
: 6.542 1
: 6.543 1 THIS ROUTINE PRINTS (EXTENDED) THE OPCODE AND COMMAND MODIFIER (IF
: 6.544 1 PRESENT) OF THE CURRENT RETURN PACKET. THESE FIELDS ARE "TRANSLATED"
: 6.545 1 INTO ENGLISH TEXT RATHER THAN PRINTED AS RAW NUMBERS.
: 6.546 1
: 6.547 1 IMPLICIT INPUTS:
: 6.548 1 RP_ADDR ADDRESS OF THE CURRENT RETURN PACKET
: 6.549 1
: 6.550 1
: 6.551 1 begin
: 6.552 1 PRINTX (EX_CMD); ! "COMMAND:"
: 6.553 1
: 6.554 1 selectoneu (,RP_ADDR [ENCODE] and OP_MSK) of
: 6.555 1 set
: 6.556 1
: 6.557 1 [OP_ONL]: PRINTX (EX_ONL); ! ONLINE
: 6.558 1
: 6.559 1 [OP_ACC]: PRINTX (EX_ACC); ! ACCEPT
: 6.560 1
: 6.561 1 [OP_RD]: begin
: 6.562 1 PRINTX (EX_RD); ! READ
: 6.563 1
: 6.564 1 if ,RP_ADDR [CMODMOD] neq 0
: 6.565 1 then
: 6.566 1 PRINTX (EX_CMP); ! COMPARE
: 6.567 1
: 6.568 1 end;
: 6.569 1
: 6.570 1 [OP_WRT]: begin
: 6.571 1 PRINTX (EX_WRT); ! WRITE
: 6.572 1
: 6.573 1 if ,RP_ADDR [CMODMOD] neq 0
: 6.574 1 then
: 6.575 1 PRINTX (EX_CMP); ! COMPARE
: 6.576 1
: 6.577 1 end;
: 6.578 1
: 6.579 1 [otherwise]: PRINTX (EX_OP, ,RP_ADDR [ENCODE]); ! ENCODE VALUE IF NO MATCH
: 6.580 1 test;
: 6.581 1 end; ! ROUTINE EMS_CMD

```

010000

EMSG_CMD ERROR MESSAGE SUBROUTINES
PROJECT CODES: RD

000000 004157 0000000
000004 01742 0000000
000001 01742 0000001
000014 01742 0000000
000014 104415

EMSG_CMD: RDR
MOV R1, R17
MOV R1, R17
MOV R1, R17
TRAP 14

6351

715

ZRQADM VOL 1.6	RD RX EXERCISER ERROR MESSAGE SUBROUTINES	11 Apr 1984 11:56:01 11 Apr 1984 11:45:02	VAX 11 H1100 16 V4.0 5/9 DISK4USER2:[POWERS]ZRQADO.BU 1:6	SEG 0197 Page 197 (66)
000030 013702 000000G		MOV	RP,ADDR,RO	
000034 116201 000011		MOVH	14(RP),R1	
000038 042701 177600		BIC	0177600,R1	
00003A 070127 000011		CMP	R1,011	
000040 001007		BNE	15	6356
000042 012716 000000G		MOV	0EX,ONL,(SP)	
000046 012746 000001		MOV	01,(SP)	
000050 010600		MOV	SP,RO	
000054 104415		TRAP	15	
000056 000464		BR	55	
000060 020127 000020	15:	CMP	R1,020	
000064 001007		BNE	25	6358
000066 012716 000000G		MOV	0EX,ACC,(SP)	
000070 012746 000001		MOV	01,(SP)	
000074 010600		MOV	SP,RO	
000100 104415		TRAP	15	
000102 000452		BR	55	
000104 020127 000041	25:	CMP	R1,041	
000110 001027		BNE	55	6360
000112 012716 000000G		MOV	0EX,RO,(SP)	
000116 012746 000001		MOV	01,(SP)	6361
000120 010600		MOV	SP,RO	
000124 104415		TRAP	15	
000126 013700 000000G		MOV	RP,ADDR,RO	
000130 005760 000012		EXT	12(RO)	6363
000136 001454		BEQ	55	
000140 012716 000000G		MOV	0EX,CMP,(SP)	
000144 012746 000001		MOV	01,(SP)	6365
000150 010600		MOV	SP,RO	
000152 104415		TRAP	15	
000154 000424		BR	45	
000156 020127 000043	55:	CMP	R1,043	
000160 001024		BNE	65	6367
000164 012716 000000G		MOV	0EX,WRT,(SP)	
000170 012746 000001		MOV	01,(SP)	6370
000174 010600		MOV	SP,RO	
000176 104415		TRAP	15	
000200 013700 000000G		MOV	RP,ADDR,RO	
000204 005760 000012		EXT	12(RO)	6372
000210 001407		BEQ	55	
000212 012716 000000G		MOV	0EX,CMP,(SP)	
000216 012746 000001		MOV	01,(SP)	6374
000220 010600		MOV	SP,RO	
000224 104415		TRAP	15	
000226 005726	45:	EXT	12(RO)	
000230 005726	55:	EXT	12(RO)	
000232 000412		BR	75	6376
000234 005016	65:	CLR	00	6378
000236 112716 000014		MOVH	14(RP),SP	
000240 012746 000000G		MOV	0EX,RP,(SP)	
000244 012746 000002		MOV	01,(SP)	
000246 010600		MOV	SP,RO	
000254 104415		TRAP	15	

[] []

ZRQADM
VOL.6

RD RX EXERCISER
ERROR MESSAGE SUBROUTINES

11 Apr 1984 11:46:01
11 Apr 1984 11:45:02

VAX 11 B1144 16 V4.0 5/79
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0198
Page 198
(66)

000050 000026
000060 000026
000062 000020

74:

CHP (SP)+,(SP)+
CMP (SP)+,(SP)+
RTS PC

1
1

6750
6339

; Routine size: 90 words, Routine Base: \$CODE\$ + 10520
; Maximum stack depth per invocation: 9 words

ZRQAD0
VOL.6

RD RX EXERCISER
ERROR MESSAGE SUBROUTINES

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX 11 B1100 16 V4.0 579
DISK\$USER2:[POWER5]ZRQAD0.BC1:6

SEQ 0199
Page 199
(67)

```

: 6382 1 GLOBAL ROUTINE EMS.DBN : NOVALUE :
: 6383 1 : :
: 6384 1 : THIS ROUTINE PRINTS THE PRESENT DBN :
: 6385 1 : :
: 6386 1 : IMPLICIT INPUTS; :
: 6387 1 : CST ADDR = ADDRESS OF CONTROLLER STATUS TABLE :
: 6388 1 : :
: 6389 1 : :
: 6390 1 BEGIN :
: 6391 2 PRINTB (XX13, .CDISK); !"DISK XXX" :
: 6392 2 PRINTB (XX23, .CST ADDR [.C00FF + OF DBN, D.DBN], .CST ADDR :
: 6393 2 [.C00FF + OF DBN, D.DBN]); !"DBN: XXXXXX." :
: 6394 2 PRINTB (XX32, .S.DUPPKT); !PRINT BYTE COUNT :
: 6395 2 PRINTB (XX33, .S.PATTERN); !PRINT THE PATTERN :
: 6396 2 PRINTB (XX34, (.DUPPKT + .S.DUPPKT), (.DUPPKT + .S.DUPPKT)); !PRINT THE WORD READ :
: 6397 2 EMS.BLK (DUPPKT * 2, 256); !PRINT WHOLE BLOCK READ :
: 6398 1 END; !IN OCTAL :

```

			.SBYTL. EMS.DBN ERROR MESSAGE SUBROUTINES	
000000	013746	0000000	EMS.DBN:	
			MOV CDISK, (SP)	6391
000004	012746	0000000	MOV XXX13, (SP)	
000010	012746	0000002	MOV XXX23, (SP)	
000014	010600		MOV SP, R0	: SP, *
000016	104414		TRAP 14	
000020	013700	0000000	MOV C00FF, R0	6393
000024	006300		ASL R0	
000026	065700	0000000	ADD CST ADDR, R0	
000032	005016		CLR (SP)	
000034	116016	000020	MOVH PC(R0), (SP)	
000040	005046		CLR (SP)	
000042	116016	000020	MOVH PC(R0), (SP)	
000044	012746	0000000	MOV XXX23, (SP)	
000046	012746	0000003	MOV XXX32, (SP)	
000052	012746		MOV XXX33, (SP)	
000054	010600		MOV SP, R0	: SP, *
000060	104414		TRAP 14	
000062	013716	0000000	MOV S.DUPPKT, (SP)	6394
000064	162716	0000002	MOV XXX32, (SP)	
000066	012746	0000000	MOV XXX33, (SP)	
000068	012746	0000003	MOV XXX34, (SP)	
000074	010600		MOV SP, R0	: SP, *
000076	104414		TRAP 14	
000078	013716	0000000	MOV S.PATTERN, (SP)	6395
000080	012746	0000000	MOV XXX33, (SP)	
000082	012746	0000003	MOV XXX34, (SP)	
000084	010600		MOV SP, R0	: SP, *
000086	104414		TRAP 14	
000088	013700	0000000	MOV S.DUPPKT, R0	6396
000090	016016	0000000	MOV DUPPKT(R0), (SP)	
000092	011646		MOV (SP), (SP)	
000094	012746	0000000	MOV XXX34, (SP)	
000096	012746	0000003	MOV XXX35, (SP)	

11

ZRQADM
VOL 1

RD RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11-B1155 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0200
Page 200
(67)

000150	010600		MOV	SP,RO
000152	104414		TRAP	14
000154	012716	0000026	MOV	0DUPPCT+2,(SP)
000160	012746	000400	MOV	0400,(SP)
000164	004737	000000V	JSR	PC,FMS BLK
000170	002706	000034	ADD	034,SP
000174	000207		RTS	PC

; SP, +

6397

; ;

6390
6382

; Routine size: 15 words, Routine Base: \$CODE\$ + 11004
; Maximum stack depth per invocation: 15 words

; 0400 1
; 0400 1

(5) 15

ZRQAM,
V01.6

RD RX EXERCISER
ERROR MESSAGE SUBROUTINES

11 Apr 1984 11:56:01
11 Apr 1984 11:45:02

VAX 11 H1144 16 V4.0 5/79
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0201
Page 201
(68)

```

: 6401 1
: 6402 1 GLOBAL ROUTINE EMS_BLK (ADDR, LENGTH) : NOVALUE
: 6403 1
: 6404 1 !
: 6405 1 ! THIS ROUTINE WILL PRINTX A BLOCK OF MEMORY, WHICH IS LENGTH
: 6406 1 ! WORDS LONG STARTING AT ADDRESS ADDR. PRINTING IS DONE IN OCTAL
: 6407 1 ! 8 WDS TO A LINE.
: 6408 1 !
: 6409 1
: 6410 2 BEGIN
: 6411 2 LITERAL
: 6412 2 MASK = #0177;
: 6413 2
: 6414 2 PRINTX (CRLF);
: 6415 2 INCR COUNT FROM 1 TO LENGTH DO !FOR EACH WD TO PRINT
: 6416 3 BEGIN
: 6417 3 IF ((COUNT - 1) AND MASK) EQ 0 !IF START OF NEW LINE
: 6418 3 THEN
: 6419 3 PRINTX (SPACE4); !PRINT 4 BLANKS
: 6420 3
: 6421 3 PRINTX (EX_WRD, ADDR); !PRINTX A WORD
: 6422 3 ADDR = ADDR + 2; !TO NEXT ADDRESS
: 6423 3
: 6424 4 IF (((COUNT AND MASK) EQ 0) OR !END OF LINE OR
: 6425 4 (COUNT EQ LENGTH)) !WHEN DONE
: 6426 3 THEN
: 6427 3 PRINTX (CRLF); !PRINT CR LF
: 6428 2
: 6429 1 END;
END;
```

000000 010146		SBTTL EMS_BLK ERROR MESSAGE SUBROUTINES	
EMS_BLK:			
000002 012746 0000006	MOV R1, (SP)	:	6401
000006 012746 000001	MOV @CRLF, (SP)	:	6404
000012 010600	MOV @1, (SP)	:	
000014 104415	MOV SP, R0	:	SP, *
000016 005001	TRAP 15	:	
000020 000445	CLR R1	:	COUNT
000022 010100	BR 55	:	
000024 005300	15: MOV R1, R0	:	COUNT, *
000026 032700 000007	DEC R0	:	
000032 001007	HLT @2, R0	:	
000034 012716 0000006	ERR 25	:	
000040 012746 000001	MOV @SPACE4, (SP)	:	6419
000044 010600	MOV @1, (SP)	:	
000046 104415	MOV SP, R0	:	SP, *
000050 005300	TRAP 15	:	
000052 012616 000012	TEST (SP),	:	
000056 012746 0000006	25: MOV @1, (SP), (SP)	:	ADDR, *
000060 012746 000002	MOV @EX_WRD, (SP)	:	6421
000066 010600	MOV @2, (SP)	:	
	MOV SP, R0	:	SP, *

FILE

TRQAMP	RD/RY EXERCISER	11 Apr 1984 11:50:11	VAX 11 B11-86 16 V4.0 579	SEQ 0202
VC1,6	ERROR MESSAGE SUBROUTINES	11 Apr 1984 11:45:02	DISK\$USER2:[POWERS]ZRDADO.BL1;6	Page 202
				(68)

000070	104415		TRAP	15		
000072	062766	000007	ADD	27,16(SP)	:	*.ADDR
000100	032701	000007	BIT	27,R1	:	*.COUNT
000104	001403		BEQ	33		
000106	020166	000014	CMP	R1,14(SP)	:	COUNT,LENGTH
000112	001007		BNE	43		
000114	012716	0000006	MOV	2CR1,16(SP)	:	
000120	012746	000001	MOV	21,16(SP)	:	
000124	010600		MOV	SP,R0	:	SP,*
000126	104415		TRAP	15		
000130	005726		TXI	16(SP),		
000132	022626	43:	CMP	(SP),16(SP),	:	
000134	005201	53:	INC	R1	:	COUNT
000136	020166	000010	CMP	R1,10(SP)	:	COUNT,LENGTH
000142	005727		BLE	13		
000144	022626		CMP	(SP),16(SP),	:	
000146	012601		MOV	(SP),R1	:	
000150	000207		RTS	PC	:	

; Routine Size: 53 words, Routine Base: \$CODE\$ + 11202
; Maximum stack depth per invocation: 8 words

; 6430 1
; 6431 1

TRQAM,
VOL.6

RD RX EXERCISER
ERROR MESSAGE SUBROUTINES

11 Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX 11 B11ss 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRGADO.BU1;6

SEQ 0203
Page 203
(59)

```

: 6432 1 routine EMS_LBN : novalue *
: 6433 1
: 6434 1
: 6435 1
: 6436 1
: 6437 1
: 6438 1
: 6439 1
: 6440 1
: 6441 1
: 6442 1
: 6443 1
: 6444 1
: 6445 1
: 6446 1
: 6447 1
: 6448 1
: 6449 1
: 6450 1
: 6451 1
: 6452 1
: 6453 1
: 6454 1
: 6455 1
: 6456 1
: 6457 1
: 6458 1
: 6459 1
: 6460 1
: 6461 1
: 6462 1
: 6463 1
: 6464 1
: 6465 1
: 6466 1
: 6467 1

THIS ROUTINE PRINTS (EXTENDED) ONE OF TWO BLOCK NUMBERS APPEARING IN
THE CURRENT RETURN PACKET. NORMALLY, THE LBN FIELD IS PRINTED; THIS
FIELD WAS COPIED INTO THE RETURN PACKET FROM THE ASSOCIATED COMMAND
PACKET. HOWEVER, IF THE "FLAGS" FIELD OF THE CURRENT RETURN PACKET
INDICATES "BAD BLOCK REPORTED", THEN THE "FIRST BAD BLOCK" FIELD IS
PRINTED.

IMPLICIT INPUTS:
    RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET

begin
    if (not BIT_TST (RP_ADDR [FLAGS], EF_BBR)) and
        (not BIT_TST (RP_ADDR [FLAGS], EF_BBU))
    then
        PRINTX (EX_LBN, RP_ADDR [LBN_LO], RP_ADDR [LBN_HI]);
    if (not BIT_TST (RP_ADDR [FLAGS], EF_BBR)) and
        (BIT_TST (RP_ADDR [FLAGS], EF_BBU))
    then
        PRINTX (EX_BBU, RP_ADDR [BBUK_LO], RP_ADDR [BBUK_HI]);
    if (BIT_TST (RP_ADDR [FLAGS], EF_BBR)) and
        (not BIT_TST (RP_ADDR [FLAGS], EF_BBU))
    then
        PRINTX (EX_BB, RP_ADDR [BBUK_LO], RP_ADDR [BBUK_HI]);
    if (BIT_TST (RP_ADDR [FLAGS], EF_BBR)) and
        (BIT_TST (RP_ADDR [FLAGS], EF_BBU))
    then
        PRINTX (EX_BB1, RP_ADDR [BBUK_LO], RP_ADDR [BBUK_HI]);
end;
```

000000	013700	0000000	EMS_LBN: MOV	RP_ADDR, R0		6444
000004	105760	000015	TEST	15(R0)		
000010	100417		BMI	15		
000012	132760	000100 000015	BITB	0100, 15(R0)		6449
000020	001015		BNE	15		
000022	016046	000050	MOV	50(R0), (SP)		6451
000026	011646		MOV	(SP), (SP)		
000030	012746	0000000	MOV	0EX_LBN, (SP)		
000034	012746	000003	MOV	03, (SP)		
000040	010600		MOV	SP, R0		
000042	104415		TRAP	15		
000044	062706	000010	ADD	010, SP		
000050	013700	0000006	MOV	RP_ADDR, R0		6453
000054	105760	000015	TEST	15(R0)		

J16

ZRQAM2 V01.6	RD-RX EXERCISER ERROR MESSAGE SUBROUTINES	11-Apr-1984 11:56:01 11-Apr-1984 11:45:02	VAX-11 B11gs 16 V4.0 579 DISK\$USER2:[POWER5]ZRQADO.BU 1;6	SEQ 0204 Page 204 (69)

000060	100417		BMI	2\$		
000062	132760	000100 000015	BITB	0100,15(R0)	:	6454
000070	001413		BEQ	2\$		
000072	016046	000040	MOV	40(R0), (SP)	:	6456
000076	011646		MOV	(SP), (SP)		
000100	012746	0000006	MOV	0EX,BBU, -(SP)		
000104	012746	000003	MOV	03, (SP)		
000110	010600		MOV	SP,R0	:	SP,4
000112	104415		TRAP	15		
000114	062706	000010	ADD	010,SP		
000120	013700	0000006	2\$: MOV	RP,ADDR,R0	:	6458
000124	105760	000015	TSTB	15(R0)		
000130	100017		BPL	3\$		
000132	132760	000100 000015	BITB	0100,15(R0)	:	6459
000140	001013		BNE	3\$		
000142	016046	000040	MOV	40(R0), (SP)	:	6461
000146	011646		MOV	(SP), (SP)		
000150	012746	0000006	MOV	0EX,BB, -(SP)		
000154	012746	000003	MOV	03, (SP)		
000160	010600		MOV	SP,R0	:	SP,4
000162	104415		TRAP	15		
000164	062706	000010	ADD	010,SP		
000170	013700	0000006	3\$: MOV	RP,ADDR,R0	:	6463
000174	105760	000015	TSTB	15(R0)		
000180	100017		BPL	4\$		
000202	132760	000100 000015	BITB	0100,15(R0)	:	6464
000210	001413		BEQ	4\$		
000212	016046	000040	MOV	40(R0), (SP)	:	6466
000216	011646		MOV	(SP), (SP)		
000220	012746	0000006	MOV	0EX,BB1, -(SP)		
000224	012746	000003	MOV	03, (SP)		
000230	010600		MOV	SP,R0	:	SP,4
000232	104415		TRAP	15		
000234	062706	000010	ADD	010,SP		
000240	000207	4\$:	RTS	PC	:	6432

; Routine Size: 81 words, Routine Base: \$CODE\$ + 11354
; Maximum stack depth per invocation: 6 words

K16

ZRQAMP
V01.6

RDZRX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11ss 16 V4.0 579
DISK\$USER2:[POWER5]ZRQADO.BL1:6

SEQ 0205
Page 205
(70)

```

: 6468 1 routine EMS_BC : novalue =
: 6469 1
: 6470 1 !
: 6471 1 ! THIS ROUTINE PRINTS (EXTENDED) BOTH BYTE COUNT FIELDS OF THE CURRENT
: 6472 1 ! RETURN PACKET: THE BYTE COUNT FROM THE COMMAND PACKET AND THE
: 6473 1 ! ACTUAL NUMBER OF BYTES TRANSFERRED (FROM THE RESPONSE PACKET).
: 6474 1 !
: 6475 1 ! IMPLICIT INPUTS:
: 6476 1 ! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 6477 1 !
: 6478 1
: 6479 2 begin
: 6480 2 PRINTX (EX_CBC, RP_ADDR [CBCNT_LO]); ! "BYTE COUNT IN COMMAND: xxxxx."
: 6481 2 PRINTX (EX_BC, RP_ADDR [BCNT_LO]); ! "ACTUAL # OF BYTES TRANSFERRED: xxxxx."
: 6482 1 end; ! ROUTINE EMS_BC

```

			SBTTL	EMS_BC ERROR MESSAGE SUBROUTINES	
000000	013700	000000G	EMS_BC:	MOV RP_ADDR,R0	
000004	016046	000044		MOV 44(R0),-(SP)	6480
000010	012746	000000G		MOV 0EX_CBC,-(SP)	
000014	012746	000002		MOV 02,-(SP)	
000020	010600			MOV SP,R0	: SP,+
000022	104415			TRAP 15	
000024	013700	000000G		MOV RP_ADDR,R0	
000030	016016	000020		MOV 20(R0),-(SP)	6481
000034	012746	000000G		MOV 0EX_BC,-(SP)	
000040	012746	000002		MOV 02,-(SP)	
000044	010600			MOV SP,R0	: SP,+
000046	104415			TRAP 15	
000050	062706	000012		ADD 012,SP	
000054	000207			RTS PC	6468

: Routine size: 23 words, Routine Base: \$CODE\$ + 11616
: Maximum stack depth per invocation: 7 words

ZRQAM, RD/RX EXERCISER 11-Apr-1984 11:56:01 VAX-11 Bliss 16 V4.0-579 SEQ 0206
 VOL.6 ERROR MESSAGE SUBROUTINES 11-Apr-1984 11:45:02 DISK\$USER2:[POWERS]ZRQADO.BL1;6 Page 206
 (71)

```

: 6483 1 routine EMS.BD : novalue *
: 6484 1
: 6485 1 !
: 6486 1 ! THIS ROUTINE PRINTS (EXTENDED) THE TWO WORD I/O BUFFER DESCRIPTOR
: 6487 1 ! APPEARING IN THE CURRENT RETURN PACKET.
: 6488 1 !
: 6489 1 ! IMPLICIT INPUTS:
: 6490 1 ! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 6491 1 !
: 6492 1
: 6493 1 PRINTX (EX_BD, .RP_ADDR [BUFF_1], .RP_ADDR [BUFF_0]); ! "I/O BUFFER DESCRIPTOR: XXXXXX XXXXXX"

```

000000	013700	000000G	.SBITL	EMS.BD ERROR MESSAGE SUBROUTINES	
000004	016046	000024	EMS.BD: MOV	RP_ADDR, R0	6493
000010	016046	000026	MOV	24(R0), -(SP)	
000014	012746	000000G	MOV	26(R0), (SP)	
000020	012746	000003	MOV	EX_BD, (SP)	
000024	010600		MOV	03, (SP)	
000026	104415		MOV	SP, R0	: SP, *
000030	062706	000010	TRAP	15	
000034	000207		ADD	010, SP	
			RTS	PC	6483

: Routine Size: 15 words, Routine Base: \$CODE\$ + 11674
 : Maximum stack depth per invocation: 6 words

M16

RUAM,
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-57
DISK\$USER2:[POWERS]ZRQA IL1;6

SEQ 0207
Page 207
(72)

```

: 6494 1
: 6495 1
: 6496 1 routine EMS_RP : novalue =
: 6497 1
: 6498 1
: 6499 1
: 6500 1 !+
: 6501 1 ! THIS ROUTINE IS RESPONSIBLE FOR PRINTING (EXTENDED) THE RELEVANT FIELDS
: 6502 1 ! OF THE CURRENT RETURN PACKET.
: 6503 1 !-
: 6504 1
: 6505 2 begin
: 6506 2 EMS_SBC (); ! SUB CODE
: 6507 2 EMS_CMD (); ! COMMAND (AND MODIFIER)
: 6508 2
: 6509 2 if (.RP_ADDR [ENDCOD] and OP_MSK) neq OP_ONL
: 6510 2
: 6511 2 then
: 6512 2 EMS_LBN (); ! LBN OR BAD BLOCK NUMBER
: 6513 2
: 6514 2 if ((.RP_ADDR [ENDCOD] and OP_MSK) eq1 OP_RD) or
: 6515 3 ((.RP_ADDR [ENDCOD] and OP_MSK) eq1 OP_WRT)
: 6516 3
: 6517 2 then
: 6518 3 begin
: 6519 3 EMS_BC (); ! BYTE COUNTS
: 6520 3 EMS_BD (); ! I/O BUFFER DESCRIPTOR
: 6521 2 end;
: 6522 2
: 6523 2 EMS_TIM (); ! TIME
: 6524 1 end; ! ROUTINE EMS_RP

```

000000	010146		SBTTL	EMS_RP ERROR MESSAGE SUBROUTINES	
000002	004737	010010'	EMS_RP;	MOV R1, -(SP)	6496
000006	004737	010520'		JSR PC, EMS_SBC	6506
000012	013700	000000G		JSR PC, EMS_CMD	6507
000016	116000	000014		MOV RP_ADDR, R0	6509
000022	042700	177600		MOVB 14(R0), R0	
000026	020027	000011		BIC #177600, R0	
000032	001402			CMP R0, #11	
000034	004737	011354'		BEQ 1\$	
000040	013700	000000G	1\$:	JSR PC, EMS_LBN	6512
000044	116001	000014		MOV RP_ADDR, R0	6514
000050	042701	177600		MOVB 14(R0), R1	
000054	020127	000041		BIC #177600, R1	
000060	001407			CMP R1, #41	
000062	116000	000014		BEQ 2\$	
000066	042700	177600		MOVB 14(R0), R0	6515
000072	020027	000042		BIC #177600, R0	
000076	001004			CMP R0, #42	
000100	004737	011616'	2\$:	BNE 3\$	
				JSR PC, EMS_BC	6519

ZRQAM1	RD/R....B1	ZRQAM1	RD/R....B5	ZRQAM1	RD/R....B9	ZRQAM2	RD/R....B13
ZRQAM1	RD/R....C1	ZRQAM1	RD/R....C5	ZRQAM1	RD/R....C9	ZRQAM2	RD/R....C13
ZRQAM1	RD/R....D1	ZRQAM1	RD/R....D5	ZRQAM1	RD/R....D9	ZRQAM2	RD/R....D13
ZRQAM1	RD/R....E1	ZRQAM1	RD/R....E5	ZRQAM1	RD/R....E9	ZRQAM2	RD/R....E13
ZRQAM1	RD/R....F1	ZRQAM1	RD/R....F5	ZRQAM1	RD/R....F9	ZRQAM2	RD/R....F13
ZRQAM1	RD/R....G1	ZRQAM1	RD/R....G5	ZRQAM1	RD/R....G9	ZRQAM2	RD/R....G13
ZRQAM1	RD/R....H1	ZRQAM1	RD/R....H5	ZRQAM1	RD/R....H9	ZRQAM2	RD/R....H13
ZRQAM1	RD/R....I1	ZRQAM1	RD/R....I5	ZRQAM1	RD/R....I9	ZRQAM2	RD/R....I13
ZRQAM1	RD/R....J1	ZRQAM1	RD/R....J5	ZRQAM2	RD/R....J9	ZRQAM2	RD/R....J13
ZRQAM1	RD/R....K1	ZRQAM1	RD/R....K5	ZRQAM2	RD/R....K9	ZRQAM2	RD/R....K13
ZRQAM1	RD/R....L1	ZRQAM1	RD/R....L5	ZRQAM2	RD/R....L9	ZRQAM2	RD/R....L13
ZRQAM1	RD/R....M1	ZRQAM1	RD/R....M5	ZRQAM2	RD/R....M9	ZRQAM2	RD/R....M13
ZRQAM1	RD/R....N1	ZRQAM1	RD/R....N5	ZRQAM2	RD/R....N9	ZRQAM2	RD/R....N13
ZRQAM1	RD/R....B2	ZRQAM1	RD/R....B6	ZRQAM2	RD/R....B10	ZRQAM2	RD/R....B14
ZRQAM1	RD/R....C2	ZRQAM1	RD/R....C6	ZRQAM2	RD/R....C10	ZRQAM2	RD/R....C14
ZRQAM1	RD/R....D2	ZRQAM1	RD/R....D6	ZRQAM2	RD/R....D10	ZRQAM2	RD/R....D14
ZRQAM1	RD/R....E2	ZRQAM1	RD/R....E6	ZRQAM2	RD/R....E10	ZRQAM2	RD/R....E14
ZRQAM1	RD/R....F2	ZRQAM1	RD/R....F6	ZRQAM2	RD/R....F10	ZRQAM2	RD/R....F14
ZRQAM1	RD/R....G2	ZRQAM1	RD/R....G6	ZRQAM2	RD/R....G10	ZRQAM2	RD/R....G14
ZRQAM1	RD/R....H2	ZRQAM1	RD/R....H6	ZRQAM2	RD/R....H10	ZRQAM2	RD/R....H14
ZRQAM1	RD/R....I2	ZRQAM1	RD/R....I6	ZRQAM2	RD/R....I10	ZRQAM2	RD/R....I14
ZRQAM1	RD/R....J2	ZRQAM1	RD/R....J6	ZRQAM2	RD/R....J10	ZRQAM2	RD/R....J14
ZRQAM1	RD/R....K2	ZRQAM1	RD/R....K6	ZRQAM2	RD/R....K10	ZRQAM2	RD/R....K14
ZRQAM1	RD/R....L2	ZRQAM1	RD/R....L6	ZRQAM2	RD/R....L10	ZRQAM2	RD/R....L14
ZRQAM1	RD/R....M2	ZRQAM1	RD/R....M6	ZRQAM2	RD/R....M10	ZRQAM2	RD/R....M14
ZRQAM1	RD/R....N2	ZRQAM1	RD/R....N6	ZRQAM2	RD/R....N10	ZRQAM2	RD/R....N14
ZRQAM1	RD/R....B3	ZRQAM1	RD/R....B7	ZRQAM2	RD/R....B11	ZRQAM2	RD/R....B15
ZRQAM1	RD/R....C3	ZRQAM1	RD/R....C7	ZRQAM2	RD/R....C11	ZRQAM2	RD/R....C15
ZRQAM1	RD/R....D3	ZRQAM1	RD/R....D7	ZRQAM2	RD/R....D11	ZRQAM2	RD/R....D15
ZRQAM1	RD/R....E3	ZRQAM1	RD/R....E7	ZRQAM2	RD/R....E11	ZRQAM2	RD/R....E15
ZRQAM1	RD/R....F3	ZRQAM1	RD/R....F7	ZRQAM2	RD/R....F11	ZRQAM2	RD/R....F15
ZRQAM1	RD/R....G3	ZRQAM1	RD/R....G7	ZRQAM2	RD/R....G11	ZRQAM2	RD/R....G15
ZRQAM1	RD/R....H3	ZRQAM1	RD/R....H7	ZRQAM2	RD/R....H11	ZRQAM2	RD/R....H15
ZRQAM1	RD/R....I3	ZRQAM1	RD/R....I7	ZRQAM2	RD/R....I11	ZRQAM2	RD/R....I15
ZRQAM1	RD/R....J3	ZRQAM1	RD/R....J7	ZRQAM2	RD/R....J11	ZRQAM2	RD/R....J15
ZRQAM1	RD/R....K3	ZRQAM1	RD/R....K7	ZRQAM2	RD/R....K11	ZRQAM2	RD/R....K15
ZRQAM1	RD/R....L3	ZRQAM1	RD/R....L7	ZRQAM2	RD/R....L11	ZRQAM2	RD/R....L15
ZRQAM1	RD/R....M3	ZRQAM1	RD/R....M7	ZRQAM2	RD/R....M11	ZRQAM2	RD/R....M15
ZRQAM1	RD/R....N3	ZRQAM1	RD/R....N7	ZRQAM2	RD/R....N11	ZRQAM2	RD/R....N15
ZRQAM1	RD/R....B4	ZRQAM1	RD/R....B8	ZRQAM2	RD/R....B12	ZRQAM2	RD/R....B16
ZRQAM1	RD/R....C4	ZRQAM1	RD/R....C8	ZRQAM2	RD/R....C12	ZRQAM2	RD/R....C16
ZRQAM1	RD/R....D4	ZRQAM1	RD/R....D8	ZRQAM2	RD/R....D12	ZRQAM2	RD/R....D16
ZRQAM1	RD/R....E4	ZRQAM1	RD/R....E8	ZRQAM2	RD/R....E12	ZRQAM2	RD/R....E16
ZRQAM1	RD/R....F4	ZRQAM1	RD/R....F8	ZRQAM2	RD/R....F12	ZRQAM2	RD/R....F16
ZRQAM1	RD/R....G4	ZRQAM1	RD/R....G8	ZRQAM2	RD/R....G12	ZRQAM2	RD/R....G16
ZRQAM1	RD/R....H4	ZRQAM1	RD/R....H8	ZRQAM2	RD/R....H12	ZRQAM2	RD/R....H16
ZRQAM1	RD/R....I4	ZRQAM1	RD/R....I8	ZRQAM2	RD/R....I12	ZRQAM2	RD/R....I16
ZRQAM1	RD/R....J4	ZRQAM1	RD/R....J8	ZRQAM2	RD/R....J12	ZRQAM2	RD/R....J16
ZRQAM1	RD/R....K4	ZRQAM1	RD/R....K8	ZRQAM2	RD/R....K12	ZRQAM2	RD/R....K16
ZRQAM1	RD/R....L4	ZRQAM1	RD/R....L8	ZRQAM2	RD/R....L12	ZRQAM2	RD/R....L16
ZRQAM1	RD/R....M4	ZRQAM1	RD/R....M8	ZRQAM2	RD/R....M12	ZRQAM2	RD/R....M16
ZRQAM1	RD/R....N4	ZRQAM1	RD/R....N8	ZRQAM2	RD/R....N12	ZRQAM2	RD/R....N16

B1

ZRQAM2 RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11-16 V4.0-579
DISK#USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0208
Page 208
(72)

000104	004737	011674		JSR	PC,EMS,JD		
000110	004737	00C000V	31:	JSR	PC,EMS,TIM		
000114	012601			MOV	(SP)+,R1		
000116	000207			RTS	PC		

6520
6523
6496

; Routine Size: 4^ words, Routine Base: \$CODE\$ + 11732
; Maximum stack depth per invocation: 2 words

; 6525 1
; 6526 1

C1

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0209
Page 209
(73)

```

; 6527 1 global routine EMS_RP1 : novalue *
; 6528 1
; 6529 1
; 6530 1
; 6531 1
; 6532 1
; 6533 1
; 6534 1
; 6535 2 begin
; 6536 2 PRINTX (EX_RP);
; 6537 2 EMS_BLK (,RP_ADDR, RP_LEN);
; 6538 1 end;

```

THIS ROUTINE IS CALLED TO PRINT THE ENTIRE CONTENTS OF THE RETURN PACKET DESIGNATED BY THE GLOBAL DATUM "RP_ADDR". HOWEVER, THE PRINTING WILL ONLY OCCUR IF EXTENDED ERROR PRINTING IS ENABLED.

! "CONTENTS OF RETURN PACKET:"
! PRINT BLOCK OF WORDS

			.SBTTL	EMS_RP1 ERROR MESSAGE SUBROUTINES	
000000	012746	000000G	EMS_RP1::		
000004	012746	000001	MOV	0EX,RP, -(SP)	6536
000010	010600		MOV	01, (SP)	
000012	104415		MOV	SP,RO	SP, *
000014	013716	000000G	TRAP	15	
000020	012746	000026	MOV	RP_ADDR, (SP)	6537
000024	004137	011202	MOV	026, -(SP)	
000030	062706	000006	JSR	PC,EMS.BLK	
000034	000207		ADD	06,SP	6535
			RTS	PC	6527

; Routine Size: 15 words, Routine Base: \$CODE\$ + 12052
; Maximum stack depth per invocation: 4 words

D1

ZRGAM2
V01.6RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK1USER2:[POWERS]ZRGADO.BL1;6SEQ 0210
Page 210
(74)

```

: 6539 1 global routine EMS_EL (index) : novalue -
: 6540 1
: 6541 1 !+
: 6542 1 ! THIS ROUTINE IS CALLED FROM 'SEQUEN' AND 'DATAGM' AND PRINTS THE CONTENTS OF THE
: 6543 1 ! ERROR-LOG PACKET
: 6544 1 !-
: 6545 1
: 6546 2 begin
: 6547 2
: 6548 2 local
: 6549 2 ELOG_ADDR : ref block (EP_LFN, word) field (EP_FIELDS),
: 6550 2 REASON : word,
: 6551 2 DISK_NUM : byte,
: 6552 2 ELOG_CODE : byte,
: 6553 2 ELOG_SUB : word;
: 6554 2
: 6555 2 ELOG_ADDR = ELOG_PKT + (.index * EP_LEN * 2);
: 6556 2 REASON = .ELOG_ADDR [EL_FORMAT];
: 6557 2 DISK_NUM = .ELOG_ADDR [EL_DK_NUM];
: 6558 2 ELOG_CODE = .ELOG_ADDR [EL_CODE];
: 6559 2 ELOG_SUB = .ELOG_ADDR [EL_SUBCODE];
: 6560 2 PRINTB (ELG_00);
: 6561 2
: 6562 2 if (.REASON eq1 FORMAT_CNTR) or
: 6563 3 (.REASON eq1 FORMAT_HOST)
: 6564 2 then
: 6565 3 PRINTB (.ELG_FMT [.REASON])
: 6566 2 else
: 6567 2 PRINTB (.ELG_FMT [.REASON], .DISK_NUM);
: 6568 2
: 6569 2 if (.ELOG_CODE gtru 0) and
: 6570 3 (.ELOG_CODE lequ 11)
: 6571 2 then
: 6572 3 begin
: 6573 3 PRINTX (ASTERISK);
: 6574 3 PRINTX (.ERR_COD [.ELOG_CODE - 1]);
: 6575 3 end
: 6576 2 else
: 6577 2
: 6578 2 if .ELOG_CODE eq1 ST_DIA
: 6579 2 then
: 6580 3 begin
: 6581 3 PRINTX (ASTERISK);
: 6582 3 PRINTX (.ERR_COD [12]);
: 6583 3 end;
: 6584 2
: 6585 2 if (.ELOG_CODE eq1 ST_MFE) and
: 6586 3 (.ELOG_SUB lequ 10)
: 6587 2 then
: 6588 3 begin
: 6589 3 PRINTX (CR LF);
: 6590 3 PRINTX (ASTERISK);
: 6591 3 PRINTX (.TBL_MFE [.ELOG_SUB]);

```

! ERROR LOG PACKET'S ADDRESS
! FORMAT
! DISK NUMBER
! CODE
! SUBCODE
! ERROR-LOG MESSAGE RECEIVED

! PRINT BASIC REASON
! PRINT BASIC REASON WITH DISK NUMBER

! CODE

! MESSAGE FROM INTERNAL DIAGNOSTICS

! MEDIA FORMAT ERROR

ZRQAM,
V01.6

RD-RX EXERCISER
ERROR MESSAGE SUBROUTINES

[1]

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11gs-16 V4.0-579
DISK:USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0211
Page 211
(74)

```

: 6592 2      end;
: 6593 2
: 6594 2      if (.ELOG_CODE eq1 ST_DAT) and
: 6595 3          (.ELOG_SUB lequ 15)
: 6596 2      then
: 6597 3          begin
: 6598 3              PRINTX (CRLF);
: 6599 3              PRINTX (ASTERISK);
: 6600 3              PRINTX (.TBL_DAT [.ELOG_SUB]);
: 6601 2          end;
: 6602 2
: 6603 2      if (.ELOG_CODE eq1 ST_HST) and
: 6604 3          (.ELOG_SUB lequ 4)
: 6605 2      then
: 6606 3          begin
: 6607 3              PRINTX (CRLF);
: 6608 3              PRINTX (ASTERISK);
: 6609 3              PRINTX (.TBL_HST [.ELOG_SUB]);
: 6610 2          end;
: 6611 2
: 6612 2      if (.ELOG_CODE eq1 ST_CNT) and
: 6613 3          (.ELOG_SUB lequ 3)
: 6614 2      then
: 6615 3          begin
: 6616 3              PRINTX (CRLF);
: 6617 3              PRINTX (ASTERISK);
: 6618 3              PRINTX (.TBL_CNT [.ELOG_SUB]);
: 6619 2          end;
: 6620 2
: 6621 2      if (.ELOG_CODE eq1 ST_DRV) and
: 6622 3          (.ELOG_SUB lequ 8)
: 6623 2      then
: 6624 3          begin
: 6625 3              PRINTX (CRLF);
: 6626 3              PRINTX (ASTERISK);
: 6627 3              PRINTX (.TBL_DRV [.ELOG_SUB]);
: 6628 2          end;
: 6629 2
: 6630 2      if .REASON eq1 FORMAT_XFER
: 6631 2      then
: 6632 2
: 6633 2          if .ELOG_ADDR [EL_BLOCK_TYPE] eq1 TYPE_PBN
: 6634 2          then
: 6635 3              PRINTX (EX_PBN, .ELOG_ADDR [EL_BLOCK], .ELOG_ADDR [EL_BLOCK]);
: 6636 2          else
: 6637 3              PRINTX (EX_RBN, .ELOG_ADDR [EL_BLOCK], .ELOG_ADDR [EL_BLOCK]);
: 6638 2
: 6639 2      EMS_TIM ();
: 6640 2      EMS_BLK ((.ELOG_ADDR + 2), ((.ELOG_ADDR [EL_MSGLEN] + 1) / 2) + 2);
: 6641 2      ELOG_ADDR [EL_CONTENTS] = EMPTY;
: 6642 2
: 6643 1      end;

```

! DATA ERROR

! HOST ACCESS ERROR

! CONTROLLER ERROR

! DRIVE ERROR

! IF DISK XFER INVOLVED

! PRINT PBN OR RBN

! TIME

! PRINTX CONTENTS OF PACKET

! DECLARE SAVE AREA FREE

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0212
Page 212
(74)

			SBTTL	EMS,EL ERROR MESSAGE SUBROUTINES	
000000	004137	000000G	EMS,EL::JSR	R1,\$SAVE5	6539
000004	005746		TST	-(SP)	
000006	016646	000020	MOV	20(SP),-(SP)	: INDEX,*
000012	012746	000102	MOV	#102,-(SP)	
000016	004737	000000G	JSR	PC,BL\$MUL	
000022	062700	000000G	ADD	#ELOG.PKT,R0	
000026	010001		MOV	R0,R1	: *,ELOG.ADDR
000030	116166	000016 000004	MOVB	16(R1),4(SP)	: *(ELOG.ADDR),REASON
000036	105066	000005	CLRB	5(SP)	: REASON
000042	116105	000012	MOVB	12(R1),R5	: *(ELOG.ADDR),DISK.NUM
000046	116100	000020	MOVB	20(R1),R0	: *(ELOG.ADDR),*
000052	042700	177740	BIC	#177740,R0	
000056	105004		CLRB	R4	: ELOG.CODE
000060	050004		BIS	R0,R4	: *,ELOG.CODE
000062	016103	000020	MOV	20(R1),R3	: *(ELOG.ADDR),ELOG.SUB
000066	006203		ASR	R3	: ELOG.SUB
000070	006203		ASR	R3	: ELOG.SUB
000072	006203		ASR	R3	: ELOG.SUB
000074	006203		ASR	R3	: ELOG.SUB
000076	006203		ASR	R3	: ELOG.SUB
000100	042703	174000	BIC	#174000,R3	: *,ELOG.SUB
000104	012716	000000G	MOV	#ELG.00,(SP)	
000110	012746	000001	MOV	#1,-(SP)	
000114	010600		MOV	SP,R0	: SP,*
000116	104414		TRAP	14	
000120	016602	000006	MOV	6(SP),R2	: REASON,*
000124	006302		ASL	R2	
000126	005766	000006	TST	6(SP)	: REASON
000132	001404		BEQ	15	
000134	026627	000006 000001	CMF	6(SP),#1	: REASON,*
000142	001007		BNE	25	
000144	016216	000000G	15: MOV	ELG.FMT(R2),(SP)	
000150	012746	000001	MOV	#1,(SP)	
000154	010600		MOV	SP,R0	: SP,*
000156	104414		TRAP	14	
000160	000411		BR	35	
000162	005016		25: CLR	(SP)	
000164	110516		MOVB	R5,(SP)	: DISK.NUM,*
000166	016246	000000G	MOV	ELG.FMT(R2),-(SP)	
000172	012746	000002	MOV	#2,(SP)	
000176	010600		MOV	SP,R0	: SP,*
000200	104414		TRAP	14	
000202	005726		TST	(SP)	
000204	105704		35: TSTB	R4	: ELOG.CODE
000206	001423		BEQ	45	
000210	120427	000013	CMFB	R4,#15	: ELOG.CODE,*
000214	101020		BFI	45	
000216	012716	000000G	MOV	#ASTERISK,(SP)	
000222	012746	000001	MOV	#1,(SP)	
000226	010600		MOV	SP,R0	: SP,*
000230	104414		TRAP	15	

(31)

ZRQAMP
V01.6RD/RX EXERCISE
ERROR MESSAGE SUBROUTINES11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO,BL1;6SEQ 0213
Page 213
(74)

000232	005000		CLR	R0	:		6574	
000234	150400		BISB	R4,R0	:	ELOG.CODE,*		
000236	006300		ASL	R0	:			
000240	016016	177776G	MOV	ERR.COD-2(R0),(SP)	:			
000244	012746	000001	MOV	01,-(SP)	:			
000250	010600		MOV	SP,R0	:	SP,*		
000252	104415		TRAP	15	:			
000254	020417		BR	5:	:		6572	
000256	120427	000037	4:	CMPB	R4,037	:	ELOG.CODE,*	6578
000262	001015		BNE	6:	:			
000264	012716	000000G	MOV	0ASTERISK,(SP)	:		6581	
000270	012746	000001	MOV	01,-(SP)	:			
000274	010600		MOV	SP,R0	:	SP,*		
000276	104415		TRAP	15	:			
000300	013716	000030G	MOV	ERR.COD+30,(SP)	:		6582	
000304	012746	000001	MOV	01,-(SP)	:			
000310	010600		MOV	SP,R0	:	SP,*		
000312	104415		TRAP	15	:			
000314	022626		5:	CMP	(SP),(SP)+	:	6580	
000316	120427	000005	6:	CMPB	R4,05	:	ELOG.CODE,*	6585
000322	001031		BNE	7:	:			
000324	020327	000012	CMP	R3,012	:	ELOG.SUB,*	6586	
000330	101026		BHI	7:	:			
000332	012716	000000G	MOV	0CRLF,(SP)	:		6589	
000336	012746	000001	MOV	01,-(SP)	:			
000342	010600		MOV	SP,R0	:	SP,*		
000344	104415		TRAP	15	:			
000346	012716	000000G	MOV	0ASTERISK,(SP)	:		6590	
000352	012746	000001	MOV	01,-(SP)	:			
000356	010600		MOV	SP,R0	:	SP,*		
000360	104415		TRAP	15	:			
000362	010300		MOV	R3,R0	:	ELOG.SUB,*	6591	
000364	006300		ASL	R0	:			
000366	016016	000064	MOV	TBL.MFE(R0),(SP)	:			
000372	012746	0000C1	MOV	01,(SP)	:			
000376	010600		MOV	SP,R0	:	SP,*		
000400	104415		TRAP	15	:			
000402	062706	000006	ADD	06,SP	:		6583	
000406	120427	000010	7:	CMPB	R4,010	:	ELOG.CODE,*	6594
000412	001031		BNL	8:	:			
000414	020327	000017	CMP	R3,017	:	ELOG.SUB,*	6595	
000420	101026		BHI	8:	:			
000422	012716	000000G	MOV	0CRLF,(SP)	:		6598	
000426	012746	000001	MOV	01,-(SP)	:			
000432	010600		MOV	SP,R0	:	SP,*		
000434	104415		TRAP	15	:			
000436	012716	000000G	MOV	0ASTERISK,(SP)	:		6599	
000442	012746	000001	MOV	01,-(SP)	:			
000446	010600		MOV	SP,R0	:	SP,*		
000450	104415		TRAP	15	:			
000452	010300		MOV	R3,R0	:	ELOG.SUB,*	6600	
000454	006300		ASL	R0	:			
000456	016016	000120	MOV	TBL.DAT(R0),(SP)	:			

[H]

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0214
Page 214
(74)

000462	012746	000001		MOV	01, -(SP)		
000466	010600			MOV	SP, R0	; SP, *	
000470	104415			TRAP	15		
000472	062706	000006		ADD	06, SP		6597
000476	120427	000011	8\$:	CMPB	R4, 011	; ELOG.CODE, *	6603
000502	001031			BNE	9\$		
000504	020327	000004		CMP	R3, 04	; ELOG.SUB, *	6604
000510	101026			BHI	9\$		
000512	012716	000000G		MOV	0CRLE, (SP)		6607
000516	012746	000001		MOV	01, -(SP)		
000522	010600			MOV	SP, R0	; SP, *	
000524	104415			TRAP	15		
000526	012716	000000G		MOV	0ASTERISK, (SP)		6608
000532	012746	000001		MOV	01, -(SP)		
000536	010600			MOV	SP, R0	; SP, *	
000540	104415			TRAP	15		
000542	010300			MOV	R3, R0	; ELOG.SUB, *	6609
000544	006300			ASL	R0		
000546	016016	000160'		MOV	TBL.HST(R0), (SP)		
000552	012746	000001		MOV	01, (SP)		
000556	010600			MOV	SP, R0	; SP, *	
000560	104415			TRAP	15		
000562	062706	000006		ADD	06, SP		6606
000566	120427	000012	9\$:	CMPB	R4, 012	; ELOG.CODE, *	6612
000572	001031			BNE	10\$		
000574	020327	000003		CMP	R3, 03	; ELOG.SUB, *	6613
000600	101026			BHI	10\$		
000602	012716	000000G		MOV	0CRLE, (SP)		6616
000606	012746	000001		MOV	01, -(SP)		
000612	010600			MOV	SP, R0	; SP, *	
000614	104415			TRAP	15		
000616	012716	000000G		MOV	0ASTERISK, (SP)		6617
000622	012746	000001		MOV	01, -(SP)		
000626	010600			MOV	SP, R0	; SP, *	
000630	104415			TRAP	15		
000632	010300			MOV	R3, R0	; ELOG.SUB, *	6618
000634	006300			ASL	R0		
000636	016016	000172'		MOV	TBL.CNT(R0), (SP)		
000642	012746	000001		MOV	01, (SP)		
000646	010600			MOV	SP, R0	; SP, *	
000650	104415			TRAP	15		
000652	062706	000006		ADD	06, SP		6615
000656	120427	000013	10\$:	CMPB	R4, 013	; ELOG.CODE, *	6621
000662	001031			BNE	11\$		
000664	020327	000010		CMP	R3, 010	; ELOG.SUB, *	6622
000670	101026			BHI	11\$		
000672	012716	000000G		MOV	0CRLE, (SP)		6625
000676	012746	000001		MOV	01, (SP)		
000702	010600			MOV	SP, R0	; SP, *	
000704	104415			TRAP	15		
000706	012716	000000G		MOV	0ASTERISK, (SP)		6626
000712	012746	000001		MOV	01, (SP)		
000716	010600			MOV	SP, R0	; SP, *	

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0215
Page 215
(74)

000720	104415			TRAP	15			
000722	010320			MOV	R3,R0		; ELOG.SUB,*	6627
000724	006300			ASL	R0			
000726	016016	000202		MOV	TBL,DRV(R0),(SP)			
000732	012746	000001		MOV	#1,-(SP)			
000736	010600			MOV	SP,R0		; SP,*	
000740	104415			TRAP	15			
000742	062706	000006		ADD	#6,SP			6624
000746	026627	000010	000002	11\$: CMP	10(SP),#2		; REASON,*	6630
000754	001031			BNE	14\$			
000756	032761	170000	000060	BIT	#170000,60(R1)		; *,*(ELOG.ADDR)	6633
000764	001012			BNE	12\$			
000766	016116	000056		MOV	56(R1),(SP)		; *(ELOG.ADDR),*	6635
000772	011646			MOV	(SP),-(SP)			
000774	012746	000000G		MOV	#EX.PBN,-(SP)			
001000	012746	000003		MOV	#*,-(SP)			
001004	010600			MOV	SP,R0		; SP,*	
001006	104415			TRAP	15			
001010	000411			BR	13\$			6633
001012	013116	000056		12\$: MOV	56(R1),(SP)		; *(ELOG.ADDR),*	6637
001016	011646			MOV	(SP),-(SP)			
001020	012746	000000G		MOV	#EX.RBN,-(SP)			
001024	012746	000003		MOV	#3,(SP)			
001030	010600			MOV	SP,R0		; SP,*	
001032	104415			TRAP	15			
001034	062706	000006		13\$: ADD	#6,SP			6633
001040	004737	000000V		14\$: JSR	PC,FMS,TIM			6639
001044	012716	000002		MOV	#2,(SP)			6640
001050	060116			ADD	R1,(SP)		; ELOG.ADDR,*	
001052	016146	000002		MOV	2(R1),-(SP)		; *(ELOG.ADDR),*	
001056	005216			INC	(SP)			
001060	012746	000002		MOV	#2,-(SP)			
001064	004737	000000G		JSR	PC,BL\$DIV			
001070	010066	000002		MOV	R0,2(SP)			
001074	062766	000002	000002	ADD	#2,2(SP)			
001102	005726			TST	(SP)			
001104	004737	011202		JSR	PC,FMS,BLK			
001110	105061	000001		CLRB	1(R1)		; *(ELOG.ADDR)	6641
001114	062706	000014		ADD	#14,SP			6539
001120	000207			RTS	PC			

; Routine Size: 297 words, Routine Base: \$CODE\$ + 12110
; Maximum stack depth per invocation: 16 words

7 QM
1 1

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0216
Page 215
(75)

```

: 6644 1 global routine EMS_CMP (ADDR) : novalue =
: 6645 1
: 6646 1 !+
: 6647 1 ! THIS ROUTINE IS CALLED FROM 'HOST_WRT_CHK' AND PRINTS RELEVANT DATA ON A HOST
: 6648 1 ! COMPARE ERROR
: 6649 1 !-
: 6650 1
: 6651 2 begin
: 6652 2
: 6653 2 local
: 6654 2 ORIG_ADDR : ref block [RP_LEN, word] field (RP_FIELDS);
: 6655 2
: 6656 2 ORIG_ADDR = ADDR;
: 6657 2 PRINTB (ERR_00, CDISK);
: 6658 2 PRINTB (DASH);
: 6659 2 PRINTB (ERR_COD [12]);
: 6660 2 PRINTX (EX_LBW, ORIG_ADDR [LBN_LO], ORIG_ADDR [LBN_LO]);
: 6661 2 PRINTX (EX_LBR, RP_ADDR [LBN_LO], RP_ADDR [LBN_LO]);
: 6662 2 PRINTX (EX_CBW, ORIG_ADDR [CBCNT_LO]);
: 6663 2 PRINTX (EX_BC, ORIG_ADDR [BCNT_LO]);
: 6664 2 PRINTX (EX_CBR, RP_ADDR [CBCNT_LO]);
: 6665 2 PRINTX (EX_BC, RP_ADDR [BCNT_LO]);
: 6666 2 PRINTX (EX_BDW, ORIG_ADDR [BUFF_1], ORIG_ADDR [BUFF_0]);
: 6667 2 PRINTX (EX_BDR, RP_ADDR [BUFF_1], RP_ADDR [BUFF_0]);
: 6668 2 EMS_TIM ();
: 6669 1 end;

```

```

! ADDRESS OF THE WRITE RETPKT
! "DISK XXX"
! " - HOST COMPARE ERROR"
! LBN (WRITE)
! LBN (READ)
! BYTE COUNT (WRITE)
! BYTE COUNT XMITTED (WRITE)
! BYTE COUNT (READ);
! BYTE COUNT XMITTED (READ)
! BUFFER ADDRESS (WRITE)
! BUFFER ADDRESS (READ)
! TIME

```

```

000000 010146 .SBTTL EMS_CMP ERROR MESSAGE SUBROUTINES
EMS_CMP:
MOV R1, (SP)
MOV 4(SP), R1
MOV CDISK, (SP)
MOV ERR_00, (SP)
MOV 02, (SP)
MOV SP, R0
TRAP 14
MOV DASH, (SP)
MOV 01, (SP)
MOV SP, R0
TRAP 14
MOV ERR_COD+30, (SP)
MOV 01, (SP)
MOV SP, R0
TRAP 14
MOV 50(R1), (SP)
MOV 05P, (SP)
MOV 0EX_LBW, (SP)
MOV SP, R0
TRAP 15
MOV RP_ADDR, R0
MOV 50(R0), (SP)

```

6644
6656
6657

6658

6659

6660

6661

K7

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0295
Page 64
(15)

000004	005003		CLR	R3	; RESULT	2884
000006	012702	000001	MOV	#1,R2	; *,PASS	
000012	005037	000000G	CLR	SB.CODE		2891
000016	005037	000000G	CLR	ST.CODE		
000022	013700	000000G	MOV	L\$LUN,R0		2892
000026	006300		ASL	R0		
000030	016000	000054'	MOV	MAX,LBN(R0),R0		
000034	060200		ADD	R2,R0		
000036	006200		ASH	R0		
000040	010004		MOV	R0,R4	; *,LBN	
000042	042704	100000	BIC	#100000,R4	; *,LBN	
000046	005304		DEC	R4	; LBN	
000050	013746	000000G	1\$: MOV	CCTLR,-(SP)		2897
000054	004737	000000G	JSR	PC,GET.PKT		
000060	010037	000000G	MOV	R0,P.INDEX		
000064	010016		MOV	R0,(SP)	; P.INDEX, *	2898
000066	012746	000106	MOV	#106,-(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)		2899
000112	012760	001000 000026G	MOV	#1000,MSCP.PKT+26(R0)		2900
000120	010460	000046G	MOV	R4,MSCP.PKT+46(R0)	; LBN, *	2901
000124	112760	000002 000004G	MOVB	#2,MSCP.PKT+4(R0)		2902
000132	013716	000000G	MOV	P.INDEX,(SP)		2904
000136	004737	000000G	JSR	PC,SEND		
000142	005700		TST	R0		
000144	001007		BNE	2\$		
000146	013716	000000G	MOV	P.INDEX,(SP)		2907
000152	004737	000000G	JSR	PC,PUT.PKT		
000156	012702	000002	MOV	#2,R2	; *,PASS	2908
000162	000522		BR	9\$		2904
000164	004737	000000G	2\$: JSR	PC,WAIT		2915
000170	004737	000000G	JSR	PC,OUT. IODQ		2916
000174	010037	000000G	MOV	R0,RP.INDX		
000200	010016		MOV	R0,(SP)	; RP.INDX, *	2917
000202	012746	000054	MOV	#54,-(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)		2919
000230	001404		BEQ	3\$		
000232	013716	000000G	MOV	RP.INDX,(SP)		2921
000236	004737	000000G	JSR	PC,PUT.RETPKT		
000242	005726		3\$: TST	(SP)+		2914
000244	013701	000000G	MOV	RP.ADDR,R1		2924
000250	005000		CLR	R0		
000252	126127	000003 000003	CMPB	3(R1),#3		
000260	001002		BNE	4\$		
000262	005200		INC	R0		
000264	000407		BR	5\$		
000266	132761	000360 000002	4\$: BITB	#360,2(R1)		2925
000274	001333		BNE	2\$		
000276	105761	000014	TSTB	14(R1)		2926

1 /

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0218
Page 218
(76)

```

: 6670 1 global routine EMS.ERR : novalue =
: 6671 1
: 6672 2 begin
: 6673 2
: 6674 2 ! TABLE OF BASIC, HARD ERROR MESSAGE ADDRESSES, INDEXED BY STATUS CODE
: 6675 2 !
: 6676 2 PRINTB (ERR_00, .CDISK); ! "DISK xxx"
: 6677 2 PRINTB (DASH); !
: 6678 2
: 6679 2 if (.ST_CODE gtru 0) and ! IF STATUS CODE IS WITHIN RANGE
: 6680 3 (.ST_CODE lequ 11)
: 6681 2 then
: 6682 3 PRINTB (.ERR_COD [.ST_CODE - 1]) ! PRINTB APPROPRIATE MESSAGE
: 6683 2 else
: 6684 2
: 6685 2 if .ST_CODE eql ST_DIA
: 6686 2 then
: 6687 3 PRINTB (.ERR_COD [11]) ! MESSAGE FROM INTERNAL DIAGNOSTICS
: 6688 2 else
: 6689 2 PRINTB (EX_SC, .ST_CODE); ! JUST PRINT STATUS CODE WHEN NO MATCH
: 6690 2
: 6691 2 EMS_RP (); ! PRINTX OTHER RETPKT FIELDS
: 6692 2
: 6693 1 end;
```

			.SBTTL	EMS.ERR ERROR MESSAGE SUBROUTINES		
000000	013746	000000G	EMS.ERR:			
			MOV	CDISK, -(SP)	6676	
000004	012746	000000G	MOV	ERR_00, -(SP)		
000010	012746	000002	MOV	02, -(SP)		
000014	010600		MOV	SP, R0	: SP, +	
000016	104414		TRAP	14		
000020	012716	000000G	MOV	DASH, (SP)	6677	
000024	012746	000001	MOV	01, -(SP)		
000030	010600		MOV	SP, R0	: SP, +	
000032	104414		TRAP	14		
000034	013700	000000G	MOV	ST_CODE, R0	6679	
000040	001413		BEQ	15		
000042	020027	000013	CMP	R0, 013	6680	
000044	101010		BHI	15		
000050	006300		ASL	R0	6682	
000052	016016	177776G	MOV	ERR_COD - 2(R0), (SP)		
000056	012746	000001	MOV	01, (SP)		
000062	010600		MOV	SP, R0	: SP, +	
000064	104414		TRAP	14		
000066	006422		BR	15	6679	
000070	020027	000037	15:	CMP	R0, 037	6685
000074	001007		BNE	25		
000076	013716	000026G	MOV	ERR_COD - 26, (SP)	6687	
000102	012746	000001	MOV	01, (SP)		
000106	010600		MOV	SP, R0	: SP, +	
000110	104414		TRAP	14		

M1

ZRQAMP
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0219
Page 219
(76)

000112	000410		BR	3:					
000114	010016		MOV	R0,(SP)					6685
000116	012746	000000G	MOV	4EX.SC,-(SP)					6689
000122	012746	000002	MOV	02,-(SP)					
000126	010600		MOV	SP,R0					
000130	104414		TRAP	14					
000132	005726		TST	(SP)+					
000134	004737	011732	JSR	PC,EMS.RP					6691
000140	062706	000012	ADD	012,SP					6672
000144	000207		RTS	PC					6670

; Routine Size: 51 words, Routine Base: \$CODE\$ + 13560
; Maximum stack depth per invocation: 8 words

N1

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4 : 579
DISK\$USER2:[POWERS]ZRQAD0.BL1;6

SEQ 0220
Page 220
(77)

```
; 6694 1 routine EMS_TIM : novalue =
; 6695 1
; 6696 1 !+
; 6697 1 ! THIS ROUTINE PRINTS THE TIME-OF-DAY MESSAGE
; 6698 1 !-
; 6699 1
; 6700 1 PRINTX (EX_TIM, .HOURS, .MINUTES);
```

000000	005046		.SBTTL	EMS_TIM ERROR MESSAGE SUBROUTINES	
000002	113716	000000G	EMS_TIM:CLR	-(SP)	6700
000006	005046		MOVB	MINUTES,(SP)	
000010	113716	000000G	CLR	-(SP)	
000014	012746	000000G	MOVB	HOURS,(SP)	
000020	012746	000003	MOV	EX_TIM, -(SP)	
000024	010600		MOV	03, -(SP)	
000026	104415		MOV	SP,R0	: SP,▲
000030	062700	000010	TRAP	15	
000034	000207		ADD	010,SP	
			RTS	PC	6694

```
; Routine Size: 15 words, Routine Base: $CODE$ + 13726
; Maximum stack depth per invocation: 6 words
```

B.2

ZRQAM, RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11:16 V4.0-579
DISK:USER2:[POWERS]ZRQADO.BL1:6

; 6701 1 BGNMSG (EMS_01);

000000	004737	000000V	EMS_01::	.SBTTL EMS_01 ERROR MESSAGE SUBROUTINES	
000004	104423		JSR	PC,M\$EMS_01	6701
000006	000207		TRAP	23	
			RTS	PC	

; Routine Size: 4 words, Routine Base: \$CODE\$ + 13764
; Maximum stack depth per invocation: 2 words

; 6702 2 PRINTB (EBS_01, MAX_UNITS); ! "MORE THAN XX UNITS SPECIFIED"
; 6703 1 ENDMMSG;

000000	012746	000004	M\$EMS_01:	.SBTTL M\$EMS_01 ERROR MESSAGE SUBROUTINES	
000004	012746	000000G	MOV	\$4, -(SP)	6702
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	6701
			RTS	PC	

; Routine Size: 11 words, Routine Base: \$CODE\$ + 13774
; Maximum stack depth per invocation: 5 words

C.

ZRQAMP
VO1.6

RDRX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11:16 V4.0-579
DISK\$USER2:[POWER5]ZRQAD0.BL1:6

SEQ 0222
Page 222
(79)

; 6704 1 BGNMSG (EMS_10);

000000	004737	000000V	.SBTTL	EMS_10 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS_10::JSR	PC,M\$EMS_10	
000006	000207		TRAP	23	
			RTS	PC	

6704

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14022
; Maximum stack depth per invocation: 2 words

; 6705 2 PRINTB (EBD_10, RDRX_ADDR + OF_RC); ! "NO RESPONSE AT ADDRESS XXXXXX"
; 6706 1 ENDMSG;

000000	013746	000000G	.SBTTL	M\$EMS_10 ERROR MESSAGE SUBROUTINES	
			M\$EMS_10:		
000004	063716	000000G	MOV	RDRX_ADDR, -(SP)	
000010	012746	000000G	ADD	OF_RC, (SP)	
000014	012746	000000G	MOV	EBD_10, -(SP)	
000020	010600	0000002	MOV	02, -(SP)	
000022	104414		MOV	SP, R0	; SP, *
000024	062706	0000006	TRAP	14	
000030	000207		ADD	06, SP	
			RTS	PC	

6705

6704

; Routine Size: 13 words, Routine Base: \$CODE\$ + 14032
; Maximum stack depth per invocation: 5 words

[D]

ZRQAM2 RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B11-16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0223
Page 223
(80)

; 6707 1 BGNMSG (EMS_12);

000000	004737	000000V		.SBTTL	EMS.12 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS.12::	JSR	PC,114EMS.12	
000006	000207			TRAP	23	
				RTS	PC	

6707

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14064
; Maximum stack depth per invocation: 2 words

; 6708 2 PRINTB (EBD_12, .RDRX_ADDR);
; 6709 1 ENDMSG;

! "INCORRECT BR LEVEL GIVEN FOR DEVICE XXXXXX"

000000	013746	000000G		.SBTTL	M\$EMS.12 ERROR MESSAGE SUBROUTINES	
			M\$EMS.12:			
000004	012746	000000G		MOV	RDRX_ADDR, -(SP)	
000010	012746	000002		MOV	EBD_12, -(SP)	
000014	010600			MOV	02, -(SP)	
000016	104414			MOV	SP, R0	; SP, 4
000020	062706	000006		TRAP	14	
000024	000207			ADD	06, SP	
				RTS	PC	

6708

6707

; Routine Size: 11 words, Routine Base: \$CODE\$ + 14074
; Maximum stack depth per invocation: 5 words

E2

SEQ 0224
Page 224
(81)ZRQAM: RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES11-Apr-1984 11:56:01
11-Apr-1984 11:45:02VAX-11 B111-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

; 6710 1 BGNMSG (EMS_13);

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

6710

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14122
; Maximum stack depth per invocation: 2 words

; 6711	2	PRINTB (EBD_13, .STEP);	! "STEP X READ ERROR"
; 6712	2	EMS_SA ();	! PRINTX SA CONTENTS
; 6713	1	ENDMSG;	

000000	013746	000000G	.SBTTL	M\$EMS.13 ERROR MESSAGE SUBROUTINES	
			M\$EMS.13:		
000004	012746	000000G	MOV	STEP, -(SP)	;
000010	012746	000002	MOV	EBD_13, -(SP)	
000014	010600		MOV	02, -(SP)	
000016	104414		MOV	SP, R0	; SP, 4
000020	004737	007476	TRAP	14	
000024	062706	000006	JSR	PC, EMS_SA	;
000030	000207		ADD	06, SP	;
			RTS	PC	

6711

6712
6710; Routine Size: 13 words, Routine Base: \$CODE\$ + 14132
; Maximum stack depth per invocation: 5 words

F2

ZRQAM2 RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0225
Page 225
(82)

: 6714 1 BGNMSG (EMS_14);

```

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

```

6714

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14164
; Maximum stack depth per invocation: 2 words

```

: 6715 2 PRINTB (EBD_14, ,IRDRX_ADDR);      ! "BAD SA CODE FROM DEVICE XXXXXX"
: 6716 2 EMS_SA ();                          ! PRINTX SA REGISTER CONTENTS
: 6717 1 ENDMMSG;

```

```

000000 013746 000000G      .SBTTL M$EMS.14 ERROR MESSAGE SUBROUTINES
                                M$EMS.14:
000004 012746 000000G      MOV IRDRX_ADDR, -(SP)
000010 012746 000002      MOV EBD_14, -(SP)
000014 010600      MOV #2, -(SP)
000016 104414      MOV SP, R0
000020 004737 007476      TRAP 14
000024 062706 000006      JSR PC, EMS_SA
000030 000207      ADD #6, SP
                        RTS PC

```

6715

6716
6714

; Routine Size: 13 words, Routine Base: \$CODE\$ + 14174
; Maximum stack depth per invocation: 5 words

62

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 B1:as-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1:6

SEQ 0226
Page 226
(83)

; 6718 1 BGNMSG (EMS_18);

000000	004737	000000V		.SBTTL	EMS_18 ERROR MESSAGE SUBROUTINES	
000004	104423		EMS_18::	JSR	PC,M\$EMS_18	;
000006	000207			TRAP	23	
				RTS	PC	

6718

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14226
; Maximum stack depth per invocation: 2 words

; 6719	2	PRINTB (ERD_18, .CDISK);	!	"DISK XXX WENT OFFLINE"
; 6720	2	EMS_RP ();	!	PRINTX RELEVANT RETPKT FIELDS
; 6721	1	ENDMSG;		

000000	013746	000000G		.SBTTL	M\$EMS_18 ERROR MESSAGE SUBROUTINES	
			M\$EMS_18:			
000004	012746	000000G		MOV	CDISK, -(SP)	;
000010	012746	0000002		MOV	0FBD.18, -(SP)	
000014	010600			MOV	02, -(SP)	
000016	104414			MOV	SP, R0	; SP, *
000020	004737	011732'		TRAP	14	
000024	062706	000006		JSR	PC, EMS_RP	;
000030	000207			ADD	06, SP	;
				RTS	PC	

6719

6720

6718

; Routine Size: 13 words, Routine Base: \$CODE\$ + 14236
; Maximum stack depth per invocation: 5 words

H2

ZRQAM2 RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0227
Page 227
(84)

: 6722 1 BGNMSG (EMS_21);

000000	004737	000000V	EMS.21::	.SBTTL EMS.2 ERROR MESSAGE SUBROUTINES	
000004	104423			JSR PC,M\$EMS.21	;
000006	000207			TRAP 23	
				RTS PC	

6722

: Routine Size: 4 words, Routine Base: \$CODE\$ + 14270
: Maximum stack depth per invocation: 2 words

: 6723 2 EMS_RP1 ();
: 6724 1 ENDMSG;

! CONTENTS OF RETURN PACKET

000000	004737	012052'	M\$EMS.21:	.SBTTL M\$EMS.21 ERROR MESSAGE SUBROUTINES	
000004	000207			JSR PC,EMS.RP1	;
				RTS PC	;

6723

6722

: Routine Size: 3 words, Routine Base: \$CODE\$ + 14300
: Maximum stack depth per invocation: 1 word

ZRQAM, RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0228
Page 228
(85)

; 6725 1 BGNMSG (EMS_22) !CONTENTS OF DUP BUFFER ZZZ

000000 004737 000000V EMS_22: .SBTTL EMS_22 ERROR MESSAGE SUBROUTINES
000004 104423 :JSR PC,M\$EMS_22 ;
000006 000207 TRAP 23
RTS PC

6725

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14306
; Maximum stack depth per invocation: 2 words

; 6726 2 EMS_DBN (); !ZZZ
; 6727 1 ENDMSG; !ZZZ

000000 004737 011004' M\$EMS_22: .SBTTL M\$EMS_22 ERROR MESSAGE SUBROUTINES
000004 000207 :JSR PC,EMS_DBN ;
RTS PC ;

6726
6725

; Routine Size: 3 words, Routine Base: \$CODE\$ + 14316
; Maximum stack depth per invocation: 1 word

JP

ZRQAMP
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0229
Page 229
(86)

; 6728 1 BGNMSG (EMS_24);

000000	004737	000000V	EMS.24::	.SBTTL EMS.24 ERROR MESSAGE SUBROUTINES	
000004	104423			JSR PC,M\$EMS.24	;
000006	000207			TRAP 23	
				RTS PC	

6728

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14324
; Maximum stack depth per invocation: 2 words

; 6729	2	PRINTB (EBD_24, .CDISK);	! "DISK XXX WENT TO THE AVAILABLE STATE"
; 6730	2	EMS_RP ();	! PRINTX RELEVANT RETPKT FIELDS
; 6731	1	ENDMSG;	

000000	013746	000000G	M\$EMS.24:	.SBTTL M\$EMS.24 ERROR MESSAGE SUBROUTINES	
000004	012746	000000G		MOV CDISK, -(SP)	;
000010	012746	000002		MOV EBD_24, -(SP)	
000014	010600			MOV #2, (SP)	
000016	104414			MOV SP, R0	; SP, +
000020	004737	011732		TRAP 14	
000024	062706	000006		JSR PC, EMS_RP	;
000030	000207			ADD #6, SP	;
				RTS PC	

6729

6730
6728

; Routine Size: 13 words, Routine Base: \$CODE\$ + 14334
; Maximum stack depth per invocation: 5 words

ZRQAMP RD/RX EXERCISER
V01.6 ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0230
Page 230
(87)

; 6732 1 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

```

6732

; Routine Size: 4 words, Routine Base: \$CODE\$ + 14366
; Maximum stack depth per invocation: 2 words

; 6733 2 EMS_ERR ();
; 6734 1 ENDMSG;

! PRINT ALL RELEVANT DATA ON DETECTING AN ERROR

```

000000 004737 013560'      .SBTTL M$EMS_30 ERROR MESSAGE SUBROUTINES
000004 000207      M$EMS_30: JSR PC,EMS_ERR
      RTS PC

```

6733

6732

; Routine Size: 3 words, Routine Base: \$CODE\$ + 14376
; Maximum stack depth per invocation: 1 word

; 6735 1
; 6736 1 end
; 6737 1
; 6738 0 eludom

OTS external references

```

.GLOBAL $SAVE5, $SAVE4, $SAVE3, $SAVE2
.GLOBAL BL$DIV, BL$MOD, BL$MUL

```

PSECT SUMMARY

Psect Name	Words	Attributes
\$OWN\$	74	RW, D, LCL, REL, CON
\$CODE\$	3202	RO, I, LCL, REL, CON
\$PLIT\$	12	RO, D, LCL, REL, CON

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
DISK\$USER2:[POWERS]ZRQADO.L16;3	404	294	72	21	00:00.2

L.2

ZRQAM2
V01.6

RD/RX EXERCISER
ERROR MESSAGE SUBROUTINES

11-Apr-1984 11:56:01
11-Apr-1984 11:45:02

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL1;6

SEQ 0231
Page 231
(87)

COMMAND QUALIFIERS

BLISS/PDP11 ZRQADO.BL1/LIST=ZRQADO.LS1/OBJECT=ZRQADO.OB1/SOURCE=PAGE:53

; Size: 3032 code + 6503 data words
; Run Time: 02:48.2
; Elapsed Time: 05:13.5
; Lines/CPU Min: 2403
; Lexemes/CPU-Min: 23961
; Memory Used: 677 pages
; Compilation Complete

M2

ZRQAM3

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ_0232
Page 1
(1)

```

: 0001 0 module ZRQAM3 (
: 0002 0
: 0003 0 *title 'RD/RX EXERCISER'
: 0004 0 ident = 'V01.6',
: 0005 0 addressing_mode (absolute),
: 0006 0 environment (noeie)
: 0007 0 ) =
: 0008 0
: 0009 1 begin
: 0010 1
: 0011 1 *sbttl 'DECLARATIONS'
: 0012 1
: 0013 1 library 'ZRQADO.L16'; ! RDRX EXERCISER GLOBAL LIBRARY
: 0014 1
: 0015 1 require 'BLSMAC.REQ'; ! DIAGNOSTIC SUPERVISOR LIBRARY
: 1506 1
: 1507 1 EQUALS;
: 1508 1
: 1509 1 forward routine ! ROUTINES APPEAR IN THIS ORDER
: 1510 1 INIT_TEST : novalue, ! INDENTATION IMPLIES CALLED SUBROUTINE
: 1511 1 DRIVER_INIT : novalue,
: 1512 1 CTLR_INIT : novalue,
: 1513 1 INI_CTLR_DAT : novalue,
: 1514 1 REG_EXIST,
: 1515 1 VEC_BR_TEST,
: 1516 1 INT_GEN,
: 1517 1 HARD_INIT,
: 1518 1 INI_RRING : novalue,
: 1519 1 SET_CTLR_CHAR,
: 1520 1 UNIT_INIT : novalue,
: 1521 1 DR_ERR : novalue,
: 1522 1 ACCESS : novalue,
: 1523 1 MULTI_DRIVE : novalue,
: 1524 1 MD_INIT : novalue,
: 1525 1 INIT_IO_BUFF : novalue,
: 1526 1 FATAL_ERROR : novalue,
: 1527 1 QIO_OK,
: 1528 1 QIO_OUT,
: 1529 1 QIO_GEN : novalue,
: 1530 1 GET_RANDOM : novalue,
: 1531 1 QIO_UNIT : novalue,
: 1532 1 QIO_FUNC : novalue,
: 1533 1 DUP : NOVALUE, !???
: 1534 1 DUPWRITDBN : NOVALUE, !???
: 1535 1 DUPREDDBN : NOVALUE, !???
: 1536 1 DUPCOMMAND : NOVALUE, !???
: 1537 1 DUPIDLE : NOVALUE, !???
: 1538 1 QIO_LBN : novalue,
: 1539 1 QIO_SIZE : novalue,
: 1540 1 FILL_BUFF : novalue,
: 1541 1 PROC RETPKT : novalue,
: 1542 1 QIO_RETPKT : NOVALUE, !???
: 1543 1 DUP_COMPARE : NOVALUE, !???

```

N2

ZRQAM3
V01.6RD/RX EXERCISER
DECLARATIONS11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0233
Page 2
(1)

```

: 1544 1      IO_RETPKT : novalue,
: 1545 1      FSET_UPAR : novalue,
: 1546 1      HARD_ERROR : novalue,
: 1547 1      ERR_HRD_RTNE : novalue,
: 1548 1      ERR_HRD_RTNE_APT : novalue,
: 1549 1      UPD_IO_TALLY : novalue,
: 1550 1      OVF_CHK : novalue,
: 1551 1      ROUND_OUTPUT : novalue,
: 1552 1      HOST_WRT_CHK,
: 1553 1      !      ERR_HRD_RTNE : novalue,
: 1554 1      !      ERR_HRD_RTNE_APT : novalue,
: 1555 1      SWEEP : novalue,
: 1556 1      RPS_REM,
: 1557 1      DR_RETPKT : novalue,
: 1558 1      AZINT0 : L$ISR novalue,
: 1559 1      AZINT : novalue,
: 1560 1      !      FATAL_ERROR : novalue,
: 1561 1      !      POLL_CPING : novalue,
: 1562 1      !      POLL_RRING : novalue,
: 1563 1      !      DUP_RSP : NOVALUE,
: 1564 1      !ZZZ
: 1565 1      DISK_RSP : novalue,
: 1566 1      SEQUEN : novalue,
: 1567 1      SCAN_ERRLOG : novalue,
: 1568 1      ERR_SOFT_RTNE : novalue,
: 1569 1      ERR_SOFT_RTNE_APT : novalue,
: 1570 1      SOFT_ERROR : novalue,
: 1571 1      DATAGM : novalue,
: 1572 1      !      ERR_SOFT_RTNE : novalue,
: 1573 1      !      ERR_SOFT_RTNE_APT : novalue,
: 1574 1      !      SOFT_ERROR : novalue,
: 1575 1      external
: 1576 1      CST : blockvector [MAX_CTLR, CST_LEN, word] field (CST_FIELDS),
: 1577 1      ! RUN-TIME CONTROLLER STATUS TABLES
: 1578 1      CST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 1579 1      ! CONTROLLER STATUS TABLE ADDRESS OF "CURRENT" CONTROLLER
: 1580 1      DCT : blockvector [MAX_CTLR, DCT_LEN, word] field (DCT_FIELDS),
: 1581 1      ! DRIVER CONTROLLER TABLES
: 1582 1      DCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 1583 1      ! ADDRESS OF "CURRENT" DRIVER CONTROLLER TABLE
: 1584 1      RDRX_ADDR : ref rdx field (RC_REG),
: 1585 1      ! DEVICE ADDRESS OF "CURRENT" CONTROLLER
: 1586 1      IRDRX_ADDR : ref rdx field (RC_REG),
: 1587 1      ! DEVICE ADDRESS OF INTERRUPTING CONTROLLER
: 1588 1      BST : BLOCKVECTOR [MAX_UNITS, 2, WORD],
: 1589 1      ! BLOCK SEQUENCE TABLE FOR SEQUENTIAL LBN (VS
: 1590 1      ! RANDOM SEEK) MODE
: 1591 1      TALLY : vector [MAX_UNITS * TALLY_LEN, word] field (T_FIELDS),
: 1592 1      ! STATISTICS TABLES
: 1593 1      T_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 1594 1      ! ADDRESS OF STATISTICS TABLE (TALLY) FOR CURRENT UNIT
: 1595 1      DUPPKT : BLOCK [257, WORD] FIELD (DP_FIELDS),
: 1596 1      ! INFO FROM RECEIVE AND SEND COMMANDS
: 1596 1      ! BUFFER FOR DUP
: 1596 1      ZZZ

```

B3

 ZRQAMS
 V01.6

 RD/RX EXERCISER
 DECLARATIONS

 11-Apr-1984 11:08:35
 11-Apr-1984 11:08:22

 VAX-11 B1:99-16 V4.0-579
 DISK1USER2:[POWERS]ZRQADO.BL2:6

```

1597 1   TRK_SGN : VECTOR [MAX_UNITS, BYTE, SIGNED], !CURRENT TK DIRECTION      ZZZ
1598 1   RDM_CNT : WORD, !NO. OF RANDOM NOS. KEEP\ \      ZZZ
1599 1   RANDOM : VECTOR [RDM_LEN, WORD], !RAND NO TABLE TOGET//HER      ZZZ
1600 1   C_ERR_TBL : blockvector [MAX_CTLR, C_ERR_LEN, word] field (C_ERR_FIELDS),
1601 1   ! STATISTICS TABLE FOR CONTROLLER ERRORS
1602 1   MSCP_PKT : blockvector [PKT_CNT, PKT_LEN, word] field (PKT_FIELDS),
1603 1   ! MSCP PACKET POOL
1604 1   IPKT_ADDR : ref block [PKT_LEN, word] field (PKT_FIELDS),
1605 1   ! ADDRESS OF AN MSCP PACKET (INTERRUPT PROCESSING)
1606 1   PKT_USE : vector [PKT_CNT, byte, signed],
1607 1   ! MSCP PACKET POOL ALLOCATION TABLE
1608 1   RETPKT : blockvector [RP_CNT, RP_LEN, word] field (RP_FIELDS),
1609 1   ! RETURN PACKET POOL
1610 1   RP_USE : vector [RP_CNT, byte, signed],
1611 1   ! RETURN PACKET POOL ALLOCATION TABLE
1612 1   RP_INDX : word, ! CURRENT RETURN PACKET INDEX
1613 1   RP_ADDR : ref block [RP_LEN, word] field (RP_FIELDS),
1614 1   ! CURRENT RETURN PACKET ADDRESS
1615 1   LOG_PKT : blockvector [EP_CNT + 1, EP_LEN, word] field (EP_FIELDS),
1616 1   ! ERROR-LOG PACKET SAVE AREA
1617 1   BUFF_ADDR : vector [MAX_BUF_CNT],
1618 1   ! TABLE OF I/O BUFFER DESCRIPTORS
1619 1   BUFF_OWN : vector [MAX_BUF_CNT, byte, signed],
1620 1   ! I/O BUFFER OWNERSHIP (CONTROLLER NUMBER)
1621 1   IODQ : vector [IODQ_LEN, byte],
1622 1   ! I/O DONE QUEUE - CIRCULAR QUEUE OF RETPKT INDECS
1623 1   IODQ_IN : word,
1624 1   ! I/O DONE QUEUE IN POINTER
1625 1   IODQ_OUT : word,
1626 1   ! I/O DONE QUEUE OUT POINTER
1627 1   ENTRY_REASON : byte,
1628 1   ! CURRENT OPERATOR COMMAND
1629 1   EOP_FLAG : byte,
1630 1   ! END-OF-PASS FLAG
1631 1   DUP_FLAGS : WORD,
1632 1   ! DUP FLAGS ZZZ
1633 1   CCTLR : word,
1634 1   ! NUMBER OF "CURRENT" CONTROLLER
1635 1   CDISK : word,
1636 1   ! CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
1637 1   CUOFF : word,
1638 1   ! CURRENT UNIT CST OFFSET
1639 1   CTLR_CNT : word,
1640 1   ! TOTAL NUMBER OF CONFIGURED CONTROLLERS
1641 1   DUR : vector [MAX_UNITS, byte],
1642 1   ! DROP UNIT REASON
1643 1   QIO : vector [MAX_CTLR, byte],
1644 1   ! NUMBER OF OUTSTANDING QIOs PER CONTROLLER
1645 1   FREE_MEM_ADDR,
1646 1   ! START OF FREE MEMORY
1647 1   BYTS_PER_QIO : word,
1648 1   ! SIZE (BYTES) OF AN I/O BUFFER
1649 1   ST_CODE : word,
1650 1   ! CURRENT STATUS CODE
1651 1   SB_CODE : word,
1652 1   ! CURRENT SUB-CODE
1653 1   STEP : word,
1654 1   ! CURRENT STEP IN HARD_INIT
1655 1   OF_RC : signed word,
1656 1   ! OFFSET (0 OR 2) TO READ IP OR SA
1657 1   SA_REG : word,
1658 1   ! STORAGE FOR SA REGISTER READS AND WRITES
1659 1   CMD_TIME : word,
1660 1   ! COMMAND TIMEOUT VALUE (IN SECONDS)
1661 1   NEX : word,
1662 1   ! NON-EXISTENT MEMORY TRAP INDICATOR
1663 1   CRN_LOW : word,
1664 1   ! COMMAND REF NUMBER OF LAST COMMAND SENT
1665 1   CRN_HIGH : word,
1666 1   ! COMMAND REF NUMBER (HI ORDER)
1667 1   CREDIT_BAL : word,
1668 1   ! CREDIT BALANCE
1669 1   NEXT_PKT_USE : byte,
1670 1   ! POINTER TO NEXT ENTRY IN PKT_USE TABLE
1671 1   HOURS : byte,
1672 1   ! TIME OF DAY (HOURS)
1673 1   MINUTES : byte,
1674 1   ! TIME OF DAY (MINUTES)
1675 1   CLK_TICKS : word,
1676 1   ! TIME OF DAY (LINE-CLOCK TICKS)
1677 1   CLK_PRESENT : byte,
1678 1   ! FLAG INDICATES IF LINE-CLOCK PRESENT
1679 1   HOE_FLAG : byte,
1680 1   ! FLAG INDICATES IF "HALT ON ERROR" FLAG SET
1681 1   FORCED_ERROR : byte,
1682 1   ! "FORCED ERROR" DETECTED IN LAST READ

```

C3

ZRQAM3
V01.6

RD/RX EXERCISER
DECLARATIONS

11 Apr 1984 11:08:35
11 Apr 1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK1USER2:[POWERS]ZRQADO.BL2;6

SEQ 0235
Page 4
(1)

```

: 1650 1 FER_LBN : word,
: 1651 1 FER_BC : word,
: 1652 1 INIT_OCCURED : byte,
: 1653 1 ADDR_VECT_OK : byte,
: 1654 1 S_PATTERN : WORD,
: 1655 1 S_DUPPKT : WORD,
: 1656 1 P_INDEX : SIGNED WORD,
: 1657 1 DBM12,
: 1658 1 DBM18,
: 1659 1 DBM19,
: 1660 1 DBM20,
: 1661 1 DBM21,
: 1662 1 DBM22,
: 1663 1 DBM23,
: 1664 1 DBM25,
: 1665 1 DBM26,
: 1666 1 DBM27,
: 1667 1 DBM29,
: 1668 1 DBM108,
: 1669 1 DBM109,
: 1670 1 DBM111,
: 1671 1 DBM112,
: 1672 1 DBM120,
: 1673 1 DBM121,
: 1674 1 EH_0,
: 1675 1 EH_1,
: 1676 1 EH_2,
: 1677 1 EH_3,
: 1678 1 EH_4,
: 1679 1 EH_5,
: 1680 1 EH_6,
: 1681 1 EH_7,
: 1682 1 EH_8,
: 1683 1 EH_9,
: 1684 1 EH_10,
: 1685 1 EH_12,
: 1686 1 EH_13,
: 1687 1 MSG_02,
: 1688 1 MSG_03,
: 1689 1 EGS_02,
: 1690 1 EGD_10,
: 1691 1 EGD_11,
: 1692 1 EGD_12,
: 1693 1 EGD_13,
: 1694 1 EGD_14,
: 1695 1 EGD_15,
: 1696 1 EGD_16,
: 1697 1 EGD_17,
: 1698 1 EGD_18,
: 1699 1 EGD_19,
: 1700 1 EGD_20,
: 1701 1 EGD_21,
: 1702 1 EGD_22,

```

```

      ! LBN OF THE "FORCED ERROR" BLOCK
      ! BYTE COUNT OF THE "FORCED ERROR" BLOCK
      ! EXERCISER INITIALIZATION COMPLETE
      ! FLAG INDICATES IF ADDRESS/VECTOR TEST PASSED
!PATTERN WRITTEN TO DBNS          ZZZ
!DBN BYTE COUNTER                 ZZZ
!CURRENT MESSAGE PACKET INDEX     ZZZ

```

[illegible]

03

ZRQAM3
V01.6

RD/RX EXERCISER
DECLARATIONS

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11:16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0236
Page 5
(1)

```

: 1703 1      EGD_23,
: 1704 1      EGD_24,
: 1705 1      EGH_30,
: 1706 1      DF_MSG,
: 1707 1      HRD_MSG,
: 1708 1      SFT_MSG,
: 1709 1      HRD_SUB,
: 1710 1      CRLF,
: 1711 1      SWP_ERROR : word,          ! HARD ERROR LIMIT FOR DROPPING UNIT
: 1712 1      SWP_XFER : word,          ! TRANSFER LIMIT FOR DROPPING UNIT
: 1713 1      SWP_FLAGS : word,         ! FLAGS (SEE DOCUMENTATION)
: 1714 1      DUPROUND : word,         ! DUP TESTING RATIO
: 1715 1      SWP_RAT : word,          ! RD51/52 OPERATION RATIO
: 1716 1      SWP_DPAT : word,         ! DATA PATTERN NUMBER
: 1717 1      SWP_UCNT : word,         ! USER DATA PATTERN COUNT
: 1718 1      SWP_TIME : word,         ! TIME OF DAY
: 1719 1      SWP_UDPAT : vector [MAX_UDP_CNT, word], ! USER DATA PATTERN
: 1720 1      L$LUN,
: 1721 1      L$UNIT;
: 1722 1
: 1723 1      psect
: 1724 1          own = $GGG$(read, nowrite, execute, local, concatenate);
: 1725 1
: 1726 1      own
: 1727 1          COMM_AREA : blockvector [MAX_CTLR, COMM_LEN, word] field (COM_FIELDS),
: 1728 1                      ! COMMUNICATIONS AREA BETWEEN HOST AND AZTEC CONTROLLERS
: 1729 1      !!ZZZ    BST : vector [MAX_UNITS, word, signed],
: 1730 1                      ! BLOCK SEQUENCE TABLE FOR SEQUENTIAL LBN (VS. RANDOM SEEK) MODE
: 1731 1      DPST : vector [MAX_UNITS, byte], ! DATA PATTERN SEQUENCE TABLE
: 1732 1      MAX_LBN : vector [MAX_UNITS, word], ! LARGEST LBN ALLOWED
: 1733 1      STORAGE : vector [MAX_UNITS, word], ! DUMMY STORAGE
: 1734 1      ICOM_ADDR : ref block [COMM_LEN, word] field (COM_FIELDS),
: 1735 1                      ! ADDRESS OF INTERRUPTING CONTROLLER'S COMMUNICATION AREA
: 1736 1      ICST_ADDR : ref block [CST_LEN, word] field (CST_FIELDS),
: 1737 1                      ! ADDRESS OF INTERRUPTING CONTROLLER'S CST
: 1738 1      IDCT_ADDR : ref block [DCT_LEN, word] field (DCT_FIELDS),
: 1739 1                      ! ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 1740 1      INT_ADDR : vector [MAX_CTLR] initial (AZINT0 * (AZINT1, AZINT2, AZINT3) *),
: 1741 1                      ! INTERRUPT SERVICE ROUTINE ADDRESS TABLE
: 1742 1      !!ZZZ    RDM_CNT : word initial (RDM_LEN), ! NUMBER OF RANDOM NUMBERS \ KEEP
: 1743 1      !!ZZZ    RANDOM : vector [RDM_LEN, word], ! RANDOM NUMBER TABLE / TOGETHER
: 1744 1      ICTLR : word, ! INTERRUPTING CONTROLLING NUMBER
: 1745 1      MX1 : signed word, ! MSCP PKT INDEX FOR FIRST QIO
: 1746 1      MX2 : signed word, ! MSCP PKT INDEX FOR SECOND QIO
: 1747 1      MAD1 : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 1748 1                      ! ADDRESS OF MSCP PACKET FOR FIRST QIO
: 1749 1      MAD2 : ref block [PKT_LEN, word] field (PKT_FIELDS),
: 1750 1                      ! ADDRESS OF MSCP PACKET FOR SECOND QIO
: 1751 1      LAST_PKT : blockvector [MAX_CTLR, LAST_PKT_LEN, word] field (LAST_PKT_FIELDS),
: 1752 1                      ! SAVE AREA FOR INFO ABOUT LAST RESPONSE PACKET
: 1753 1      PAT02 : vector [2] initial (1, ! PATTERN 2
: 1754 1          %0'000000'),
: 1755 1      PAT03 : vector [2] initial (1, ! PATTERN 3

```

F3

ZRGAM3
V01.6RD/RX EXERCISER
DECLARATIONS11-Apr-1984 11:08:35 VAX-11 B11ss-16 V4.0-579
11-Apr-1984 11:08:22 DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0237
Page 6
(1)

```

: 1756 1      #o'177777'),
: 1757 1      PAT04 : vector [2] initial (1,          ! PATTERN 4
: 1758 1      #o'105613'),
: 1759 1      PAT05 : vector [2] initial (1,          ! PATTERN 5
: 1760 1      #o'031463'),
: 1761 1      PAT06 : vector [2] initial (1,          ! PATTERN 6
: 1762 1      #o'030221'),
: 1763 1      PAT07 : vector [17] initial (16,        ! PATTERN 7
: 1764 1      #o'000001', #o'000003', #o'000007', #o'000017',
: 1765 1      #o'000037', #o'000077', #o'000177', #o'000377',
: 1766 1      #o'000777', #o'001777', #o'003777', #o'007777',
: 1767 1      #o'017777', #o'037777', #o'077777', #o'177777'),
: 1768 1      PAT08 : vector [17] initial (16,        ! PATTERN 8
: 1769 1      #o'177776', #o'177774', #o'177770', #o'177760',
: 1770 1      #o'177740', #o'177700', #o'177600', #o'177400',
: 1771 1      #o'177000', #o'176000', #o'174000', #o'170000',
: 1772 1      #o'160000', #o'140000', #o'100000', #o'000000'),
: 1773 1      PAT09 : vector [17] initial (16,        ! PATTERN 9
: 1774 1      rep 3 of (#o'000000'), rep 3 of (#o'177777'),
: 1775 1      rep 2 of (#o'000000'), rep 2 of (#o'177777'),
: 1776 1      #o'000000', #o'177777', #o'000000', #o'177777',
: 1777 1      #o'000000', #o'177777'),
: 1778 1      PAT10 : vector [2] initial (1,          ! PATTERN 10
: 1779 1      #o'133331'),
: 1780 1      PAT11 : vector [17] initial (16,        ! PATTERN 11
: 1781 1      rep 3 of (#o'052525'), rep 3 of (#o'125252'),
: 1782 1      rep 2 of (#o'052525'), rep 2 of (#o'125252'),
: 1783 1      #o'052525', #o'125252', #o'052525', #o'125252',
: 1784 1      #o'052525', #o'125252'),
: 1785 1      PAT12 : vector [21] initial (20,        ! PATTERN 12
: 1786 1      rep 3 of (#o'026455'), rep 3 of (#o'151322'),
: 1787 1      rep 2 of (#o'026455'), rep 2 of (#o'151322'),
: 1788 1      rep 2 of (#o'026455'),
: 1789 1      #o'151322', #o'026455', #o'151322', #o'026455',
: 1790 1      #o'151322', #o'026455', #o'151322', #o'026455'),
: 1791 1      PAT13 : vector [2] initial (1,          ! PATTERN 13
: 1792 1      #o'066666'),
: 1793 1      PAT14 : vector [17] initial (16,        ! PATTERN 14
: 1794 1      #o'000001', #o'000002', #o'000004', #o'000010',
: 1795 1      #o'000020', #o'000040', #o'000100', #o'000200',
: 1796 1      #o'000400', #o'001000', #o'002000', #o'004000',
: 1797 1      #o'010000', #o'020000', #o'040000', #o'100000'),
: 1798 1      PAT15 : vector [17] initial (16,        ! PATTERN 15
: 1799 1      #o'177776', #o'177775', #o'177773', #o'177767',
: 1800 1      #o'177757', #o'177737', #o'177677', #o'177577',
: 1801 1      #o'177377', #o'176777', #o'175777', #o'173777',
: 1802 1      #o'167777', #o'157777', #o'137777', #o'077777'),
: 1803 1      PAT16 : vector [17] initial (16,        ! PATTERN 16
: 1804 1      rep 3 of (#o'133331'), rep 3 of (#o'155554'),
: 1805 1      rep 2 of (#o'133331'), rep 2 of (#o'155554'),
: 1806 1      #o'133331', #o'155554', #o'133331', #o'155554',
: 1807 1      #o'133331', #o'155554'),
: 1808 1      PAT17 : vector [22] initial (21,        ! PATTERN 17

```


F3

ZRRQM3
VO1.6RD/RX EXERCISER
DECLARATIONS11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1:ss-16 V4.0-579
DISK\$USER2:[POWER5]ZRRQADO.BL2;6SEQ 0238
Page 7
(1)

```

: 1809 1      %o'000000', rep 2 of (%o'106466'),
: 1810 1      rep 3 of (%o'071311'), rep 4 of (%o'106466'),
: 1811 1      rep 5 of (%o'071311'), rep 6 of (%o'106466')),
: 1812 1      PAT18 : vector [22] initial (21,                ! PATTERN 18
: 1813 1      %o'106466', %o'000000', %o'071311',
: 1814 1      rep 3 of (%o'106466'), rep 4 of (%o'071311'),
: 1815 1      rep 5 of (%o'106466'), rep 6 of (%o'071311')),
: 1816 1      PAT19 : vector [22] initial (21,                ! PATTERN 19
: 1817 1      %o'000000', rep 2 of (%o'134631'),
: 1818 1      rep 3 of (%o'043146'), rep 4 of (%o'134631'),
: 1819 1      rep 5 of (%o'043146'), rep 6 of (%o'134631')),
: 1820 1      PAT20 : vector [22] initial (21,                ! PATTERN 20
: 1821 1      %o'134631', %o'000000', %o'043146',
: 1822 1      rep 3 of (%o'134631'), rep 4 of (%o'043146'),
: 1823 1      rep 5 of (%o'134631'), rep 6 of (%o'043146')),
: 1824 1      PAT21 : vector [2] initial (1,                  ! PATTERN 21
: 1825 1      %o'000000'),                                     ! (LBN)
: 1826 1      DPA_TBL : vector [DP_CNT] initial              ! DATA PATTERN ADDRESS TABLE
: 1827 1      (RDM_CNT, PAT02, PAT03, PAT04, PAT05,
: 1828 1      PAT06, PAT07, PAT08, PAT09, PAT10, PAT11,
: 1829 1      PAT12, PAT13, PAT14, PAT15, PAT16, PAT17,
: 1830 1      PAT18, PAT19, PAT20, PAT21),
: 1831 1      BST_CNT : word initial (0),                    ! CURRENT SEQUENTIAL BLOCK COUNT
: 1832 1      BST_DEV : word initial (0),                    ! CURRENT SEQUENTIAL BLOCK DEVICE
: 1833 1      CURRENT_VECTOR : word,                        ! CURRENT DEVICE'S VECTOR ADDRESS
: 1834 1      BRLEVEL : word,                               ! CURRENT DEVICE'S BR LEVEL
: 1835 1      DUOFF : word,                                  ! DUP OFFSET INTO CST                ZZZ
: 1836 1      DRS_START,                                     ! START OF THE SUPERVISOR
: 1837 1      APT_MODE : byte initial (byte (FALSE)),       ! FLAG SET IF EXERCISER RUNNING UNDER APT
: 1838 1      MAIL_BOX_TESTNUM,                             ! ADDRESS OF TEST NUMBER LOCATION IN APT MAIL-BOX
: 1839 1      MAIL_BOX_SUBST,                               ! ADDRESS OF SUB-TEST NUMBER LOCATION IN APT MAIL-BOX
: 1840 1      COMPARE_DATA : byte,                          ! FLAG CLEARED TO BYPASS HOST COMPARES
: 1841 1      DRS_FLAGS : word,                             ! FLAGS USED IN START/RESTART OF THE EXERCISER
: 1842 1      RD_MAX_SEQ_CNT : word,                        ! COUNT USED IN SEQUENTIAL ACCESS OPERATIONS
: 1843 1      RX_MAX_SEQ_CNT : word;
: 1844 1
: 1845 1      external routine
: 1846 1      NEX_TRAP : L$ISR novalue,
: 1847 1      TIME : L$ISR novalue,
: 1848 1      SET_CPAR : novalue,
: 1849 1      SET_UPAR : novalue,
: 1850 1      OUT_IODQ,
: 1851 1      IN_IODQ : novalue,
: 1852 1      GET_PKT,
: 1853 1      PUT_PKT : novalue,
: 1854 1      GET_RETPKT,
: 1855 1      PUT_RETPKT : novalue,
: 1856 1      GET_IO_BUFF : novalue,
: 1857 1      PUT_IO_BUFF : novalue,
: 1858 1      PUTA_BUFF : novalue,
: 1859 1      SEND,
: 1860 1      WAIT : novalue,
: 1861 1      MODULAS,

```

!ZZZ

63

ZRQAMS
V01.6

RD/RX EXERCISER
DECLARATIONS

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1:99-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0239
Page 8
(1)

:	1862	1	DROP_CTLR : novalue.
:	1863	1	DRV_CTLERR : novalue.
:	1864	1	EMS_RP1 : novalue.
:	1865	1	EMS_EL : novalue.
:	1866	1	EMS_CMP : novalue.
:	1867	1	EMS_ERR : novalue.
:	1868	1	EMS_10 : novalue.
:	1869	1	EMS_12 : novalue.
:	1870	1	EMS_13 : novalue.
:	1871	1	EMS_14 : novalue.
:	1872	1	EMS_18 : novalue.
:	1873	1	EMS_21 : novalue.
:	1874	1	EMS_22 : NOVALUE.
:	1875	1	EMS_24 : novalue.
:	1876	1	EMS_30 : novalue.

!ZZZ

H3

ZRQAM3
V01.6RD/RX EXERCISER
TEST SECTION11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0240
Page 9
(2)

```

: 1877 1 *sbttl 'TEST SECTION'
: 1878 1
: 1879 1 !+
: 1880 1 !      THIS SECTION CONTAINS THE TOP-LEVEL TEST CODE FOR THE RDRX EXERCISER.
: 1881 1 !      THE EXERCISER CONSISTS OF ONE TEST WHICH IS SUBDIVIDED INTO A NUMBER OF
: 1882 1 !      SUBTESTS. ALL SUBTESTS ARE DECLARED WITHIN THIS BLOCK.
: 1883 1 !-
: 1884 1
: 1885 3 RGNTST;
: 1886 3
: 1887 3 local
: 1888 3     DUMMY_0 : word;
: 1889 3     DUMMY_1 : word;
: 1890 3
: 1891 3 EOP_FLAG = TRUE;
: 1892 3 COMPARE_DATA = TRUE;
: 1893 3 DUP_FLAGS = .DUP_FLAGS AND (NOT SWP_DINT);
: 1894 3 HOE_FLAG = FALSE;
: 1895 3 FORCED_ERROR = FALSE;
: 1896 3
: 1897 3   incr I from 0 to PKT_CNT - 1 do
: 1898 4     begin
: 1899 4
: 1900 4       incr J from 0 to PKT_LEN - 1 do
: 1901 4         MSCP_PKT [.I, .J, 0, 16, 0] = 0;
: 1902 4
: 1903 4       MSCP_PKT [.I, RSP_RECEIVED] = FALSE;
: 1904 3     end;
: 1905 3
: 1906 3   incr I from 0 to RP_CNT - 1 do
: 1907 3     incr J from 0 to RP_LEN - 1 do
: 1908 3       RETPKT [.I, .J, 0, 16, 0] = 0;
: 1909 3
: 1910 3   incr I from 0 to EP_CNT do
: 1911 4     begin
: 1912 4
: 1913 4       incr J from 0 to EP_LEN - 1 do
: 1914 4         ELOG_PKT [.I, .J, 0, 16, 0] = 0;
: 1915 4
: 1916 4       ELOG_PKT [.I, EL_CONTENTS] = EMPTY;
: 1917 3     end;
: 1918 3
: 1919 4   IF BIT_TST (SWP_FLAGS, SWF_CWC)
: 1920 3   then
: 1921 3     SWP_FLAGS = .SWP_FLAGS and (not SWF_HWC);
: 1922 3
: 1923 4   IF BIT_TST (SWP_FLAGS, SWF_RDM)
: 1924 3   then
: 1925 3     SWP_FLAGS = .SWP_FLAGS and (not SWF_SEQ);
: 1926 3
: 1927 3   IF not .INIT_OCCURED
: 1928 3   then
: 1929 4     begin

```

! ASSUME NO UNIT AVAILABLE
! ALLOW HOST COMAPRES IF ASKED FOR
! CLEAR DUP INIT FLAG ZZZ
! ASSUME 'HOE' FLAG NOT SET
! INITIALIZE "FORCED ERROR" FLAG

! INITIALIZE PACKET AREA

! INITIALIZE RESPONSE SAVE AREA

! INITIALIZE ERROR-LOG SAVE AREA

! NO SIMULTANEOUS CNTR/HOST WRIE CHECKS

! NO SIMULTANEOUS RANDOM/SEQUENTIAL SELECTS

ZRGAM3
V01.6RD/RX EXERCISER
TEST SECTION11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0241
Page 10
(2)

```

1930 4      DRS_START = .FREE_MEM_ADDR + 2 * (.FREE_MEM_ADDR * 2);      ! START OF SUPERVISOR
1931 4      !
1932 4      ! CAUTION... THE FOLLOWING CODE IS 'KLUGED' TO DETERMINE IF THE EXERCISER IS RUNNING UNDER THE
1933 4      ! APT SUPERVISOR, AND IS DEPENDENT FOR IT'S SUCCESS ON THE KNOWLEDGE OF THE ACTUAL
1934 4      ! APT SUPERVISOR AND THE ADDRESS OF THE MAIL-BOX WITHIN THE SUPERVISOR.
1935 4      !
1936 4      !
1937 4      if (.DRS_START eq1 %0'167') and      ! APT DRS STARTS WITH A JMP INSTRUCTION
1938 5      (not MANUAL)
1939 4      then
1940 5          begin
1941 5              APT_MODE = TRUE;
1942 5              MAIL_BOX_TESTNUM = .DRS_START + %0'62' + %0'6';      ! APT MAIL-BOX IS OFFSET AT OCTAL 62 FROM
1943 5              MAIL_BOX_SUBTST = .DRS_START + %0'62' + %0'4';      ! BEGINNING OF SUPERVISOR
1944 5              end;
1945 4      !
1946 4      NEX = FALSE;
1947 4      CLK_PRESENT = FALSE;
1948 4      SETVEC (4, NEX_TRAP, PRI07);
1949 4      DUMMY_0 = .LINE_CLOCK;
1950 4      DUMMY_1 = 0;
1951 4      CLRVEC (4);
1952 4      !
1953 4      if not .NEX
1954 4      then
1955 5          begin
1956 5              CLK_PRESENT = TRUE;
1957 5              CLK_TICKS = 0;
1958 5              HOURS = .SWP_TIME / 100;
1959 5              MINUTES = (.SWP_TIME mod 100) * 1;
1960 5              !
1961 5              while .MINUTES gequ 60 do
1962 6                  begin
1963 6                      MINUTES = .MINUTES - 60;
1964 6                      HOURS = .HOURS + 1;
1965 6                  end;
1966 5              !
1967 5              HOURS = .HOURS mod 24;
1968 5              end;
1969 4      !
1970 3      end;
1971 3      !
1972 3      if .CLK_PRESENT
1973 3      then
1974 4          begin
1975 4              SETVEC (%0'100', TIME, PRI06);
1976 4              LINE_CLOCK = BIT6;
1977 4              end;
1978 3      !
1979 3      RFLAGS (DRS_FLAGS);
1980 3      !
1981 3      if BIT_TST (DRS_FLAGS, HQE) eq1 HQE
1982 3      then

```

J3

ZRQAM3
V01.6

RD/RX EXERCISER
TEST SECTION

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 811ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0242
Page 11
(2)

```

: 1983 3      HOE_FLAG = TRUE;                ! SET FLAG IF 'HOE' SET
: 1984 3
: 1985 3      INIT_TEST ();                    ! INITIALIZE TEST ENVIRONMENT
: 1986 3
: 1987 3      incr CTLR from 0 to (MAX_CTLR - 1) do  ! FOR EVERY CONTROLLER
: 1988 3
: 1989 3          if (.CST [.CTLR, STATE] eq1 ONLINE) and  ! IF CONTROLLER ONLINE
: 1990 3              (.DCT [.CTLR, STAT] eq1 ONLINE) and
: 1991 4              (.CST [.CTLR, U_CNT] geau 0)
: 1992 3      then
: 1993 3          incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + 4) by UNIT_SIZE do
: 1994 3
: 1995 3              if .CST [.CTLR, .OFFSET + OF_DATA, D_STAT] eq1 ONLINE  ! IF AT LEAST ONE UNIT ALIVE
: 1996 3      then
: 1997 4          begin
: 1998 4              EOP_FLAG = FALSE;
: 1999 4              exitloop;
: 2000 3          end;
: 2001 3
: 2002 3      if not .EOP_FLAG
: 2003 3      then
: 2004 3          MULTI_DRIVE ();
: 2005 1      ENDTST;

```

.TITLE ZRQAM3 RD/RX EXERCISER
.IDENT /V01.6/
.ENABL AMA

```

000000      .PSECT $GGG$, RO
000000      COMM, AREA:
000050          .BLKW 24
000054      DPST: .BLKW 2
000064      MAX_LBN: .BLKW 4
000074      STORAGE: .BLKW 4
000076      ICOM, ADDR:
000100          .BLKW 1
000100      ICST, ADDR:
000102      IDCT, ADDR:
000102      INT, ADDR:
000104          .WORD AZINT0
000106      ICTLR: .BLKW 1
000110      MX1: .BLKW 1
000112      MX2: .BLKW 1
000114      MAD1: .BLKW 1
000116      MAD2: .BLKW 1
000124      LAST, PKT:
000126      PAT02: .BLKW 3
000126      .WORD 1
000126      .WORD 0

```

K3

ZRQAM3
V01.6RD/RX EXERCISER
TEST SECTION11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0243
Page 12
(2)

000130 000001
 000132 177777
 000134 000001
 000136 105613
 000140 000001
 000142 031463
 000144 000001
 000146 030221
 000150 000020
 000152 000001
 000154 000003
 000156 000007
 000160 000017
 000162 000037
 000164 000077
 000166 000177
 000170 000377
 000172 000777
 000174 001777
 000176 003777
 000200 007777
 000202 017777
 000204 037777
 000206 077777
 000210 177777
 000212 000020
 000214 177776
 000216 177774
 000220 177770
 000222 177760
 000224 177740
 000226 177700
 000230 177600
 000232 177400
 000234 177000
 000236 176000
 000240 174000
 000242 170000
 000244 160000
 000246 140000
 000250 100000
 000252 000000
 000254 000020
 000256 000000
 000260 000000
 000262 000000
 000264 177777
 000266 177777
 000270 177777
 000272 000000
 000274 000000
 000276 177777
 000300 177777

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
 PAT08: .WORD -1
 .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
 PAT09: .WORD 0
 .WORD 20
 .WORD 0
 .WORD 0
 .WORD 0
 .WORD 0
 .WORD -1
 .WORD -1
 .WORD 1
 .WORD 0
 .WORD 0
 .WORD -1
 .WORD -1

ZRQAM5
V01.6RD/RX EXERCISER
TEST SECTION11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO,BL2;6SEQ 0244
Page 13
(2)

000302	000000	.WORD	0
000304	177777	.WORD	-1
000306	000000	.WORD	0
000310	177777	.WORD	-1
000312	000000	.WORD	0
000314	177777	.WORD	-1
000316	000001	PAT10: .WORD	1
000320	133331	.WORD	-44447
000322	000020	PAT11: .WORD	20
000324	052525	.WORD	52525
000326	052525	.WORD	52525
000330	052525	.WORD	52525
000332	125252	.WORD	-52526
000334	125252	.WORD	-52526
000336	125252	.WORD	-52526
000340	052525	.WORD	52525
000342	052525	.WORD	52525
000344	125252	.WORD	-52526
000346	125252	.WORD	-52526
000350	052525	.WORD	52525
000352	125252	.WORD	-52526
000354	052525	.WORD	52525
000356	125252	.WORD	-52526
000360	052525	.WORD	52525
000362	125252	.WORD	-52526
000364	000024	PAT12: .WORD	24
000366	026455	.WORD	26455
000370	026455	.WORD	26455
000372	026455	.WORD	26455
000374	151322	.WORD	-26456
000376	151322	.WORD	-26456
000400	151322	.WORD	-26456
000402	026455	.WORD	26455
000404	026455	.WORD	26455
000406	151322	.WORD	-26456
000410	151322	.WORD	-26456
000412	026455	.WORD	26455
000414	026455	.WORD	26455
000416	151322	.WORD	-26456
000420	026455	.WORD	26455
000422	151322	.WORD	-26456
000424	026455	.WORD	26455
000426	151322	.WORD	-26456
000430	026455	.WORD	26455
000432	151322	.WORD	-26456
000434	026455	.WORD	26455
000436	000001	PAT13: .WORD	1
000440	066666	.WORD	66666
000442	000020	PAT14: .WORD	20
000444	000001	.WORD	1
000446	000002	.WORD	2
000450	000004	.WORD	4
000452	000010	.WORD	10

M3

ZRQAM3
V01.6

RD/RX EXERCISER
TEST SECTION

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2:6

SEQ 0245
Page 14
(2)

000454	000020		.WORD	20
000456	000040		.WORD	40
000460	000100		.WORD	100
000462	000200		.WORD	200
000464	000400		.WORD	400
000466	001000		.WORD	1000
000470	002000		.WORD	2000
000472	004000		.WORD	4000
000474	010000		.WORD	10000
000476	020000		.WORD	20000
000500	040000		.WORD	40000
000502	100000		.WORD	-100000
000504	000020	PAT15:	.WORD	20
000506	177776		.WORD	-2
000510	177775		.WORD	-3
000512	177773		.WORD	-5
000514	177767		.WORD	-11
000516	177757		.WORD	-21
000520	177737		.WORD	-41
000522	177677		.WORD	-101
000524	177577		.WORD	-201
000526	177377		.WORD	-401
000530	176777		.WORD	-1001
000532	175777		.WORD	-2001
000534	173777		.WORD	-4001
000536	167777		.WORD	-10001
000540	157777		.WORD	-20001
000542	137777		.WORD	-40001
000544	077777		.WORD	77777
000546	000020	PAT16:	.WORD	20
000550	133331		.WORD	-44447
000552	133331		.WORD	-44447
000554	133331		.WORD	-44447
000556	155554		.WORD	-22224
000560	155554		.WORD	-22224
000562	155554		.WORD	-22224
000564	133331		.WORD	-44447
000566	133331		.WORD	-44447
000570	155554		.WORD	-22224
000572	155554		.WORD	-22224
000574	133331		.WORD	-44447
000576	155554		.WORD	-22224
000600	133331		.WORD	-44447
000602	155554		.WORD	-22224
000604	133331		.WORD	-44447
000606	155554		.WORD	-22224
000610	000025	PAT17:	.WORD	25
000612	000000		.WORD	0
000614	106466		.WORD	-71312
000616	106466		.WORD	-71312
000620	071311		.WORD	71311
000622	071311		.WORD	71311
000624	071311		.WORD	71311

N3

ZRQAM3
V01.6

RD/RX EXERCISER
TEST SECTION

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0246
Page 15
(2)

000626	106466	WORD	-71312
000630	106466	.WORD	-71312
000632	106466	.WORD	-71312
000634	106466	.WORD	-71312
000636	071311	.WORD	71311
000640	071311	.WORD	71311
000642	071311	.WORD	71311
000644	071311	.WORD	71311
000646	071311	.WORD	71311
000650	106466	.WORD	-71312
000652	106466	.WORD	-71312
000654	106466	.WORD	-71312
000656	106466	.WORD	-71312
000660	106466	.WORD	-71312
000662	106466	.WORD	-71312
000664	000025	PAT18: .WORD	25
000666	106466	.WORD	-71312
000670	000000	.WORD	0
000672	071311	.WORD	71311
000674	106466	.WORD	-71312
000676	106466	.WORD	-71312
000700	106466	.WORD	-71312
000702	071311	.WORD	71311
000704	071311	.WORD	71311
000706	071311	.WORD	71311
000710	071311	.WORD	71311
000712	106466	.WORD	-71312
000714	106466	.WORD	-71312
000716	106466	.WORD	-71312
000720	106466	.WORD	-71312
000722	106466	.WORD	-71312
000724	071311	.WORD	71311
000726	071311	.WORD	71311
000730	071311	.WORD	71311
000732	071311	.WORD	71311
000734	071311	.WORD	71311
000736	071311	.WORD	71311
000740	000025	PAT19: .WORD	25
000742	000000	.WORD	0
000744	134631	.WORD	-43147
000746	134631	.WORD	-43147
000750	043146	.WORD	43146
000752	043146	.WORD	43146
000754	043146	.WORD	43146
000756	134631	.WORD	-43147
000760	134631	.WORD	-43147
000762	134631	.WORD	-43147
000764	134631	.WORD	-43147
000766	043146	.WORD	43146
000770	043146	.WORD	43146
000772	043146	.WORD	43146
000774	043146	.WORD	43146
000776	043146	.WORD	43146

134

ZRQAM3
V01.6

RD/RX EXERCISER
TEST SECTION

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ms-16 V4.0-579
DISK#USER2:[POWERS]ZRQADO.BL2;6

SEQ 0247
Page 16
(2)

001000	134631		.WORD	-43147
001002	134631		.WORD	-43147
001004	134631		.WORD	-43147
001006	134631		.WORD	-43147
001010	134631		.WORD	-43147
001012	134631		.WORD	-43147
001014	000025	PAT20:	.WORD	25
001016	134631		.WORD	-43147
001020	000000		.WORD	0
001022	043146		.WORD	43146
001024	134631		.WORD	-43147
001026	134631		.WORD	-43147
001030	134631		.WORD	-43147
001032	043146		.WORD	43146
001034	043146		.WORD	43146
001036	043146		.WORD	43146
001040	043146		.WORD	43146
001042	134631		.WORD	-43147
001044	134631		.WORD	-43147
001046	134631		.WORD	-43147
001050	134631		.WORD	-43147
001052	134631		.WORD	-43147
001054	043146		.WORD	43146
001056	043146		.WORD	43146
001060	043146		.WORD	43146
001062	043146		.WORD	43146
001064	043146		.WORD	43146
001066	043146		.WORD	43146
001070	000001	PAT21:	.WORD	1
001072	000000		.WORD	0
001074	000000G	DPA.TBL:	.WORD	RDM.CNT
001076	000124		.WORD	PAT02
001100	000130		.WORD	PAT03
001102	000134		.WORD	PAT04
001104	000140		.WORD	PAT05
001106	000144		.WORD	PAT06
001110	000150		.WORD	PAT07
001112	000212		.WORD	PAT08
001114	000254		.WORD	PAT09
001116	000316		.WORD	PAT10
001120	000322		.WORD	PAT11
001122	000364		.WORD	PAT12
001124	000436		.WORD	PAT13
001126	000442		.WORD	PAT14
001130	000504		.WORD	PAT15
001132	000546		.WORD	PAT16
001134	000610		.WORD	PAT17
001136	000664		.WORD	PAT18
001140	000740		.WORD	PAT19
001142	001014		.WORD	PAT20
001144	001070		.WORD	PAT21
001146	000000	BST.CNT:	.WORD	0
001150	000000	BST.DEV:	.WORD	0

C4

ZRQAM3
V01.6

RD/RX EXERCISER
TEST SECTION

11 Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1155-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL216

SEQ 0248
Page 17
(2)

001152
001154
001156
001160

001162
001162

000

001164

001166

001170

001172

001174

001176

CURRENT VECTOR:
.BLKW 1
BRLEVEL: .BLKW 1
DUOFF: .BLKW 1
DRS START:
.BLKW 1
APT MODE:
.BYTE 0
.EVEN
MAIL BOX TESTNUM:
.BLKW 1
MAIL BOX SUBTST:
.BLKW 1
COMPARE DATA:
.BLKB 1
.EVEN
DRS FLAGS:
.BLKW 1
RD MAX SEQ CNT:
.BLKW 1
RX MAX SEQ CNT:
.BLKW 1

.GLOBL CST, CST.ADDR, DCT, DCT.ADDR, RDRX.ADDR
.GLOBL IRDRX.ADDR, BST, TALLY, T.ADDR
.GLOBL DUPPKT, TRK.SGN, RDM.CNT, RANDOM
.GLOBL C.ERR.TBL, MSCP.PKT, IPKT.ADDR
.GLOBL PKT.USE, RETPKT, RP.USE, RP.INDX
.GLOBL RP.ADDR, ELOG.PKT, BUFF.ADDR, BUFF.OWN
.GLOBL IOOQ, IOOQ.IN, IOOQ.OUT, ENTRY.REASON
.GLOBL EOP.FLAG, DUP.FLAGS, CCTL, CDISK
.GLOBL CUOFF, CCLR.CN, 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, CREDIT.BAL
.GLOBL NEXT.PKT.USE, HOURS, MINUTES, CLK.TICKS
.GLOBL CLK.PRESENT, MOE.FLAG, FORCED.ERROR
.GLOBL FER.LBN, FER.BC, INIT.OCCURED
.GLOBL ADDR.VECT.OK, S.PATTERN, S.DUPPKT
.GLOBL P.INDEX, DBM12, DBM18, DBM19, DBM20
.GLOBL DBM21, DBM22, DBM23, DBM25, DBM26
.GLOBL DBM27, DBM29, DBM108, DBM109, DBM111
.GLOBL DBM112, DBM120, DBM121, EH.0, EH.1
.GLOBL EH.2, EH.3, EH.4, EH.5, EH.6, EH.7
.GLOBL EH.8, EH.9, EH.10, EH.12, EH.13
.GLOBL MSG.02, MSG.03, EGS.02, EGD.10
.GLOBL EGD.11, EGD.12, EGD.13, EGD.14
.GLOBL EGD.15, EGD.16, EGD.17, EGD.18
.GLOBL EGD.19, EGD.20, EGD.21, EGD.22
.GLOBL EGD.23, EGD.24, EGH.30, DF.MSG
.GLOBL HRD.MSG, SFT.MSG, HRD.SUB, CRIF
.GLOBL SWP.ERROR, SWP.XFER, SWP.FLAGS

D4

ZRQAMS
V01.6

RD/RX EXERCISER
TEST SECTION

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11es-16 V4.0-579
DISK#USER2:[POWERS]ZRQADO.BL2;6

SEQ 0249
Page 18
(2)

.GLOBL DUPROUND, SWP,RAT, SWP,DPAT, SWP,UCNT
.GLOBL SWP,TIME, SWP,UDPAT, L#LUN, L#UNIT
.GLOBL NEX,TRAP, TIME, SET,CPAR, SET,UPAR
.GLOBL OUT,IODQ, IN,IODQ, GET,PKT, PUT,PKT
.GLOBL GET,RETPKT, PUT,RETPKT, GET,IO,BUFF
.GLOBL PUT,IO,BUFF, PUTA,BUFF, SEND, WAIT
.GLOBL MODULAS, DROP,CTLR, DRV,CTLEERR
.GLOBL EMS,RP1, EMS,EL, EMS,CMP, EMS,ERR
.GLOBL EMS,10, EMS,12, EMS,13, EMS,14
.GLOBL EMS,18, EMS,21, EMS,22, EMS,24
.GLOBL EMS,30

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

ZRQAM3
V01.6RD/RX EXERCISER
TEST SECTION11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0250
Page 19
(2)

000010	LOT**	10
000020	ADR**	20
000040	IDU**	40
000100	ISR**	100
000200	UAM**	200
000400	BOE**	400
001000	PN1**	1000
002000	PR1**	2000
004000	IXE**	4000
010000	IRE**	10000
020000	IER**	20000
040000	LOE**	40000
100000	HOE**	-100000

				.SBITL	\$11 TEST SECTION		
				.PSECT	\$CODE\$, RO		
000000							
000000	004137	000000G		\$T1:	JSR R1,\$SAVE3	:	1876
000004	112737	000001 000000G			MOV8 #1,ECP.FLAG	:	1891
000012	112737	000001 001170'			MOV8 #1,COMPARE.DATA	:	1892
000020	042737	000002 000000G			BIC #2,DUP.FLAGS	:	1893
000026	105037	000000G			CLRB HOE.FLAG	:	1894
000032	105037	000000G			CLRB FORCED.ERROR	:	1895
000036	005002				CLR R2	: I	1897
000040	010246					: I,*	1901
000042	012746	000043		1\$:	MOV R2,-(SP)		
000046	004737	000000G			MOV #43,-(SP)		
000052	005001				JSR PC,BL\$MUL		
000054	010003				CLR R1	: J	1900
000056	060103			2\$:	MOV R0,R3	:	1901
000060	006303				ADD R1,R3	: J,*	
000062	005063	000000G			ASL R3		
000066	005201				CLRB HSCP,PKT(R3)		
000070	020127	000042			INC R1	: J	1900
000074	003767				CMP R1,#42	: J,*	
000076	010216				BLE 2\$		
000100	012746	000106			MOV R2,(SP)	: I,*	1903
000104	004737	000000G			MOV #106,-(SP)		
000110	105060	000005G			JSR PC,BL\$MUL		
000114	062706	000006			CLRB HSCP,PKT+5(R0)		
000120	005202				ADD #6,SP	:	1898
000122	020227	000013			INC R2	: I	1897
000126	003744				CMP R2,#13	: I,*	
000130	005002				BLE 1\$		
000132	005001				CLR R2	: I	1906
000134	010200			3\$:	CLR R1	: J	1907
000136	060100			4\$:	MOV R2,R0	: I,*	1908
000140	006300				ADD R1,R0	: J,*	
000142	005060	000000G			ASL R0		
000146	005201				CLRB RETPRT(R0)		
000150	020127	000025			INC R1	: J	1907
000154	003767				CMP R1,#25	: J,*	
					BLE 4\$		

F4

ZRQAM3 V01.6	RD/RX EXERCISER TEST SECTION		11-Apr-1984 11:08:35 11-Apr-1984 11:08:22	VAX-11 B11ss-16 V4.0-579 DISK\$USER2:[POWERS]ZRQAD0.BL2;6	SEQ 0251 Page 20 (2)	
000156	062702	000026	ADD	#26,R2	; *,I	1906
000162	020227	000232	CMP	R2,#232	; I,*	
000166	003761		BLE	3\$		
000170	005002		CLR	R2	; I	1910
000172	010246		5\$: MOV	R2,-(SP)	; I,*	1914
000174	012746	000041	MOV	#41,(SP)		
000200	004737	000000G	JSR	PC,BL\$MUL		
000204	005001		CLR	R1	; J	1913
000206	010003		6\$: MOV	R0,R3		1914
000210	060103		ADD	R1,R3	; J,*	
000212	006303		ASL	R3		
000214	005063	000000G	CLR	ELOG,PKT(R3)		
000220	005201		INC	R1	; J	1913
000222	020127	000040	CMP	R1,#40	; J,*	
000226	003767		BLE	6\$		
000230	010216		MOV	R2,(SP)	; I,*	1916
000232	012746	000102	MOV	#102,-(SP)		
000236	004737	000000G	JSR	PC,BL\$MUL		
000242	105060	000001G	CLRB	ELOG,PKT+1(R0)		
000246	062706	000006	ADD	#6,SP		1911
000252	005202		INC	R2	; I	1910
000254	020227	000014	CMP	R2,#14	; I,*	
000260	003744		BLE	5\$		
000262	032737	000020 000000G	BIT	#20,SWP.FLAGS		1919
000270	001403		BEG	7\$		
000272	042737	000040 000000G	BIC	#40,SWP.FLAGS		1921
000300	032737	000002 000000G	7\$: BIT	#2,SWP.FLAGS		1923
000306	001403		BEG	8\$		
000310	042737	001000 000000G	BIC	#1000,SWP.FLAGS		1925
000316	132737	000001 000000G	8\$: BITB	#1,INIT.OCCURED		1927
000324	001145		BNE	13\$		
000326	017700	000000G	MOV	#FREE,MEM.ADDR,R0		1930
000332	006300		ASL	R0		
000334	063700	000000G	ADD	#FREE,MEM.ADDR,R0		
000340	010037	001160'	MOV	R0,DRS.START		
000344	062737	000002 001160'	ADD	#2,DRS.START		
000352	027727	001160' 000167	CMP	#DRS.START,#167		1937
000360	001021		BNE	9\$		
000362	104450		TRAP	50		1938
000364	103417		BOS	9\$		
000366	112737	000001 001162'	MOV8	#1,AFT.MODE		1941
000374	013737	001160' 001164'	MOV	#DRS.START,MAIL.BOX.TESTNUM		1942
000402	062737	000070 001164'	ADD	#10,MAIL.BOX.TESTNUM		
000410	013737	001160' 001166'	MOV	#DRS.START,MAIL.BOX.SUBTST		1943
000416	062737	000066 001166'	ADD	#6,MAIL.BOX.SUBTST		
000424	005037	000000G	9\$: CLR	R0		1946
000430	105037	000000G	CLRB	CLK.PRESENT		1947
000434	012746	000340	MOV	#340,-(SP)		1948
000440	012746	000000G	MOV	#TRAP,-(SP)		
000444	012746	000004	MOV	#1,-(SP)		
000450	012746	000003	MOV	#3,-(SP)		
000454	104457		TRAP	3		
000456	012700	000004	MOV	#4,R0		1951

G4

ZRQAM3
V01.6RD/RX EXERCISER
TEST SECTION11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0252
Page 21
(2)

000462	104436			TRAP	36		
000464	032737	000001	000000G	BIT	#1,NEX	:	1953
000472	001060			BNE	177		
000474	112737	000001	000000G	MOVB	#1,CLK.PRESENT	:	1956
000502	005037	000000G		CLR	CLK.TICKS	:	1957
000506	013716	000000G		MOV	SWP.TIME,(SP)	:	1958
000512	012746	000144		MOV	#14#,-(SP)		
000516	004737	000000G		JSR	PC,HL#DIV		
000522	110037	000000G		MOVB	RO,HOURS		
000526	013716	000000G		MOV	SWP.TIME,(SP)	:	1959
000532	012746	000144		MOV	#14#,-(SP)		
000536	004737	000000G		JSR	PC,HL#MOD		
000542	010001			MOV	RO,R1		
000544	005201			INC	R1		
000546	110137	000000G		MOVB	R1,MINUTES		
000552	123727	000000G	000074	10\$: CMPB	MINUTES,474	:	1961
000560	103412			BLO	117		
000562	005000			CLR	RO	:	1963
000564	153700	000000G		BTSB	MINUTES,RO		
000570	162700	000074		SUB	474,RO		
000574	110037	000000G		MOVB	RO,MINUTES		
000600	105237	000000G		INCB	HOURS	:	1964
000604	000762			BR	107	:	1961
000606	005016			11\$: CLR	(SP)	:	1967
000610	113716	000000G		MOVB	HOURS,(SP)		
000614	012746	000030		MOV	#30,-(SP)		
000620	004737	000000G		JSR	PC,HL#MOD		
000624	110037	000000G		MOVB	RO,HOURS		
000630	062706	000006		ADD	#6,SP	:	1955
000634	062706	000010		12\$: ADD	#10,SP	:	1929
000640	132737	000001	000000G	13\$: BITB	#1,CLK.PRESENT	:	1972
000646	001416			BEQ	147		
000650	012746	000300		MOV	#300,-(SP)	:	1975
000654	012746	000000G		MOV	#100,-(SP)		
000660	012746	000100		MOV	#100,-(SP)		
000664	012746	000003		MOV	#3,-(SP)		
000670	104437			TRAP	37		
000672	012737	000100	177546	MOV	#100,#0177546	:	1976
000700	062706	000010		ADD	#10,SP	:	1974
000704	104421			14\$: TRAP	21	:	1979
000706	010037	001172		MOV	RO,DNS.FLAGS		
000712	042700	077777		BIC	#77777,RO	:	1981
000716	020027	100000		CMP	RO,#100000		
000722	001003			BNE	177		
000724	012700	000001		MOV	RO,RO		
000730	000401			BR	107		
000732	005000			15\$: CLR	RO		
000734	020027	100000		16\$: CMP	RO,#100000		
000740	001003			BNE	177		
000742	112737	000001	000000G	MOVB	#1,DNS.FLAG	:	1983
000750	004737	000000G		17\$: JSR	PC,INIT.TEST	:	1985
000754	005002			CLR	R1	:	1987
000756	010246			18\$: MOV	R2,-(SP)	:	1989
						:	CTLR,+

H4

ZRRQAM5
V01.6RD/RX EXERCISER
TEST SECTION11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11:gs-16 V4.0-579
DISK\$USER2:[POWERS]ZRRQAD0.BL2;6SEQ 0253
Page 22
(2)

000760	012746	000126		MOV	\$226, -(SP)		
000764	004737	000000G		JSR	PC, BL \$MUL		
000770	022626			CMP	(SP)+, (SP)+		
000772	005760	000002G		TST	CST(R0)		
000776	100040			BPL	224		
001000	010246			MOV	R2, -(SP)	; CTRL, *	1990
001002	012746	000022		MOV	R2, -(SP)		
001006	004737	000000G		JSR	PC, BL \$MUL		
001012	022626			CMP	(SP)+, (SP)+		
001014	005760	000000G		TST	PC(R0)		
001020	100027			BPL	224		
001022	010246			MOV	R2, -(SP)	; CTRL, *	1995
001024	012746	000053		MOV	R2, -(SP)		
001030	004737	000000G		JSR	PC, BL \$MUL		
001034	012701	000003		MOV	R3, R1	; *, OFF SET	1993
001040	010003		194:	MOV	R0, R3		1995
001042	060103			ADD	R1, R3	; OFFSET, *	
001044	006303			ASL	R3		
001046	032763	020000	000000G	BIT	420000, CST(R3)		
001054	001403			BEQ	204		
001056	105037	000000G		CLRB	EOP.FLAG		1998
001062	000405			BR	214		1997
001064	062701	000012	204:	ADD	412, P1	; *, OFF SET	1993
001070	020127	000042		CMP	R1, 442	; OFFSET, *	
001074	003761			BLE	194		
001076	022626		214:	CMP	(SP)+, (SP)+		
001100	005202		224:	INC	R2	; CTRL	1987
001102	000243			.WORD	CLV:CLC		
001104	003724			BLE	184		
001106	132737	000001	000000G	BLTB	01, EOP.FLAG		2002
001114	001002			BNE	274		
001116	004737	000000V		JSR	PC, MULTI.DRIVE		2004
001122	000207		234:	RTS	PL		1876

; Routine Size: 298 words, Routine Base: \$CODE\$ + 0000
; Maximum stack depth per invocation: 12 words

000000	004737	000000V		.SBYTE	11 TEST SECTION		
000000			11::	JSR	PC, 111		
000004	104466		14:	TRAP	0		2004
000006	006000			ROR	R0		
000010	103773			BLO	14		
000012	000207			RTS	PC		

; Routine Size: 6 words, Routine Base: \$CODE\$ + 1124
; Maximum stack depth per invocation: 2 words

ZRQAM3
VO1.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0254
Page 23
(3)

```

: 2006 1 *sbt11 'INITIALIZATION TEST ROUTINES'
: 2007 1
: 2008 1 GLOBAL routine INIT_TEST : novalue *
: 2009 1
: 2010 1 !!
: 2011 1 !! THE INITIALIZATION TEST IS DESIGNED TO VERIFY THE EXISTENCE OF THE
: 2012 1 !! DEVICES AS CONFIGURED BY THE OPERATOR DURING THE HW DIALOG, AND TO
: 2013 1 !! BRING EACH DEVICE ONLINE IN PREPARATION FOR EITHER THE MULTI-DRIVE TEST
: 2014 1 !! OR THE DM EXERCISER.
: 2015 1 !!
: 2016 1 !! BASICALLY, THE DEVICES ARE BROUGHT ONLINE VIA "DRIVER_INIT", WHICH IS
: 2017 1 !! INVOKED IMMEDIATELY. ANY DEVICES WHICH FAIL DURING THIS PHASE WILL BE
: 2018 1 !! MARKED OFFLINE IN THEIR DCT AND CST. FOR THOSE DEVICES WHICH SURVIVE
: 2019 1 !! THE INITIALIZATION, THIS ROUTINE WILL ATTEMPT 1 OR 2 ACCESS COMMANDS TO
: 2020 1 !! EACH DISK VIA ROUTINE "ACCESS". THE INITIALIZATION TEST IS DEEMED A
: 2021 1 !! SUCCESS IF A BLOCK ON THE INNER TRACK OF EACH DISK CAN BE ACCESSED.
: 2022 1 !!-
: 2023 1
: 2024 2 begin
: 2025 2 DRIVER_INIT (); ! INIT DRIVER DATA AND DEVICES
: 2026 2
: 2027 2 incr CTLR from 0 to (MAX_CTLR - 1) do ! FOR EACH CONTROLLER
: 2028 3 begin
: 2029 3 SET_CPAR (.CTLR); ! SET UP COMMONLY-USED CONTROLLER-RELATED DATA ITEMS
: 2030 3
: 2031 3 if .CST_ADDR [STAT] eq1 ONLINE ! IF CONTROLLER IS STILL ALIVE
: 2032 3 then ! FOR EACH DISK
: 2033 3
: 2034 3 incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 2035 3
: 2036 3 if (.CST_ADDR [.OFFSET + OF_DATA, D_PRE] eq1 PRESENT) and
: 2037 3 (.CST_ADDR [.OFFSET + OF_DATA, D_STAT] eq1 ONLINE) and
: 2038 4 (not .CST_ADDR [.OFFSET + OF_DATA, D_FATAL])
: 2039 3 then
: 2040 4 begin
: 2041 4 SET_UPAR (.OFFSET); ! SET UP UNIT-RELATED DATA ITEMS
: 2042 5 IF SWP_DINT NEQ (.DUP_FLAGS AND SWP_DINT) !ZZZ
: 2043 4 THEN ACCESS (); !ZZZ
: 2044 4 !SKIP IF DUP CAUSED INIT ZZZ
: 2045 4
: 2046 3 end; ! IF UNIT IS PRESENT AND ONLINE
: 2047 3
: 2048 2 end; ! CONTROLLER LOOP
: 2049 2
: 2050 1 end; ! ROUTINE INIT_TEST

```

```

000000 004137 000000G      .SBI11  INIT_TEST INITIALIZATION TEST ROUTINES
                                INIT_TEST::
                                JSR      R1,SAVE2
                                JSR      PC,DRIVER_INIT
                                CLR      R2
000010 005002              13:  MOV      R2, -(SP)
000012 010246

```

2008
2005
2007
2009

J4

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0255
Page 24
(3)

000014	004737	000000G	JSR	PC,SET,CFAR		
000020	013700	000000G	MOV	CST,ADDR,RO	:	2031
000024	005760	000002	ISI	2(RO)		
000030	100035		BPL	4\$		
000032	012701	000003	MOV	03,R1	: *,OFFSET	2034
000036	010100		MOV	R1,RO	: OFFSET,*	2036
000040	006300		ASL	RO		
000042	063700	000000G	ADD	CST,ADDR,RO		
000046	032710	040000	BIT	040000,(RO)		
000052	001417		BEQ	3\$		
000054	032710	020000	BIT	020000,(RO)	:	2037
000060	001414		BEQ	3\$		
000062	032710	010000	BIT	010000,(RO)	:	2038
000066	001011		BNE	3\$		
000070	010116		MOV	R1,CSP	: OFFSET,*	2041
000072	004737	000000G	JSR	PC,SET,UPAR		
000076	032737	000002 000000G	BIT	02,DOWN,FLAGS	:	2042
000104	001002		BNE	3\$		
000106	004737	000000V	JSR	PC,ACCESS	:	2043
000112	062701	000012	ADD	012,R1	: *,OFFSET	2034
000116	020127	000041	CLD	R1,041	: OFFSET,*	
000122	003745		BLE	2\$		
000124	005726		ISI	(SP):		2028
000126	005202		INC	R2	: CTRL	2027
000130	000243		.WORD	CLV:CLC		
000132	003727		BLE	1\$		
000134	000207		RTS	PC	:	2008

; Routine Size: 47 words, Routine Base: \$CODE + 1140
; Maximum stack depth per invocation: 5 words

ZRRQM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRRQADO.BL2:6SEQ 0256
Page 25
(4)

```

: 2051 1 GLOBAL routine DRIVER_INIT : novalue *
: 2052 1
: 2053 1 !!
: 2054 1 !! THIS ROUTINE IS EQUIVALENT IN FUNCTION TO THE INITIALIZATION ENTRY
: 2055 1 !! POINT OF A STANDARD DEVICE DRIVER. ITS RESPONSIBILITY IS TO INITIALIZE
: 2056 1 !! DRIVER DATA, AND TO BRING EACH RDRX CONTROLLER AND UNIT (DISK)
: 2057 1 !! ONLINE.
: 2058 1 !!
: 2059 1 !!
: 2060 2 begin
: 2061 2
: 2062 2 local
: 2063 2 PKT_ADDR;
: 2064 2
: 2065 2 PKT_ADDR = MSCP_PKT + 10; ! ADDR (TEXT + 0) OF 1ST MSCP PKT
: 2066 2 NEXT_PKT_USE = 0; ! NEXT PACKET TO ALLOCATE
: 2067 2
: 2068 2 incr COUNT from 0 to (PKT_CNT - 1) do ! FOR EACH MSCP PACKET
: 2069 3 begin
: 2070 3 PKT_USE [.COUNT] = -1; ! MARK PACKET FREE
: 2071 3 MSCP_PKT [.COUNT, PKT_LO] = .PKT_ADDR; ! LOAD ADDR INTO BUFFER DESCRIPTOR
: 2072 3 MSCP_PKT [.COUNT, PKT_HI] = 0;
: 2073 3 MSCP_PKT [.COUNT, CONNID] = CID_DISK; ! SET CONNECTION ID TO MSCP ID
: 2074 3 PKT_ADDR = .PKT_ADDR + (PKT_LEN * 2); ! ADVANCE ADDR TO NEXT PACKET
: 2075 2 end;
: 2076 2
: 2077 2 incr CTRLR from 0 to (MAX_CTRLR - 1) do ! FOR EACH CONTROLLER
: 2078 2
: 2079 2 if .CST [.CTRLR, IP_ADDR] neq 0 ! IF CONTROLLER IS PRESENT
: 2080 2 then
: 2081 3 begin
: 2082 3 SET_CPAR (.CTRLR); ! CURRENT CONTROLLER PARAMETERS
: 2083 3 CURRENT_VECTOR = .CST_ADDR [VEC_ADDR]; ! CURRENT CONTROLLER'S VECTOR
: 2084 3 BRLEVEL = .CST_ADDR [BR_LEV] + 5; ! SET CURRENT CONTROLLER'S BR LEVEL
: 2085 3 CTRLR_INIT (); ! INIT DEVICE AND CTRLR DATA
: 2086 3
: 2087 3 if .DCT_ADDR [STAT] eq 1 ONLINE ! IF CONTROLLER IS STILL ALIVE
: 2088 3 then ! FOR EACH DIK UNIT
: 2089 3
: 2090 3 incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 2091 3
: 2092 3 if (.CST_ADDR [.OFFSET + OF_DATA, D_PRESENT] eq 1 PRESENT) and ! IF UNIT EXISTS
: 2093 4 (not .CST_ADDR [.OFFSET + OF_DATA, D_FATAL])
: 2094 3 then
: 2095 4 begin
: 2096 4 CST_ADDR [.OFFSET + OF_NAME_0, D_NAME_0] = %0'40'; ! BLANK DEVICE NAME
: 2097 4 CST_ADDR [.OFFSET + OF_NAME_0, D_NAME_1] = %0'40';
: 2098 4 CST_ADDR [.OFFSET + OF_NAME_2, D_NAME_2] = %0'40';
: 2099 4 CST_ADDR [.OFFSET + OF_NAME_2, D_NAME_3] = %0'40';
: 2100 4 SET_UPAR (.OFFSET); ! SET UP UNIT RELATED DATA ITEMS
: 2101 4 UNIT_INIT (); ! BRING UNIT ONLINE
: 2102 3 end; ! IF UNIT EXISTS
: 2103 3

```

ZRQAM3
V01.6

PD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0257
Page 26
(4)

: 2104 2
: 2105 2
: 2106 1

end;
end;

! IF CONTROLLER IS PRESENT

! ROUTINE DRIVER_INIT

			.SBTTL	DRIVER_INIT INITIALIZATION TEST ROUTINES	
000000	004137	000000G	DRIVER_INIT::		
000004	012702	000012G	JSR	R1,\$SAVE3	2051
000010	105037	000000G	MOV	\$MSCP.PKT+12,R2	2065
000014	005001		CLRB	NEXT.PKT,USE	2066
000016	112761	000377 000000G	1\$: MOV	R1,\$377,PKT,USE(R1)	2068
000024	010146		MOV	R1,-(SP)	2070
000026	012746	000106	MOV	\$106,-(SP)	2071
000032	004737	000000	JSR	PC,BL\$MUL	
000036	010260	000000	MOV	R2,\$MSCP.PKT(R0)	; PKT,ADDR,*
000042	005060	000000	CLRB	\$MSCP.PKT+2(R0)	2072
000046	105060	000011G	CLRB	\$MSCP.PKT+11(R0)	2073
000052	062702	000106	ADD	\$106,R2	2074
000056	022626		CMR	(SP)+,(SP)+	2069
000060	005201		INC	R1	2068
000062	020127	000013	CMR	R1,\$13	; COUNT,*
000066	003753		BLE	1\$	
000070	005003		CLRB	R3	; CTRL,*
000072	010346		2\$: MOV	R3,-(SP)	2077
000074	012746	000126	MOV	\$126,-(SP)	2079
000100	004737	000000G	JSR	PC,BL\$MUL	
000104	022626		CMR	(SP)+,(SP)+	
000106	005760	000000G	TST	CST(R0)	
000112	001503		BEQ	6\$	
000114	010346		MOV	R3,-(SP)	; CTRL,*
000116	004737	000000G	JSR	PC,SET,CPAR	2082
000122	013700	000000G	MOV	CST,ADDR,R0	2083
000126	016037	000002 001152'	MOV	2(R0),CURRENT,VECTOR	
000134	042737	177000 001152'	BIC	\$177000,CURRENT,VECTOR	
000142	005016		CLRB	(SP)	2084
000144	116016	000004	MOVB	4(R0),(SP)	
000150	012746	000005	MOV	\$5,(SP)	
000154	004737	000000G	JSR	PC,BL\$SHF	
000160	010037	001154'	MOV	R0,BRLEVEL	
000164	004737	000000V	JSR	PC,CTRL,INIT	2085
000170	005777	000000G	TST	\$DOCT,ADDR	2087
000174	100051		BPL	5\$	
000176	012701	000003	MOV	\$3,R1	; *,OFFSET
000202	013702	000000G	3\$: MOV	CST,ADDR,R2	2090
000206	010100		MOV	R1,R0	2092
000210	006300		ASL	R0	
000212	060200		ADD	R2,R0	
000214	032710	040000	BIT	\$40000,(R0)	
000220	001432		BEQ	4\$	
000222	032710	010000	BIT	\$10000,(R0)	2093
000226	001027		BNE	4\$	
000230	010100		MOV	R1,R0	2096
				; OFFSET,*	

M4

ZRQAI.3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0258
Page 27
(4)

000232	006300			ASL	R0		
000234	060200			ADD	R2,R0		
000236	112760	000040	000012	MOVB	#40,12(R0)		
000244	112760	000040	000013	MOVB	#40,13(R0)		
000252	010100			MOV	R1,R0	; OFFSET,*	2097
000254	006300			ASL	R0		2098
000256	060200			ADD	R2,R0		
000260	112760	000040	000014	MOVB	#40,14(R0)		
000266	112760	000040	000015	MOVB	#40,15(R0)		
000274	010116			MOV	R1,(SP)	; OFFSET,*	2099
000276	004737	000000G		JSR	PC,SET.UPAR		2100
000302	004737	000000V		JSR	PC,UNIT.INIT		
000306	062701	000012	4\$:	ADD	#12,R1	; *,OFFSET	2101
000312	020127	000041		CMP	R1,#41	; OFFSET,*	2090
000316	003731			BLE	3\$		
000320	022626		5\$:	CMP	(SP)+,(SP)+		2081
000322	005203		6\$:	INC	R3	; CTLR	2077
000324	000243			.WORD	CLV!CLC		
000326	003661			BLE	2\$		
000330	000207			RTS	PC		2051

; Routine Size: 109 words, Routine Base: \$CODE\$ + 1276
; Maximum stack depth per invocation: 7 words

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

```

: 2107 1 GLOBAL routine CTRL_INIT : novalue =
: 2108 1
: 2109 1 !*
: 2110 1 ! THIS "DRIVER" ROUTINE IS CALLED FROM DRIVER_INIT FOR EACH CONTROLLER
: 2111 1 ! CONFIGURED FOR TESTING. ITS GENERAL PURPOSE IS TO BRING THE RDRX ONLINE
: 2112 1 ! TO THE HOST. SPECIFICALLY, IT IS WRITTEN TO:
: 2113 1 !
: 2114 1 ! 1. INITIALIZE DRIVER CONTROLLER DATA, INCLUDING THE DCT,
: 2115 1 ! 2. SET UP THE DEVICE'S INTERRUPT VECTOR ADDRESS,
: 2116 1 ! 3. PERFORM A REGISTER EXISTENCE TEST TO VERIFY THE DEVICE'S PRESENCE,
: 2117 1 ! 4. PERFORM A VECTOR AND BR LEVEL TEST TO VERIFY THE DEVICE'S VECTOR
: 2118 1 ! ADDRESS AND INTERRUPT REQUEST LEVEL,
: 2119 1 ! 5. DO A HARD INITIALIZATION (FOUR STEPS) ON THE DEVICE.
: 2120 1 !
: 2121 1 ! IF ANY OF THESE INITIAL TESTS FAIL, THEN ALL UNITS ASSOCIATED WITH THE
: 2122 1 ! DEVICE ARE DROPPED.
: 2123 1 !-
: 2124 1
: 2125 2 begin
: 2126 2
: 2127 2 local
: 2128 2 RESULT : byte;
: 2129 2
: 2130 2 INI_CTRL_DAT (); ! INITIALIZE CONTROLLER DATA
: 2131 2 SETVEC (.CURRENT_VECTOR, .INT_ADDR (.CTRLR), PRI04); ! SET DEVICE'S ASSUMED VECTOR ADDRESS
: 2132 2 DCT_ADDR [IG_INT] = TRUE; ! SET "IGNORE INTERRUPT" BIT
: 2133 2 L$UN = .CST_ADDR [OF_UN + OF_DATA, D_UNIT]; ! GET FIRST UNIT NUMBER OF CONTROLLER
: 2134 2 ! (USED BY DRS FOR DEVICE-FATAL CTRLR ERRORS)
: 2135 3 IF SWP_DINT NEQ (.DUP_FLAGS AND SWP_DINT) ! IF DUP ZZZ
: 2136 2 THEN !CAUSED INIT, SKIP THIS CODE ZZZ
: 2137 2
: 2138 2 IF REG_EXIST () eq1 FAILURE ! REGISTER EXISTENCE TEST
: 2139 2 then
: 2140 3 begin
: 2141 3 DROP_CTRLR (.CTRLR, DU_INIT); ! DROP ALL CONTROLLER'S UNITS
: 2142 3 return;
: 2143 2 end;
: 2144 2
: 2145 3 IF SWP_DINT NEQ (.DUP_FLAGS AND SWP_DINT) ! IF DUP ZZZ
: 2146 2 THEN !CAUSED INIT, SKIP THIS CODE ZZZ
: 2147 2
: 2148 2 IF VEC_BR_TEST () eq1 FAILURE ! VECTOR ADDR AND BR LEVEL TEST
: 2149 2 then
: 2150 3 begin
: 2151 3 DROP_CTRLR (.CTRLR, DU_INIT); ! DROP ALL CONTROLLER'S UNITS
: 2152 3 return;
: 2153 2 end;
: 2154 2
: 2155 2 RESULT = HARD_INIT (); ! ATTEMPT HARD DEVICE INIT
: 2156 2 DCT_ADDR [IG_INT] = FALSE; ! CLEAR "IGNORE INTERRUPT" BIT
: 2157 2
: 2158 2 IF .RESULT eq1 SUCCESS ! IF HARD INIT WAS SUCCESSFUL
: 2159 2 then

```

85

ZRQAM3
VOL.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK#USER2:(POWERS)ZRQADO,BL2,6

SEQ 0260
Page 29
(5)

```

: 2160 3      begin
: 2161 3      ADDR_VECT_OK = TRUE;
: 2162 3      INI_RRING ();
: 2163 3      WRT_RDRX (RCSA, RC_ALL, SA_GO);
: 2164 3
: 2165 3      if SET_CTLR_CHAR () eq1 SUCCESS
: 2166 3      then
: 2167 4          begin
: 2168 4              DCT_ADDR [STAT] = ONLINE;
: 2169 4              CST_ADDR [STATE] = ONLINE;
: 2170 3          end;
: 2171 3      end
: 2172 3
: 2173 2      else
: 2174 3          begin
: 2175 3              DROP_CTLR (.CCTL, DU_INIT);
: 2176 2          end;
: 2177 2
: 2178 1      end;

```

! ADDRESS/VECTOR TEST PASSED
! INITIALIZE RESPONSE RING
! SET "GO" BIT (START CTLR POLLING)
! SET CONTROLLER CHARACTERISTICS
! MARK CONTROLLER ONLINE IN "DRIVER"
! MARK CONTROLLER ONLINE IN "PROGRAM"
! HARD INIT FAILED
! DROP ALL CONTROLLER'S UNITS
! ROUTINE CTLR INIT

		.SBTTL	CTLR,INIT	CTLR,INIT INITIALIZATION TEST ROUTINES	
000000	010146		CTLR,INIT::		
000002	004737	000000V	MOV	R1, -(SP)	2107
000006	012746	000200	JSR	PC,INI,CTLR,DAT	2130
000012	013700	000000G	MOV	#200, -(SP)	2131
000016	006300		MOV	CCTL,R0	
000020	016046	000102'	ASL	R0	
000024	013746	001152'	MOV	INT,ADDR(R0), -(SP)	
000030	012746	000003	MOV	CURRENT,VECTOR, -(SP)	
000034	104437		MOV	#3, -(SP)	
000036	052777	040000 000000G	TRAP	37	
000044	013700	000000G	BIS	#40000, DCT, ADDR	2132
000050	016001	000006	MOV	CST, ADDR, R0	2133
000054	000301		MOV	6(R0), R1	
000056	042701	177760	SWAB	R1	
000062	010137	000000G	BIC	#177760, R1	
000066	032737	000002 000000G	MOV	R1, LUN	
000074	001025		BIT	#2, DUP, FLAGS	2135
000076	004737	000000V	BNE	21	
000102	005700		JSR	PC, REG, EXIST	2138
000104	001410		TST	R0	
000106	032737	000002 000000G	BEQ	11	2141
000114	001015		BIT	#2, DUP, FLAGS	2145
000116	004737	000000V	BNE	21	
000122	005700		JSR	PC, VEC, BR, TEST	2148
000124	001011		TST	R0	
000126	013716	000000G	BNE	21	
000132	012746	000002	MOV	CCTL, (SP)	2151
000136	004737	000000G	MOV	#2, (SP)	
000142	062706	000012	JSR	PC, DROP, CTLR	
000146	000453		ADD	#12, SP	2152
			BR	51	2150

05

ZRQAM3	RD/RX EXERCISER				11-Apr-1984 11:08:35	VAX-11 Bliss-16 V4.0 579	SEQ 0261
V01.6	INITIALIZATION TEST ROUTINES				11 Apr-1984 11:08:22	DISK#USER2:[POWERS]ZRQADO.BL2;6	Page 30
							(5)
000150	004737	000000V	2\$:	JSR	PC,HARD.INIT	:	2155
000154	110001			MOVB	R0,R1	: *,RESULT	
000156	042777	040000 000000G		BIC	#40000,8OCT.ADDR	:	2156
000164	120127	000001		CMPI	R1,#1	: RESULT,*	2158
000170	001031			BNE	3\$	:	
000172	112737	000001 000000G		MOVB	#1,ADDR.VECT.OK	:	2161
000200	004737	000000V		JSR	PC,INI.RRING	:	2162
000204	012701	000001		MOV	#1,R1	: *,RC.REG	2163
000210	013700	000000G		MOV	RDRX,ADDR,R0	:	
000214	010160	000002		MOV	R1,2(R0)	: RC.REG,*	
000220	004737	000000V		JSR	PC,SET.CTLR.CHAR	:	2165
000224	020027	000001		CMPI	R0,#1	:	
000230	001020			BNE	4\$	:	
000232	052777	100000 000000G		BIS	#100000,8OCT.ADDR	:	2168
000240	013700	000000G		MOV	CST,ADDR,R0	:	2169
000244	052760	100000 000002		BIS	#100000,2(R0)	:	
000252	000407			BR	4\$	:	2158
000254	013716	000000G	3\$:	MOV	CCTLR,(SP)	:	2175
000260	012746	000002		MOV	#2,-(SP)	:	
000264	004737	000000G		JSR	PC,DROP.CTLR	:	
000270	005726			TST	(SP),	:	2174
000272	062706	000010	4\$:	ADD	#10,SP	:	2125
000276	012601		5\$:	MOV	(SP),R1	:	2107
000300	000207			RTS	PC	:	

: Routine Size: 97 words, Routine Base: \$CODE\$ + 1630
: Maximum stack depth per invocation: 7 words

D5

ZRQ4M3
VOL.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1155-16 V4.0-579
DISK\$USER2:[POWERS]ZRQ4D0.BL2;6

SEQ 0262
Page 31
(6)

```

: 2179 1 GLOBAL routine INI_CTLR_DAT : novalue *
: 2180 1
: 2181 1 !
: 2182 1 ! THIS ROUTINE IS RESPONSIBLE FOR INITIALIZING ALL CONTROLLER-RELATED
: 2183 1 ! DATA IN THE "DRIVER" PORTION OF THE EXERCISER. THIS INCLUDES THE
: 2184 1 ! CONTROLLER'S DCT AND OUTSTANDING COMMAND LIST.
: 2185 1 !
: 2186 1 ! IMPLICIT INPUTS:
: 2187 1 ! CCTLR - CURRENT CONTROLLER NUMBER
: 2188 1 ! DCT_ADDR - ADDRESS OF CURENT CONTROLLER'S DCT
: 2189 1 !
: 2190 1
: 2191 2 begin
: 2192 2 DCT_ADDR [WORD0] = 0;
: 2193 2 DCT_ADDR [RR_BEG] = COMM_AREA + 8 + ((CCTLR * COMM_LEN * 2);
: 2194 2 DCT_ADDR [RR_END] = DCT_ADDR [RR_BEG] + ((RRING_LEN - 1) * 4);
: 2195 2 DCT_ADDR [CR_BEG] = DCT_ADDR [RR_END] + 4;
: 2196 2 DCT_ADDR [CR_END] = DCT_ADDR [CR_BEG] + ((CRING_LEN - 1) * 4);
: 2197 2 DCT_ADDR [RR_POLL] = DCT_ADDR [RR_BEG];
: 2198 2 DCT_ADDR [CR_POLL] = DCT_ADDR [CR_NEXT] = DCT_ADDR [CR_BEG];
: 2199 1 end;

```

			.SBTTL	INI_CTLR_DAT INITIALIZATION TEST ROUTINES	
000000	004137	000000G	INI_CTLR_DAT::		
			JSR	R1, \$SAVE2	2179
000004	013701	000000G	MOV	DCT_ADDR, R1	2192
000010	005011		CLR	(R1)	
000012	012702	000004	MOV	#4, R2	2193
000016	060102		ADD	R1, R2	
000020	013746	000000G	MOV	CCTLR, -(SP)	
000024	012746	000050	MOV	#50, -(SP)	
000030	004737	000000G	JSR	PC, BL \$MUL	
000034	062700	000010'	ADD	#COMM_AREA+10, R0	
000040	010012		MOV	R0, (R2)	
000042	010061	000006	MOV	R0, 6(R1)	2194
000046	062761	000014 000006	ADD	#14, 6(R1)	
000054	012700	000010	MOV	#10, R0	2195
000060	060100		ADD	R1, R0	
000062	016110	000006	MOV	6(R1), (R0)	
000066	062710	000004	ADD	#4, (R0)	
000072	011061	000012	MOV	(R0), 12(R1)	2196
000076	062761	000014 000012	ADD	#14, 12(R1)	
000104	011261	000014	MOV	(R2), 14(R1)	2197
000110	011061	000020	MOV	(R0), 20(R1)	2198
000114	011061	000016	MOV	(R0), 16(R1)	
000120	022626		CMP	(SP), (SP)	2191
000122	000207		RTS	PC	2179

; Routine Size: 42 words, Routine Base: \$CODE\$ + 2132
; Maximum stack depth per invocation: 6 words

85

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11gg-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0263
Page 32
(7)

```

: 2200 1 GLOBAL routine REG_EXIST *
: 2201 1 !+
: 2202 1 ! THIS IS THE REGISTER EXISTENCE (OR "PROBE") TEST DESIGNED TO VERIFY
: 2203 1 ! THE PRESENCE OF AN RDRX DEVICE. THIS OBJECTIVE IS ACCOMPLISHED BY
: 2204 1 ! SETTING UP THE NON-EXISTENT MEMORY (NEX) TRAP VECTOR (LOCATION 4) AND
: 2205 1 ! ATTEMPTING TO READ WHAT IS ASSUMED TO BE THE DEVICE'S SA AND IP
: 2206 1 ! REGISTERS. IF THE NEX TRAP HANDLER IS INVOKED DUE TO AN ABSENT DEVICE,
: 2207 1 ! THEN THE GLOBAL DATUM "NEX" WILL BE SET TO "TRUE". THIS DATUM
: 2208 1 ! DETERMINES THE SUCCESS / FAILURE VALUE OF THIS ROUTINE.
: 2209 1 !-
: 2210 2 begin
: 2211 2
: 2212 2 local
: 2213 2 DUMMY_0 : word, ! TEMP FOR READING SA AND IP
: 2214 2 DUMMY_1 : word; !
: 2215 2
: 2216 2 if .ENTRY_REASON eal NEW_PASS
: 2217 2 then
: 2218 2 return SUCCESS; ! SKIP TEST FOR NEXT PASS
: 2219 2
: 2220 2 OF_RC + 2; ! SET UP TO READ SA FIRST
: 2221 2
: 2222 2 do
: 2223 3 begin
: 2224 3 NEX = FALSE; ! SET TO "TRAP NOT RECEIVED"
: 2225 3 SETVEC (4, NEX_TRAP, PRI07); ! SET LOCATION 4 TRAP VECTOR ADDRESS
: 2226 3 DUMMY_0 = (.RDRX_ADDR + .OF_RC); ! READ REGISTER (THEN TRAP OR CONTINUE)
: 2227 3 DUMMY_1 = 0; ! DUMMY INSTRUCTION TO COVER TRAP RETURN BUG
: 2228 3 ! (TRAP RETURNS TO NEXT INSTRUCTION)
: 2229 3 CLRVEC (4); ! CLEAR LOCATION 4 TRAP VECTOR ADDRESS
: 2230 3
: 2231 3 if .NEX ! IF NEX TRAP OCCURRED
: 2232 3 then
: 2233 4 begin
: 2234 4 C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 2235 4
: 2236 4 if .APT_MODE
: 2237 4 then
: 2238 5 begin
: 2239 5 .MAIL_BOX_TESTNUM + 1;
: 2240 5 .MAIL_BOX_SUBTST = 0;
: 2241 5 end;
: 2242 4
: 2243 4 ERRDF (10, EGD_10, EMS_10); ! REGISTER EXISTENCE TEST FAILED
: 2244 4 SEIPRI (PRI00); ! LOWER PRIORITY
: 2245 4 return FAILURE;
: 2246 4 end
: 2247 3 else
: 2248 3 OF_RC = .OF_RC - 2; ! SET UP FOR IP REG OR QUIT
: 2249 3
: 2250 3 end
: 2251 2 until .OF_RC lss 0;
: 2252 2

```


F5

ZRQAM5
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0264
Page 33
(7); 2253 2 return SUCCESS;
; 2254 1 end;

		.SBTTL REG.EXIST INITIALIZATION TEST ROUTINES		
000000	004137	000000G	REG.EXIST::	
000004	123727	000000G 000005	JSR R1,\$SAVE2	2200
000012	001472		CMPB ENTRY,REASON,#5	2216
000014	012737	000002 000000G	BEQ 4\$	2218
000022	005037	000000G	MOV #2,OF,RC	2220
000026	012746	000340	1\$: CLR NEX	2224
000032	012746	000000G	MOV #340,-(SP)	2225
000036	012746	000004	MOV #NEX,TRAP,-(SP)	
000042	012746	000003	MOV #4,-(SP)	
000046	104437		MOV #3,-(SP)	
000050	013700	000000G	TRAP 37	
000054	063700	000000G	MOV RDRX,ADDR,RO	2226
000060	011001		ADD OF,RC,RO	
000062	005002		MOV (RO),R1	; *,DUMMY.0
000064	012700	000004	CLR R2	; DUMMY.1
000070	104436		MOV #4,RO	2227
000072	032737	000001 000000G	TRAP 36	2229
000100	001427		BIT #1,NEX	
000102	013700	000000G	BEQ 3\$	2231
000106	006300		MOV CCTL,RO	2234
000110	105260	000000G	ASL RO	
000114	032737	000001 001162'	INCB C.ERR,TBL(RO)	
000122	001405		BIT #1,APT,MODE	2236
000124	012777	000001 001164'	BEQ 2\$	
000132	005077	001166'	MOV #1,MAIL.BOX.TESTNUM	2239
000136	104455		CLR MAIL.BOX.SUBTST	2240
000140	000012		2\$: TRAP 55	2243
000142	000000G		.WORD 12	
000144	000000G		.WORD EGO.10	
000146	005000		.WORD EMS.10	
000150	104441		CLR RO	2244
000152	062706	000010	TRAP 41	
000156	000413		ADD #10,SP	2245
000160	162737	000002 000000G	BR 5\$	2253
000166	062706	000010	3\$: SUB #2,OF,RC	2248
000172	005737	000000G	ADD #10,SP	2253
000176	002311		TST OF,RC	2251
000200	012700	000001	BGE 1\$	
000204	000207		MOV #1,RO	2210
000206	005000		RTS PC	
000210	000207		5\$: CLR RO	2200
			RTS PC	

; Routine Size: 69 words, Routine Base: \$CODE\$ + 2256
; Maximum stack depth per invocation: 9 words

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0265
Page 34
(8)

```

: 2255 1 GLOBAL routine VEC_BR_TEST *
: 2256 1
: 2257 1 !+
: 2258 1 !
: 2259 1 ! THIS ROUTINE ATTEMPTS TO VERIFY (A) THAT THE RDRX VECTOR ADDRESS GIVEN
: 2260 1 ! BY THE USER DURING THE HW DIALOG IS VALID, AND (B) THAT THE
: 2261 1 ! USER-SPECIFIED BUS REQUEST LEVEL FOR THE DEVICE IS CORRECT. THE FIRST
: 2262 1 ! OBJECTIVE IS ACCOMPLISHED BY SETTING THE CPU PRIORITY TO 0 AND FORCING
: 2263 1 ! AN RDRX INTERRUPT. IF THE USER SPECIFIED AN INCORRECT VECTOR ADDRESS,
: 2264 1 ! THEN THE RESULT MAY BE UNPREDICTABLE. FOR THIS REASON, THE MESSAGE
: 2265 1 ! "FUNCTIONAL TEST STARTED" IS PRINTED BEFORE THE TEST, AND
: 2266 1 ! "EXERCISER STARTED" IS PRINTED AT ITS SUCCESSFUL CONCLUSION. IF
: 2267 1 ! EITHER "FUNCTIONAL TEST ..." OR "EXERCISER ..." DOES NOT APPEAR, THEN
: 2268 1 ! PROGRAM CONTROL IS ASSUMED LOST AND A FATAL TRAP IS LIKELY TO OCCUR. AT
: 2269 1 ! THIS POINT, THE EXERCISER MUST BE STARTED AGAIN.
: 2270 1 !
: 2271 1 ! IF THIS TEST SUCCEEDS, THEN THE BR LEVEL TEST IS RUN BY SETTING THE
: 2272 1 ! PROCESSOR PRIORITY TO THE ASSUMED INTERRUPT PRIORITY GIVEN BY THE
: 2273 1 ! USER. A FORCED INTERRUPT SHOULD NOT OCCUR. THEN, BY LOWERING THE
: 2274 1 ! PRIORITY BY ONE, THE DELAYED INTERRUPT SHOULD OCCUR.
: 2275 1 !-
: 2276 2 begin
: 2277 2
: 2278 2 if .ENTRY_REASON eq1 NEW_PASS
: 2279 2 then
: 2280 3 begin
: 2281 3 SETPRI (PRI00); ! LOWER PRIORITY
: 2282 3 return SUCCESS; ! SKIP TEST IF NEXT PASS
: 2283 2 end;
: 2284 2
: 2285 2 PRINTF (MSG_02); ! "FUNCTIONAL TEST STARTED"
: 2286 2
: 2287 2 if INT_GEN () eq1 FALSE ! FORCE AN INTERRUPT
: 2288 2 then
: 2289 3 begin ! IF INTERRUPT DID NOT OCCUR
: 2290 3 C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 2291 3
: 2292 3 if .APT_MODE
: 2293 3 then
: 2294 4 begin
: 2295 4 .MAIL_BOX_TESTNUM = 1;
: 2296 4 .MAIL_BOX_SUBTST = 0;
: 2297 3 end;
: 2298 3
: 2299 3 ERRDF (11, EGD_11, 0); ! VECTOR TEST FAILED
: 2300 3 return FAILURE;
: 2301 3 end
: 2302 2 else
: 2303 3 begin ! INTERRUPT DID OCCUR
: 2304 3 PRINTF (MSG_03); ! "EXERCISER STARTED"
: 2305 3 SETPRI (.BRLEVEL); ! SET PRIORITY TO ASSUMED BR LEVEL
: 2306 3
: 2307 3 if INT_GEN () eq1 FALSE ! FORCE AN INTERRUPT (SHOULD NOT OCCUR)

```

H5

ZRQAM5
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0266
Page 35
(8)

```

: 2308 3      then
: 2309 4      begin
: 2310 4          SETPRI (.BRLEVEL - %0'40');
: 2311 4          DELAY (1);
: 2312 4
: 2313 4          if .DCT_ADDR [SA_SAVE] neq 0
: 2314 4      then
: 2315 5      begin
: 2316 5          SETPRI (PRI00);
: 2317 5          return SUCCESS;
: 2318 4      end;
: 2319 4
: 2320 3      end;
: 2321 3
: 2322 2      end;
: 2323 2
: 2324 2      SETPRI (PRI00);
: 2325 2      C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 2326 2
: 2327 2      if .APT_MODE
: 2328 2      then
: 2329 3      begin
: 2330 3          .MAIL_BOX_TESTNUM = 1;
: 2331 3          .MAIL_BOX_SUBTST = 0;
: 2332 2      end;
: 2333 2
: 2334 2      ERRDF (12, EGD_12, EMS_12);
: 2335 2      return FAILURE;
: 2336 1      end;

```

.GLOBL L3DLY

			.SBTTL	VEC.BR.TEST INITIALIZATION TEST ROUTINES	
000000	010146		VEC.BR.TEST::		
000002	005746		MOV	R1, -(SP)	2255
000004	123727	000000G 000005	TST	(SP)	
000012	001003		CMPB	ENTRY, REASON, 05	2258
000014	005000		BNE	1\$	
000016	104441		CLR	R0	2281
000020	000504		TRAP	41	
000022	012746	000000G	BR	8\$	2280
000026	012746	000001	1\$: MOV	0MSG, 02, -(SP)	2285
000032	010600		MOV	01, (SP)	
000034	104417		MOV	SP, R0	
000036	004737	000000V	TRAP	17	
000042	005700		JSR	PC, INT_GEN	2287
000044	001023		TST	R0	
000046	013700	000000G	BNE	3\$	
000052	006300		MOV	CCTLR, R0	2290
000054	105260	000000G	AND	R0	
			INCB	C_ERR_TBL(R0)	

ZRRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRRQAD0,BL2;6SEQ 0267
Page 36
(8)

000060	032737	000001	001162'	BIT	01,APT.MODE	:	2292
000066	001405			BEQ	2\$	:	
000070	012777	000001	001164'	MOV	01,0MAIL.BOX.TESTNUM	:	2295
000076	005077	001166'		CLR	0MAIL.BOX.SUBTST	:	2296
000102	104455			TRAP	55	:	2299
000104	000013			.WORD	13	:	
000106	000000G			.WORD	EGD.11	:	
000110	000000			.WORD	0	:	
000112	000477			BR	11\$	:	2300
000114	012716	000000G		MOV	0MSG.03,(SP)	:	2304
000120	012746	000001		MOV	01,-(SP)	:	
000124	010600			MOV	SP,R0	: SP,*	
000126	104417			TRAP	17	:	
000130	013700	001154'		MOV	BRLEVEL,R0	:	2305
000134	104441			TRAP	41	:	
000136	004737	000000V		JSR	PC,INT.GEN	:	2307
000142	005700			TST	R0	:	
000144	001035			BNE	9\$	:	
000146	013700	001154'		MOV	BRLEVEL,R0	:	2310
000152	162700	000040		SUB	040,R0	:	
000156	104441			TRAP	41	:	
000160	012701	000001		MOV	01,R1	: *,\$\$TMP2	2311
000164	001411			BEQ	7\$	:	
000166	013700	000000G		MOV	L\$DLY,R0	: *,\$\$TMP1	
000172	001404			BEQ	6\$	:	
000174	005066	000006		CLR	6(SP)	: \$\$TMP	
000200	005300			DEC	R0	: \$\$TMP1	
000202	001374			BNE	5\$	:	
000204	005301			DEC	R1	: \$\$TMP2	
000206	000766			BR	4\$	:	
000210	013700	000000G		MOV	DCT.ADDR,R0	:	2313
000214	005760	000002		TST	2(R0)	:	
000220	001407			BEQ	1\$	:	
000222	005000			CLR	F)	:	2316
000224	104441			TRAP	11	:	
000226	062706	000006		ADD	15,SP	:	2317
000232	012700	000001		MOV	01,R0	:	2315
000236	000427			BR	12\$	:	
000240	005726			TST	(SP),	:	2303
000242	005000			CLR	R0	:	2324
000244	104441			TRAP	41	:	
000246	013700	000000G		MOV	CCTLR,R0	:	2325
000252	006300			ASL	R0	:	
000254	105260	000000G		INCB	C.ERR,1BL(R0)	:	
000260	032737	000001	001162'	BIT	01,APT.MODE	:	2327
000266	001405			BEQ	10\$	:	
000270	012777	000001	001164'	MOV	01,0MAIL.BOX.TESTNUM	:	2330
000276	005077	001166'		CLR	0MAIL.BOX.SUBTST	:	2331
000302	104455			TRAP	55	:	2334
000304	000014			.WORD	14	:	
000306	000000G			.WORD	EGD.12	:	
000310	000000G			.WORD	EMS.12	:	
000312	022626			CMP	(SP)*,(SP)*	:	2335

J5

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1:gs-16 V4.0-579
DISK\$USER2:([POWERS])ZRQADO.BL2;6

SEQ 0258
Page 37
(8)

000314	005000		CLR	R0	
000316	005726	12\$:	TST	(SP)+	
000320	012601		MOV	(SP)+,R1	
000322	000207		RTS	PC	

; Routine Size: 106 words, Routine Base: \$CODE\$ + 2470
; Maximum stack depth per invocation: 7 words

2255

ZRQAM5
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0269
Page 38
(9)

```

: 2337 1 GLOBAL routine INT_GEN *
: 2338 1
: 2339 1 !+
: 2340 1 ! THIS ROUTINE BEGINS AN RDRX INITIALIZATION SEQUENCE, BUT ONLY
: 2341 1 ! COMPLETES THROUGH THE STEP 1 WRITE. ITS PURPOSE IS TO CREATE AN RDRX
: 2342 1 ! INTERRUPT (AT THE COMPLETION OF STEP 1) IN ORDER TO HELP VERIFY THE
: 2343 1 ! THE USER-SPECIFIED VECTOR ADDRESS AND BUS REQUEST INTERRUPT LEVEL.
: 2344 1 ! A VALUE OF "TRUE" IS RETURNED TO THE CALLER IF AN INTERRUPT OCCURS,
: 2345 1 ! AND "FALSE" OTHERWISE. THE INTERRUPT IS VERIFIED BY A NON-ZERO VALUE
: 2346 1 ! IN THE "SA SAVE" WORD IN THE DEVICE'S DCT.
: 2347 1 !-
: 2348 1
: 2349 2 begin
: 2350 2
: 2351 2 local
: 2352 2 SA : word;
: 2353 2 ! STORAGE FOR STEP 1 READ AND WRITE
: 2354 2 DCT_ADDR [SA_SAVE] * 0;
: 2355 2 WRT_RDRX (RCIP, RC_ALL, ALL_ONES);
: 2356 2 DELAY (2);
: 2357 2 SA * .RDRX_ADDR [RCSA, RC_ALL];
: 2358 2 SA * (WR_RING + 8) or (.CURRENT_VECTOR + -2) or SA_INT;
: 2359 2 WRT_RDRX (RCSA, RC_ALL, .SA);
: 2360 2 ! ZERO OUT SA SAVE WORD IN DCT
: 2361 2 ! WRITE IP TO START INIT SEQUENCE
: 2362 2 ! WAIT
: 2363 2 ! STEP 1 READ
: 2364 2 ! STEP 1 WRITE VALUE
: 2365 2 ! STEP 1 WRITE
: 2366 2
: 2367 2 incr COUNT from 1 to 8000 do
: 2368 2 begin
: 2369 2 DELAY (1);
: 2370 2 ! TOTAL DELAY COUNT OF 8,000
: 2371 2 if .DCT_ADDR [SA_SAVE] neq 0
: 2372 2 then
: 2373 2 return TRUE;
: 2374 2 ! IF SA WAS CHANGED
: 2375 2 ! INTERRUPT OCCURED
: 2376 2 BREAK;
: 2377 2 end;
: 2378 2 return FALSE;
: 2379 2 ! IF INTERRUPT DID NOT OCCUR
: 2380 2 end;

```

			.SBTTL	INT_GEN INITIALIZATION TEST ROUTINES	
000000	004137	000000G	INT_GEN:		
000004	024646		JSR	R1, \$SAVE2	2337
000006	013700	000000G	CMP	-(SP), (SP)	
000012	005060	000002	MOV	DCT_ADDR, R0	2354
000016	012700	177777	CLR	2(R0)	
000022	010077	000000G	MOV	#1, R0	2355
000026	012701	000002	MOV	R0, .RDRX_ADDR	
000032	001411		MOV	#2, R1	2356
000034	013700	000000G	BEQ	4\$	
000040	001404		MOV	1\$DLY, R0	
000042	005066	000002	BEQ	3\$	
000046	005300		CLR	2(SP)	
			DEC	R0	

L5

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0270
Page 39
(9)

000050	001374		BNE	2\$			
000052	005301	3\$:	DEC	R1		; \$\$TMP2	
000054	000766		BR	1\$			
000056	013700	000000G	MOV	RDRX.ADDR,R0			
000062	016016	000002	MOV	2(R0),(SP)		; *,RC.REG	2357
000066	013701	001152'	MOV	CURRENT.VECTOR,R1			
000072	006201		ASR	R1			2358
000074	006201		ASR	R1			
000076	010102		MOV	R1,R2		; *,SA	
000100	052702	111200	BIS	0111200,R2		; *,SA	
000104	010201		MOV	R2,R1		; SA,RC.REG	2359
000106	010160	000002	MOV	R1,2(R0)		; RC.REG,*	
000112	012702	017500	MOV	017500,R2		; *,COUNT	2361
000116	012701	000001	MOV	01,R1		; *,\$\$TMP2	2363
000122	001411		BEQ	9\$			
000124	013700	000000G	MOV	L\$DLY,R0		; *,\$\$TMP1	
000130	001404		BEQ	8\$			
000132	005066	000002	CLR	2(SP)		; \$\$TMP	
000136	005300		DEC	R0		; \$\$TMP1	
000140	001374		BNE	7\$			
000142	005301		DEC	R1		; \$\$TMP2	
000144	000766		BR	6\$			
000146	013700	000000G	MOV	DCT.ADDR,R0			2365
000152	005760	000002	TST	2(R0)			
000156	001403		BEQ	10\$			
000160	012700	000001	MOV	01,R0			2367
000164	000404		BR	11\$			
000166	104422		TRAP	22			
000170	005302		DFC	R2		; COUNT	2361
000172	001351		BNE	5\$			
000174	005000		CLR	R0			2349
000176	022626		CMP	(SP)+,(SP)+			2337
000200	000207		RTS	PC			

; Routine Size: 65 words, Routine Base: \$CODE\$ + 3014
; Maximum stack depth per invocation: 7 words

M5

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0271
Page 40
(10)

```

: 2374 1 GLOBAL routine HARD_INIT =
: 2375 1
: 2376 1 !+
: 2377 1 ! THIS ROUTINE PERFORMS THE FOUR READ / WRITE STEPS REQUIRED TO
: 2378 1 ! INITIALIZE AN RDRX DEVICE. IF NO READ ERRORS ARE DETECTED IN ANY OF
: 2379 1 ! THE FOUR STEPS, THEN A SUCCESS VALUE IS RETURNED TO THE CALLER.
: 2380 1 ! OTHERWISE, ADDITIONAL ATTEMPTS MAY BE MADE TO INITIALIZE THE DEVICE.
: 2381 1 ! IF ALL ATTEMPTS FAIL, A FAILURE INDICATION IS RETURNED.
: 2382 1 !-
: 2383 1
: 2384 2 begin
: 2385 2
: 2386 2 local
: 2387 2 IE_VEC : word;
: 2388 2
: 2389 2
: 2390 2 IE_VEC = .CURRENT_VECTOR + -2;
: 2391 2
: 2392 2 incr ATTEMPTS from 1 to INI_ATT do
: 2393 3 begin
: 2394 3
: 2395 3 label
: 2396 3 STEP_1_READ,
: 2397 3 STEP_2_READ,
: 2398 3 STEP_3_READ,
: 2399 3 STEP_4_READ;
: 2400 3
: 2401 3 WRT_RDRX (RCIP, RC_ALL, ALL_ONES);
: 2402 3
: 2403 3 STEP 1 READ
: 2404 3
: 2405 3 STEP = 1;
: 2406 3 STEP_1_READ:
: 2407 4 begin
: 2408 4
: 2409 4 incr COUNT from 1 to 500 do
: 2410 5 begin
: 2411 5 DELAY (1);
: 2412 5 SA_REG = .RDRX_ADDR (RCSA, RC_ALL);
: 2413 5
: 2414 5 if (.SA_REG and S1_MASK) eq1 SA_S1
: 2415 5 then
: 2416 5 leave STEP_1_READ;
: 2417 5
: 2418 5 BREAK;
: 2419 4 end;
: 2420 4
: 2421 4 exitloop;
: 2422 3 end;
: 2423 3
: 2424 3
: 2425 3 STEP 1 WRITE
: 2426 3

```

! IE-BIT-AND-VECTOR-ADDRESS/4 BYTE
! (USED IN STEP 1 WRITE AND STEP 3 READ)
! GET VECTOR ADDR/4 (IE - 0)
! WRITE IP TO START INIT SEQUENCE
! TOTAL DELAY COUNT OF 500 FOR STEP 1
! READ SA
! IF STEP 1 READ IS O.K.

N5

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0272
Page 41
(10)

```

: 2427 3      SA_REG = (WR_RING + 8) or .IF_VEC;      ! STEP 1 WRITE VALUE
: 2428 3      WRT_RDRX (RCSA, RC_ALL, .SA_REG);      ! STEP 1 WRITE
: 2429 3      !
: 2430 3      ! STEP 2 READ
: 2431 3      !
: 2432 3      STEP = .STEP + 1;
: 2433 3      STEP_2_READ:
: 2434 4          begin
: 2435 4
: 2436 4          incr COUNT from 1 to 10000 do
: 2437 5              begin
: 2438 5                  DELAY (1);
: 2439 5                  SA_REG = .RDRX_ADDR [RCSA, RC_ALL];      ! TOTAL DELAY COUNT OF 10,000 FOR STEP 2
: 2440 5                  ! READ SA
: 2441 6                  if (.SA_REG and S2_MASK) eq1 (SA_S2 or WR_RING) ! IF STEP 2 READ IS O.K.
: 2442 5                  then
: 2443 5                      leave STEP_2_READ;
: 2444 5
: 2445 5                  BREAK;
: 2446 4                  end;
: 2447 4
: 2448 4          exitloop;
: 2449 3          end;
: 2450 3
: 2451 3      !
: 2452 3      ! STEP 2 WRITE
: 2453 3      !
: 2454 3      WRT_RDRX (RCSA, RC_ALL, .DCT_ADDR [RR_BEG]);      ! RINGBASE-LO, PI = 0
: 2455 3      !
: 2456 3      ! STEP 3 READ
: 2457 3      !
: 2458 3      STEP = .STEP + 1;
: 2459 3      STEP_3_READ:
: 2460 4          begin
: 2461 4
: 2462 4          incr COUNT from 1 to 10000 do
: 2463 5              begin
: 2464 5                  DELAY (1);
: 2465 5                  SA_REG = .RDRX_ADDR [RCSA, RC_ALL];      ! TOTAL DELAY COUNT OF 10,000 FOR STEP 3 READ
: 2466 5                  ! READ SA
: 2467 6                  if (.SA_REG and S3_MASK) eq1 (SA_S3 or .IE_VEC) ! IF STEP 3 READ IS O.K.
: 2468 5                  then
: 2469 5                      leave STEP_3_READ;
: 2470 5
: 2471 5                  BREAK;
: 2472 4                  end;
: 2473 4
: 2474 4          exitloop;
: 2475 3          end;
: 2476 3
: 2477 3      !
: 2478 3      ! STEP 3 WRITE
: 2479 3      !

```

136

ZRQAM3
VOL.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1100-16 V4.0-579
DISK#USER2:[POWERS]ZRQAD0.BL2:6

SEQ 0273
Page 42
(10)

```

: 2480 3      WRT_RDRX (RCSA, RC_ALL, 0);      ! PP, RINGBASE-HI = 0
: 2481 3      !
: 2482 3      ! STEP 4 READ
: 2483 3      !
: 2484 3      STEP = STEP + 1;
: 2485 3      STEP_4_READ:
: 2486 4          begin
: 2487 4              incr COUNT from 1 to 10000 do
: 2488 4                  begin
: 2489 5                      DELAY (1);
: 2490 5                      SA_REG = RDRX_ADDR (RCSA, RC_ALL);
: 2491 5                      ! TOTAL DELAY COUNT OF 10,000 FOR STEP 4 READ
: 2492 5                      ! READ SA
: 2493 5                      if (.SA_REG and SA_MASK) eq1 SA_SA
: 2494 5                          then
: 2495 5                              leave STEP_4_READ;
: 2496 5                      ! IF STEP 4 READ IS O.K.
: 2497 5                      BREAK;
: 2498 4                      end;
: 2499 4              endloop;
: 2500 4              end;
: 2501 3      !
: 2502 3      ! STEP 4 WRITE
: 2503 3      !
: 2504 3      !
: 2505 3      CREDIT_BAL = 1;
: 2506 3      WRT_RDRX (RCSA, RC_ALL, 0);
: 2507 3      return SUCCESS;
: 2508 3      ! START WITH A CREDIT BALANCE = 1
: 2509 2      ! BURST, LF, GO = 0
: 2510 2      ! SUCCESS EXIT POINT
: 2511 2      ! TRY AGAIN OR GIVE UP
: 2512 2      CREDIT_BAL = 0;
: 2513 2      C_ERR_TBL (.CCTLR, C_ERR_HRD) = C_ERR_TBL (.CCTLR, C_ERR_HRD) + 1;
: 2514 2      ! NO CREDIT BALANCE
: 2515 2      if .APT_MODE
: 2516 2      then
: 2517 3          begin
: 2518 3              MAIL_BOX_TESTNUM = 1;
: 2519 3              MAIL_BOX_SUBTST = 0;
: 2520 2          end;
: 2521 2      ERRDF (13, EGD_13, EMS_13);
: 2522 2      return FAILURE;
: 2523 1      ! INIT SEQUENCE FAILED
: 2523 1      ! ROUTINE HARD_INIT

```

000000	004137	000000G	.SBTTL	HARD_INIT INITIALIZATION TEST ROUTINES	
			HARD_INIT::		
000004	162706	000012	JSR	R1,SAVE5	2374
000010	013704	001152	SUB	#12,SP	
000014	006204		MOV	CURRENT.VECTOR,R4	2390
000016	006204		ASR	R4	
			ASR	R4	

C6

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0 579
DISK#USER2:[POWERS]ZRQADO,BL2:6SEQ 0274
Page 43
(10)

000020	012705	000002		MOV	02,R5	*,ATTEMPTS	2392
000024	012700	177777		MOV	0-1,R0	*,RC,REG	2401
000030	010077	000000G		MOV	R0,RDRX,ADDR	RC,REG,*	
000034	012737	000001	000000G	MOV	01,STEP		2405
000042	012702	000764		MOV	0764,R2	*,COUNT	2409
000046	012701	000001		MOV	01,R1	*,\$\$TMP2	2411
000052	001411		1\$:	BEQ	5\$		
000054	013700	000000G	2\$:	MOV	L\$DLY,R0	*,\$\$TMP1	
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	013700	000000G	5\$:	MOV	RDRX,ADDR,R0		2412
000102	016016	000002		MOV	2(R0),(SP)	*,RC,REG	
000106	011637	000000G		MOV	(SP),SA,REG	RC,REG,*	
000112	011600			MOV	(SP),R0	SA,REG,*	2414
000114	042700	001777		BIC	01777,R0		
000120	020027	004000		CMP	R0,04000		
000124	001404			BEQ	6\$		2416
000126	104422			TRAP	22		
000130	005302			DEC	R2	COUNT	2409
000132	001345			BNE	1\$		
000134	000532			BR	10\$		2393
000136	010437	000000G	6\$:	MOV	R4,SA,REG	IE,VFC,*	2427
000142	052737	111000	000000G	BIS	0111000,SA,REG		
000150	013701	000000G		MOV	SA,REG,R1	*,RC,REG	2428
000154	013700	000000G		MOV	RDRX,ADDR,R0		
000160	010160	000002		MOV	R1,2(R0)	RC,REG,*	
000164	005237	000000G		INC	STEP		2432
000170	012702	023420		MOV	023420,R2	*,COUNT	2436
000174	012701	000001	7\$:	MOV	01,R1	*,\$\$TMP2	2438
000200	001411		8\$:	BEQ	11\$		
000202	013700	000000G		MOV	L\$DLY,R0	*,\$\$TMP1	
000206	001404			BEQ	10\$		
000210	005066	000010	9\$:	CLR	10(SP)	\$\$TMP	
000214	005300			DEC	R0	\$\$TMP1	
000216	001374			BNE	9\$		
000220	005301		10\$:	DEC	R1	\$\$TMP2	
000222	000766			BR	8\$		
000224	013700	000000G	11\$:	MOV	RDRX,ADDR,R0		2439
000230	016066	000002	000002	MOV	2(R0),2(SP)	*,RC,REG	
000236	016637	000002	000000G	MOV	2(SP),SA,REG	RC,REG,*	
000244	016600	000002		MOV	2(SP),R0	SA,REG,*	2441
000250	042700	003400		BIC	03400,R0		
000254	020027	010222		CMP	R0,010222		
000260	001404			BEQ	12\$		2443
000262	104422			TRAP	22		
000264	005302			DEC	R2	COUNT	2436
000266	001342			BNE	7\$		
000270	000537			BR	26\$		2443
000272	013700	000000G	12\$:	MOV	DCT,ADDR,R0		2454

106

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0275
Page 44
(10)

000276	016001	000004		MOV	4(R0),R1	; *,RC,REG	
000302	013700	000000G		MOV	RDRX,ADDR,R0		
000306	010160	000002		MOV	R1,2(R0)	; RC,REG,*	
000312	005237	000000G		INC	STEP		
000316	010403			MOV	R4,R3	; IE,VEC,*	2458
000320	052703	020000		BIS	#20000,R3		2467
000324	012702	023420		MOV	#23420,R2	; *,COUNT	2462
000330	012701	000001	13:	MOV	#1,R1	; *,\$\$TMP2	2464
000334	001411		14:	BEQ	17		
000336	013700	000000G		MOV	L\$DLY,R0	; *,\$\$TMP1	
000342	001404			BEQ	16		
000344	005066	000010	15:	CLR	10(SP)	; \$\$TMP	
000350	005300			DEC	R0	; \$\$TMP1	
000352	001374			BNE	15		
000354	005301		16:	DEC	R1	; \$\$TMP2	
000356	000766			BR	14		
000360	013700	000000G	17:	MOV	RDRX,ADDR,R0		2465
000364	016066	000002	000004	MOV	2(R0),4(SP)	; *,RC,REG	
000372	016637	000004	000000G	MOV	4(SP),SA,REG	; RC,REG,*	
000400	016600	000004		MOV	4(SP),R0	; SA,REG,*	2467
000404	042700	003400		BIC	#3400,R0		
000410	020003			CMP	R0,R3		
000412	001404			BEQ	19		2469
000414	104422			TRAP	22		
000416	005302			DEC	R2	; COUNT	2462
000420	001343			BNE	13		
000422	000462		18:	BR	26		2393
000424	013700	000000G	19:	MOV	RDRX,ADDR,R0		2480
000430	005060	000002		CLR	2(R0)		
000434	005237	000000G		INC	STEP		2484
000440	012703	023420		MOV	#23420,R3	; *,COUNT	2488
000444	012701	000001	20:	MOV	#1,R1	; *,\$\$TMP2	2490
000450	001411		21:	BEQ	24		
000452	013700	000000G		MOV	L\$DLY,R0	; *,\$\$TMP1	
000456	001404			BEQ	23		
000460	005066	000010	22:	CLR	10(SP)	; \$\$TMP	
000464	005300			DEC	R0	; \$\$TMP1	
000466	001374			BNE	27		
000470	005301		23:	DEC	R1	; \$\$TMP2	
000472	000766			BR	21		
000474	013700	000000G	24:	MOV	RDRX,ADDR,R0		2491
000500	016066	000002	000006	MOV	2(R0),6(SP)	; *,RC,REG	
000506	016637	000006	000000G	MOV	6(SP),SA,REG	; RC,REG,*	
000514	016600	000006		MOV	6(SP),R0	; SA,REG,*	2493
000520	042700	003777		BIC	#3777,R0		
000524	020027	040000		CMP	R0,#40000		
000530	001404			BEQ	25		2495
000532	104422			TRAP	22		
000534	005303			DEC	R3	; COUNT	2488
000536	001342			BNE	20		
000540	000413			BR	26		2393
000542	012737	000001	000000G	25:	MOV	#1,CREDIT,BAL	2505
000550	005001			CLR	R1	; RC,REG	2506

EE6

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0276
Page 45
(10)

000552	013700	000000G		MOV	RDRX,ADDR,R0		
000556	005060	000002		CLR	2(R0)		
000562	012700	000001		MOV	01,R0	:	2393
000566	000425			BR	28:		
000570	005037	000000G	26:	CLR	CREDIT,BAL	:	2511
000574	013700	000000G		MOV	CCTLR,R0	:	2512
000600	006300			ASL	R0		
000602	105260	000000G		INCB	C.ERR.TBL(R0)		
000606	032737	000001	001162'	BIT	01,APT.MODE	:	2514
000614	001405			BEQ	27:		
000616	012777	000001	001164'	MOV	01,0MAIL.BOX.TESTNUM	:	2517
000624	005077	001166'		CLR	0MAIL.BOX.SUBTST	:	2518
000630	104455		27:	TRAP	55	:	2521
000632	000015			.WORD	15		
000634	000000G			.WORD	EGD.13		
000636	000000G			.WORD	EMS.13		
000640	005000			CLR	R0	:	2384
000642	062706	000012	28:	ADD	012.SP	:	2374
000646	000207			RTS	PC		

; Routine Size: 212 words, Routine Base: \$CODE\$ + 3216
; Maximum stack depth per invocation: 13 words

F6

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0277
Page 46
(11)

```

: 2524 1 GLOBAL routine INI_RRING : novalue =
: 2525 1
: 2526 1 !+
: 2527 1 ! THIS ROUTINE IS RESPONSIBLE FOR ALLOCATING ENOUGH MSCP PACKETS TO
: 2528 1 ! FILL AN RDRX RESPONSE RING. THE BUFFER DESCRIPTOR OF EACH PACKET
: 2529 1 ! (LOCATED IN FRONT OF THE PACKET ITSELF) IS LOADED INTO SUCCESSIVE
: 2530 1 ! RRING SLOTS. NOTE THAT THE BUFFER DESCRIPTORS HAVE BEEN INITIALIZED
: 2531 1 ! WITH THE FLAG AND OWNERSHIP BITS SET TO "1", MAKING EACH SLOT
: 2532 1 ! CONTROLLER-OWNED.
: 2533 1 !
: 2534 1 ! IMPLICIT INPUTS:
: 2535 1 ! CCTLR - CURRENT CONTROLLER NUMBER
: 2536 1 ! DCT_ADDR - ADDRESS OF CURRENT CONTROLLER'S DCT
: 2537 1 !-
: 2538 1
: 2539 2 begin
: 2540 2
: 2541 2 local
: 2542 2 index : word,
: 2543 2 RRING_ADDR;
: 2544 2
: 2545 2 RRING_ADDR = .DCT_ADDR [RR_BEG]; ! FIRST RESPONSE RING SLOT
: 2546 2
: 2547 2 incr COUNT from 1 to RRING_LEN do
: 2548 3 begin
: 2549 3 index = GET_PKT (.CCTLR); ! GET AN MSCP PACKET
: 2550 3 .RRING_ADDR = .MSCP_PKT [.index, PKT_LO]; ! LOAD LO-ORDER BUFF DESC INTO SLOT
: 2551 3 RRING_ADDR = .RRING_ADDR + 2; ! ADVANCE TO SECOND WORD
: 2552 3 .RRING_ADDR = .MSCP_PKT [.index, PKT_HI]; ! LOAD HI-ORDER BUFF DESC INTO SLOT
: 2553 3 PKT_USE [.index] = .CCTLR; ! PACKET IN USE
: 2554 3 .RRING_ADDR = .RRING_ADDR or ED_OWN or ED_FLAG; ! GIVE OWNERSHIP TO CONTRLLER
: 2555 3 RRING_ADDR = .RRING_ADDR + 2; ! ADVANCE TO NEXT SLOT
: 2556 2 end;
: 2557 2
: 2558 1 end;

```

			.SBITL	INI_RRING INITIALIZATION TEST ROUTINES	
000000	004137	000000G	INI_RRING::		
			JSR	R1,\$SAVE4	2524
000004	013700	000000G	MOV	DCT_ADDR,R0	2545
000010	016001	000004	MOV	4(R0),R1	
000014	013703	000000G	MOV	CCTLR,R3	
000020	012704	000004	MOV	#4,R4	2549
000024	010346		MOV	R3,(SP)	2549
000026	004737	000000G	JSR	PC,GET_PKT	
000032	010002		MOV	R0,R2	
000034	010216		MOV	R2,(SP)	
000036	012746	000106	MOV	#106,(SP)	2550
000042	004737	000000G	JSR	PC,BL\$MUL	
000046	016021	000000G	MOV	MSCP_PKT(R0),(R1)	
000052	016011	000002G	MOV	MSCP_PKT+2(R0),(R1)	2552
000056	013703	000000G	MOV	CCTLR,R3	2553

G6

ZPQAM3 RD/RX EXERCISER
V01.6 INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 81199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0278
Page 47
(11)

000062	110362	000000G	MOVB	R3,PKT.USE(R2)	; *,*(INDEX)	
000066	052721	140000	BIS	#140000,(R1)+	; *.RRING.ADDR	2554
000072	022626		CMP	(SP)+,(SP)+		2548
000074	005304		DEC	R4	; COUNI	2547
000076	001352		BNE	1;		
000100	000207		RTS	PC		2524

; Routine Size: 33 words, Routine Base: \$CODE\$ + 4066
; Maximum stack depth per invocation: 8 words

H6

ZRGAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0279
Page 48
(12)

```

: 2559 1 GLOBAL routine SET_CTLR_CHAR =
: 2560 1
: 2561 1 !+
: 2562 1 ! THIS ROUTINE IS CALLED BY CTLR_INIT AFTER THE RDRX HAS BEEN HARD-
: 2563 1 ! INITIALIZED. ITS PURPOSE IS TO FORMAT AND SEND THE "SET CONTROLLER
: 2564 1 ! CHARACTERISTICS" COMMAND, AND TO VALIDATE THE RESPONSE (END MESSAGE).
: 2565 1 !
: 2566 1 ! IMPLICIT INPUTS:
: 2567 1 !     CCTLR = CURRENT CONTROLLER NUMBER
: 2568 1 !-
: 2569 1
: 2570 2     begin
: 2571 2
: 2572 2     local
: 2573 2         P_INDEX : word;
: 2574 2
: 2575 2
: 2576 2     ! MISCELLANEOUS INITIALIZATION
: 2577 2
: 2578 2     QIO [CCTLR] = 0;           !INIT NO OF OUTSTANDING QIOS
: 2579 2     CST [CCTLR, U_CNT] = 0;   !CLEAR UNITS IN CST TABLE
: 2580 2     INCR COUNT FROM 0 TO (RP_CNT - 1) DO !INIT RETURN PACKET POOL
: 2581 2         RP_USE [COUNT] = -1;
: 2582 2
: 2583 2     IODQ_IN = IODQ_OUT = 0;    !INIT I/O DONE QUEUE POINTERS
: 2584 2
: 2585 2
: 2586 2     P_INDEX = GET_PKT (CCTLR);    ! GET AN MSCP PACKET
: 2587 2     MSCP_PKT [P_INDEX, MSGLEN] = SZ_SCC;    ! PACKET SIZE
: 2588 2     MSCP_PKT [P_INDEX, OPCODE] = OP_SCC;    ! OPCODE = SET CTLR CHAR
: 2589 2     MSCP_PKT [P_INDEX, C_FLAGS] = CF_MASK; ! CONTROLLER FLAGS
: 2590 2     MSCP_PKT [P_INDEX, CMD_TYPE] = IMM_CMD; ! IMMEDIATE COMMAND
: 2591 2
: 2592 2     if SEND (P_INDEX) eq 1 FAILURE    ! ATTEMPT SEND
: 2593 2     then
: 2594 3         begin
: 2595 3             C_ERR_TBL [CCTLR, C_ERR_HRD] = C_ERR_TBL [CCTLR, C_ERR_HRD] + 1; ! IF SEND WAS UNSUCCESSFUL
: 2596 3
: 2597 3             if .APT_MODE
: 2598 3             then
: 2599 4                 begin
: 2600 4                     .MAIL_BOX_TESTNUM = 1;
: 2601 4                     .MAIL_BOX_SUBTST = 0;
: 2602 3                 end;
: 2603 3
: 2604 3                 ERROF (20, EGD_20, 0);    ! FATAL ERROR
: 2605 3                 PUT_PKT (P_INDEX);        ! RETURN PACKET TO POOL
: 2606 3                 DROP_CTLR (CCTLR, DU_CFATAL); ! DROP CONTROLLER
: 2607 3                 return FAILURE;
: 2608 3             end
: 2609 2         else
: 2610 3             begin
: 2611 3

```


ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11:16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0280
Page 49
(12)

```

: 2612 3      do
: 2613 4      begin
: 2614 4      WAIT ();
: 2615 4      RP_INDX * OUT_100Q ();
: 2616 4      RP_ADDR * RETPKT * (.RP_INDX * RP_LEN * 2);
: 2617 4
: 2618 4      if .RP_ADDR [MESTYP] neq MT_SEQ
: 2619 4      then
: 2620 4      PUT_RETPKT (.RP_INDX);
: 2621 4
: 2622 4      end
: 2623 3      until (.RP_ADDR [CONID] eq CID_DRIVER) or
: 2624 4      ((.RP_ADDR [MESTYP] eq MT_SEQ) and
: 2625 3      ((.RP_ADDR [ENDCOD] and OP_END) eq OP_END));
: 2626 3
: 2627 3      if .RP_ADDR [CONID] eq CID_DRIVER
: 2628 3      then
: 2629 4      begin
: 2630 4      PRINTF (DBM23);
: 2631 4      PUT_RETPKT (.RP_INDX);
: 2632 4      DR_ERR ();
: 2633 4      return FAILURE;
: 2634 4      end
: 2635 3      else
: 2636 4      begin
: 2637 4
: 2638 4      if (.RP_ADDR [ENDCOD] neq (OP_SCC or OP_END)) or
: 2639 5      ((.RP_ADDR [C_FLGS] and CF_MASK) neq CF_MASK)
: 2640 4      then
: 2641 5      begin
: 2642 5      C_ERR_TBL [.CCTLR, C_ERR_HRD] * .C_ERR_TBL [.CCTLR, C_ERR_HRD] * 1;
: 2643 5
: 2644 5      if .APT_MODE
: 2645 5      then
: 2646 6      begin
: 2647 6      .MAIL_BOX_TESTNUM * 1;
: 2648 6      .MAIL_BOX_SUBTST * 0;
: 2649 5      end;
: 2650 5
: 2651 5      ERRDF (21, EGD_21, EMS_21);
: 2652 5      DROP_CTLR (.CCTLR, DU_CFATAL);
: 2653 5      PUT_RETPKT (.RP_INDX);
: 2654 5      return FAILURE;
: 2655 5      end
: 2656 4      else
: 2657 5
: 2658 5      CTD_TIME = .RP_ADDR [C_TIME] * 2;
: 2659 5
: 2660 6      if BIT_TST (SWP_FLAGS, SWP_TRC)
: 2661 5      then
: 2662 5      PRINTF (DBM25, .RP_ADDR [C_TIME]);
: 2663 5
: 2664 4      end;

```

! WAIT FOR RETPKT RESPONSE
! GET INDEX OF RETPKT
! CALCULATE RETPKT ADDRESS
! RETURN ALL RETPKTS NOT SENT BY CONTROLLER
! IF RETPKT IS FROM "DRIVER"
! "ERROR IN SET_CTLR_CHAR"
! RELEASE RETURN PACKET
! DROP CONTROLLER
! ELSE - RETPKT IS FROM DISK MSCP
! IF WRONG ENDCODE
! OR FLAGS IN ERROR
! FATAL ERROR
! DROP CONTROLLER
! RELEASE RETURN PACKET
! RETPKT HAS CORRECT ENDCODE
! RETPKT HAS CORRECT ENDCODE

J6

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0281
Page 50
(12)

```

; 2665 4
; 2666 3          end;
; 2667 3
; 2668 3          PUT_RETPKT (.RP_INDX);
; 2669 3          return SUCCESS;
; 2670 2          end;
; 2671 2
; 2672 1          end;

```

! IF RETPKT WAS SENT BY DISK MSCP

! IF SEND WAS SUCCESSFUL

! ROUTINE SET_CTLR_CHAR

```

000000 010146          .SBTTL SET_CTLR_CHAR INITIALIZATION TEST ROUTINES
                                SET_CTLR_CHAR::
000002 013701 000000G      MOV      R1, -(SP)
000006 105061 000000G      MOV      CCTL, R1
000012 010146          CLR      QIO(R1)
000014 012746 000126      MOV      R1, -(SP)
000020 004737 000000G      MOV      #126, -(SP)
000024 105060 000005G      JSR      PC, BL$MUL
000030 005000          CLR      CST+5(R0)
000032 112760 000377 000000G      CLR      R0
000040 005200          1$:      MOV      #377, RP, USE(R0)
000042 020027 000007      INC      R0
000046 003771          CMP      R0, #7
000050 005037 000000G      BLE      1$
000054 005037 000000G      CLR      IODQ, OUT
000060 010116          CLR      IODQ, IN
000062 004737 000000G      MOV      R1, (SP)
000066 010001          JSR      PC, GET, PKT
000070 010116          MOV      R0, R1
000072 012746 000106      MOV      R1, (SP)
000076 004737 000000G      MOV      #106, -(SP)
000102 012760 000040 000006G      JSR      PC, BL$MUL
000110 112760 000004 000022G      MOV      #40, MSCP, PKT+6(R0)
000116 012760 000120 000030G      MOV      #4, MSCP, PKT+22(R0)
000124 105060 000004G      MOV      #120, MSCP, PKT+30(R0)
000130 010116          CLR      MSCP, PKT+4(R0)
000132 004737 000000G      MOV      R1, (SP)
000136 005700          JSR      PC, SEND
000140 001036          TST      R0
000142 013700 000000G      BNE      3$
000146 006300          MOV      CCTL, R0
000150 105260 000000G      ASL      R0
000154 032737 000001 001162'      INCB      C, ERR, TBL(R0)
000162 001405          BIT      #1, APT, MODE
000164 012777 000001 001164'      BEQ      2$
000172 005077 001166'      MOV      #1, @MAIL, BOX, TESTNUM
000176 104455          CLR      @MAIL, BOX, SUBTST
000200 000024          2$:      TRAP      55
000202 000000G          .WORD      24
000204 000000          .WORD      EGD, 20
000206 010116          .WORD      0
000210 004737 000000G      MOV      R1, (SP)
                                JSR      PC, PUT, PKT

```

2557

2578

2579

2580

2581

2580

2583

2586

2587

2588

2589

2590

2592

2595

2597

2600

2601

2604

2605

K6

ZRQAM5
V01 6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0282
Page 51
(12)

000214	013716	000000G		MOV	CCTLR,(SP)	:	2606
000220	012746	000006		MOV	#6,-(SP)	:	
000224	004737	000000G		JSR	PC,DROP,CTLR	:	
000230	005726			TST	(SP)+	:	2594
000232	005000			CLR	R0	:	2610
000234	000571			BR	13\$	:	
000236	004737	000000G	3\$:	JSR	PC,WAIT	:	2614
000242	004737	000000G		JSR	PC,OUT,IODQ	:	2615
000246	010037	000000G		MOV	R0,RP,INDX	:	
000252	010016			MOV	R0,(SP)	:	2616
000254	012746	000054		MOV	#54,-(SP)	:	
000260	004737	000000G		JSR	PC,BL\$MUL	:	
000264	062700	000000G		ADD	#RETPKT,R0	:	
000270	010037	000000G		MOV	R0,RP,ADDR	:	
000274	132760	000360	000002	BITB	#360,2(R0)	:	2618
000302	001404			BEQ	4\$	:	
000304	013716	000000G		MOV	RP,INDX,(SP)	:	2620
000310	004737	000000G		JSR	PC,PUT,RETPKT	:	
000314	005726		4\$:	TST	(SP)+	:	2613
000316	013701	000000G		MOV	RP,ADDR,R1	:	2623
000322	005000			CLR	R0	:	
000324	126127	000003	000003	CMPB	3(R1),#3	:	
000332	001002			BNE	5\$	:	
000334	005200			INC	R0	:	
000336	000407			BR	6\$	:	
000340	132761	000360	000002	BITB	#360,2(R1)	:	2624
000346	001333			BNE	3\$	:	
000350	105761	000014		TSTB	14(R1)	:	2625
000354	100330			BPL	3\$	:	
000356	006000		6\$:	ROR	R0	:	2627
000360	103015			BCC	7\$	:	
000362	012716	000000G		MOV	#DBM23,(SP)	:	2630
000366	012746	000001		MOV	#1,-(SP)	:	
000372	010600			MOV	SP,R0	:	
000374	104417			TRAP	17	:	
000376	013716	000000G		MOV	RP,INDX,(SP)	:	2631
000402	004737	000000G		JSR	PC,PUT,RETPKT	:	
000406	004737	000000V		JSR	PC,DR,ERR	:	2632
000412	000447			BR	10\$	:	2633
000414	126127	000014	000204	CMPB	14(R1),#204	:	2638
000422	001007			BNE	8\$	:	
000424	016100	000022		MOV	22(R1),R0	:	2639
000430	042700	177657		BIC	#177657,R0	:	
000434	020027	000120		CMP	R0,#120	:	
000440	001437			BEQ	11\$	:	
000442	013700	000000G	8\$:	MOV	CCTLR,R0	:	2642
000446	006300			ASL	R0	:	
000450	105260	000000G		INCB	C,ERR,TBL(R0)	:	
000454	032737	000001	001162'	BIT	#1,APT,MODE	:	2644
000462	001405			BEQ	9\$	:	
000464	012777	000001	001164'	MOV	#1,EMAIL,BOX,TESTNUM	:	2645
000472	005077	001166'		CLR	EMAIL,BOX,SUBTST	:	2648
000476	104455		9\$:	TRAP	55	:	2651

L6

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2:6

SEQ 0283
Page 52
(12)

000500	000025			.WORD	25		
000502	000000G			.WORD	EGD.21		
000504	000000G			.WORD	EMS.21		
000506	013716	000000G		MOV	CCTL R,(SP)	:	2652
000512	012746	000006		MOV	#6,-(SP)		
000516	004737	000000G		JSR	PC,DROP.CTLR		
000522	013716	000000G		MOV	RP,INDX,(SP)	:	2653
000526	004737	000000G		JSR	PC,PUT.RETPKT		
000532	062706	000010	10\$:	ADD	#10,SP	:	2654
000536	000433			BR	14\$	:	2641
000540	016137	000024	000000G	11\$:	MOV	24(R1),CMD.TIME	2658
000546	006337	000000G		ASL	CMD.TIME		
000552	032737	000001	000000G	BIT	#1,SWP.FLAGS	:	2660
000560	001411			BEQ	12\$		
000562	016116	000024		MOV	24(R1),(SP)	:	2662
000566	012746	000000G		MOV	#DBM25,-(SP)		
000572	012746	000002		MOV	#2,-(SP)		
000576	010600			MOV	SP,R0	: SP, +	
000600	104417			TRAP	17		
000602	022626			CMP	(SP)+,(SP)+		
000604	013716	000000G	12\$:	MOV	RP,INDX,(SP)	:	2668
000610	004737	000000G		JSR	PC,PUT.RETPKT		
000614	012700	000001		MOV	#1,R0	:	2610
000620	062706	000006	13\$:	ADD	#6,SP	:	2592
000624	000401			BR	15\$	:	2570
000626	005000		14\$:	CLR	R0	:	2559
000630	012601		15\$:	MOV	(SP)+,R1		
000632	000207			RTS	PC		

; Routine Size: 206 words, Routine Base: \$CODE\$ + 4170
; Maximum stack depth per invocation: 8 words

M6

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0284
Page 53
(13)

```

: 2673 1 routine UNIT_INIT : novalue *
: 2674 1
: 2675 1
: 2676 1
: 2677 1
: 2678 1
: 2679 1
: 2680 1
: 2681 1
: 2682 1
: 2683 1
: 2684 1
: 2685 1
: 2686 1
: 2687 1
: 2688 2
: 2689 2
: 2690 2
: 2691 2
: 2692 2
: 2693 2
: 2694 2
: 2695 2
: 2696 2
: 2697 2
: 2698 2
: 2699 2
: 2700 2
: 2701 2
: 2702 3
: 2703 3
: 2704 3
: 2705 3
: 2706 3
: 2707 3
: 2708 3
: 2709 3
: 2710 2
: 2711 3
: 2712 3
: 2713 3
: 2714 4
: 2715 4
: 2716 4
: 2717 4
: 2718 4
: 2719 4
: 2720 4
: 2721 4
: 2722 4
: 2723 4
: 2724 3
: 2725 4

THIS ROUTINE IS CALLED FROM DRIVER_INIT FOR EACH CONFIGURED UNIT
(DISK) WHICH IS ATTACHED TO A CONTROLLER THAT SURVIVED
INITIALIZATION. ITS PURPOSE IS TO FORMAT AND SEND AN "ONLINE"
MESSAGE, AND TO VERIFY THE RESPONSE.

IMPLICIT INPUTS:
CCTL - CURRENT CONTROLLER NUMBER
CDISK - CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
L$LUN - CURRENT (DRS) UNIT NUMBER
CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST

begin
local
  MAX0_LBNS : WORD UNSIGNED;      ! UNIT'S MAXIMUM LO WORD LBN
  MAX1_LBNS : WORD UNSIGNED;      ! UNIT'S MAXIMUM HI WORD LBN

  P_INDEX = GET_PKT (.CCTL);      ! GET AN MSCP PACKET
  MSCP_PKT [.P_INDEX, MSGLEN] = SZ_ONL; ! PACKET SIZE
  MSCP_PKT [.P_INDEX, DK_NUM] = .CDISK; ! SET DISK ADDRESS (RD/RX DISK NUMBER)
  MSCP_PKT [.P_INDEX, OPCODE] = OP_ONL; ! OPCODE FOR "ONLINE"
  MSCP_PKT [.P_INDEX, DDPAR] = BIT00; ! SHOW ALL ECC ERRORS IN ERROR LOG MESSAGES
  MSCP_PKT [.P_INDEX, CMD_TYPE] = SEQ_CMD; ! SEQUENTIAL COMMAND

  if SEND (.P_INDEX) eq FAILURE ! ATTEMPT TO SEND; IF CTLR IS OFFLINE
  then
    begin
      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
      CST_ADDR [.CUOFF, D_FATAL] = TRUE; ! FATAL ERROR
      ERRDF (22, EGD_22, 0);
      DUR [.L$LUN] = DU_ONLINE; ! SETUP REASON TO DROP UNIT
      DODU (.L$LUN); ! DROP UNIT
      PUT_PKT (.P_INDEX); ! RETURN PACKET TO POOL
    end
  else
    begin
      ! OTHERWISE (SEND WAS SUCCESSFUL)

      do
        begin
          WAIT (); ! WAIT FOR RETPKT RESPONSE
          RP_INDX = OUT_IODQ (); ! GET INDEX OF RETPKT
          RP_ADDR = RETPKT + (.RP_INDX * RP_LEN * 2); ! CALCULATE RETPKT ADDRESS

          if .RP_ADDR [MESTYP] neq MT_SEQ ! RETURN ALL RETPKTS NOT SENT BY CONTROLLER
          then
            PUT_RETPKT (.RP_INDX);
          end
        end
      until (.RP_ADDR [CONID] eq CID_DRIVER) or
        ((.RP_ADDR [MESTYP] eq MT_SEQ) and

```

N6

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0285
Page 54
(13)

```

: 2726 3      ((.RP_ADDR [ENDCOD] and OP_END) eq1 OP_END));
: 2727 3
: 2728 3      if .RP_ADDR [CONID] eq1 CID_DRIVER      ! IF RETPKT IS FROM "DRIVER"
: 2729 3      then
: 2730 4          begin
: 2731 4              PRINYE (DBM26);                  ! "ERROR IN UNIT INIT"
: 2732 4              DR_ERR ();                      ! DROP CONTROLLER
: 2733 4          end
: 2734 3      else
: 2735 3
: 2736 4          if .RP_ADDR [ENDCOD] neq (OP_ONL or OP_END) ! IF RETPKT IS FROM DISK MSCP
: 2737 3          then
: 2738 4              begin
: 2739 4                  T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 2740 4                  CST_ADDR [.CUOFF, D_FATAL] = TRUE;
: 2741 4                  ERRDF (23, EGD_23, EMS_21);    ! FATAL ERROR
: 2742 4                  DUR [.L$LUN] = DU_ONLINE;    ! SETUP REASON TO DROP UNIT
: 2743 4                  DODU (.L$LUN);              ! DROP UNIT
: 2744 4              end
: 2745 3          else
: 2746 4              begin
: 2747 4                  ST_CODE = .RP_ADDR [STSCOD];    ! RETPKT HAS GOOD ENDCODE
: 2748 4                  SB_CODE = .RP_ADDR [SUBCOD];  ! GET STATUS CODE
: 2749 4                  ! GET SUB-CODE
: 2750 4                  if .ST_CODE neq ST_SUC      ! IF STATUS CODE IS NOT SUCCESSFUL
: 2751 4                  then
: 2752 5                      begin
: 2753 5                          T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 2754 5                          CST_ADDR [.CUOFF, D_FATAL] = TRUE;
: 2755 5                          ERRDF (15, EGD_15, EMS_30); ! ONLINE FAILED
: 2756 5                          DUR [.L$LUN] = DU_ONLINE; ! SET UP REASON FOR DROPPING UNIT
: 2757 5                          DODU (.L$LUN);          ! DROP UNIT
: 2758 5                      end
: 2759 4                  else
: 2760 5                      begin
: 2761 5                          ! SUCCESSFUL OPERATION
: 2762 5                          MAX0_LBNS = .RP_ADDR [SIZE0]; ! LOAD LOWER WORD OF UNIT SIZE
: 2763 5                          MAX1_LBNS = .RP_ADDR [SIZE1]; ! LOAD UPPER WORD OF UNIT SIZE
: 2764 5
: 2765 6                          if (.MAX0_LBNS eq1 0)      ! THIS SUBTRACTS ONE FROM THE TOTAL
: 2766 5                          then                      ! BECAUSE EVERYTHING STARTS AT 0
: 2767 6                              begin                ! THROUGH (MAXIMUM - 1)
: 2768 6                                  MAX0_LBNS = #0'177777';
: 2769 6                                  MAX1_LBNS = .MAX1_LBNS - 1;
: 2770 6                              end
: 2771 5                          else
: 2772 5                              MAX0_LBNS = .MAX0_LBNS - 1;
: 2773 5
: 2774 5                          if (.CST_ADDR [.CUOFF + 2, D_BEG1] gtru .MAX1_LBNS) or      ! THIS SECTION CHECKS TO SEE
: 2775 5                          ((.CST_ADDR [.CUOFF + 2, D_BEG1] eq1u .MAX1_LBNS) and      ! IN SOFTWARE QUESTIONS WERE
: 2776 6                          (.CST_ADDR [.CUOFF + 1, D_BEG0] gtru (.MAX0_LBNS - 1)))    ! DEVICE SPECIFIED
: 2777 6                          ! note 1 less than max. on diagnosti
: 2778 6
: 2779 6
: 2780 6
: 2781 6
: 2782 6
: 2783 6
: 2784 6
: 2785 6
: 2786 6
: 2787 6
: 2788 6
: 2789 6
: 2790 6
: 2791 6
: 2792 6
: 2793 6
: 2794 6
: 2795 6
: 2796 6
: 2797 6
: 2798 6
: 2799 6
: 2800 6
: 2801 6
: 2802 6
: 2803 6
: 2804 6
: 2805 6
: 2806 6
: 2807 6
: 2808 6
: 2809 6
: 2810 6
: 2811 6
: 2812 6
: 2813 6
: 2814 6
: 2815 6
: 2816 6
: 2817 6
: 2818 6
: 2819 6
: 2820 6
: 2821 6
: 2822 6
: 2823 6
: 2824 6
: 2825 6
: 2826 6
: 2827 6
: 2828 6
: 2829 6
: 2830 6
: 2831 6
: 2832 6
: 2833 6
: 2834 6
: 2835 6
: 2836 6
: 2837 6
: 2838 6
: 2839 6
: 2840 6
: 2841 6
: 2842 6
: 2843 6
: 2844 6
: 2845 6
: 2846 6
: 2847 6
: 2848 6
: 2849 6
: 2850 6
: 2851 6
: 2852 6
: 2853 6
: 2854 6
: 2855 6
: 2856 6
: 2857 6
: 2858 6
: 2859 6
: 2860 6
: 2861 6
: 2862 6
: 2863 6
: 2864 6
: 2865 6
: 2866 6
: 2867 6
: 2868 6
: 2869 6
: 2870 6
: 2871 6
: 2872 6
: 2873 6
: 2874 6
: 2875 6
: 2876 6
: 2877 6
: 2878 6
: 2879 6
: 2880 6
: 2881 6
: 2882 6
: 2883 6
: 2884 6
: 2885 6
: 2886 6
: 2887 6
: 2888 6
: 2889 6
: 2890 6
: 2891 6
: 2892 6
: 2893 6
: 2894 6
: 2895 6
: 2896 6
: 2897 6
: 2898 6
: 2899 6
: 2900 6
: 2901 6
: 2902 6
: 2903 6
: 2904 6
: 2905 6
: 2906 6
: 2907 6
: 2908 6
: 2909 6
: 2910 6
: 2911 6
: 2912 6
: 2913 6
: 2914 6
: 2915 6
: 2916 6
: 2917 6
: 2918 6
: 2919 6
: 2920 6
: 2921 6
: 2922 6
: 2923 6
: 2924 6
: 2925 6
: 2926 6
: 2927 6
: 2928 6
: 2929 6
: 2930 6
: 2931 6
: 2932 6
: 2933 6
: 2934 6
: 2935 6
: 2936 6
: 2937 6
: 2938 6
: 2939 6
: 2940 6
: 2941 6
: 2942 6
: 2943 6
: 2944 6
: 2945 6
: 2946 6
: 2947 6
: 2948 6
: 2949 6
: 2950 6
: 2951 6
: 2952 6
: 2953 6
: 2954 6
: 2955 6
: 2956 6
: 2957 6
: 2958 6
: 2959 6
: 2960 6
: 2961 6
: 2962 6
: 2963 6
: 2964 6
: 2965 6
: 2966 6
: 2967 6
: 2968 6
: 2969 6
: 2970 6
: 2971 6
: 2972 6
: 2973 6
: 2974 6
: 2975 6
: 2976 6
: 2977 6
: 2978 6
: 2979 6
: 2980 6
: 2981 6
: 2982 6
: 2983 6
: 2984 6
: 2985 6
: 2986 6
: 2987 6
: 2988 6
: 2989 6
: 2990 6
: 2991 6
: 2992 6
: 2993 6
: 2994 6
: 2995 6
: 2996 6
: 2997 6
: 2998 6
: 2999 6
: 3000 6

```

IF LBNS LISTED

TO LARGE FOR

c will error

ZRGAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss 16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2:6SEQ 0286
Page 55
(13)

```

: 2779 5      then
: 2780 6      begin
: 2781 6      CST_ADDR [ .CUOFF + 2, D_BEG1 ] = 0;
: 2782 6      CST_ADDR [ .CUOFF + 1, D_BEG0 ] = 0;
: 2783 5      end;
: 2784 5
: 2785 5      if
: 2786 5      ( .CST_ADDR [ .CUOFF + 4, D_END1 ] gtru .MAX1_LBNS ) or
: 2787 5      ( ( .CST_ADDR [ .CUOFF + 4, D_END1 ] eqlo .MAX1_LBNS ) and
: 2788 6      ( .CST_ADDR [ .CUOFF + 3, D_END0 ] gtru .MAX0_LBNS ) )
: 2789 6
: 2790 5      then
: 2791 6      begin
: 2792 6      CST_ADDR [ .CUOFF + 4, D_END1 ] = .MAX1_LBNS;
: 2793 6      CST_ADDR [ .CUOFF + 3, D_END0 ] = .MAX0_LBNS;
: 2794 5      end;
: 2795 5
: 2796 5
: 2797 5
: 2798 7      if ( ( .ENTRY_REASON eq1 RESTART ) or
: 2799 6      ( .ENTRY_REASON eq1 START ) ) and
: 2800 6
: 2801 6      ( .CRN_LOW leq 8 ) and
: 2802 6      ( .CRN_HIGH eq1 0 )
: 2803 6
: 2804 5      THEN
: 2805 6      begin
: 2806 6      BST [ .L$LUN, LO_WRD ] = .CST_ADDR [ .CUOFF + 1, D_BEG0 ];
: 2807 6      BST [ .L$LUN, HI_WRD ] = .CST_ADDR [ .CUOFF + 2, D_BEG1 ];
: 2808 6      TRK_SGN [ .L$LUN ] = 1;
: 2809 5      end;
: 2810 5
: 2811 5      selectoneu .WPH_ADDR [ R_MODEL ] of
: 2812 5
: 2813 5      set
: 2814 5
: 2815 5      [ #0'6' ] : CST_ADDR [ .CUOFF, D_TYPE ] = RD_51;
: 2816 5      [ #0'7' ] : CST_ADDR [ .CUOFF, D_TYPE ] = RX_50;
: 2817 5      [ #0'10' ] : CST_ADDR [ .CUOFF, D_TYPE ] = RD_52;
: 2818 5
: 2819 6      [ otherwise ] : BEGIN
: 2820 6      ERRDF (25, EGD_24, EMS_30);
: 2821 5      END;
: 2822 5
: 2823 5      tes;
: 2824 5
: 2825 6      if ( ( .RP_ADDR [ U_FLAGS ] and ( .WPH ) eq1 ( .WPH ) and
: 2826 5      ( .CST_ADDR [ .CUOFF, D_PROT ] eq1 UNPROTECTED ) )
: 2827 6      then
: 2828 6      begin
: 2829 6      T_ADDR [ ERR_HRD_DRV ] = .T_ADDR [ ERR_HRD_DRV ] + 1;
: 2830 6      CST_ADDR [ .CUOFF, D_FATAL ] = TRUE;
: 2831 6      ERRDF (16, EGD_16, EMS_30);
: 2831 6      DUR [ .L$LUN ] = DU_PROTECT;

```

! 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 DIRECTION

! 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

C/

ZRGAM5
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL216SEQ 0287
Page 56
(13)

```

: 2832 6          DODU (.L$LUN);          ! DROP UNIT
: 2833 6          end
: 2834 5          else
: 2835 6          begin
: 2836 6          CST_ADDR [.CUOFF, D_STAT] = ONLINE;          ! WRITE PROTECT SWITCH IS O.K.
: 2837 6          CST [.CCTLR, U_CNT] = .CST [.CCTLR, U_CNT] + 1;          ! SET ONLINE FLAG
: 2838 5          end;          ! ADD UNIT TO CTRL TABLE
: 2839 4          end;
: 2840 3          end;          ! IF RETPKT HAS CORRECT ENCODE
: 2841 3          PUT_RETPKT (.RP_INDX);
: 2842 3          end;          ! IF SEND WAS SUCCESSFUL
: 2843 2
: 2844 2
: 2845 1          end;          ! ROUTINE UNIT-INIT

```

			SBTTL	UNIT.INIT INITIALIZATION TEST ROUTINES	
000000	004137	000000G	UNIT.INIT:		
			JSR	R1, \$SAVE5	2673
000004	024646		CMP	-(SP), -(SP)	
000006	013746	000000G	MOV	CCTLR, -(SP)	2693
000012	004737	000000C	JSR	PC, GET.PKT	
000016	010037	000000G	MOV	R0, P.INDEX	
000022	010016		MOV	R0, (SP)	
000024	012746	000106	MOV	#106, -(SP)	2694
000030	004737	000000G	JSR	PC, BL \$INUL	
000034	012760	000044 000006G	MOV	#44, MSCP.PKT.6(R0)	
000042	013760	000000G 000016G	MOV	CUISK, MSCP.PKT.16(R0)	2695
000050	112760	000011 000022G	MOVB	#11, MSCP.PKT.22(R0)	2696
000056	012760	000001 000046G	MOV	#1, MSCP.PKT.46(R0)	2697
000064	112760	000001 000004G	MOVB	#1, MSCP.PKT.4(R0)	2698
000072	013716	000000G	MOV	P.INDEX, (SP)	2700
000076	004737	000000G	JSR	PC, SEND	
000102	005700		TST	R0	
000104	001033		BNE	14	
000106	013700	000000G	MOV	T.ADDR, R0	2703
000112	105260	000051	INCB	51(R0)	
000116	013700	000000G	MOV	CUOFF, R0	2704
000122	006300		ASL	R0	
000124	063700	000000G	ADD	CST.ADDR, R0	
000130	052710	010000	BIS	#10000, (R0)	
000134	104455		TRAP	55	2705
000136	000026		.WORD	26	
000140	000000G		.WORD	EGD.22	
000142	006000		.WORD	0	
000144	013700	000000G	MOV	1 \$UN, R0	2706
000150	112760	000007 000000G	MOVB	#7, DUR(R0)	
000156	104451		TRAP	51	2707
000160	013716	000000G	MOV	P.INDEX, (SP)	2708
000164	004737	000000G	JSR	PC, PUT.PKT	
000170	000137	006330'	JMP	244	2709
000174	004737	000000G	JSR	PC, WAIT	2710
000200	004737	000000G	JSR	PC, OUT.1000	2716

17

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11 Apr-1984 11:08:35
11 Apr-1984 11:08:22

VAX-11 B1199-16 v4.0-579
DISK#USER2:[POWERS]ZRQADO.BL2;6

SEQ 0288
Page 57
(13)

000204	010037	000000G		MOV	R0,RP,INDX		
000210	010016			MOV	R0,(SP)	; RP,INDX,*	2717
000212	012746	000054		MOV	#54,-(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)		2719
000240	001404			BEQ	2#		
000242	013716	000000G		MOV	RP,INDX,(SP)		2721
000246	004737	000000G		JSR	PC,PUT,RETPKT		
000252	005726		2#:	TST	(SP),		2714
000254	013702	000000G		MOV	RP,ADDR,R2		2724
000260	005000			CLR	R0		
000262	126227	000003	000003	CMPB	3(R2),#3		
000270	001002			BNE	3#		
000272	005200			INC	R0		
000274	000407			BR	4#		
000276	132762	000360	000002	BITB	#360,2(R2)		2725
000304	001333			BNE	1#		
000306	105762	000014		TSTB	14(R2)		2726
000312	100330			BPL	1#		
000314	006000		4#:	ROR	R0		2728
000316	103012			BCC	5#		
000320	012716	000000G		MOV	#0BM26,(SP)		2731
000324	012746	000001		MOV	#1,-(SP)		
000330	010600			MOV	SP,R0	; SP,*	
000332	104417			TRAP	17		
000334	004737	000000V		JSR	PC,DR,ERR		2732
000340	000137	006316'		JMP	23#		2730
000344	013705	000000G	5#:	MOV	CST,ADDR,R5		2740
000350	013766	000000G	000006	MOV	CUOFF,6(SP)		
000356	006366	000006		ASL	6(SP)		
000362	060566	000006		ADD	R5,6(SP)		
000366	126227	000014	000211	CMPB	14(R2),#211		2736
000374	001422			BEQ	6#		
000376	013700	000000G		MOV	T,ADDR,R0		2739
000402	105260	000050		INCB	50(R0)		
000406	052776	010000	000006	BIS	#10000,#6(SP)		2740
000414	104455			TRAP	55		2741
000416	000027			.WORD	27		
000420	000000G			.WORD	EGD,23		
000422	000000G			.WORD	EMS,21		
000424	013706	000000G		MOV	L#LUN,R0		2742
000430	112760	000007	000000G	MOVB	#7,DUR(R0)		
000436	104451			TRAP	51		2743
000440	000445			BR	7#		2736
000442	116237	000016	000000G	MOVB	16(R2),ST.CODE		2747
000450	042737	177740	000000G	BIC	#177740,ST.CODE		
000456	016200	000016		MOV	16(R2),R0		2748
000462	006200			ASR	R0		
000464	006200			ASR	R0		
000466	006200			ASR	R0		
000470	006200			ASR	R0		

E7

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0289
Page 58
(13)

000472	006200		ASR	R0		
000474	042700	174000	BIC	#174000,R0		
000500	010037	000000G	MOV	R0,SB.CODE		
000504	005737	000000G	TST	ST.CODE		
000510	001423		BEQ	8\$		2750
000512	013700	000000G	MOV	T.ADDR,R0		
000516	105260	000050	INCB	50(R0)		2753
000522	052776	010000 000006	BIS	#10000,#6(SP)		2754
000530	104455		TRAP	55		2755
000532	000017		.WORD	17		
000534	000000G		.WORD	EGD.15		
000536	000000G		.WORD	EMS.30		
000540	013700	000000G	MOV	L\$LUN,R0		2756
000544	112760	000007 000000G	MOVB	#7,DUR(R0)		
000552	104451		TRAP	51		2757
000554	000137	006320	7\$: JMP	23\$		2750
000560	016203	000044	8\$: MOV	44(R2),R3	; *,MAX0.LBNS	2762
000564	016204	000046	MOV	46(R2),R4	; *,MAX1.LBNS	2763
000570	005703		TST	R3	; MAX0.LBNS	2765
000572	001004		BNE	9\$		
000574	012703	177777	MOV	#-1,R3	; *,MAX0.LBNS	2768
000600	005304		DEC	R4	; MAX1.LBNS	2769
000602	000401		BR	10\$		2765
000604	005303		9\$: DEC	R3	; MAX0.LBNS	2772
000606	013700	000000G	10\$: MOV	CUOFF,R0		2774
000612	006300		ASL	R0		
000614	060500		ADD	R5,R0		
000616	011766	000004	MOV	(PC),4(SP)		
000622	060066	000004	ADD	R0,4(SP)		
000626	027604	000004	CMP	#4(SP),R4	; *,MAX1.LBNS	
000632	101012		BHI	11\$		
000634	001021		BNE	12\$		2776
000636	013700	000000G	MOV	CUOFF,R0		2777
000642	006300		ASL	R0		
000644	060500		ADD	R5,R0		
000646	010301		MOV	R3,R1	; MAX0.LBNS,*	
000650	005301		DEC	R1		
000652	026001	000002	CMP	2(R0),R1		
000656	101410		BLOS	12\$		
000660	005076	000004	11\$: CLR	#4(SP)		2781
000664	013700	000000G	MOV	CUOFF,R0		2782
000670	006300		ASL	R0		
000672	060500		ADD	R5,R0		
000674	005060	000002	CLR	2(R0)		
000700	013700	000000G	12\$: MOV	CUOFF,R0		2786
000704	006300		ASL	R0		
000706	060500		ADD	R5,R0		
000710	026004	000010	CMP	10(R0),R4	; *,MAX1.LBNS	
000714	101010		BHI	13\$		
000716	001017		BNE	14\$		2788
000720	013701	000000G	MOV	CUOFF,R1		2789
000724	006301		ASL	R1		
000726	060501		ADD	R5,R1		

F7

ZRQAM3 V01.6	RD/RX EXERCISER INITIALIZATION TEST ROUTINES	11-Apr-1984 11:08:35 11-Apr-1984 11:08:22	VAX-11 Bliss-16 V4.0-579 DISK\$USER2:[POWERS]ZRQAD0.BL2;6	SEQ 029 Page 59 (13)
000730 026103 000006		CMP 6(R1),R3	; MAX0.LBNS	
000734 101410		BLOS 14;		
000736 010460 000010	13\$:	MOV R4,10(R0)	; MAX1.LBNS, +	2792
000742 013700 000000G		MOV CUOFF,R0	;	2793
000746 006300		ASL R0		
000750 060500		ADD R5,R0		
000752 010360 000006		MOV R3,6(R0)	; MAX0.LBNS, +	
000756 123727 000000G 000002	14\$:	CMPB ENTRY.REASON,#2	;	2798
000764 001404		BEQ 15;		
000766 123727 000000G 000001		CMPB ENTRY.REASON,#1	;	2799
000774 001031		BNE 16;		
000776 023727 000000G 000010	15\$:	CMP CRN.LOW,#10	;	2801
001004 003025		BGT 16;		
001006 005737 000000G		TST CRN.HIGH	;	2802
001012 001022		BNE 16;		
001014 013704 000000G		MOV L\$LOW,R4	;	2806
001020 010400		MOV R4,R0		
001022 006300		ASL R0		
001024 006300		ASL R0		
001026 013701 000000G		MOV CUOFF,R1		
001032 006301		ASL R1		
001034 060501		ADD R5,R1		
001036 016160 000002 000000G		MOV 2(R1),BST(R0)		
001044 017660 000004 000002G		MOV 84(SP),BST+2(R0)	;	2807
001052 112764 000001 000000G		MOVB #1,TRK.SGN(R4)	;	2808
001060 005001	16\$:	CLR R1	;	2811
001062 156201 000032		BISB 32(R2),R1	;	
001066 020127 000006		CMP R1,#6	;	2815
001072 001007		BNE 17;		
001074 013700 000000G		MOV CUOFF,R0		
001100 006300		ASL R0		
001102 060500		ADD R5,R0		
001104 152710 000020		BISB #20,(R0)		
001110 000421		BR 20;	;	2811
001112 020127 000007	17\$:	CMP R1,#7	;	2816
001116 001403		BEQ 18;		
001120 020127 000010		CMP R1,#10	;	2817
001124 001007		BNE 19;		
001126 013700 000000G	18\$:	MOV CUOFF,R0		
001132 006300		ASL R0		
001134 060500		ADD R5,R0		
001136 142710 000020		BICB #20,(R0)		
001142 000404		BR 20;	;	2811
001144 104455	19\$:	TRAP 55	;	2820
001146 000031		.WORD 31		
001150 000000G		.WORD EGD.24		
001152 000000G		.WORD EMS.30		
001154 032762 020000 000022	20\$:	BIT #20000,22(R2)	;	2824
001162 001430		BEQ 21;		
001164 013700 000000G		MOV CUOFF,R0	;	2825
001170 006300		ASL R0		
001172 060500		ADD R5,R0		
001174 005710		TST (R0)		

G7

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0291
Page 60
(13)

001176	100022			BPL	21\$		
001200	013700	000000G		MOV	T,ADDR,RO	:	2828
001204	105260	000050		INCB	50(RO)		
001210	052776	010000	000006	BIS	010000,06(SP)	:	2829
001216	104455			TRAP	55	:	2830
001220	000020			.WORD	20		
001222	000000G			.WORD	EGD.16		
001224	000000G			.WORD	EMS.30		
001226	013700	000000G		MOV	L\$LUN,RO	:	2831
001232	112760	000011	000000G	MOVB	011,DUR(RO)		
001240	104451			TRAP	51	:	2832
001242	000414			BR	23\$	:	2824
001244	052776	020000	000006	21\$: BIS	020000,06(SP)	:	2836
001252	013716	000000G		MOV	CCTLR,(SP)	:	2837
001256	012746	000126		MOV	0126,-(SP)		
001262	004737	000000G		JSR	PC,BL\$MUL		
001266	105260	000005G		INCB	CST:5(RO)		
001272	005726		22\$:	TST	(SP).	:	2835
001274	013716	000000G	23\$:	MOV	RP,INDX,(SP)	:	2842
001300	004737	000000G		JSR	PC,PUT,RETPKT		
001304	062706	000010	24\$:	ADD	010,SP	:	2673
001310	000207			RTS	PC		

; Routine Size: 357 words, Routine Base: \$CODE\$ + 5024
; Maximum stack depth per invocation: 13 words

H7

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0,BL2;6SEQ 0292
Page 61
(14)

```

; 2846 1 GLOBAL routine DR_ERR : novalue *
; 2847 1
; 2848 1
; 2849 1
; 2850 1
; 2851 1
; 2852 1
; 2853 1
; 2854 1
; 2855 1
; 2856 1
; 2857 1
; 2858 1
; 2859 1
; 2860 2
; 2861 2
; 2862 2
; 2863 2
; 2864 2
; 2865 2
; 2866 2
; 2867 2
; 2868 2
; 2869 1

    !!
    !! THIS ROUTINE IS DESIGNED TO PROCESS RETURN PACKETS THAT ORIGINATE AT
    !! THE "DRIVER" RATHER THAN THE DEVICE. DRIVER-ORIGINATED PACKETS INDICATE
    !! EITHER A FATAL DEVICE ERROR OR A COMMAND TIMEOUT. SINCE THIS ROUTINE IS
    !! ONLY CALLED DURING THE INITIALIZATION TEST, IT TREATS A COMMAND TIMEOUT
    !! AS AN INITIALIZATION ERROR.
    !!
    !! IMPLICIT INPUTS:
    !!     RP_ADDR - ADDRESS OF A RETPKT THAT ORIGINATED AT THE "DRIVER"
    !!               (I.E., CONNECTION ID = CID_DRIVER)
    !!
    begin
    local
    REASON : word initial (DU_TIME);      ! ASSUME COMMAND TIMEOUT
    if .RP_ADDR [MESTYP] eq1 MT_FATAL      ! IF FATAL DEVICE ERROR
    then
    DROP_CTLR (.CCTL, .REASON);           ! DROP ALL UNITS ON CONTROLLER
    end;

```

			.SBTTL	DR.ERR INITIALIZATION TEST ROUTINES	
000000	010146		DR.ERR::	MOV R1, -(SP)	2846
000002	012701	000012		MOV #12, R1	2840
000006	013700	000000G		MOV RP_ADDR, R0	2865
000012	116000	000002		MOVB 2(R0), R0	
000016	042700	177417		BIC #177417, R0	
000022	020027	000060		CMP R0, #60	
000026	001006			BNE 1\$	
000030	013746	000000G		MOV CCTL, -(SP)	2868
000034	010146			MOV R1, -(SP)	
000036	004737	000000G		JSR PC, DROP_CTLR	
000042	022626			CMP (SP)+, (SP)+	
000044	012601		1\$:	MOV (SP)+, R1	2846
000046	000207			RTS PC	

```

; Routine Size: 20 words,      Routine Base: $CODE$ + 6336
; Maximum stack depth per invocation: 4 words

```

ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0293
Page 62
(15)

```

: 2870 1 routine ACCESS : novalue =
: 2871 1
: 2872 1 !
: 2873 1 ! THIS ROUTINE IS CALLED BY INIT_TEST TO VERIFY THAT THE CURRENT DISK
: 2874 1 ! CAN BE ACCESSED. THIS OBJECTIVE IS ACCOMPLISHED BY FORMATTING AND
: 2875 1 ! SENDING ONE OR TWO MSCP ACCESS COMMANDS TO THE DISK, AND CHECKING
: 2876 1 ! THE STATUS FIELD OF THE RESPONSE MESSAGE(S).
: 2877 1 !
: 2878 1 ! IMPLICIT INPUTS:
: 2879 1 ! CCTLN - CURRENT CONTROLLER NUMBER
: 2880 1 ! CDISK - CURRENT DISK ADDRESS (RD/RX DISK NUMBER)
: 2881 1 ! L$LUN - CURRENT (DRS) UNIT NUMBER
: 2882 1 !
: 2883 1
: 2884 2 begin
: 2885 2
: 2886 2 local
: 2887 2 RESULT : word initial (FAILURE), ! GUILTY UNTIL PROVEN INNOCENT
: 2888 2 LBN : word,
: 2889 2 PASS : word initial (1); ! LOOP PASS COUNT
: 2890 2
: 2891 2 ST_CODE = SB_CODE = 0; ! STATUS CODE AND SUB-CODE
: 2892 2 LBN = (((MAX_LBN [L$LUN] + 1) + -1) and #0'77777') - 1;
: 2893 2 ! START WITH LAST LBN ON TOP SURFACE: [(X+1)/2] -1
: 2894 2
: 2895 2 do
: 2896 3 begin ! LOOP STARTS HERE
: 2897 3 P_INDEX = GET_PKT (.CCTLN); ! GET AN MSCP PACKET
: 2898 3 MSCP_PKT [.P_INDEX, DK_NUM] = .CDISK; ! SET DISK ADDR (RD/RX DISK NUMBER)
: 2899 3 MSCP_PKT [.P_INDEX, OPCODE] = OP_ACC; ! ACCESS OPCODE
: 2900 3 MSCP_PKT [.P_INDEX, BC_LU] = 512; ! BYTE COUNT (1 BLOCK)
: 2901 3 MSCP_PKT [.P_INDEX, LBN_L] = .LBN; ! LOGICAL BLOCK NUMBER
: 2902 3 MSCP_PKT [.P_INDEX, CMD_TYPE] = NON_SEQ_CMD; ! NON-SEQUENTIAL COMMAND
: 2903 3
: 2904 3 if SEND (.P_INDEX) eq FAILURE ! ATTEMPT TO SEND; IF CCTLN NOT ONLINE
: 2905 3 then
: 2906 4 begin
: 2907 4 PUT_PKT (.P_INDEX); ! RETURN PACKET TO POOL
: 2908 4 PASS = 2; ! NO MORE TRIES
: 2909 4 end
: 2910 3 else
: 2911 4 begin ! IF SEND WAS SUCCESSFUL
: 2912 4
: 2913 4 do
: 2914 5 begin
: 2915 5 WAIT (); ! WAIT FOR RESPONSE
: 2916 5 RP_INDEX = OUT_IODQ (); ! GET RETPKT (RESPONSE) INDEX
: 2917 5 RP_ADDR = RETPKT + (.RP_INDEX * RP_LEN + 2); ! CALCULATE RETPKT ADDRESS
: 2918 5
: 2919 5 if .RP_ADDR [MESTIP] neq MT_SEQ ! RETURN ALL RETPKTS NOT SENT BY CONTROLLER
: 2920 5 then
: 2921 5 PUT_RETPKT (.RP_INDEX);
: 2922 5

```


ZRQAM3
V01.6RD/RX EXERCISER
INITIALIZATION TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0294
Page 63
(15)

```

: 2923 5      end
: 2924 4      until (.RP_ADDR [CONID] eq1 CID_DRIVER) or
: 2925 5          ((.RP_ADDR [MESTYP] eq1 MT_SEQ) and
: 2926 4          ((.RP_ADDR [ENDCOD] and OP_END) eq1 OP_END));
: 2927 4
: 2928 4      if .RP_ADDR [CONID] eq1 CID_DRIVER ! IF RETPKT CAME FROM "DRIVER"
: 2929 4      then
: 2930 4          PASS = 2 ! NO MORE TRIES
: 2931 4      else
: 2932 4
: 2933 5          if .RP_ADDR [ENDCOD] neq (OP_ACC or OP_END)
: 2934 4          then
: 2935 5              begin
: 2936 5                  PRINTF (DBM29); ! "RETPKT HAS BAD ENDCODE"
: 2937 5                  EMSCMD ();
: 2938 5              end
: 2939 4          else
: 2940 5              begin ! RETPKT HAS CORRECT ENDCODE
: 2941 5                  ST_CODE = .RP_ADDR [STSCOD]; ! GET STATUS CODE FROM PACKET
: 2942 5                  SB_CODE = .RP_ADDR [SUBCOD]; ! GET SUB-CODE FROM PACKET
: 2943 5
: 2944 5                  if .ST_CODE eq1 ST_SUC ! IF STATUS CODE INDICATES SUCCESS
: 2945 5                  then
: 2946 6                      begin
: 2947 6                          RESULT = SUCCESS;
: 2948 6                          PASS = 2; ! NO NEED TO TRY AGAIN
: 2949 5                      end;
: 2950 5
: 2951 4                  end; ! IF RETPKT HAS CORRECT ENDCODE
: 2952 4
: 2953 4                  PUT_RETPKT (.RP_INDX);
: 2954 3                  end; ! IF SEND WAS SUCCESSFUL
: 2955 3
: 2956 3                  LBN = .LBN + 1; ! ADVANCE TO FIRST LBN OF BOTTOM SURFACE
: 2957 3                  PASS = .PASS + 1; ! SECOND PASS
: 2958 3                  end ! END OF PASS LOOP
: 2959 2      until .PASS gequ 3;
: 2960 2
: 2961 2      if .RESULT eq1 FAILURE
: 2962 2      then
: 2963 3          begin
: 2964 3              T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 2965 3              CST_ADDR [.CUOFF, D_FATAL] = TRUE; ! FATAL ERROR
: 2966 3              ERROF (17, EGD_17, EMS_30); ! ACCESS FAILED
: 2967 3              DUR [.L$LUN] = DU_ACCESS; ! SET REASON TO DROP UNIT
: 2968 3              DODU (.L$LUN); ! DROP UNIT
: 2969 2              end; ! IF ACCESS FAILED
: 2970 2
: 2971 1      end; ! ROUTINE ACCESS

```

000000 004137 000000G

 .SRITL ACCESS INITIALIZATION TEST ROUTINES
 ACCESS: JSR R1,\$SAVE4 ;

2870

K7

ZRQAM3
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0295
Page 64
(15)

000004	005003		CLR	R3	; RESULT	2884
000006	012702	000001	MOV	#1,R2	; *,PASS	
000012	005037	000000G	CLR	SB.CODE		2891
000016	005037	000000G	CLR	ST.CODE		
000022	013700	000000G	MOV	L\$LUN,R0		2892
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	CCTL,R,-(SP)		2897
000054	004737	000000G	JSR	PC,GET.PKT		
000060	010037	000000G	MOV	R0,P.INDEX		
000064	010016		MOV	R0,(SP)	; P.INDEX, *	2898
000066	012746	000106	MOV	#106,-(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)		2899
000112	012760	001000 000026G	MOV	#1000,MSCP.PKT+26(R0)		2900
000120	010460	000046G	MOV	R4,MSCP.PKT+46(R0)	; LBN, *	2901
000124	112760	000002 000004G	MOVB	#2,MSCP.PKT+4(R0)		2902
000132	013716	000000G	MOV	P.INDEX,(SP)		2904
000136	004737	000000G	JSR	PC,SEND		
000142	005700		TST	R0		
000144	001007		BNE	2\$		
000146	013716	000000G	MOV	P.INDEX,(SP)		2907
000152	004737	000000G	JSR	PC,PUT.PKT		
000156	012702	000002	MOV	#2,R2	; *,PASS	2908
000162	000522		BR	9\$		2904
000164	004737	000000G	JSR	PC,WAIT		2915
000170	004737	000000G	JSR	PC,OUT.IODQ		2916
000174	010037	000000G	MOV	R0,RP.INDX		
000200	010016		MOV	R0,(SP)	; RP,INDX, *	2917
000202	012746	000054	MOV	#54,(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)		2919
000230	001404		BEQ	3\$		
000232	013716	000000G	MOV	RP,INDX,(SP)		2921
000236	004737	000000G	JSR	PC,PUT.RETPKT		
000242	005726		TST	(SP)+		2914
000244	013701	000000G	MOV	RP.ADDR,R1		2924
000250	005000		CLR	R0		
000252	126127	000003 000003	CMPB	3(R1),#3		
000260	001002		BNE	4\$		
000262	005200		INC	R0		
000264	000407		BR	5\$		
000266	132761	000360 000002	BITB	#360,2(R1)		2925
000274	001333		BNE	2\$		
000276	105761	000014	TSTB	14(R1)		2926

L7

ZRQAM5
V01.6

RD/RX EXERCISER
INITIALIZATION TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0296
Page 65
(15)

000302	100330			BPL	2\$		
000304	006000		5\$:	ROR	R0	:	2928
000306	103442			BLO	7\$	:	2930
000310	126127	000014	000220	CMPE	14(R1),#220	:	2933
000316	001410			BEQ	6\$		
000320	012716	000000G		MOV	#DBM29,(SP)	:	2936
000324	012746	000001		MOV	#1,-(SP)		
000330	010600			MOV	SP,R0	: SP,*	
000332	104417			TRAP	17		
000334	005726			TST	(SP)+	:	2935
000336	000430			BR	8\$	:	2933
000340	116137	000016	000000G	6\$:	MOVB	16(R1),ST.CODE	2941
000346	042737	177740	000000G		BIC	#177740,ST.CODE	
000354	016100	000016			MOV	16(R1),R0	2942
000360	006200				ASR	R0	
000362	006200				ASR	R0	
000364	006200				ASR	R0	
000366	006200				ASR	R0	
000370	006200				ASR	R0	
000372	042700	174000			BIC	#174000,R0	
000376	010037	000000G			MOV	R0,SB.CODE	
000402	005737	000000G			TST	ST.CODE	2944
000406	001004				BNE	8\$	
000410	012703	000001			MOV	#1,R3	2947
000414	012702	000002	7\$:		MOV	#2,R2	2948
000420	013716	000000G	8\$:		MOV	RP,INDX,(SP)	2953
000424	004737	000000G			JSR	PC,PUT.RETPKT	
000430	005204		9\$:		INC	R4	2956
000432	005202				INC	R2	2957
000434	022626				CMP	(SP)+,(SP)+	2896
000436	020227	000003			CMP	R2,#3	2959
000442	103602				BLO	1\$	
000444	005703				TST	R3	2961
000446	001025				BNE	10\$	
000450	013700	000000G			MOV	T,ADDR,R0	2964
000454	105260	000050			INCB	50(R0)	
000460	013700	000000G			MOV	CUOFF,R0	2965
000464	006300				ASL	R0	
000466	063700	000000G			ADD	CST,ADDR,R0	
000472	052710	010000			BIS	#10000,(R0)	
000476	104455				TRAP	55	2966
000500	000021				.WORD	21	
000502	000000G				.WORD	EGD,17	
000504	000000G				.WORD	EMS,30	
000506	013700	000000G			MOV	L\$UN,R0	2967
000512	112760	000010	000000G		MOVB	#10,DUR(R0)	
000520	104451				TRAP	51	2968
000522	000207		10\$:		RTS	PC	2870

; Routine Size: 170 words, Routine Base: \$CODE\$ + 6406
; Maximum stack depth per invocation: 10 words

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11gs-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0297
Page 66
(16)

```

: 2972 1 *sbttl 'MULTI-DRIVE TEST ROUTINES'
: 2973 1
: 2974 1 GLOBAL routine MULTI_DRIVE : novalue =
: 2975 1
: 2976 1 !
: 2977 1 ! THIS SUBTEST IS THE MOST SIGNIFICANT PART OF THE ENTIRE PROGRAM. THE
: 2978 1 ! MULTI-DRIVE TEST IS A HOST-CONTROLLED EXERCISER DESIGNED TO GIVE THE
: 2979 1 ! USER AN INDICATION OF HOW ONE OR SEVERAL RDRX DRIVES WOULD PERFORM IN
: 2980 1 ! AN OPERATING SYSTEM ENVIRONMENT.
: 2981 1 !
: 2982 1 ! THIS ROUTINE ACTS AS AN "EXECUTIVE" TO THE WHOLE PROCESS. AFTER
: 2983 1 ! INVOKING MD_INIT TO INITIALIZE MULTI-DRIVE TEST DATA, THIS ROUTINE
: 2984 1 ! ENTERS A LOOP WHICH ISSUES QIOs TO ALL ACTIVE CONTROLLERS AND PROCESSES
: 2985 1 ! ANY RESPONSES. IN ADDITION, ALL OUTSTANDING COMMANDS ARE TIMED IN
: 2986 1 ! DRV_TIMCHK WHICH IS INVOKED EVERY SECOND. NORMAL TERMINATION OF THIS
: 2987 1 ! LOOP OCCURS WHEN QIOs ARE NO LONGER BEING ISSUED, AND ALL OUTSTANDING
: 2988 1 ! QIOs HAVE COMPLETED.
: 2989 1 !
: 2990 1
: 2991 2 begin
: 2992 2
: 2993 2 local
: 2994 2   CUR_PRIORITY : word;
: 2995 2
: 2996 2 label
: 2997 2   SEND_COMMANDS;
: 2998 2
: 2999 2 MD_INIT ();
: 3000 2 INIT_OCCURED = TRUE;
: 3001 2
: 3002 3 do begin
: 3003 3
: 3004 3   incr CTLR from 0 to (MAX_CTLR - 1) do
: 3005 4     begin
: 3006 4       SET_CPAR (.CTLR);
: 3007 4       GETPRI (CUR_PRIORITY);
: 3008 4       SETPRI (PRI04);
: 3009 4       ICTLR = .CCTLR;
: 3010 4       ICST_ADDR = .CST_ADDR;
: 3011 4       IDCT_ADDR = .DCT_ADDR;
: 3012 4       IRDRX_ADDR = .ICST_ADDR [IP_ADDR];
: 3013 4       IDCT_ADDR [SA_SAVE] = .IRDRX_ADDR [RCSA, RC_ALL];
: 3014 4
: 3015 5       if BIT_TST (IDCT_ADDR [SA_SAVE], SA_ERR)
: 3016 4       then
: 3017 5         begin
: 3018 5           FATAL_ERROR ();
: 3019 5           SETPRI (.CUR_PRIORITY);
: 3020 5           exitloop;
: 3021 5         end
: 3022 4       else
: 3023 4         SETPRI (.CUR_PRIORITY);
: 3024 4
: 3025 4
: 3026 4
: 3027 4
: 3028 4
: 3029 4
: 3030 4
: 3031 4
: 3032 4
: 3033 4
: 3034 4
: 3035 4
: 3036 4
: 3037 4
: 3038 4
: 3039 4
: 3040 4
: 3041 4
: 3042 4
: 3043 4
: 3044 4
: 3045 4
: 3046 4
: 3047 4
: 3048 4
: 3049 4
: 3050 4
: 3051 4
: 3052 4
: 3053 4
: 3054 4
: 3055 4
: 3056 4
: 3057 4
: 3058 4
: 3059 4
: 3060 4
: 3061 4
: 3062 4
: 3063 4
: 3064 4
: 3065 4
: 3066 4
: 3067 4
: 3068 4
: 3069 4
: 3070 4
: 3071 4
: 3072 4
: 3073 4
: 3074 4
: 3075 4
: 3076 4
: 3077 4
: 3078 4
: 3079 4
: 3080 4
: 3081 4
: 3082 4
: 3083 4
: 3084 4
: 3085 4
: 3086 4
: 3087 4
: 3088 4
: 3089 4
: 3090 4
: 3091 4
: 3092 4
: 3093 4
: 3094 4
: 3095 4
: 3096 4
: 3097 4
: 3098 4
: 3099 4
: 3100 4
: 3101 4
: 3102 4
: 3103 4
: 3104 4
: 3105 4
: 3106 4
: 3107 4
: 3108 4
: 3109 4
: 3110 4
: 3111 4
: 3112 4
: 3113 4
: 3114 4
: 3115 4
: 3116 4
: 3117 4
: 3118 4
: 3119 4
: 3120 4
: 3121 4
: 3122 4
: 3123 4
: 3124 4
: 3125 4
: 3126 4
: 3127 4
: 3128 4
: 3129 4
: 3130 4
: 3131 4
: 3132 4
: 3133 4
: 3134 4
: 3135 4
: 3136 4
: 3137 4
: 3138 4
: 3139 4
: 3140 4
: 3141 4
: 3142 4
: 3143 4
: 3144 4
: 3145 4
: 3146 4
: 3147 4
: 3148 4
: 3149 4
: 3150 4
: 3151 4
: 3152 4
: 3153 4
: 3154 4
: 3155 4
: 3156 4
: 3157 4
: 3158 4
: 3159 4
: 3160 4
: 3161 4
: 3162 4
: 3163 4
: 3164 4
: 3165 4
: 3166 4
: 3167 4
: 3168 4
: 3169 4
: 3170 4
: 3171 4
: 3172 4
: 3173 4
: 3174 4
: 3175 4
: 3176 4
: 3177 4
: 3178 4
: 3179 4
: 3180 4
: 3181 4
: 3182 4
: 3183 4
: 3184 4
: 3185 4
: 3186 4
: 3187 4
: 3188 4
: 3189 4
: 3190 4
: 3191 4
: 3192 4
: 3193 4
: 3194 4
: 3195 4
: 3196 4
: 3197 4
: 3198 4
: 3199 4
: 3200 4
: 3201 4
: 3202 4
: 3203 4
: 3204 4
: 3205 4
: 3206 4
: 3207 4
: 3208 4
: 3209 4
: 3210 4
: 3211 4
: 3212 4
: 3213 4
: 3214 4
: 3215 4
: 3216 4
: 3217 4
: 3218 4
: 3219 4
: 3220 4
: 3221 4
: 3222 4
: 3223 4
: 3224 4
: 3225 4
: 3226 4
: 3227 4
: 3228 4
: 3229 4
: 3230 4
: 3231 4
: 3232 4
: 3233 4
: 3234 4
: 3235 4
: 3236 4
: 3237 4
: 3238 4
: 3239 4
: 3240 4
: 3241 4
: 3242 4
: 3243 4
: 3244 4
: 3245 4
: 3246 4
: 3247 4
: 3248 4
: 3249 4
: 3250 4
: 3251 4
: 3252 4
: 3253 4
: 3254 4
: 3255 4
: 3256 4
: 3257 4
: 3258 4
: 3259 4
: 3260 4
: 3261 4
: 3262 4
: 3263 4
: 3264 4
: 3265 4
: 3266 4
: 3267 4
: 3268 4
: 3269 4
: 3270 4
: 3271 4
: 3272 4
: 3273 4
: 3274 4
: 3275 4
: 3276 4
: 3277 4
: 3278 4
: 3279 4
: 3280 4
: 3281 4
: 3282 4
: 3283 4
: 3284 4
: 3285 4
: 3286 4
: 3287 4
: 3288 4
: 3289 4
: 3290 4
: 3291 4
: 3292 4
: 3293 4
: 3294 4
: 3295 4
: 3296 4
: 3297 4
: 3298 4
: 3299 4
: 3300 4
: 3301 4
: 3302 4
: 3303 4
: 3304 4
: 3305 4
: 3306 4
: 3307 4
: 3308 4
: 3309 4
: 3310 4
: 3311 4
: 3312 4
: 3313 4
: 3314 4
: 3315 4
: 3316 4
: 3317 4
: 3318 4
: 3319 4
: 3320 4
: 3321 4
: 3322 4
: 3323 4
: 3324 4
: 3325 4
: 3326 4
: 3327 4
: 3328 4
: 3329 4
: 3330 4
: 3331 4
: 3332 4
: 3333 4
: 3334 4
: 3335 4
: 3336 4
: 3337 4
: 3338 4
: 3339 4
: 3340 4
: 3341 4
: 3342 4
: 3343 4
: 3344 4
: 3345 4
: 3346 4
: 3347 4
: 3348 4
: 3349 4
: 3350 4
: 3351 4
: 3352 4
: 3353 4
: 3354 4
: 3355 4
: 3356 4
: 3357 4
: 3358 4
: 3359 4
: 3360 4
: 3361 4
: 3362 4
: 3363 4
: 3364 4
: 3365 4
: 3366 4
: 3367 4
: 3368 4
: 3369 4
: 3370 4
: 3371 4
: 3372 4
: 3373 4
: 3374 4
: 3375 4
: 3376 4
: 3377 4
: 3378 4
: 3379 4
: 3380 4
: 3381 4
: 3382 4
: 3383 4
: 3384 4
: 3385 4
: 3386 4
: 3387 4
: 3388 4
: 3389 4
: 3390 4
: 3391 4
: 3392 4
: 3393 4
: 3394 4
: 3395 4
: 3396 4
: 3397 4
: 3398 4
: 3399 4
: 3400 4
: 3401 4
: 3402 4
: 3403 4
: 3404 4
: 3405 4
: 3406 4
: 3407 4
: 3408 4
: 3409 4
: 3410 4
: 3411 4
: 3412 4
: 3413 4
: 3414 4
: 3415 4
: 3416 4
: 3417 4
: 3418 4
: 3419 4
: 3420 4
: 3421 4
: 3422 4
: 3423 4
: 3424 4
: 3425 4
: 3426 4
: 3427 4
: 3428 4
: 3429 4
: 3430 4
: 3431 4
: 3432 4
: 3433 4
: 3434 4
: 3435 4
: 3436 4
: 3437 4
: 3438 4
: 3439 4
: 3440 4
: 3441 4
: 3442 4
: 3443 4
: 3444 4
: 3445 4
: 3446 4
: 3447 4
: 3448 4
: 3449 4
: 3450 4
: 3451 4
: 3452 4
: 3453 4
: 3454 4
: 3455 4
: 3456 4
: 3457 4
: 3458 4
: 3459 4
: 3460 4
: 3461 4
: 3462 4
: 3463 4
: 3464 4
: 3465 4
: 3466 4
: 3467 4
: 3468 4
: 3469 4
: 3470 4
: 3471 4
: 3472 4
: 3473 4
: 3474 4
: 3475 4
: 3476 4
: 3477 4
: 3478 4
: 3479 4
: 3480 4
: 3481 4
: 3482 4
: 3483 4
: 3484 4
: 3485 4
: 3486 4
: 3487 4
: 3488 4
: 3489 4
: 3490 4
: 3491 4
: 3492 4
: 3493 4
: 3494 4
: 3495 4
: 3496 4
: 3497 4
: 3498 4
: 3499 4
: 3500 4
: 3501 4
: 3502 4
: 3503 4
: 3504 4
: 3505 4
: 3506 4
: 3507 4
: 3508 4
: 3509 4
: 3510 4
: 3511 4
: 3512 4
: 3513 4
: 3514 4
: 3515 4
: 3516 4
: 3517 4
: 3518 4
: 3519 4
: 3520 4
: 3521 4
: 3522 4
: 3523 4
: 3524 4
: 3525 4
: 3526 4
: 3527 4
: 3528 4
: 3529 4
: 3530 4
: 3531 4
: 3532 4
: 3533 4
: 3534 4
: 3535 4
: 3536 4
: 3537 4
: 3538 4
: 3539 4
: 3540 4
: 3541 4
: 3542 4
: 3543 4
: 3544 4
: 3545 4
: 3546 4
: 3547 4
: 3548 4
: 3549 4
: 3550 4
: 3551 4
: 3552 4
: 3553 4
: 3554 4
: 3555 4
: 3556 4
: 3557 4
: 3558 4
: 3559 4
: 3560 4
: 3561 4
: 3562 4
: 3563 4
: 3564 4
: 3565 4
: 3566 4
: 3567 4
: 3568 4
: 3569 4
: 3570 4
: 3571 4
: 3572 4
: 3573 4
: 3574 4
: 3575 4
: 3576 4
: 3577 4
: 3578 4
: 3579 4
: 3580 4
: 3581 4
: 3582 4
: 3583 4
: 3584 4
: 3585 4
: 3586 4
: 3587 4
: 3588 4
: 3589 4
: 3590 4
: 3591 4
: 3592 4
: 3593 4
: 3594 4
: 3595 4
: 3596 4
: 3597 4
: 3598 4
: 3599 4
: 3600 4
: 3601 4
: 3602 4
: 3603 4
: 3604 4
: 3605 4
: 3606 4
: 3607 4
: 3608 4
: 3609 4
: 3610 4
: 3611 4
: 3612 4
: 3613 4
: 3614 4
: 3615 4
: 3616 4
: 3617 4
: 3618 4
: 3619 4
: 3620 4
: 3621 4
: 3622 4
: 3623 4
: 3624 4
: 3625 4
: 3626 4
: 3627 4
: 3628 4
: 3629 4
: 3630 4
: 3631 4
: 3632 4
: 3633 4
: 3634 4
: 3635 4
: 3636 4
: 3637 4
: 3638 4
: 3639 4
: 3640 4
: 3641 4
: 3642 4
: 3643 4
: 3644 4
: 3645 4
: 3646 4
: 3647 4
: 3648 4
: 3649 4
: 3650 4
: 3651 4
: 3652 4
: 3653 4
: 3654 4
: 3655 4
: 3656 4
: 3657 4
: 3658 4
: 3659 4
: 3660 4
: 3661 4
: 3662 4
: 3663 4
: 3664 4
: 3665 4
: 3666 4
: 3667 4
: 3668 4
: 3669 4
: 3670 4
: 3671 4
: 3672 4
: 3673 4
: 3674 4
: 3675 4
: 3676 4
: 3677 4
: 3678 4
: 3679 4
: 3680 4
: 3681 4
: 3682 4
: 3683 4
: 3684 4
: 3685 4
: 3686 4
: 3687 4
: 3688 4
: 3689 4
: 3690 4
: 3691 4
: 3692 4
: 3693 4
: 3694 4
: 3695 4
: 3696 4
: 3697 4
: 3698 4
: 3699 4
: 3700 4
: 3701 4
: 3702 4
: 3703 4
: 3704 4
: 3705 4
: 3706 4
: 3707 4
: 3708 4
: 3709 4
: 3710 4
: 3711 4
: 3712 4
: 3713 4
: 3714 4
: 3715 4
: 3716 4
: 3717 4
: 3718 4
: 3719 4
: 3720 4
: 3721 4
: 3722 4
: 3723 4
: 3724 4
: 3725 4
: 3726 4
: 3727 4
: 3728 4
: 3729 4
: 3730 4
: 3731 4
: 3732 4
: 3733 4
: 3734 4
: 3735 4
: 3736 4
: 3737 4
: 3738 4
: 3739 4
: 3740 4
: 3741 4
: 3742 4
: 3743 4
: 3744 4
: 3745 4
: 3746 4
: 3747 4
: 3748 4
: 3749 4
: 3750 4
: 3751 4
: 3752 4
: 3753 4
: 3754 4
: 3755 4
: 3756 4
: 3757 4
: 3758 4
: 3759 4
: 3760 4
: 3761 4
: 3762 4
: 3763 4
: 3764 4
: 3765 4
: 3766 4
: 3767 4
: 3768 4
: 3769 4
: 3770 4
: 3771 4
: 3772 4
: 3773 4
: 3774 4
: 3775 4
: 3776 4
: 3777 4
: 3778 4
: 3779 4
: 3780 4
: 3781 4
: 3782 4
: 3783 4
: 3784 4
: 3785 4
: 3786 4
: 3787 4
: 3788 4
: 3789 4
: 3790 4
: 3791 4
: 3792 4
: 3793 4
: 3794 4
: 3795 4
: 3796 4
: 3797 4
: 3798 4
: 3799 4
: 3800 4
: 3801 4
: 3802 4
: 3803 4
: 3804 4
: 3805 4
: 3806 4
: 3807 4
: 3808 4
: 3809 4
: 3810 4
: 3811 4
: 3812 4
: 3813 4
: 3814 4
: 3815 4
: 3816 4
: 3817 4
: 3818 4
: 3819 4
: 3820 4
: 3821 4
: 3822 4
: 3823 4
: 3824 4
: 3825 4
: 3826 4
: 3827 4
: 3828 4
: 3829 4
: 3830 4
: 3831 4
: 3832 4
: 3833 4
: 3834 4
: 3835 4
: 3836 4
: 3837 4
: 3838 4
: 3839 4
: 3840 4
: 3841 4
: 3842 4
: 3843 4
: 3844 4
: 3845 4
: 3846 4
: 3847 4
: 3848 4
: 3849 4
: 3850 4
: 3851 4
: 3852 4
: 3853 4
: 3854 4
: 3855 4
: 3856 4
: 3857 4
: 3858 4
: 3859 4
: 3860 4
: 3861 4
: 3862 4
: 3863 4
: 3864 4
: 3865 4
: 3866 4
: 3867 4
: 3868 4
: 3869 4
: 3870 4
: 3871 4
: 3872 4
: 3873 4
: 3874 4
: 3875 4
: 3876 4
: 3877 4
: 3878 4
: 3879 4
: 3880 4
: 3881 4
: 3882 4
: 3883 4
: 3884 4
: 3885 4
: 3886 4
: 3887 4
: 3888 4
: 3889 4
: 3890 4
: 3891 4
: 3892 4
: 3893 4
: 3894 4
: 3895 4
: 3896 4
: 3897 4
: 3898 4
: 3899 4
: 3900 4
: 3901 4
: 3902 4
: 3903 4
: 3904 4
: 3905 4
: 3906 4
: 3907 4
: 3908 4
: 3909 4
: 3910 4
: 3911 4
: 3912 4
: 3913 4
: 3914 4
: 3915 4
: 3916 4
: 3917 4
: 3918 4
: 3919 4
: 3920 4
: 3921 4
: 3922 4
: 3923 4
: 3924 4
: 3925 4
: 3926 4
: 3927 4
: 3928 4
: 3929 4
: 3930 4
: 3931 4
: 3932 4
: 3933 4
: 3934 4
: 3935 4
: 3936 4
: 3937 4
: 3938 4
: 3939 4
: 3940 4
: 3941 4
: 3942 4
: 3943 4
: 3944 4
: 3945 4
: 3946 4
: 3947 4
: 3948 4
: 3949 4
: 3950 4
: 3951 4
: 3952 4
: 3953 4
: 3954 4
: 3955 4
: 3956 4
: 3957 4
: 3958 4
: 3959 4
: 3960 4
: 3961 4
: 3962 4
: 3963 4
: 3964 4
: 3965 4
: 3966 4
: 3967 4
: 3968 4
: 3969 4
: 3970 4
: 3971 4
: 3972 4
: 3973 4
: 3974 4
: 3975 4
: 3976 4
: 3977 4
: 3978 4
: 3979 4
: 3980 4
: 3981 4
: 3982 4
: 3983 4
: 3984 4
: 3985 4
: 3986 4
: 3987 4
: 3988 4
: 3989 4
: 3990 4
: 3991 4
: 3992 4
: 3993 4
: 3994 4
: 3995 4
: 3996 4
: 3997 4
: 3998 4
: 3999 4
: 4000 4
: 4001 4
: 4002 4
: 4003 4
: 4004 4
: 4005 4
: 4006 4
: 4007 4
: 4008 4
: 4009 4
: 4010 4
: 4011 4
: 4012 4
: 4013 4
: 4014 4
: 4015 4
: 4016 4
: 4017 4
: 4018 4
: 4019 4
: 4020 4
: 4021 4
: 4022 4
: 4023 4
: 4024 4
: 4025 4
: 4026 4
: 4027 4
: 4028 4
: 4029 4
: 4030 4
: 4031 4
: 4032 4
: 4033 4
: 4034 4
: 4035 4
: 4036 4
: 4037 4
: 4038 4
: 4039 4
: 4040 4
: 4041 4
: 4042 4
: 4043 4
: 4044 4
: 4045 4
: 4046 4
: 4047 4
: 4048 4
: 4049 4
: 4050 4
: 4051 4
: 4052 4
: 4053 4
: 4054 4
: 4055 4
: 4056 4
: 4057 4
: 4058 4
: 4059 4
: 4060 4
: 4061 4
: 4062 4
: 4063 4
: 4064 4
: 4065 4
: 4066 4
: 4067 4
: 4068 4
: 4069 4
: 4070 4
: 4071 4
: 4072 4
: 4073 4
: 4074 4
: 4075 4
: 4076 4
: 4077 4
: 4078 4
: 4079 4
: 4080 4
: 4081 4
: 4082 4
: 4083 4
: 4084 4
: 4085 4
: 4086 4
: 4087 4
: 4088 4
: 4089 4
: 4090 4
: 4091 4
: 4092 4
: 4093 4
: 4094 4
: 4095 4
: 4096 4
: 4097 4
: 4098 4
: 4099 4
: 4100 4
: 4101 4
: 4102 4
: 4103 4
: 4104 4
: 4105 4
: 4106 4
: 4107 4
: 4108 4
: 4109 4
: 4110 4
: 4111 4
: 4112 4
: 4113 4
: 4114 4
: 4115 4
: 4116 4
: 4117 4
: 4118 4
: 4119 4
: 4120 4
: 4121 4
: 4122 4
: 4123 4
: 4124 4
: 4125 4
: 4126 4
: 4127 4
: 4128 4
: 4129 4
: 4130 4
: 4131 4
: 4132 4
: 4133 4
: 4134 4
: 4135 4
: 4136 4
: 4137 4
: 4138 4
: 4139 4
: 4140 4
: 4141 4
: 4142 4
: 4143 4
: 4144 4
: 4145 4
: 4146 4
: 4147 4
: 4148 4
: 4149 4
: 4150 4
: 4151 4
: 4152 4
: 4153 4
: 4154 4
: 4155 4
: 4156 4
: 4157 4
: 4158 4
: 4159 4
: 4160 4
: 4161 4
: 4162 4
: 4163 4
: 4164 4
: 4165 4
: 4166 4
: 4167 4
: 4168 4
: 4169 4
: 4170 4
: 4171 4
: 4172 4
: 4173 4
: 4174 4
: 4175 4
: 4176 4
: 4177 4
: 4178 4
: 4179 4
: 4180 4
: 4181 4
: 4182 4
: 4183 4
: 4184 4
: 4185 4
: 4186 4
: 4187 4
: 4188 4
: 4189 4
: 4190 4
: 4191 4
: 4192 4
: 4193 4
: 4194 4
: 4195 4
: 4196 4
: 4197 4
: 4198 4
: 4199 4
: 4200 4
: 4201 4
: 4202 4
: 4203 4
: 4204 4
: 4205 4
: 4206 4
: 4207 4
: 4208 4
: 4209 4
: 4210 4
: 4211 4
: 4212 4
: 4213 4
: 4214 4
: 4215 4
: 4216 4
: 4217 4
: 4218 4
: 4219 4
: 4220 4
: 4221 4
: 4222 4
: 4223 4
: 4224 4
: 4225 4
: 4226 4
: 4227 4
: 4228 4
: 4229 4
: 4230 4
: 4231 4
: 4232 4
: 4233 4
: 4234 4
: 4235 4
: 4236 4
: 4237 4
: 4238 4
: 4239 4
: 4240 4
: 4241 4
: 4242 4
: 4243 4
: 4244 4
: 4245 4
: 4246 4
: 4247 4
: 4248 4
: 4249 4
: 4250 4
: 4251 4
: 4252 4
: 4253 4
: 4254 4
: 4255 4
: 4256 4
: 4257 4
: 4258 4
: 4259 4
: 4260 4
: 4261 4
: 4262 4
: 4263 4
: 4264 4
: 4265 4
: 426
```

N7

ZRQAM3
VOL.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0298
Page 67
(16)

```

; 3025 4      if QIO_OK ( )
; 3026 4      then
; 3027 4          SEND_COMMANDS;
; 3028 5          begin
; 3029 5              QIO_GEN ( );
; 3030 5
; 3031 5              if ( .MX1 geq 0 ) and
; 3032 6                  ( not .EOP_FLAG )
; 3033 5              then
; 3034 5
; 3035 5                  if SEND ( .MX1 ) eq 1 SUCCESS
; 3036 5                  then
; 3037 5                      QIO [ .CTLR ] = .QIO [ .CTLR ] + 1
; 3038 5                  else
; 3039 6                      begin
; 3040 6                          PUT_PKT ( .MX1 );
; 3041 6                          leave SEND_COMMANDS;
; 3042 5                      end;
; 3043 5
; 3044 5              if ( .MX2 geq 0 ) and
; 3045 6                  ( not .EOP_FLAG )
; 3046 5              then
; 3047 6                  begin
; 3048 6
; 3049 6                      do
; 3050 6                          BREAK
; 3051 6                      until ( .DCT_ADDR [ CRING_CNT ] lssu CRING_LEN );
; 3052 6
; 3053 6                      if SEND ( .MX2 ) eq 1 SUCCESS
; 3054 6                      then
; 3055 6                          QIO [ .CTLR ] = .QIO [ .CTLR ] + 1
; 3056 6                      else
; 3057 7                          begin
; 3058 7                              PRINTF ( DBM121, .CRN HIGH, .CRN LOW );
; 3059 7                              COMPARE_DATA = FALSE;
; 3060 7                              PUT_PKT ( .MX2 );
; 3061 6                              end;
; 3062 5                          end;
; 3063 5
; 3064 4                  end;
; 3065 3              end;
; 3066 3
; 3067 3          BREAK;
; 3068 3          PROC_RETPKT ( );
; 3069 3
; 3070 3      end
; 3071 3      until ((not QIO_OUT ( )) or
; 3072 4          ((.DCT_ADDR [ CRING_CNT ] eq 0) and
; 3073 2              (.EOP_FLAG)));
; 3074 2
; 3075 2      DCT_ADDR [ IG_INT ] = TRUE;
; 3076 2
; 3077 1      end;

```

! IF O.K. TO ISSUE QIO(S) TO CONTROLLER

! GENERATE 1 OR 2 QIOs

! IF SUCCESS ON FIRST QIO

! ATTEMPT TO SEND IT. IF SUCCESS

! INCR OUTSTANDING QIO COUNT

! RETURN PACKET TO POOL

! IF SUCCESS ON SECOND QIO

! WAIT TILL 1 MORE SLOT AVAILABLE IN CRING

! ATTEMPT TO SEND IT.

! IF SUCCESS, INCR OUTSTANDING QIO COUNT

! NO SENSE IN COMPARING WRITE DATA

! RETURN PACKET TO POOL

! O.K. TO ISSUE QIO(S)

! CONTROLLER LOOP

! LET SUPERVISOR CATCH USER REQUESTS

! PROCESS AND RETURN PACKETS

! EXECUTIVE PROCESSING LOOP

! NO FURTHER INTERRUPTS ON THIS CONTROLLER

! EXERCISER

ZRQAM5
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11-16 V4.0-579
DISK#USER2:[POWERS]ZRQADO,BL2;6

SEQ 0299
Page 68
(16)

		.SBTTL MULTI-DRIVE MULTI-DRIVE TEST ROUTINES		
000000	004137	000000G	MULTI-DRIVE::	
000004	005746		JSR	R1,SAVE3
000006	004737	000000V	YST	(SP)
000012	112737	000001 000000G	JSR	PC,MD,INIT
000020	005001		MOVB	#1,INIT.OCCURED
000022	010146		14: CLR	R1
000024	004737	000000G	24: MOV	R1,(SP)
000030	104440		JSR	PC,SET,CPAR
000032	010003		TRAP	40
000034	012700	000200	MOV	R0,R3
000040	104441		MOV	#200,R0
000042	013737	000000G 000104'	TRAP	41
000050	013737	000000G 000076'	MOV	CCTLR,ICTLR
000056	013737	000000G 000100'	MOV	CST.ADDR,ICST.ADDR
000064	017737	000076' 000000G	MOV	DCT.ADDR,IDCT.ADDR
000072	013700	000100'	MOV	WICST.ADDR,IRDRX.ADDR
000076	013702	000000G	MOV	IDCT.ADDR,R0
000102	016266	000002 000002	MOV	IRDRX.ADDR,R2
000110	016660	000002 000002	MOV	2(R2),2(SP)
000116	016600	000002	MOV	2(SP),2(R0)
000122	042700	077777	MOV	2(SP),R0
000126	020027	100000	BIC	#77777,R0
000132	001006		CMF	R0,#-100000
000134	004737	000000V	BNE	31
000140	010300		JSR	PC,FATAL,ERROR
000142	104441		MOV	R3,R0
000144	005726		TRAP	41
000146	000511		IST	(SP),
000150	010300		BR	91
000152	104441		31: MOV	R3,R0
000154	004737	000000V	TRAP	41
000160	006000		JSR	PC,QIO,OK
000162	103077		ROR	R0
000164	004737	000000V	BCC	81
000170	013700	000106'	JSR	PC,QIO,GEN
000174	002422		MOV	MX1,R0
000176	132737	000001 000000G	BLT	51
000204	001016		BITB	#1,EOP,FLAG
000206	010016		BNE	51
000210	004737	000000G	MOV	R0,(SP)
000214	020027	000001	JSR	PC,SEND
000220	001003		CMF	R0,#1
000222	105261	000000G	BNE	41
000226	000405		INCB	QIO(R1)
000230	013716	000106'	BR	51
000234	004737	000000G	41: MOV	MX1,(SP)
000240	000450		JSR	PC,PUT,PKT
000242	005737	000110'	BR	81
000246	002445		51: TST	MX2
			BLT	81

C8

ZRQAM5 V01.6	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES	11-Apr-1984 11:08:35 11-Apr-1984 11:08:22	VAX-11 B11-16 V4.0-579 DISK:USER2:[POWERS]ZRQAD0,BL2,6	SEQ 0300 Page 69 (16)
000250 132737 000001 000000G		BITB 01,EOP,FLAG	:	3045
000256 001041		BNE 81	:	
000260 104422	61:	TRAP 22	:	3049
000262 127727 000000G 000004		CMPB 80CT,ADDR,04	:	3051
000270 103373		BHIS 61	:	
000272 013716 000110'		MOV MX2,(SP)	:	3053
000276 004737 000000G		JSR PC,SEND	:	
000302 020027 000001		CMP R0,01	:	
000306 001003		BNE 71	:	
000310 105261 000000G		INCB QIO(R1)	: *(CTLR)	3055
000314 000422		BR 81	:	3057
000316 013716 000000G	71:	MOV CRN,LOW,(SP)	:	3058
000322 013746 000000G		MOV CRN,HIGH,-(SP)	:	
000326 012746 000000G		MOV 0DBM121,-(SP)	:	
000332 012746 000003		MOV 03,-(SP)	:	
000336 010600		MOV SP,R0	: SP,*	
000340 104417		TRAP 17	:	
000342 105037 001170'		CLRB COMPARE,DATA	:	3059
000346 013716 000110'		MOV MX2,(SP)	:	3060
000352 004737 000000G		JSR PC,PUT,PKT	:	
000356 062706 000006		ADD 06,SP	:	3057
000362 005726	81:	TST (SP),	:	3005
000364 005201		INC R1	: CTLR	3004
000366 000243		.WORD CLV:CLC	:	
000370 003614		BLF 21	:	
000372 104422	91:	TRAP 22	:	3045
000374 004737 000000V		JSR PC,PROC,RETPKT	:	3068
000400 004737 000000V		JSR PC,QIO,OUT	:	3071
000404 006000		ROR R0	:	
000406 103011		BCC 101	:	
000410 105777 000000G		TSTB 80CT,ADDR	:	3072
000414 001201		BNE 11	:	
000416 132737 000001 000000G		BITB 01,EOP,FLAG	:	3073
000424 001002		BNE 101	:	
000426 000137 007152'		JMP 11	:	
000432 052777 040000 000000G	101:	BIS 040000,80CT,ADDR	:	3075
000440 005726		TST (SP),	:	2974
000442 000207		RTS PC	:	

; Routine Size: 146 words, Routine Base: 1CODE1 + 7132
; Maximum stack depth per invocation: 11 words

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0301
Page 70
(17)

```

: 3078 1 GLOBAL routine MD_INIT : novalue *
: 3079 1
: 3080 1
: 3081 1 !
: 3082 1 ! THIS ROUTINE IS CALLED BY ROUTINE MULTI_DRIVE TO INITIALIZE DATA ITEMS
: 3083 1 ! USED BY THE MULTI-DRIVE TEST.
: 3084 1 !
: 3085 2 begin
: 3086 2
: 3087 2 !!ZZZ local
: 3088 2 !!ZZZ AVG_XFER_SIZE : word, ! SIZE (BYTES) OF AN AVERAGE I/O XFER
: 3089 2 !!ZZZ QUICK_PASS_CNT : word; ! AVG NO. OF I/O OPERATIONS IN A QUICK PASS
: 3090 2
: 3091 2 if not .INIT_OCCURED ! IF THIS IS A START
: 3092 2 then ! PARTITION FREE MEMORY INTO I/O BUFFERS
: 3093 2 INIT_IO_BUFF ();
: 3094 2
: 3095 2 if (.ENTRY_REASON neq CONT) and ! IF START, RESTART, OR PWR FAIL
: 3096 3 (.ENTRY_REASON neq NEW_PASS)
: 3097 2 then
: 3098 2
: 3099 2 incr CTLR from 0 to (MAX_CTLR - 1) do
: 3100 3 begin
: 3101 3 SET_CPAR (.CTLR);
: 3102 3
: 3103 4 INCR DISK FROM (0 * OF_UN) TO (3 * UNIT_SIZE) !ZZZ !ZZZ
: 3104 3 * OF_UN) BY UNIT_SIZE DO !ZZZ
: 3105 4 BEGIN !ZZZ
: 3106 4 SET_UPAR (.DISK); !ZZZ
: 3107 4 DPST [.L$LUN] = DP_CNT; !INIT DATA PTRN SEQ TABLE !ZZZ
: 3108 3 END; !ZZZ
: 3109 3
: 3110 2 END; !ZZZ
: 3111 2 INCR COUNT FROM 0 TO (QIO_PER_CTLR * MAX_CTLR - 1) DO !INIT !ZZZ
: 3112 2 BUFF_OWN [.COUNT] = -1; !I/O BUFF ALLOC TABLE !ZZZ
: 3113 1 END; !END MD_INIT !ZZZ

```

			.SBTTL	MD_INIT MULTI-DRIVE TEST ROUTINES	
000000	004137	000000G	MD_INIT:		
000004	132737	000001 000000G	JSR	R1,\$SAVE2	3078
000012	001002		BITB	#1,INIT_OCCURED	3091
000014	004737	000000V	BNE	1\$	
000020	123727	000000G 000003	JSR	PC,INIT_IO_BUFF	3093
000026	001433		CMPB	ENTRY_REASON,#3	3095
000030	123727	000000G 000005	BEQ	4\$	
000036	001427		CMPB	ENTRY_REASON,#5	3096
000040	005002		BEQ	4\$	
000042	010246		CLR	R2	3099
000044	004737	000000G	MOV	R2, -(SP)	3101
000050	012701	000003	JSR	PC,SET_CPAR	
000054	010116		MOV	#3,R1	3103
			MOV	R1,(SP)	3106

E8

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0302
Page 71
(17)

000056	004737	000000G		JSR	PC,SET,UPAR		
000062	013700	000000G		MOV	L\$LUN,R0		3107
000066	112760	000025	000050'	MOVB	#25,DPST(R0)		
000074	062701	000012		ADD	#12,R1	;*,DISK	3103
000100	020127	000041		CMP	R1,#41	;DISK,*	
000104	003763			BLE	31		
000106	005726			TST	(SP)+		3100
000110	005202			INC	R2	;CTLR	3099
000112	000243				.WORD CLV!CLC		
000114	003752			BLE	21		
000116	005000			CLR	R0	;COUNT	3111
000120	112760	000377	000000G	4\$:	MOVB	#377,BUFF.OWN(R0)	;*,*(COUNT)
000126	005200			5\$:	INC	R0	;COUNT
000130	020027	000007			CMP	R0,#7	;COUNT,*
000134	003771				BLE	51	
000136	000207				RTS	PC	3078

; Routine Size: 48 words, Routine Base: \$CODE\$ + 7576
; Maximum stack depth per invocation: 5 words

; 3114 1

F8

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0303
Page 72
(18)

```

: 3115 1 GLOBAL routine INIT_IO_BUFF : novalue =
: 3116 1
: 3117 1 !+
: 3118 1 ! THIS ROUTINE IS CALLED BY MD_INIT WHEN THE MULTI-DRIVE TEST IS FIRST
: 3119 1 ! STARTED. IT IS RESPONSIBLE FOR PARTITIONING FREE MEMORY INTO A
: 3120 1 ! COLLECTION OF I/O BUFFERS. THE SIZE OF EACH I/O BUFFER IS DETERMINED
: 3121 1 ! BY A NUMBER OF FACTORS, INCLUDING THE NUMBER OF UNITS, THE NUMBER OF
: 3122 1 ! CONTROLLERS, AND THE SIZE OF FREE MEMORY.
: 3123 1 !
: 3124 1 ! ONCE THE BUFFER SIZE IS DETERMINED, THE NUMBER OF I/O BUFFERS IS
: 3125 1 ! CALCULATED. FINALLY, THE BUFFER ADDRESS (BUFF_ADDR) TABLE IS LOADED
: 3126 1 ! WITH FIXED BUFFER DESCRIPTORS THAT ARE USED IN THE ALLOCATION AND
: 3127 1 ! DEALLOCATION PROCESS.
: 3128 1 !
: 3129 1 ! IMPLICIT INPUTS:
: 3130 1 ! CTLR_CNT - THE NUMBER OF CONTROLLERS CONFIGURED
: 3131 1 ! $UNIT - THE NUMBER OF UNITS AVAILABLE FOR TESTING
: 3132 1 ! FREE_MEM_ADDR - START OF FREE MEMORY
: 3133 1 !-
: 3134 1
: 3135 2 begin
: 3136 2 BUFF_ADDR [0] = (.FREE_MEM_ADDR + 2 + 1) and #o'177776'; ! START OF READ/WRITE BUFFERS
: 3137 2
: 3138 2 while (.BUFF_ADDR [0] and #o'37') neq 0 do ! FORCE FIRST I/O BUFFER TO START
: 3139 2     BUFF_ADDR [0] = .BUFF_ADDR [0] + 2; ! ON EVEN BOUNDARY
: 3140 2
: 3141 2 BYTS_PER_QIO = ((.DRS_START - .BUFF_ADDR [0]) / (QIO_PER_CTLR * MAX_CTLR)) and #o'177740'; ! MAX TRANSFER SIZE
: 3142 2
: 3143 2
: 3144 2 if .BYTS_PER_QIO gtru MAX_XFER_SIZE
: 3145 2 then
: 3146 2     BYTS_PER_QIO = MAX_XFER_SIZE; ! ADJUST TRANSFER SIZE LOWER
: 3147 2
: 3148 2 if .BYTS_PER_QIO lesu 32
: 3149 2 then
: 3150 2     begin
: 3151 3     ERRSF (2, EGS_02, 0); ! ERROR IF NOT ENOUGH MEMORY
: 3152 3     DOCLN;
: 3153 2     end;
: 3154 2
: 3155 2 if (QIO_PER_CTLR * MAX_CTLR) gtru 1
: 3156 2 then
: 3157 2
: 3158 2     incr index from 1 to (QIO_PER_CTLR * MAX_CTLR - 1) do ! INIT REMAINING TABLE ENTRIES
: 3159 2     BUFF_ADDR [.index] = .BUFF_ADDR [.index - 1] + .BYTS_PER_QIO; ! FIXED BUFFER ADDRESS
: 3160 2
: 3161 1 end; ! ROUTINE INIT_IO_BUFF

```

```

000000 004137 000000G          .SBTTL  INIT_IO_BUFF MULTI-DRIVE TEST ROUTINES
                                INIT_IO_BUFF::
000004 013700 000000G          JSR    R1,$SAVE3
                                MOV     FREE_MEM_ADDR,R0

```

3115
3136

G8

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0304
Page 73
(18)

000010	062700	000003		ADD	#3,R0		
000014	010037	000000G		MOV	R0,BUFF,ADDR		
000020	042737	000001	000000G	BIC	#1,BUFF,ADDR		
000026	032737	000037	000000G	1\$: BIT	#37,BUFF,ADDR	:	3138
000034	001404			BEQ	2\$		
000036	062737	000002	000000G	ADD	#2,BUFF,ADDR	:	3139
000044	000770			BR	1\$	:	3138
000046	013746	001160		2\$: MOV	DRS.START,-(SP)	:	3141
000052	163716	000000G		SUB	BUFF,ADDR,(SP)		
000056	012746	000010		MOV	#10,-(SP)		
000062	004737	000000G		JSR	PC,BL\$DIV		
000066	010037	000000G		MOV	R0,BYTS.PER.QIO		
000072	042737	000037	000000G	BIC	#37,BYTS.PER.QIO		
000100	023727	000000G	001400	CMP	BYTS.PER.QIO,#1400	:	3144
000106	101403			BLOS	3\$		
000110	012737	001400	000000G	MOV	#1400,BYTS.PER.QIO	:	3146
000116	023727	000000G	000040	3\$: CMP	BYTS.PER.QIO,#40	:	3148
000124	103005			BHIS	4\$		
000126	104454			TRAP	54	:	3151
000130	000002			.WORD	2		
000132	000000G			.WORD	EGS.02		
000134	000000			.WORD	0		
000136	104444			TRAP	44		
000140	012702	000001		4\$: MOV	#1,R2	: *,INDEX	3155
000144	010200			5\$: MOV	R2,R0	: INDEX,*	3159
000146	006300			ASL	RC		
000150	010201			MOV	R2,R1	: INDEX,*	
000152	006301			ASL	R1		
000154	016103	177776G		MOV	BUFF,ADDR-2(R1),R3		
000160	063703	000000G		ADD	BYTS.PER.QIO,R3		
000164	010360	000000G		MOV	R3,BUFF,ADDR(R0)		
000170	005202			INC	R2	: INDEX	3155
000172	020227	000007		CMP	R2,#7	: INDEX,*	
000176	003762			BLE	5\$		
000200	022626			CMP	(SP)+,(SP)+	:	3135
000202	000207			RTS	PC	:	3115

; Routine Size: 66 words, Routine Base: \$CODE\$ + 7736
; Maximum stack depth per invocation: 8 words

H8

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6

SEQ 0305
Page 74
(19)

```

: 3162 1 GLOBAL routine QIO_OK =
: 3163 1
: 3164 1
: 3165 1
: 3166 1
: 3167 1
: 3168 1
: 3169 1
: 3170 1
: 3171 1
: 3172 1
: 3173 1
: 3174 1
: 3175 1
: 3176 1
: 3177 1
: 3178 1
: 3179 1
: 3180 1
: 3181 1
: 3182 1
: 3183 1
: 3184 1
: 3185 1
: 3186 1
: 3187 2
: 3188 2
: 3189 1
: 3190 1
: 3191 1
: 3192 1
: 3193 1

THIS ROUTINE IS CALLED BY THE MULTI_DRIVE "EXECUTIVE" IN ORDER TO
DETERMINE WHETHER OR NOT A QIO REQUEST (OR QIO PAIR) SHOULD BE
GENERATED TO THE CURRENT CONTROLLER. A VALUE OF "TRUE" IS RETURNED IF
THE CONTROLLER MEETS 3 REQUIREMENTS:

    A. THE CONTROLLER IS ONLINE;
    B. THE NUMBER OF OUTSTANDING QIOs IS AT LEAST 2 LESS THAN THE
        MAXIMUM ALLOWED FOR ANY ONE CONTROLLER;
    C. THERE IS AT LEAST ONE DISK ONLINE TO THE CONTROLLER.

IF ANY OF THESE TEST FAIL, THEN A VALUE OF "FALSE" IS RETURNED.

IMPL. IT INPUTS:
    CCTLR - CURRENT CONTROLLER NUMBER
    CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST

if (.CST_ADDR [STATE] eq1 ONLINE) and
(not .EOP_FLAG) and
((.QIO [.CCTLR] + 2) leq1 QIO_PER_CTLR) and
(.CST_ADDR [U_CNT] neq 0)
then
    return TRUE
else
    return FALSE;

! IF CONTROLLER IS ONLINE
! IF OUTSTANDING QIO COUNT IS O.K.
! IF THERE IS VALID UNIT
! "TRUE" EXIT POINT
! "FALSE" EXIT POINT

```

000000	013700	000000G	QIO.OK: SBTTL	QIO.OK MULTI-DRIVE TEST ROUTINES	
000004	005760	000002	MOV	CST.ADDR,R0	3184
000010	100027		TST	2(R0)	
000012	132737	000001 000000G	BPL	1\$	
000020	001023		BITB	#1,EOP.FLAG	3185
000022	013700	000000G	BNE	1\$	
000026	116000	000000G	MOV	CCTLR,R0	3186
000032	042700	177400	MOVB	QIO(R0),R0	
000036	062700	000002	BIC	#177400,R0	
000042	020027	000010	ADD	#2,R0	
000046	101010		CMP	R0,#10	
000050	013700	000000G	BHI	1\$	
000054	105760	000005	MOV	CST.ADDR,R0	3187
000060	001403		TSTB	5(R0)	
000062	012700	000001	BEQ	1\$	
000066	000207		MOV	#1,R0	3193
000070	005000		RTS	PC	
000072	000207		CLR	R0	
			RTS	PC	3162

ZRQAM3 RD/RX EXERCISER
V01.6 MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0306
Page 75
(19)

; Routine Size: 30 words, Routine Base: \$CODE\$ + 10142
; Maximum stack depth per invocation: 0 words

; 3194 1

ZRGAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0307
Page 76
(20)

```

: 3195 1 GLOBAL routine QIO_OUT =
: 3196 1
: 3197 1 !
: 3198 1 ! THIS ROUTINE IS CALLED BY THE MULTI_DRIVE EXECUTIVE FOR DETERMNING THE
: 3199 1 ! END OF THE MULTI-DRIVE TEST, ITS PURPOSE IS TO EXAMINE THE QIO VECTOR
: 3200 1 ! FOR ANY OUTSTANDING QIOs ON ANY CONTROLLER. A VALUE OF "TRUE" IS
: 3201 1 ! RETURNED IF THERE IS AT LEAST ONE QIO OUTSTANDING ON ANY CONTROLLER.
: 3202 1 ! OTHERWISE, "FALSE" IS RETURNED INDICATING NO OUTSTANDING QIOs.
: 3203 1 !
: 3204 1
: 3205 2 begin
: 3206 2
: 3207 2 incr CTLR from 0 to (MAX_CTLR - 1) do
: 3208 3 begin
: 3209 3 SET_CPAR (.CTLR); ! SET UP CURRENT CONTROLLER PARAMETERS
: 3210 3
: 3211 3 IF .CST_ADDR [STATE] eq1 ONLINE ! IF CONTROLLER IS ONLINE
: 3212 3 then
: 3213 3 return TRUE;
: 3214 3
: 3215 2 end;
: 3216 2
: 3217 2 return FALSE; ! EXIT - NO CONTROLLERS ONLINE
: 3218 1 end;

```

		.SBTTL QIO.OUT MULTI-DRIVE TEST ROUTINES		
000000	010146	QIO.OUT::		
000002	005001	MOV R1, -(SP)	:	3195
000004	010146	CLR R1	:	3207
000006	004737	1\$: MOV R1, -(SP)	:	3209
000012	013700	JSR PC, SET_CPAR	:	
000016	005760	MOV CST_ADDR, R0	:	3211
000022	100004	TST 2(R0)	:	
000024	005726	BPL 2\$	:	
000026	012700	TST (SP)+	:	3213
000032	000405	MOV #1, R0	:	
000034	005726	BR 3\$	:	
000036	005201	2\$: TST (SP)+	:	3208
000040	000243	INC R1	:	3207
		.WORD CLV!CLC	:	
000042	003760	BLE 1\$	:	
000044	005000	CLR R0	:	3205
000046	012601	3\$: MOV (SP)+, R1	:	3195
000050	000207	RTS PC	:	

```

; Routine Size: 21 words, Routine Base: $CODE$ + 10236
; Maximum stack depth per invocation: 3 words

```

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0308
Page 77
(21)

```

: 3219 1 GLOBAL routine QIO_GEN : novalue =
: 3220 1
: 3221 1 !
: 3222 1 ! THIS ROUTINE IS CALLED BY THE MULTI_DRIVE EXECUTIVE FOR AN ONLINE
: 3223 1 ! CONTROLLER ELIGIBLE TO RECEIVE I/O TRANSFER REQUESTS. IT IS
: 3224 1 ! RESPONSIBLE FOR SECURING ONE OR TWO MSCP PACKETS AND LOADING THEM
: 3225 1 ! WITH VARIOUS PARAMETERS COMPRISING THE I/O REQUEST. THE I/O REQUEST
: 3226 1 ! GENERATED HERE IS DESTINED TO A PARTICULAR UNIT SELECTED AT RANDOM FROM
: 3227 1 ! THOSE CONFIGURED UNDER THE CURRENT CONTROLLER.
: 3228 1 !
: 3229 1 ! EACH FIELD OF THE PACKET(S) IS LOADED WITHIN INDIVIDUAL ROUTINES
: 3230 1 ! (QIO_FUNC, QIO_LBN, QIO_SIZE, ETC.). MOST OF THE VALUES SELECTED FOR
: 3231 1 ! EACH FIELD ARE BASED ON A SET OF RANDOM NUMBER GENERATED AT THE START.
: 3232 1 !
: 3233 1 ! UNDER NORMAL CIRCUMSTANCES, ONLY ONE I/O REQUEST IS GENERATED. HOWEVER,
: 3234 1 ! IF THIS I/O REQUEST IS A "WRITE", AND IF THE OPERATOR SELECTED THE
: 3235 1 ! OPTION FOR HOST WRITE-COMPARES, THEN A SECOND "READ" REQUEST WILL BE
: 3236 1 ! GENERATED WITH THE SAME LBN AND BYTE COUNT.
: 3237 1 !
: 3238 1 ! AFTER THE PACKET(S) HAVE BEEN LOADED, THIS ROUTINE REGAINS CONTROL
: 3239 1 ! AND ATTEMPTS TO GET ONE OR TWO I/O BUFFERS FOR THE ACTUAL DATA
: 3240 1 ! TRANSFERS. THE SUCCESS / FAIL STATUS OF THIS ENTIRE OPERATION IS
: 3241 1 ! PASSED BACK TO THE CALLER THROUGH THE GLOBALS "MX1" AND "MX2"; THEY
: 3242 1 ! CONTAIN VALID MSCP PACKET INDECES, OR -1.
: 3243 1 !
: 3244 1 ! IMPLICIT INPUTS:
: 3245 1 ! CCTLR - CURRENT CONTROLLER NUMBER
: 3246 1 !
: 3247 1 !
: 3248 2 begin
: 3249 2 MX2 = -1; ! ASSUME FAILURE IN SECURING 2ND PACKET
: 3250 2
: 3251 2 if (MX1 = GET_PKT (.CCTLR)) lss 0 ! TRY TO GET 1ST PACKET. IF FAILURE
: 3252 2 then
: 3253 2 return; ! NO POINT IN CONTINUING
: 3254 2
: 3255 2 if (MX2 = GET_PKT (.CCTLR)) lss 0 ! TRY TO GET 2ND PACKET. IF FAILURE
: 3256 2 then
: 3257 3 begin
: 3258 3 PUT_PKT (.MX1); ! RETURN 1ST PACKET TO POOL
: 3259 3 MX1 = -1; ! INDICATE FAILURE
: 3260 3 return; ! DONE
: 3261 2 end;
: 3262 2
: 3263 2 MAD1 = MSCP_PKT + (.MX1 * PKT_LEN * 2); ! CALCULATE STARTING ADDRESSES
: 3264 2 MAD2 = MSCP_PKT + (.MX2 * PKT_LEN * 2); ! OF 30TH PACKETS
: 3265 2 GET_RANDOM (); ! GENERATE A SET OF RANDOM NUMBERS
: 3266 2 QIO_UNIT (); ! LOAD RANDOM UNIT NUMBER INTO PACKETS
: 3267 2
: 3268 2 if .EOP_FLAG
: 3269 2 then
: 3270 2 return;
: 3271 2

```

L8

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35 VAX-11 B1199-16 V4.0-579
11-Apr-1984 11:08:22 DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0309
Page 78
(21)

```

: 3272 2      QIO_FUNC ();          ! LOAD RANDOM FUNCTION CODE (OPCODE)
: 3273 2      QIO_LBN ();           ! LOAD LBN (RANDOM OR SEQUENTIAL)
: 3274 2      QIO_SIZE ();          ! LOAD RANDOM BYTE COUNT
: 3275 2      GET_IO_BUFF (MAD1 [BUF_0]); ! TRY TO GET AN I/O BUFFER
: 3276 2
: 3277 2      if .MX2 geq 0          ! IF TWO QIOs ARE TO BE ISSUED
: 3278 2      then
: 3279 3          begin
: 3280 3              GET_IO_BUFF (MAD2 [BUF_0]); ! TRY TO GET 2ND I/O BUFFER
: 3281 3
: 3282 3              if .MAD2 [BUF_0] eqa 0      ! IF 2ND BUFFER ALLOCATION FAILED
: 3283 3              then
: 3284 4                  begin
: 3285 4
: 3286 4                  if .MAD1 [BUF_0] neqa 0 ! IF 1ST I/O BUFFER WAS ALLOCATED
: 3287 4                  then
: 3288 5                      begin
: 3289 5                          PUT_IO_BUFF (MAD1 [BUF_0]); ! RETURN 1ST I/O BUFFER TO POOL
: 3290 5                          MAD1 [BUF_0] = 0; ! MARK IT AS FAILED
: 3291 4                      end;
: 3292 4
: 3293 4                          PUT_PKT (.MX2); ! RETURN 2ND PACKET TO POOL
: 3294 4                          MX2 = -1; ! INDICATE FAILURE
: 3295 3                          end; ! IF 2ND I/O BUFFER ALLOCATION FAILED
: 3296 3
: 3297 2          end; ! IF TWO QIOs ARE TO BE ISSUED
: 3298 2
: 3299 2      if .MAD1 [BUF_0] eqa 0      ! IF 1ST I/O BUFFER ALLOCATION FAILED
: 3300 2      then
: 3301 3          begin
: 3302 3              PUT_PKT (.MX1); ! RETURN 1ST PACKET TO POOL
: 3303 3              M1 = -1; ! INDICATE FAILURE
: 3304 3          end
: 3305 2      else
: 3306 2
: 3307 2          if .MAD1 [OPCODE] eq1 OP_WRT ! OTHERWISE, IF 1ST OPCODE IS A WRITE (ALL IS O.K.)
: 3308 2          then
: 3309 2              FILL_BUFF (); ! FILL 1ST I/O BUFFER WITH APPROPRIATE DATA PATTERN
: 3310 2
: 3311 1      end; ! ROUTINE QIO_GEN

```

000000	012737	177777	000110'	.SBTTL	QIO_GEN MULTI-DRIVE TEST ROUTINES	
				QIO_GEN::		
000006	013746	000000G		MOV	# 1,MX2	3249
000012	004737	000000G		MOV	CCTLR, -(SP)	3251
000016	010037	000106'		JSR	PC,GET_PKT	
000022	005726			MOV	R0,MX1	
000024	005700			IST	(SP)+	
000026	002563			IST	R0	; MX1
000030	013746	000000G		BLT	6\$	3253
000034	004737	000000G		MOV	CCTLR, -(SP)	3255
				JSR	PC,GET_PKT	

M8

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0310
Page 79
(21)

000040	010037	000110'	MOV	R0,MX2		
000044	005726		TST	(SP)+		
000046	005700		TST	R0	; MX2	
000050	002011		BGE	1\$		
000052	013746	000106'	MOV	MX1,-(SP)		3258
000056	004737	000000G	JSR	PC,PUT.PKT		
000062	012737	177777 000106'	MOV	#-1,MX1		3259
000070	005726		TST	(SP)+		3260
000072	000207		RTS	PC		3257
000074	013746	000106'	MOV	MX1,-(SP)		3263
000100	012746	000106	MOV	#106,-(SP)		
000104	004737	000000G	JSR	PC,BL\$MUL		
000110	062700	000000G	ADD	#MSCP.PKT,R0		
000114	010037	000112'	MOV	R0,MAD1		
000120	013716	000110'	MOV	MX2,(SP)		3264
000124	012746	000106	MOV	#106,-(SP)		
000130	004737	000000G	JSR	PC,BL\$MUL		
000134	062700	000000G	ADD	#MSCP.PKT,R0		
000140	010037	000114'	MOV	R0,MAD2		
000144	004737	000000V	JSR	PC,GET.RANDOM		3265
000150	004737	000000V	JSR	PC,QIO.UNIT		3266
000154	132737	000001 000000G	BITB	#1,EOP.FLAG		3268
000162	001103		BNE	5\$		3219
000164	004737	000000V	JSR	PC,QIO.FUNC		3272
000170	004737	000000V	JSR	PC,QIO.LBN		3273
000174	004737	000000V	JSR	PC,QIO.SIZE		3274
000200	013716	000112'	MOV	MAD1,(SP)		3275
000204	062716	000032	ADD	#32,(SP)		
000210	004737	000000G	JSR	PC,GET.IO.BUFF		
000214	005737	000110'	TST	MX2		3277
000220	002437		BCT	3\$		
000222	013716	000114'	MOV	MAD2,(SP)		3280
000226	062716	000032	ADD	#32,(SP)		
000232	004737	000000G	JSR	PC,GET.IO.BUFF		
000236	013700	000114'	MOV	MAD2,R0		3282
000242	005760	000032	TST	32(R0)		
000246	001024		BNE	3\$		
000250	013700	000112'	MOV	MAD1,R0		3286
000254	062700	000032	ADD	#32,R0		
000260	005710		TST	(R0)		
000262	001407		BEQ	2\$		
000264	010016		MOV	R0,(SP)		3289
000266	004737	000000G	JSR	PC,PUT.IO.BUFF		
000272	013700	000112'	MOV	MAD1,R0		3290
000276	005060	000032	CLR	32(R0)		
000302	013716	000110'	MOV	MX2,(SP)		3293
000306	004737	000000G	JSR	PC,PUT.PKT		
000312	012737	177777 000110'	MOV	#-1,MX2		3294
000320	013700	000112'	MOV	MAD1,R0		3299
000324	005760	000032	TST	32(R0)		
000330	001010		BNE	4\$		
000332	013716	000105'	MOV	MX1,(SP)		3302
000336	004737	000000G	JSR	PC,PUT.PKT		

N8

ZRQAM3 RD/RX EXERCISER
V01.6 MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0311
Page 80
(21)

000342	012737	177777	000106'	MOV	#-1,MX1	:	3303
000350	000410			BR	5\$	:	3299
000352	013700	000112'	4\$:	MOV	MAD1,R0	:	3307
000356	126027	000022	000042	CMPB	22(R0),#42	:	
000364	001002			BNE	5\$	:	
000366	004737	000000V		JSR	PC,FILL,BUFF	:	3309
000372	062706	000006	5\$:	ADD	#6,SP	:	3243
000376	000207		6\$:	RTS	PC	:	3219

; Routine Size: 128 words, Routine Base: \$CODE\$ + 10310
; Maximum stack depth per invocation: 4 words

{}

ZRQAMS
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11: 16 V4.0 579
DISK#USER2:[POWERS]ZRQADO.BL?16

SEQ 0312
Page 81
(22)

```

: 3312 1 GLOBAL routine GET_RANDOM : novalue =
: 3313 1
: 3314 1 !
: 3315 1 ! THIS ROUTINE IS CALLED BY QIO_GEN TO GENERATE A SET OF RANDOM NUMBERS,
: 3316 1 ! AND TO STORE THEM INTO THE RANDOM NUMBER TABLE (RANDOM). THE RANDOM
: 3317 1 ! NUMBERS ARE USED TO SELECT I/O REQUEST PARAMETERS FOR THE CURRENT QIO
: 3318 1 ! OR QIO PAIR. IN ADDITION, IF DATA PATTERN #1 IS BEING USED, THESE
: 3319 1 ! RANDOM NUMBERS WILL BE USED IN THE WRITE OPERATION.
: 3320 1 !
: 3321 1 !
: 3322 2 begin
: 3323 2
: 3324 2 own
: 3325 2 SEED : word initial (173),
: 3326 2 NEXT_RANDOM : word initial (245);
: 3327 2
: 3328 2 incr COUNT from 0 to (RDM_LEN - 1) do
: 3329 3 begin
: 3330 3 SEED = (.SEED + .NEXT_RANDOM + 1) * 4;
: 3331 3 NEXT_RANDOM = (.NEXT_RANDOM / 4) * .SEED;
: 3332 3 RANDOM [.COUNT] = .NEXT_RANDOM;
: 3333 2 end;
: 3334 2
: 3335 1 end;

```

```

001200 .PSECT $GGG$, RO
001200 000255 SEED: .WORD 255
001202 000365 NEXT_RANDOM: .WORD 365

```

```

010710 .SBTTL GET_RANDOM MULTI DRIVE TEST ROUTINES
.PSECT $CODE$, RO

```

```

000000 004137 000000G GET_RANDOM:
000004 013703 001200' JSR R1,$SAVE$ 3312
000010 013702 001202' MOV SEED,R3 3330
000014 005001 MOV NEXT_RANDOM,R2
000016 010200 14: CLR R1 COUNT 3328
000020 060300 MOV R2,RO 3330
000022 006300 ADD R3,RO
000024 006300 ASL RO
000026 010037 001200' MOV RO,SEED
000032 062737 000004 001200' ADD #4,SEED
000040 010246 MOV R2,(SP) 3331
000042 012746 000004 MOV #4,(SP)
000046 004737 000000G JSR PC,BL$DIV
000052 013703 001200' MOV SEED,R3
000056 060300 ADD R3,RO
000060 010037 001202' MOV RO,NEXT_RANDOM

```

C9

ZRQAMS
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11-16 V4.0-579
DISK#USER2:[POWERS]ZRQADO.BL2:6

SEQ 0315
Page 82
(22)

000064	010002		MOV	R0,R2	; NEXT,RANDNUM,*	3332
000066	010261	000000G	MOV	R2,RANDOM(R1)	; *,*(COUNT)	
000072	022626		CMP	(SP),*(SP)*		3329
000074	062701	000002	ADD	#2,R1	; *,COUNT	3328
000100	020127	000036	CMP	R1,#36	; COUNT,*	
000104	003744		BLE	1*		
000106	000207		RTS	PC		3312

; Routine Size: 36 words, Routine Base: \$CODE\$ + 10710
; Maximum stack depth per invocation: 7 words

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0314
Page 83
(23)

```

3336 1 GLOBAL routine QIO_UNIT : novalue *
3337 1
3338 1 !
3339 1 !
3340 1 ! THIS ROUTINE IS CALLED BY QIO_GEN TO RANDOMLY SELECT ONE UNIT
3341 1 ! CONFIGURED UNDER THE CURRENT CONTROLLER (CCTLR) TO BE USED FOR THE
3342 1 ! CURRENT QIO OR QIO PAIR. THE UNIT SELECTED IS BASED ON THE NUMBER OF
3343 1 ! UNITS ELIGIBLE TO RECEIVE AN I/O REQUEST (FROM 1 TO 4) AND THE FIRST
3344 1 ! RANDOM NUMBER IN THE RANDOM NUMBER TABLE (RANDOM).
3345 1 !
3346 1 ! IMPLICIT INPUTS:
3347 1 !     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
3348 1 !
3349 1 ! IMPLICIT OUTPUTS:
3350 1 !     THE RD/RX DISK NUMBER (DISK ADDRESS) IS LOADED INTO THE
3351 1 !     APPROPRIATE FIELD OF BOTH MSCP PACKETS.
3352 1 !
3353 2 begin
3354 2
3355 2 local
3356 2     MOD_COUNT : byte,
3357 2     TBL_COUNT : byte,
3358 2     SELECT_RD : byte initial (byte (TRUE)),
3359 2     RD_COUNT : word initial (0),
3360 2     RX_COUNT : word initial (0);
3361 2
3362 2 !
3363 2 ! THE UNITS WILL BE SELECTED ON AN ADJUSTABLE RATIO, RD51/52 TO RX50,
3364 2 ! SELECTED VIA THE SOFTWARE PARAMETERS
3365 2 !
3366 2 ! THIS MODE IS FOR SELECTING DEVICES ON THE FOLLOWING SCHEME:
3367 2 ! CHOOSE A DEVICE AND KEEP IT SELECTED FOR A CONSTANT TIME, THEN
3368 2 ! MOVE TO THE NEXT. THIS IS NON-RANDOM, FIXED SEQUENTIAL OPERATIONAL
3369 2 ! MODE
3370 2 !
3371 2
3372 2 if (not BIT_TST (SWP_FLAGS, SWF_RDM)) and          ! NOT RANDOM MODE
3373 3     (not BIT_TST (SWP_FLAGS, SWF_SEQ))            ! NOT RANDOM SEQUENTIAL MODE
3374 2 then
3375 2
3376 2     if (.BST_CNT neq 0) and
3377 2         (.CST_ADDR [.BST_DEV + OF_DATA, D_PRES] eq1 PRESENT) and
3378 2         (.CST_ADDR [.BST_DEV + OF_DATA, D_STAT] eq1 ONLINE) and
3379 3         (not .CST_ADDR [.BST_DEV + OF_DATA, D_FATAL])
3380 2     then
3381 3         begin
3382 3             BST_CNT = .BST_CNT - 1;
3383 3             SET_UPAR (.BST_DEV);
3384 3             MAD1 [DK_NUM] = .CDISK;
3385 3             MAD2 [DK_NUM] = .CDISK;
3386 3             return;
3387 3         end
3388 2     else

```

ZRQAM3
VO1.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0315
Page 84
(23)

```

: 3389 3      begin                                ! GET NEW DEVICE
: 3390 3
: 3391 3      incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3392 3
: 3393 3      if (.CST_ADDR [.OFFSET + OF_DATA, D_PRES] eq1 PRESENT) and
: 3394 3      (.CST_ADDR [.OFFSET + OF_DATA, D_STAT] eq1 ONLINE) and
: 3395 4      (not .CST_ADDR [.OFFSET + OF_DATA, D_FATAL])
: 3396 3      then
: 3397 3
: 3398 4          if (.CST_ADDR [.OFFSET + OF_DATA, D_TYPE] eq1 FIXED)
: 3399 3          then
: 3400 3              RD_COUNT = .RD_COUNT + 1          ! NUMBER OF RD51/52s UNDER TEST
: 3401 3          else
: 3402 3              RX_COUNT = .RX_COUNT + 1          ! NUMBER OF RX50s UNDER TEST
: 3403 3
: 3404 3      incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3405 4      begin
: 3406 4
: 3407 4          if (.BST_DEV eq1 0) or
: 3408 5          (.BST_DEV eq1 ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN))
: 3409 4          then
: 3410 4              BST_DEV = OF_UN
: 3411 4          else
: 3412 4              BST_DEV = .BST_DEV + UNIT_SIZE;
: 3413 4
: 3414 4          if (.CST_ADDR [.BST_DEV + OF_DATA, D_PRES] eq1 PRESENT) and
: 3415 4          (.CST_ADDR [.BST_DEV + OF_DATA, D_STAT] eq1 ONLINE) and
: 3416 5          (not .CST_ADDR [.BST_DEV + OF_DATA, D_FATAL])
: 3417 4          then
: 3418 5              begin
: 3419 5
: 3420 5                  if .CST_ADDR [.BST_DEV + OF_DATA, D_TYPE] eq1 REMOVABLE
: 3421 5                  then
: 3422 5                      BST_CNT = .RX_MAX_SEQ_CNT / .RX_COUNT
: 3423 5                  else
: 3424 5                      BST_CNT = .RD_MAX_SEQ_CNT / .RD_COUNT;
: 3425 5
: 3426 5                  if .BST_CNT eq1 0
: 3427 5                  then
: 3428 5                      BST_CNT = 1;
: 3429 5
: 3430 5                  SET_UPAR (.BST_DEV);
: 3431 5                  MAD1 [DK_NUM] = .CDISK;
: 3432 5                  MAD2 [DK_NUM] = .CDISK;
: 3433 5                  return;
: 3434 4              end;
: 3435 4
: 3436 3      end;
: 3437 3
: 3438 2      end;
: 3439 2
: 3440 2      !
: 3441 2      ! RANDOM SELECTION OF DRIVES

```

F9

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0316
Page 85
(23)

```

: 3442 2  !
: 3443 2  !
: 3444 2  ! DETERMINE IF RD51/52s ARE TO BE SELECTED
: 3445 2  !
: 3446 2  !
: 3447 2      if ((.RANDOM [RDM_LEN - 1] and %o'077777') mod 100) gequ .SWP_RAT
: 3448 2      then
: 3449 2          SELECT_RD = FALSE;
: 3450 2  !
: 3451 2  ! IF RD51/52s SELECTED
: 3452 2  !
: 3453 2  !
: 3454 2  ! COUNT NUMBER OF RD51/52s AVAILABLE
: 3455 2  !
: 3456 2  !
: 3457 2      if .SELECT_RD
: 3458 2      then
: 3459 3          begin
: 3460 3              MOD_COUNT = 0;                      ! COUNT THE NUMBER OF RDs UNDER TEST
: 3461 3              incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3462 3                  if (.CST_ADDR [.OFFSET + OF_DATA, D_PRES] eq1 PRESENT) and
: 3463 3                      (.CST_ADDR [.OFFSET + OF_DATA, D_STAT] eq1 ONLINE) and
: 3464 3                      (.CST_ADDR [.OFFSET + OF_DATA, D_TYPE] eq1 FIXED) and
: 3465 3                      (not .CST_ADDR [.OFFSET + OF_DATA, D_FATAL])
: 3466 3                  then
: 3467 4                      begin
: 3468 4                          STORAGE [.MOD_COUNT] = .OFFSET;
: 3469 4                          MOD_COUNT = .MOD_COUNT + 1;
: 3470 4                      end;
: 3471 4
: 3472 3
: 3473 3
: 3474 3  !
: 3475 3  ! SELECT ON OF THE RD51/52s
: 3476 3  !
: 3477 3  !
: 3478 3      if .MOD_COUNT neq 0                      ! IF AT LEAST ONE RD51/52 PRESENT
: 3479 3      then
: 3480 4          begin
: 3481 4              TBL_COUNT = 0;
: 3482 4
: 3483 4          do
: 3484 5              begin
: 3485 5                  SET_UPAR (.STORAGE [(RANDOM [.TBL_COUNT] and %o'077777') mod .MOD_COUNT]);
: 3486 5                  TBL_COUNT = .TBL_COUNT + 1;
: 3487 5              end
: 3488 5          until ((.CST_ADDR [.CUOFF + OF_DATA, D_PRES] eq1 PRESENT) and
: 3489 5                  (.CST_ADDR [.CUOFF + OF_DATA, D_STAT] eq1 ONLINE) and
: 3490 4                  (not .CST_ADDR [.CUOFF + OF_DATA, D_FATAL])) or
: 3491 4                  (.TBL_COUNT eq1 RDM_LEN);
: 3492 4
: 3493 4          MAD1 [DK_NUM] = .CDISK;
: 3494 4          MAD2 [DK_NUM] = .CDISK;

```

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0317
Page 86
(23)

```

: 3495 4          return;
: 3496 3          end;
: 3497 3
: 3498 2          end;
: 3499 2
: 3500 2
: 3501 2          !
: 3502 2          ! IF NO RD51/52 SELECTED, SELECT AN RX50
: 3503 2          !
: 3504 2          ! COUNT THE NUMBER OF RX50s
: 3505 2          !
: 3506 2          MOD_COUNT = 0;
: 3507 2
: 3508 2          incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3509 2
: 3510 2          if (CST_ADDR [ .OFFSET + OF_DATA, D_PRES] eq1 PRESENT) and
: 3511 2             (.CST_ADDR [ .OFFSET + OF_DATA, D_STAT] eq1 ONLINE) and
: 3512 2             (.CST_ADDR [ .OFFSET + OF_DATA, D_TYPE] eq1 REMOVABLE) and
: 3513 3             (not .CST_ADDR [ .OFFSET + OF_DATA, D_FATAL])
: 3514 2          then
: 3515 3              begin
: 3516 3                  STORAGE [ .MOD_COUNT] = .OFFSET;
: 3517 3                  MOD_COUNT = .MOD_COUNT + 1;
: 3518 2              end;
: 3519 2
: 3520 2          !
: 3521 2          ! AND CHOOSE ONE!
: 3522 2          !
: 3523 2
: 3524 2          if .MOD_COUNT neq 0
: 3525 2          then
: 3526 3              begin
: 3527 3                  TBL_COUNT = 0;
: 3528 3
: 3529 3              do
: 3530 4                  begin
: 3531 4                      SET_UPAR ( .STORAGE [( .RANDOM [ .TBL_COUNT] and %o'077777') mod .MOD_COUNT]);
: 3532 4                      TBL_COUNT = .TBL_COUNT + 1;
: 3533 4                  end
: 3534 4              until ((.CST_ADDR [ .CUOFF + OF_DATA, D_PRES] eq1 PRESENT) and
: 3535 4                     (.CST_ADDR [ .CUOFF + OF_DATA, D_STAT] eq1 ONLINE) and
: 3536 3                     (not .CST_ADDR [ .CUOFF + OF_DATA, D_FATAL])) or
: 3537 3                     (.TBL_COUNT eq1 RDM_LEN);
: 3538 3
: 3539 3              MAD1 [DK_NUM] = .CDISK;
: 3540 3              MAD2 [DK_NUM] = .CDISK;
: 3541 3              return;
: 3542 2              end;
: 3543 2
: 3544 2          !
: 3545 2          ! IF NO UNIT SELECTED SO FAR BY ABOVE METHOD, SELECT ANY ONE AT RANDOM
: 3546 2          !
: 3547 2          ! COUNT ALL UNITS AVAILABLE

```


H9

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0318
Page 87
(23)

```

: 3548 2  !
: 3549 2
: 3550 2  MOD_COUNT = 0;
: 3551 2
: 3552 2  incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 3553 2
: 3554 2  if (.CST_ADDR [.OFFSET + OF_DATA, D_PRES] eq1 PRESENT) and
: 3555 2  (.CST_ADDR [.OFFSET + OF_DATA, D_STAT] eq1 ONLINE) and
: 3556 3  (not .CST_ADDR [.OFFSET + OF_DATA, D_FATAL])
: 3557 2  then
: 3558 3  begin
: 3559 3  STORAGE [.MOD_COUNT] = .OFFSET;
: 3560 3  MOD_COUNT = .MOD_COUNT + 1;
: 3561 2  end;
: 3562 2
: 3563 2  !
: 3564 2  ! SELECT ANY ONE ONE UNIT AT RANDOM
: 3565 2  !
: 3566 2  if .MOD_COUNT neq 0
: 3567 2  then
: 3568 3  begin
: 3569 3  TBL_COUNT = 0;
: 3570 3
: 3571 3  do
: 3572 4  begin
: 3573 4  SET_UPAR (.STORAGE [(RANDOM [.TBL_COUNT] and %0'077777') mod .MOD_COUNT]);
: 3574 4  TBL_COUNT = .TBL_COUNT + 1;
: 3575 4  end
: 3576 4  until ((.CST_ADDR [.CUOFF + OF_DATA, D_PRES] eq1 PRESENT) and
: 3577 4  (.CST_ADDR [.CUOFF + OF_DATA, D_STAT] eq1 ONLINE) and
: 3578 3  (not .CST_ADDR [.CUOFF + OF_DATA, D_FATAL])) or
: 3579 3  (.TBL_COUNT eq1 RDM_LEN);
: 3580 3
: 3581 3  MAD1 [DK_NUM] = .CDISK;
: 3582 3  MAD2 [DK_NUM] = .CDISK;
: 3583 3  return
: 3584 3  end
: 3585 3
: 3586 3  !
: 3587 3  ! DECLARE END-OF-PASS IF NO UNIT ONLINE
: 3588 3  !
: 3589 3
: 3590 2  else
: 3591 2  EOP_FLAG = TRUE;
: 3592 2
: 3593 1  end;

```

! ROUTINE QIO_UNIT

000000 004137 000000G
000004 112704 000001
000010 005002

.SBTTL QIO_UNIT MULTI-DRIVE TEST ROUTINES
QIO_UNIT::

JSR R1,\$SAVE4
MOV8 R1,R4
CLR R2
; .,SELECT,RO
; RD,COUNT

3350
3353

ZRGAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0 579
DISK\$USER2:[POWERS]ZRGAD0.BL2;6SEQ 0319
Page 88
(23)

000012	005003		CLR	R3	; RX.COUNT	
000014	032737	000002 000000G	BIT	02,SWP.FLAGS		3372
000022	001003		BNE	1\$		
000024	032737	001000 000000G	BIT	01000,SWP.FLAGS		3373
000032	001402		BEQ	2\$		
000034	000137	011504'	JMP	15\$		
000040	005737	001146'	TST	BST.CNT		3376
000044	001452		BEQ	3\$		
000046	013700	001150'	MOV	BST.DEV,RO		3377
000052	006300		ASL	RO		
000054	063700	000000G	ADD	CST.ADDR,RO		
000060	032710	040000	BIT	040000,(RO)		
000064	001442		SEQ	3\$		
000066	013700	001150'	MOV	BST.DEV,RO		3378
000072	006300		ASL	RO		
000074	063700	000000G	ADD	CST.ADDR,RO		
000100	032710	020000	BIT	020000,(RO)		
000104	001432		BEQ	3\$		
000106	013700	001150'	MOV	BST.DEV,RO		3379
000112	006300		ASL	RO		
000114	063700	000000G	ADD	CST.ADDR,RO		
000120	032710	010000	BIT	010000,(RO)		
000124	001022		BNE	3\$		
000126	005337	001146'	DEC	BST.CNT		3382
000132	013746	001150'	MOV	BST.DEV, -(SP)		3383
000136	004737	000000G	JSR	PC,SET,UPAR		
000142	013700	000112'	MOV	MAD1,RO		3384
000146	013760	000000G 000016	MOV	CDISK,16(RO)		
000154	013700	000114'	MOV	MAD2,RO		3385
000160	013760	000000G 000016	MOV	CDISK,16(RO)		
000166	005726		IST	(SP)+		3386
000170	000207		RTS	PC		3381
000172	012701	000006	MOV	06,R1	; *,OFFSET	3391
000176	010100		MOV	R1,RO	; OFFSET,*	3393
000200	063700	000000G	ADD	CST.ADDR,RO		
000204	032710	040000	BIT	040000,(RO)		
000210	001414		BEQ	6\$		
000212	032710	020000	BIT	020000,(RO)		3394
000216	001411		BEQ	6\$		
000220	032710	010000	BIT	010000,(RO)		3395
000224	001006		BNE	6\$		
000226	132710	000020	BIT8	020,(RO)		3398
000232	001402		BEQ	5\$		
000234	005202		INC	R2	; RD.COUNT	3400
000236	000401		BR	6\$		3398
000240	005203		INC	R3	; RX.COUNT	3401
000242	062701	000024	ADD	024,R1	; *,OFFSET	3391
000246	020127	000102	CMP	R1,0102	; OFFSET,*	
000252	003751		BLE	4\$		
000254	012701	000003	MOV	03,R1	; *,OFFSET	3404
000260	013700	001150'	MOV	BST.DEV,RO		3405
000264	001403		BLE	8\$		
000266	020027	000041	CMP	RO,041		3408

ZRGAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0320
Page 89
(23)

000272	001004				BNE	9\$			
000274	012737	000003	001150'	8\$:	MOV	#3,BST.DEV	:		3410
000302	000403				BR	10\$	:		3407
000304	062737	000012	001150'	9\$:	ADD	#12,BST.DEV	:		3412
000312	013700	001150'		10\$:	MOV	BST.DEV,R0	:		3414
000316	006300				ASL	R0			
000320	063700	000000G			ADD	CST.ADDR,R0			
000324	032710	040000			BIT	#40000,(R0)			
000330	001450				BEQ	14\$			
000332	032710	020000			BIT	#20000,(R0)	:		3415
000336	001445				BEQ	14\$			
000340	032710	010000			BIT	#10000,(R0)	:		3416
000344	001042				BNE	14\$			
000346	132710	000020			BITB	#20,(R0)	:		3420
000352	001004				BNE	11\$			
000354	013746	001176'			MOV	RX.MAX.SEQ.CNT,-(SP)	:		3422
000360	010346				MOV	R3,-(SP)	:	RX.COUNT,*	
000362	000403				BR	12\$			
000364	013746	001174'		11\$:	MOV	RD.MAX.SEQ.CNT,-(SP)	:		3424
000370	010246				MOV	R2,-(SP)	:	RD.COUNT,*	
000372	004737	000000G		12\$:	JSR	PC,BL\$DIV			
000376	010037	001146'			MOV	R0,BST.CNT			
000402	001003				BNE	13\$	:		3426
000404	012737	000001	001146'		MOV	#1,BST.CNT	:		3428
000412	013716	001150'		13\$:	MOV	BST.DEV,(SP)	:		3430
000416	004737	000000G			JSR	PC,SET.UPAR			
000422	013700	000112'			MOV	MAD1,R0	:		3431
000426	013760	000000G	000016		MOV	CDISK,16(R0)			
000434	013700	000114'			MOV	MAD2,R0	:		3432
000440	013760	000000G	000016		MOV	CDISK,16(R0)			
000446	022626				CMP	(SP)+,(SP)+	:		3433
000450	000207				RTS	PC	:		3418
000452	062701	000012		14\$:	ADD	#12,R1	:	*,OFFSET	3404
000456	020127	000041			CMP	R1,#41	:	OFFSET,*	
000462	003676				BLE	7\$			
000464	013746	000036G		15\$:	MOV	RANDOM*36,-(SP)	:		3447
000470	042716	100000			BIC	#100000,(SP)			
000474	012746	000144			MOV	#144,-(SP)			
000500	004737	000000G			JSR	PC,BL\$MOD			
000504	022626				CMP	(SP)+,(SP)+			
000506	020037	000000G			CMP	R0,SWP.RAT			
000512	103401				BLO	16\$			
000514	105004				CLRB	R4	:	SELECT,RD	3449
000516	006004			16\$:	ROR	R4	:	SELECT,RD	3457
000520	103105				BCC	21\$			
000522	105003				CLRB	R3	:	MOD,COUNT	3460
000524	012701	000003			MOV	#3,R1	:	*,OFFSET	3462
000530	010100			17\$:	MOV	R1,R0	:	OFFSET,*	3464
000532	006300				ASL	R0			
000534	063700	000000G			ADD	CST.ADDR,R0			
000540	032710	040000			BIT	#40000,(R0)			
000544	001417				BEQ	18\$			
000546	032710	020000			BIT	#20000,(R0)	:		3465

K9

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0321
Page 90
(23)

000552	001414		BEQ	18\$			
000554	132710	000020	BITB	#20,(R0)			3466
000560	001411		BEQ	18\$			
000562	032710	010000	BIT	#10000,(R0)			3467
000566	001006		BNE	18\$			
000570	005000		CLR	R0			3470
000572	150300		BISB	R3,R0		; MOD.COUNT,*	
000574	006300		ASL	R0			
000576	010160	000064	MOV	R1,STORAGE(R0)		; OFFSET,*	
000602	105203		INCB	R3		; MOD.COUNT	3471
000604	062701	000012	ADD	#12,R1		; *,OFFSET	3462
000610	020127	000041	CMP	R1,#41		; OFFSET,*	
000614	003745		BLE	17\$			
000616	105703		TSTB	R3		; MOD.COUNT	3478
000620	001445		BEQ	21\$			
000622	105002		CLRB	R2		; TBL.COUNT	3481
000624	005000		CLR	R0			3485
000626	150200		BISB	R2,R0		; TBL.COUNT,*	
000630	006300		ASL	R0			
000632	016046	000000	MOV	RANDOM(R0),-(SP)			
000636	042716	100000	BIC	#100000,(SP)			
000642	005046		CLR	-(SP)			
000644	110316		MOVB	R3,(SP)		; MOD.COUNT,*	
000646	004737	000000G	JSR	PC,BL\$MOD			
000652	006300		ASL	R0			
000654	016016	000064	MOV	STORAGE(R0),(SP)			
000660	004737	000000G	JSR	PC,SET.UPAR			
000664	105202		INCB	R2		; TBL.COUNT	3486
000666	022626		CMP	(SP),,(SP)+			3484
000670	013700	000000G	MOV	CUOFF,R0			3488
000674	006300		ASL	R0			
000676	063700	000000G	ADD	CST,ADDR,R0			
000702	032710	040000	BIT	#40000,(R0)			
000706	001406		BEQ	20\$			
000710	032710	020000	BIT	#20000,(R0)			3489
000714	001403		BEQ	20\$			
000716	032710	010000	BIT	#10000,(R0)			3490
000722	001510		BEQ	26\$			
000724	120227	000020	CMPB	R2,#20		; TBL.COUNT,*	3491
000730	001335		BNE	19\$			
000732	000504		BR	26\$			3493
000734	105003		CLRB	R3		; MOD.COUNT	3506
000736	012701	000003	MOV	#3,R1		; *,OFFSET	3508
000742	010100		MOV	R1,R0		; OFFSET,*	3510
000744	006300		ASL	R0			
000746	063700	000000G	ADD	CST,ADDR,R0			
000752	032710	040000	BIT	#40000,(R0)			
000756	001417		BEQ	23\$			
000760	032710	020000	BIT	#20000,(R0)			3511
000764	001414		BEQ	23\$			
000766	132710	000020	BITB	#20,(R0)			3512
000772	001011		BNE	23\$			
000774	032710	010000	BIT	#10000,(R0)			3513

L9

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bligs-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0322
Page 91
(23)

001000	001006		BNE	23:			
001002	005000		CLR	R0			
001004	150300		BISB	R3,R0		; MOD.COUNT,*	3516
001006	006300		ASL	R0			
001010	010160	000064'	MOV	R1,STORAGE(R0)		; OFFSET,*	
001014	105203		INCB	R3		; MOD.COUNT	3517
001016	062701	000012	23:	ADD	#12,R1	; *,OFFSET	3508
001022	020127	000041		CMP	R1,#41	; OFFSET,'	
001026	003745		BLE	22:			
001030	105703		TSTB	R3		; MOD.COUNT	3524
001032	001445		BEQ	27:			
001034	105002		CLRB	R2		; TBL.COUNT	3527
001036	005000		24:	CLR	R0		3531
001040	150200		BISB	R2,R0		; TBL.COUNT,*	
001042	006300		ASL	R0			
001044	016046	000000G	MOV	RANDOM(R0),-(SP)			
001050	042716	100000	BIC	#100000,(SP)			
001054	005046		CLR	-(SP)			
001056	110316		MOVB	R3,(SP)		; MOD.COUNT,*	
001060	004737	000000G	JSR	PC,BL\$MOD			
001064	006300		ASL	R0			
001066	016016	000064'	MOV	STORAGE(R0),(SP)			
001072	004737	000000G	JSR	PC,SET.UPAR			
001076	105202		INCB	R2		; TBL.COUNT	3532
001100	022626		CMP	(SP)+,(SP)+			3530
001102	013700	000000G	MOV	CUOFF,R0			3534
001106	006300		ASL	R0			
001110	063700	000000G	ADD	CST.ADDR,R0			
001114	032710	040000	BIT	#40000,(R0)			
001120	001406		BEQ	25:			
001122	032710	020000	BIT	#20000,(R0)			3535
001126	001403		BEQ	25:			
001130	032710	010000	BIT	#10000,(R0)			3536
001134	001505		BEQ	32:			
001136	120227	000020	25:	CMPB	R2,#20	; TBL.COUNT,*	3537
001142	001335		BNE	24:			
001144	000501		26:	BR	32:		3539
001146	105003		27:	CLRB	R3	; MOD.COUNT	3550
001150	012701	000003		MOV	#3,R1	; *,OFFSET	3552
001154	010*00		28:	MOV	R1,R0	; OFFSET,*	3554
001156	006300		ASL	R0			
001160	063700	000000G	ADD	CST.ADDR,R0			
001164	032710	040000	BIT	#40000,(R0)			
001170	001414		BEQ	29:			
001172	032710	020000	BIT	#20000,(R0)			3555
001176	001411		BEQ	29:			
001200	032710	010000	BIT	#10000,(R0)			3556
001204	001006		BNE	29:			
001206	005000		CLR	R0			3559
001210	150300		BISB	R3,R0		; MOD.COUNT,*	
001212	006300		ASL	R0			
001214	010160	000064'	MOV	R1,STORAGE(R0)		; OFFSET,*	
001220	105203		INCB	R3		; MOD.COUNT	3560

M9

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0323
Page 92
(23)

001222	062701	000012	29\$:	ADD	#12,R1	; *,OFFSET	3552
001226	020127	000041		CMP	R1,#41	; OFFSET,*	
001232	003750			BLE	28\$		
001234	105703			TSTB	R3	; MOD.COUNT	3566
001236	001457			BEQ	33\$		
001240	105002			CLRB	R2	; TBL.COUNT	3569
001242	005000		30\$:	CLR	R0		3573
001244	150200			BISB	R2,R0	; TBL.COUNT,*	
001246	006300			ASL	R0		
001250	016046	000000G		MOV	RANDOM(R0),-(SP)		
001254	042716	100000		BIC	#100000,(SP)		
001260	005046			CLR	(SP)		
001262	110316			MOVB	R3,(SP)	; MOD.COUNT,*	
001264	004737	000000G		JSR	PC,BL\$MOD		
001270	006300			ASL	R0		
001272	016016	000064'		MOV	STORAGE(R0),(SP)		
001276	004737	000000G		JSR	PC,SET,UPAR		
001302	105202			INCB	R2	; TBL.COUNT	3574
001304	022626			CMP	(SP)+,(SP)+		3572
001306	013700	000000G		MOV	CUOFF,R0		3576
001312	006300			ASL	R0		
001314	063700	000000G		ADD	CS1,ADDR,R0		
001320	032710	040000		BIT	#40000,(R0)		
001324	001406			BEQ	31\$		
001326	032710	020000		BIT	#20000,(R0)		3577
001332	001403			BEQ	31\$		
001334	032710	010000		RIT	#10000,(R0)		3578
001340	001403			BEQ	32\$		
001342	120227	000020	31\$:	CMPB	R2,#20	; TBL.COUNT,*	3579
001346	001335			BNE	30\$		
001350	013700	000112'	32\$:	MOV	MAD1,R0		3581
001354	013760	000000G 000016		MOV	CDISK,16(R0)		
001362	013700	000114'		MOV	MAD2,R0		3582
001366	013760	000000G 000016		MUV	CDISK,16(R0)		
001374	000207			RTS	PC		3568
001376	112737	000001 000000G	33\$:	MOVB	#1,EOP.FLAG		3591
001404	000207			RTS	PC		3336

; Routine Size: 387 words, Routine Base: \$CODE\$ + 11020
 ; Maximum stack depth per invocation: 8 words

ZPQAM3
V01.6RO/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0324
Page 93
(24)

```

: 3594 1 GLOBAL routine QIO_FUNC : novalue =
: 3595 1
: 3596 1 !
: 3597 1 ! THIS ROUTINE IS CALLED BY QIO_GEN TO SELECT THE I/O FUNCTION (OPCODE)
: 3598 1 ! TO BE USED FOR THE CURRENT QIO OR QIO PAIR. THE FUNCTION IS DETERMINED
: 3599 1 ! BY THE FOLLOWING ALGORITHM:
: 3600 1 !
: 3601 1 ! IF THE CHOSEN UNIT IS PROTECTED
: 3602 1 ! THEN
: 3603 1 !     FUNCTION = READ
: 3604 1 ! ELSE (UNPROTECTED)
: 3605 1 !     FUNCTION (WRITE OR READ) IS BASED ON A RANDOM
: 3606 1 !     NUMBER
: 3607 1 !
: 3608 1 ! IN ADDITION, IF THE OPERATOR SELECTED THE OPTION OF PERFORMING WRITE-
: 3609 1 ! COMPARES AT THE HOST, AND IF A "WRITE" FUNCTION WAS CHOSEN ABOVE FOR
: 3610 1 ! THE FIRST QIO, THEN A "READ" OPCODE IS LOADED INTO THE SECOND MSCP
: 3611 1 ! PACKET. OTHERWISE, THE SECOND MSCP PACKET IS RETURNED TO THE POOL.
: 3612 1 !
: 3613 1 !
: 3614 1 ! PERIODIACALLY, THIS ROUTINE WILL CALL THE DUP ROUTINE BEFORE IT ZZZ
: 3615 1 ! BEGINS ITS OWN TASK. IF THE OPERATOR HAS SELECTED, "ALSO RUN ZZZ
: 3616 1 ! DUP EXERCISER," THEN DUP TESTING OF DBNS WILL BE INTERLEAVED ZZZ
: 3617 1 ! WITH THE REGULAR MSCP TESTING OF THE LBNS. ZZZ
: 3618 1 !
: 3619 1 ! TO AVOID LONG, CUMULATIVE INIT TIMES, THE DUP CODE IS ONLY ZZZ
: 3620 1 ! EXECUTED AFTER (25 TIMES 'DUPROUND') MSCP I/O's HAVE BEEN DONE. ZZZ
: 3621 1 ! THE NUMBER OF DUP I/O's IS 'DUPROUND'. THIS GIVES US A 25 TO 1 ZZZ
: 3622 1 ! INTERLEAVE. ZZZ
: 3623 1 !
: 3624 1 ! THE DUP TESTING IS DONE BY EXECUTING CONTROLLER LOCAL PROGRAMS ZZZ
: 3625 1 ! TO READ OR WRITE/READ DBNS. AFTER THE DUP TESTING, THE CON ZZZ
: 3626 1 ! TROLLER IS REINITIALIZED, AND QIO_FUNC ROUTINE CONTINUES FROM ZZZ
: 3627 1 ! WHERE IT LEFT OFF. ZZZ
: 3628 1 !
: 3629 1 !
: 3630 1 ! IMPLICIT INPUTS:
: 3631 1 !     CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 3632 1 !     CUOFF - CURRENT UNIT CST OFFSET
: 3633 1 !
: 3634 1 ! IMPLICIT OUTPUTS:
: 3635 1 !     THE OPCODE FIELD OF ONE OR BOTH MSCP PACKETS IS LOADED.
: 3636 1 !
: 3637 1 !
: 3638 2 begin
: 3639 2
: 3640 2 local
: 3641 2     FUNC : word;
: 3642 2
: 3643 2 DUOFF = .CUOFF;
: 3644 2
: 3645 3 IF ((.CST_ADDR [.DUOFF + OF_COUNT, D_COUNT] LEQ 0) AND !MSCP CNT=0
: 3646 3     (.CST_ADDR [.DUOFF, D_TYPE] NEQ RX_50) AND !FIXED DISK

```

! OPCODE (READ OR WRITE)

!SAVE IN CASE OTHER CMDS ZZZ

!LEFT IN QUEUE ZZZ

ZXX

ZXX

BIO

ZRQAM5
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11:16 V4.0-5/9
DISKUSER2:[POWERS]ZRQADO.BL2;6

SEQ 0325
Page 94
(24)

```

: 3647 3      (.CST_ADDR (.DUOFF + OF_DUPFLAGS, NODUPMEDIA) NEQ 1)) !MEDIA IN      ZZZ
: 3648 3      !ZZZ
: 3649 2      THEN      !ZZZ
: 3650 3      BEGIN      !ZZZ
: 3651 3      PUT_PKT (.MX2);      !RETURN 2ND ENVELOPE      !ZZZ
: 3652 3      MX2 = -1;      !INDICATE FAILURE      ZZZ
: 3653 3      DUP ();      !DO DUP TEST      ZZZ
: 3654 3      CST_ADDR (.DUOFF + OF_COUNT, D_COUNT) =      !REINIT MSCP FUN-      ZZZ
: 3655 3      (25 + .DUPROUND);      !CTION COUNTER      ZZZ
: 3656 3      !ZZZ
: 3657 3      !      THE FOLLOWING REINIT 2 ENVELOPES, SO THAT THE MSCP EXERCISER      ZZZ
: 3658 3      !      CAN PROCEED AS BEFORE THE DUP EXERCISER WAS CALLED.      ZZZ
: 3659 3      !      ZZZ
: 3660 3      DUP_FLAGS = .DUP_FLAGS OR SWP_DINT;      !SET DUP INIT FLAG      ZZZ
: 3661 3      INIT_TEST ();      !REINIT CONTROLLER      ZZZ
: 3662 3      DUP_FLAGS = .DUP_FLAGS AND (NOT SWP_DINT);      !CLR DUP INIT FLAG      ZZZ
: 3663 3      !ZZZ
: 3664 3      MX2 = 1;      !ASSUME NO 2ND ENVELOPE      ZZZ
: 3665 3      IF (MX1 = GET_PKT (.CCTLR)) LSS 0      !TRY FOR 1ST ENVELOPE      ZZZ
: 3666 4      OR (.EOP_FLAG)      !IF FAILURE      ZZZ
: 3667 3      THEN      RETURN;      !NO POINT TO GO ON      ZZZ
: 3668 3      IF (MX2 = GET_PKT (.CCTLR)) LSS 0      !TRY FOR 2ND ENVELOPE      ZZZ
: 3669 4      OR (.EOP_FLAG)      !IF FAILURE      ZZZ
: 3670 4      THEN      BEGIN      !ZZZ
: 3671 4      PUT_PKT (.MX1);      !PUT 1ST BACK IN POOL      ZZZ
: 3672 4      MX1 = -1;      !INDICATE FAILURE      ZZZ
: 3673 4      RETURN;      !DONE      ZZZ
: 3674 3      END;      !ZZZ
: 3675 3      !ZZZ
: 3676 3      MAD1 = MSCP_PKT + (.MX1 * PKT_LEN * 2); !CALC START ADDR      ZZZ
: 3677 3      MAD2 = MSCP_PKT + (.MX2 * PKT_LEN * 2); !OF BOTH ENVELOPES      ZZZ
: 3678 3      GET_RANDOM ();      !GET SET OF RANDOM NOS      ZZZ
: 3679 3      QIO_UNIT ();      !PUT RAND UNIT NO IN      ZZZ
: 3680 2      END;      !ENVELOPES      ZZZ
: 3681 2      !ZZZ
: 3682 2      !      MSCP CODE STARTS HERE      ZZZ
: 3683 2      !      ZZZ
: 3684 2      !
: 3685 2      CST_ADDR (.CUOFF + OF_COUNT, D_COUNT) =      !      ZZZ
: 3686 2      .CST_ADDR (.CUOFF + OF_COUNT, D_COUNT) - 1;      !DECR MSCP FUNCTION CNTR      ZZZ
: 3687 2      !
: 3688 2      MAD2 (OPCODE) = 0;      ! ASSUME 2ND PACKET NOT NEEDED
: 3689 2      !
: 3690 2      IF (.CST_ADDR (.CUOFF + OF_DATA, D_PROT) eq UNPROTECTED) and      ! IF "FORCED ERROR" SET IN LAST READ,
: 3691 2      (.CST_ADDR (.CUOFF + OF_DATA, D_TYPE) eq FIXED) and      !
: 3692 2      (.FORCED_ERROR)      !      REWRITE SAME BLOCK
: 3693 2      then      !
: 3694 2      FUNC = OP_WRT
: 3695 2      else
: 3696 2      !
: 3697 2      IF .CST_ADDR (.CUOFF + OF_DATA, D_PROT) eq PROTECTED      ! IF UNIT IS PROTECTED
: 3698 2      then
: 3699 2      FUNC = OP_RD      ! SET FUNCTION TO READ

```


C10

ZRQANS
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11 Apr 1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11:16 V4.0 5/9
DISK\$USER2:[POWER5]ZRQADO.BL2;6

SEQ 0326
Page 95
(24)

```

: 3700 2      else
: 3701 2
: 3702 3      if (.RANDOM [1] and 1)
: 3703 2      then
: 3704 2          FUNC = OP_RD
: 3705 2      else
: 3706 2          FUNC = OP_WRT;
: 3707 2
: 3708 2      if (MAD1 [OPCODE] = .FUNC) eq1 OP_WRT
: 3709 2      then
: 3710 3          begin
: 3711 3              MAD1 [CMD_TYPE] = NON_SEQ_CMD;
: 3712 3
: 3713 4              if BIT_TST (SWP_FLAGS, SWF_CWC)
: 3714 3              then
: 3715 3                  MAD1 [MODIFY] = MD_CMP
: 3716 3              else
: 3717 3
: 3718 4                  if BIT_TST (SWP_FLAGS, SWF_HWC)
: 3719 3                  then
: 3720 4                      begin
: 3721 4                          MAD1 [MODIFY] = MD_EXP;
: 3722 4                          MAD2 [OPCODE] = OP_RD;
: 3723 4                          MAD2 [MODIFY] = MD_EXP;
: 3724 4                          MAD2 [CMD_TYPE] = NON_SEQ_CMD;
: 3725 3                          end;
: 3726 3                      end
: 3727 2          else
: 3728 3              begin
: 3729 3                  MAD1 [CMD_TYPE] = NON_SEQ_CMD;
: 3730 3
: 3731 4                  if BIT_TST (SWP_FLAGS, SWF_CRC)
: 3732 3                  then
: 3733 3                      MAD1 [MODIFY] = MD_CMP;
: 3734 3
: 3735 2                  end;
: 3736 2
: 3737 2          if .MAD2 [OPCODE] eq1 0
: 3738 2          then
: 3739 3              begin
: 3740 3                  PUT_PKT (.MX2);
: 3741 3                  MX2 = -1;
: 3742 2              end;
: 3743 2
: 3744 1      end;

```

! USE 2ND RANDOM NUMBER TO SELECT
!
! READ
!
! WRITE
!
! LOAD CHOSEN OPCODE, IF WRITE
!
! NON-SEQUENTIAL COMMAND
!
! IF CONTROLLER DOES WRITE-COMPARES
!
! ADD COMPARE MODIFIER
!
!
! IF HOST DOES WRITE-COMPARES
!
!
! SET WRITE AS AN EXPRESS REQUEST
! SET READ OPCODE INTO 2ND MSCP PACKET
! SET READ AS AN EXPRESS REQUEST TOO
! NON-SEQUENTIAL COMMAND
!
!
! NON-SEQUENTIAL COMMAND
!
! IF READ COMPARES FUNCTION IS READ
!
! ADD COMPARE MODIFIER
!
!
! IF NO OPCODE IN 2ND PACKET
!
!
! RETURN 2ND PACKET TO POOL
! MARK IT UNUSED
!
!
! ROUTINE QIO_FUNC

000C00	004137	000000G	.SBTTL QIO_FUNC MULTI-DRIVE TEST ROUTINES
000004	013737	000000G 001156'	QIO_FUNC::
000012	013702	000000G	JSR R1,SAVE4
000016	013701	001156'	MOV CUOFF,DUOFF
			MOV CST,ADDR,R2
			MOV DUOFF,R1

3594
3643
3645

D10

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Release 16 V4.0-579
DISK#USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0327
Page 96
(24)

000022	010100		MOV	R1,R0		
000024	006300		ASL	R0		
000026	060200		ADD	R2,R0		
000030	005760	000022	TST	22(R0)		
000034	003146		BGT	4		
000036	010100		MOV	R1,R0		3646
000040	006300		ASL	R0		
000042	060200		ADD	R2,R0		
000044	132710	000020	BITB	20,(R0)		
000050	001540		BEQ	4		
000052	010100		MOV	R1,R0		3647
000054	006300		ASL	R0		
000056	060200		ADD	R2,R0		
000060	005760	000020	TST	20(R0)		
000064	100532		BMI	4		
000066	013746	000110'	MOV	MX2, -(SP)		3651
000072	004737	000000G	JSR	PC,PUT,PKT		
000076	012737	177777 000110'	MOV	4-1, MX2		3652
000104	004737	000000V	JSR	PC,DUP		3653
000110	013701	001156'	MOV	DUOFF,R1		3654
000114	006301		ASL	R1		
000116	063701	000000G	ADD	CST,ADDR,R1		
000122	013716	000000G	MOV	DUPROUND,(SP)		3655
000126	012746	000031	MOV	431, -(SP)		
000132	004737	000000G	JSR	PC,BL\$MUL		
000136	010061	000022	MOV	R0,22(R1)		
000142	052737	000002 000000G	BIS	2,DUP.FLAGS		3660
000150	004737	001140'	JSR	PC,INIT.TEST		3661
000154	042737	000002 000000G	BIC	2,DUP.FLAGS		3662
000162	012737	177777 000110'	MOV	4-1, MX2		3664
000170	013716	000000G	MOV	CCTLR,(SP)		3665
000174	004737	000000G	JSR	PC,GET,PKT		
000200	010037	000106'	MOV	R0,MX1		
000204	002426		BLT	2		
000206	132737	000001 000000G	BITB	41,EOP.FLAG		3666
000214	001022		BNE	2		3594
000216	013716	000000G	MOV	CCTLR,(SP)		3668
000222	004737	000000G	JSR	PC,GET,PKT		
000226	010037	000110'	MOV	R0,MX2		
000232	002404		BLT	1		
000234	132737	000001 000000G	BITB	41,EOP.FLAG		3669
000242	001411		BEQ	5		
000244	013716	000106'	MOV	MX1,(SP)		3671
000250	004737	000000G	JSR	PC,PUT,PKT		
000254	012737	177777 000106'	MOV	4-1, MX1		3672
000262	022626		CMF	(SP)+,(SP)+		3673
000264	000207		RTS	PC		3670
000266	013716	000106'	MOV	MX1,(SP)		3676
000272	012746	000106'	MOV	4106,(SP)		
000276	004737	000000G	JSR	PC,BL\$MUL		
000302	062700	000000G	ADD	4MSCP,PKT,R0		
000306	010037	000112'	MOV	R0,MAD1		
000312	013716	000110'	MOV	MX2,(SP)		3677

E10

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1111-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0328
Page 97
(24)

000316	012746	000106		MOV	0106, -(SP)		
000322	004737	000000G		JSR	PC, BL\$MUL		
000326	062700	000000G		ADD	0MSCP, PKT, R0		
000332	010037	000114'		MOV	R0, MAD2		
000336	004737	010710'		JSR	PC, GET, RANDOM	:	3673
000342	004737	011020'		JSR	PC, QIO, UNIT	:	3679
000346	062706	000010		ADD	010, SP	:	3650
000352	013700	000000G	4\$:	MOV	CUOFF, R0	:	3685
000356	006300			ASL	R0		
000360	063700	000000G		ADD	CST, ADDR, R0		
000364	005360	000022		DEC	22(R0)	:	3686
000370	013701	000114'		MOV	MAD2, R1	:	3688
000374	012704	000022		MOV	022, R4		
000400	060104			ADD	R1, R4		
000402	105014			CLR B	(R4)		
000404	013700	000000G		MOV	CUOFF, R0	:	3690
000410	006300			ASL	R0		
000412	063700	000000G		ADD	CST, ADDR, R0		
000416	005710			TST	(R0)		
000420	100007			BPL	5\$		
000422	132710	000020		BITB	020, (R0)	:	3691
000426	001404			BEQ	5\$		
000430	132737	000001	000000G	BITB	01, FORCED, ERROR	:	3692
000436	001012			BNE	7\$	:	3694
000440	032710	100000	5\$:	BIT	0100000, (R0)	:	3697
000444	001404			BEQ	6\$	:	3699
000446	032737	000001	000002G	BIT	01, RANDOM*2	:	3702
000454	001403			BEQ	7\$		
000458	012702	000041	6\$:	MOV	041, R2	: *, FUNC	3704
000462	000402			BR	8\$	:	3702
000464	012702	000042	7\$:	MOV	042, R2	: *, FUNC	3706
000470	013700	000112'	8\$:	MOV	MAD1, R0	:	3708
000474	013703	000000G		MOV	SWP, FLAGS, R3	:	3713
000500	110260	000022		MOVB	R2, 22(R0)	: FUNC, *	3708
000504	020227	000042		CMP	R2, 042	: FUNC, *	
000510	001025			BNE	9\$		
000512	112760	000002	000004	MOVB	02, 4(R0)	:	3711
000520	032703	000020		BIT	020, R3	:	3713
000524	001025			BNE	10\$	:	3715
000526	032703	000040		BIT	040, R3	:	3718
000532	001425			BEQ	11\$		
000534	012760	100000	000024	MOV	0-100000, 24(R0)	:	3721
000542	112714	000041		MOVB	041, (R4)	:	3722
000546	012761	100000	000024	MOV	0-100000, 24(R1)	:	3723
000554	112761	000002	000004	MOVB	02, 4(R1)	:	3724
000562	000411			BR	11\$	:	3708
000564	112760	000002	000004	MOVB	02, 4(R0)	:	3729
000572	032703	000004		BIT	04, R3	:	3731
000576	001403			BEQ	11\$		
000600	012760	040000	000024	MOV	040000, 24(R0)	:	3733
000606	105714		11\$:	ISTB	(R4)	:	3737
000610	001010			BNE	12\$		
000612	013746	000110'		MOV	MX2, -(SP)	:	3740

F10

ZRQAM3 RD/RX EXERCISER
V01.6 MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0329
Page 98
(24)

000616	004737	000000G	JSR	PC,PUT,PKT		
000622	012737	177777 000110'	MOV	#-1,MX2	:	3741
000630	005726		TST	(SP)+	:	3739
000632	000207	12\$:	RTS	PC	:	3594

; Routine Size: 206 words, Routine Base: \$CODE\$ + 12426
; Maximum stack depth per invocation: 10 words

G10

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK#USER2:[POWERS]ZRQADO.BL2;6SEQ 0330
Page 99
(25)

```

: 3745 1
: 3746 1 GLOBAL ROUTINE DUP : NOVALUE *
: 3747 1 !
: 3748 1 ! THIS ROUTINE IS CALLED BY GIO_FUNC AFTER 25 * 'DUPROUND' RD/WTS. !ZZZ
: 3749 1 ! THIS EXERCISER WAS PLACED IN THE MIDDLE OF THE MSCP EXERCISER, ZZZ
: 3750 1 ! SO COMMON INIT AND OTHER ROUTINES COULD BE USED. ZZZ
: 3751 1 ! ZZZ
: 3752 1 ! THE DUP EXERCISER WILL PERFORM EITHER READ-ONLY, OR WRITE-READ- ZZZ
: 3753 1 ! COMPARE OPERATIONS ON THE DIAGNOSTIC BLOCKS (DBNS). IT WILL ZZZ
: 3754 1 ! RECORD THE STATISTICS IN THE TALLY TABLES. ZZZ
: 3755 1 ! ZZZ
: 3756 1 ! THE PROGRAM USES CONTROLLER LOCAL PROGRAMS TO WRITE AND READ ZZZ
: 3757 1 ! DBNS. WHEN WRITING TO THE DBNS, A ONE WORD PATTERN WILL BE ZZZ
: 3758 1 ! SELECTED, AND REPLICATED THROUGH A 256 WORD BLOCK FOR DATA. ZZZ
: 3759 1 ! THE ROUTINE WILL WRITE 'DUPROUND' NUMBER OF SEQUENTIAL DBN ZZZ
: 3760 1 ! BLOCKS. IF THE CONTROLLER LOCAL PROGRAMS EXIST, AND THE OPERATOR ZZZ
: 3761 1 ! SELECTS 'WRITE TO DIAGNOSTIC AREA', WRITE-READ-COMPARES WILL BE ZZZ
: 3762 1 ! PERFORMED ON THE DBNS. OTHERWISE, READS WITH NO COMPARES WILL BE ZZZ
: 3763 1 ! DONE. BAD BLOCKS FOUND IN THE COMPARISON TESTS WILL NOT BE LIST- ZZZ
: 3764 1 ! ED IN THE RCT TABLES. ZZZ
: 3765 1 ! ZZZ
: 3766 1 ! AFTER 'DUPROUND' NUMBER OF DBNS HAVE BEEN TESTED, THE ENVELOPES ZZZ
: 3767 1 ! WILL BE REINITIATED, SO THAT THE MSCP EXERCISER CAN CONTINUE ZZZ
: 3768 1 ! AS BEFORE. ZZZ
: 3769 1 ! ZZZ
: 3770 1 ! IMPLICIT INPUTS: ZZZ
: 3771 1 ! CCTLN - CURRENT CONTROLLER NUMBER ZZZ
: 3772 1 ! CST_ADDRS - CONTAINS THE CURRENT CONTROLLER ZZZ
: 3773 1 ! STATUS TABLE ZZZ
: 3774 1 ! CUOFF - CURRENT OFFSET IN CST TABLE FOR ZZZ
: 3775 1 ! PARTICULAR DRIVE ZZZ
: 3776 1 ! ZZZ
: 3777 1 ! IMPLICIT OUTPUTS: ZZZ
: 3778 1 ! S_PATTERN - PATTERN BEING WRITTEN TO DBNS ZZZ
: 3779 1 ! ZZZ
: 3780 1 ! ZZZ
: 3781 1 ! ZZZ
: 3782 1 !

```

H10

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0331
Page 100
(26)

```

: 3783 1
: 3784 1
: 3785 2 BEGIN
: 3786 2 OWN
: 3787 2 TEMP : WORD;
: 3788 2
: 3789 2 !PRINTX (DBM110);
: 3790 2 !PRINTX (DER10);
: 3791 2
: 3792 2 until (.CRN_LOW eql .RP_ADDR [CRF_LO]) or      ! TO ENSURE THAT ALL RETURN MESSAGES HAVE BEEN PROCESSED
: 3793 2 (.EOP_FLAG eql true) do                      ! Make sure all MSCP commands are completed
: 3794 3 begin
: 3795 3 BREAK;                                         ! BREAK FOR ACT
: 3796 3 PROC_RET_PKT();                               ! PROCESS RETURN PACKET TO SEE IF OK FOR DUP
: 3797 3 RP_INDX = .RP_INDX + 1;                       ! INCREMENT RP_INDX
: 3798 3 if .RP_INDX geq RP_CNT then (RP_INDX = 0);    ! MAKE SURE THE COUNTER DOES NOT GET TO BIG
: 3799 3 RP_ADDR = RET_PKT + (.RP_INDX * RP_LEN * 2); ! CALCULATE RET_PKT ADDRESS
: 3800 2 end;
: 3801 2
: 3802 2
: 3803 2 S_PATTERN = .RANDOM [1];                      ! OTHER UNIT VARIABLES
: 3804 2
: 3805 2 IF (.CST_ADDR [.DUOFF + OF_DBN, D_DBN] + .dupround) GEQ 144 ! TEST TO SEE IF NEXT DBN'S TO LARGE
: 3806 2 THEN (.CST_ADDR [.DUOFF + OF_DBN, D_DBN] = 0); ! CIRCLE AROUND IF DBN TO LARGE
: 3807 2
: 3808 2 DUP_IDLE ();                                  ! DO A GET DUST STATUS TO FIND IF LOCAL DUP MEDIA
: 3809 2 IF .CST_ADDR [.DUOFF + OF_DBN, NODUPMEDIA] EQL 1 THEN RETURN; ! IF DUP LOCAL MEDIA NOT THERE THEN RETURN
: 3810 2
: 3811 2 TEMP = .CST_ADDR [.DUOFF + OF_DBN, D_DBN];
: 3812 2 INCR DBNCNT FROM (.TEMP + 1) TO (.TEMP + .dupround) DO ! INCREMENT FROM RELATIVE DBN TO DBN + dupro
und
: 3813 3 BEGIN
: 3814 3 IF .CST_ADDR [.DUOFF + OF_DBN, DUPWRITE] ! IF WRITE FLAG SET IN CST TABLE THE
N WRITE DBN'S
: 3815 3 THEN
: 3816 4 BEGIN
: 3817 4 DUP_IDLE (); ! MAKE SURE THE CONTROLLER IS IN AN IDLE STA
: 3818 4 DUP_WRTDBN (); ! CALL ROUTINE TO HANDLE WRITTING ROUTINES
: 3819 3 END;
: 3820 3
: 3821 3 DUP_IDLE (); ! MAKE SURE CONTROLLER IN IDLE STATE
: 3822 3 DUP_READDBN (); ! CALL ROUTINE TO HANDLE READING DBN'S
: 3823 3
: 3824 3 .CST_ADDR [.DUOFF + OF_DBN, D_DBN] = .CST_ADDR [.DUOFF + OF_DBN, D_DBN] + 1; ! INCREMENT RELATIVE DBN COUNTER
: 3825 3
: 3826 3 IF .CST_ADDR [.DUOFF + OF_DBN, DUPERROR] EQL 1 ! ERROR IN DUP REINITIALIZE
: 3827 3 THEN RETURN; ! AND RETURN
: 3828 2 END;
: 3829 1 END;

```

001204
001204TEMP: .PSECT \$GGG\$, RO
.BLKW 1

ZRGAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0332
Page 101
(26)

```

013262      .SBTTL  DUP MULTI-DRIVE TEST ROUTINES
            .PSECT $CODE$, RO

000000 004137 000000G      DUP:: JSR      R1,$SAVE3      ;      3746
000004 013700 000000G      1$:  MOV      RP,ADDR,R0      ;      3792
000010 023760 000000G 000004      CMP      CRN.LOW,4(R0)      ;
000016 001433      BEQ      3$
000020 123727 000000G 000001      CMPB     EOP.FLAG,$1      ;      3793
000026 001427      BEQ      3$
000030 105002      TRAP     22      ;      3794
000032 004137 000000V      JSR      PC,PROC.RETPKT      ;      3796
000036 005237 000000G      INC      RP,INDX      ;      3797
000042 023727 000000G 000010      CMP      RP,INDX,$10      ;      3798
000050 002402      BLT      2$
000052 005037 000000G      CLR      RP,INDX      ;
000056 013746 000000G      2$:  MOV      RP,INDX,-(SP)      ;      3799
000062 012746 000054      MOV      $54,-(SP)
000066 004737 000000G      JSR      PC,BL$MUL
000072 062700 000000G      ADD      $RETPKT,R0
000076 010037 000000G      MOV      R0,RP,ADDR
000102 022626      CMP      (SP)+,(SP)+      ;      3794
000104 000737      BR       1$      ;      3792
000106 013737 000002G 000000G      3$:  MOV      RANDOM+2,S.PATTERN      ;      3803
000114 013700 001156'      MOV      DUOFF,R0      ;      3805
000120 006300      ASL      R0
000122 063700 000000G      ADD      CST,ADDR,R0
000126 005001      CLR      R1
000130 156001 000020      BISB     20(R0),R1
000134 063701 000000G      ADD      DUPROUND,R1
000140 020127 000220      CMP      R1,$220
000144 002402      BLT      4$
000146 105060 000020      CLRB     20(R0)      ;      3806
000152 004737 000000V      4$:  JSR      PC,DUPIDLE      ;      3808
000156 013700 001156'      MOV      DUOFF,R0      ;      3809
000162 006300      ASL      R0
000164 063700 000000G      ADD      CST,ADDR,R0
000170 005760 000020      TST      20(R0)
000174 100456      BMI      8$
000176 116037 000020 001204'      MOVB     20(R0),TEMP      ;      3811
000204 105037 001205'      CLRB     TEMP+1
000210 013703 001204'      MOV      TEMP,R3      ;      3812
000214 063703 000000G      ADD      DUPROUND,R3
000220 013700 001156'      MOV      DUOFF,R0      ;      3814
000224 006300      ASL      R0
000226 063700 000000G      ADD      CST,ADDR,R0
000232 010001      MOV      R0,R1
000234 062701 000020      ADD      $20,R1
000240 013702 001204'      MOV      TEMP,R2      ; * ,DBNCNT      3812
000244 000427      BR       7$
000246 032711 010000      5$:  BIT      $10000,(R1)      ;      3814
000252 001404      BEQ      6$
000254 004737 000000V      JSR      PC,DUPIDLE      ;      3817

```

J10

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0333
Page 102
(26)

000260	004737	000000V		JSR	PC,DUPWRIDBN	:	3818
000264	004737	000000V	6\$:	JSR	PC,DUPIDLE	:	3821
000270	004737	000000V		JSR	PC,DUPREDDBN	:	3822
000274	013700	001156'		MOV	DUOFF,R0	:	3824
000300	006300			ASL	R0		
000302	063700	000000G		ADD	CST,ADDR,R0		
000306	010001			MOV	R0,R1		
000310	062701	000020		ADD	#20,R1		
000314	105211			INCB	(R1)		
000316	032711	040000		BIT	#40000,(R1)	:	3826
000322	001003			BNE	8\$	:	3827
000324	005202		7\$:	INC	R2	: DBNCNT	3812
000326	020203			CMP	R2,R3	: DBNCNT,4	
000330	003746			BLE	5\$		
000332	000207		8\$:	RTS	PC	:	3746

; Routine Size: 110 words, Routine Base: \$CODE\$ + 13262
; Maximum stack depth per invocation: 7 words

; 3830 1

K10

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0334
Page 103
(27)

```

: 3831 1 GLOBAL ROUTINE DUPWRTDBN : NOVALUE *
: 3832 1
: 3833 1 !+
: 3834 1 ! THIS ROUTINE IS CALLED BY DUP ROUTINE TO USE THE CONTROLLER LOCAL PROGRAM
: 3835 1 ! "WRTDBN". TO USE THE PROGRAM THE OPTIONAL DUP SUB-PROTOCOL IS USED TO
: 3836 1 ! COMMUNICATE WITH THE CONTROLLER. THE PROGRAM WRITES TO A DIAGNOSTIC BLOCK (DBN)
: 3837 1 ! THE WORD IN "S_PATTERN" IS WRITTEN TO THE 256 WORDS IN THE DBN. IF AN ERROR OCCURS
: 3838 1 ! WHILE RUNNING THE CONTROLLER LOCAL PROGRAM THE ERROR IS USUALLY REPORTED IN THE
: 3839 1 ! DUP BUFFER. (EX. ILLEGAL UNIT NUMBER, ILLEGAL BLK #, DEVICE ERROR, ZERO LENGHT MSG)
: 3840 1 !
: 3841 1 ! IMPLICIT INPUTS;
: 3842 1 ! CST_ADDRS - CONTAINS THE CURRENT CONTROLLER STATUS TABLE
: 3843 1 ! DUOFF - CURRENT OFFSET IN CST TABLE FOR PARTICULAR DRIVE
: 3844 1 ! S_PATTERN - CONTAINS PATTERN WORD!-
: 3845 2 BEGIN
: 3846 2 !PRINTX (DER11);
: 3847 2 T_ADDR [T_DBN_WT] = .T_ADDR [T_DBN_WT] + 1; ! INCREMENT # OF WRITES GIVEN
: 3848 2
: 3849 2 MSCP_PKT [.MX1, MSGLEN] = SZ_ELP; ! PACKET SIZE EXECUTE LOCAL PROGRAM WRT DB
N
: 3850 2 MSCP_PKT [.MX1, OPCODE] = OP_ELP; ! OPCODE = EXECUTE LOCAL PROGRAM
: 3851 2 MSCP_PKT [.MX1, L1] = *ascii'W'; ! FILL IN PROGRAM NAME WITH ASCII LETTERS
: 3852 2 MSCP_PKT [.MX1, L2] = *ascii'R';
: 3853 2 MSCP_PKT [.MX1, L3] = *ascii'T';
: 3854 2 MSCP_PKT [.MX1, L4] = *ascii'D';
: 3855 2 MSCP_PKT [.MX1, L5] = *ascii'B';
: 3856 2 MSCP_PKT [.MX1, L6] = *ascii'N';
: 3857 2 MSCP_PKT [.MX1, MODIFY] = 1; ! STANDALONE MODIFIER
: 3858 2 MSCP_PKT [.MX1, MSGTYP] = IMM_CMD; ! CALL IT IMMEDIATE
: 3859 2 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 3860 2
: 3861 2 IF .CST_ADDR [.DUOFF + OF_DBN, DUPERROR] EQL 1 !status error
: 3862 2 THEN RETURN; ! AND RETURN
: 3863 2
: 3864 3 DO (MX1 = GET_PKT (.CCTLR))
: 3865 2 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3866 2
: 3867 2 MSCP_PKT [.MX1, MSGLEN] = SZ_REC; ! PACKET SIZE RECIEVE DATA
: 3868 2 MSCP_PKT [.MX1, OPCODE] = OP_RCD; ! OPCODE = RECEIVE DATA
: 3869 2 MSCP_PKT [.MX1, BC_LO] = 80; ! BYTE COUNT TO BE TRANSFERED EQUALS 2 ***see pg 26 of DUP s
pec
: 3870 2 MSCP_PKT [.MX1, BUF_0] = DUPPKT; ! LOAD DESCRIPTOR BUFFER
: 3871 2 MSCP_PKT [.MX1, MODIFY] = 0; !
: 3872 2 MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD; ! CALL IT sequential
: 3873 2 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 3874 2
: 3875 2 IF (.CST_ADDR [.DUOFF + OF_DBN, DUPERROR] EQL 1) OR !status error
: 3876 2 (.DUPPKT [DUPTYPE] NEQU 1) OR !dup type error
: 3877 3 (.DUPPKT [DUPMSG] NEQU 6)
: 3878 2 THEN
: 3879 3 (HARD_ERROR ());
: 3880 3 CST_ADDR [.DUOFF + OF_DBN, DUPERROR] = 1; ! SET FLAG
: 3881 2 RETURN; ! NO POINT IN CONTINUING
: 3882 2
: 3883 3 DO (MX1 = GET_PKT (.CCTLR))

```

L10

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0335
Page 104
(27)

```

; 3884 2      UNTIL (.MX1 GEQ 0);                                ! TRY TO GET AN ENVELOPE, IF FAILURE LOOP PRG ERROR
; 3885 2
; 3886 2      MSCP_PKT [.MX1, MSGLEN] = SZ_SEN;                    ! PACKET SIZE                                SEND DATA
; 3887 2      MSCP_PKT [.MX1, OPCODE] = OP_SDD;                    ! OPCODE = SEND DATA
; 3888 2      MSCP_PKT [.MX1, BC_LO] = 6;                          ! BYTE COUNT TO BE TRANSFERED EQUALS 6
; 3889 2      MSCP_PKT [.MX1, BUF_0] = DUPPKT;                     ! LOAD DESCRIPTOR BUFFER
; 3890 2      DUPPKT [DUPBF0] = .CST_ADDR [.DUOFF, D_DISK_NUM];    ! LOAD UNIT NUMBER (RDRX)
; 3891 2      DUPPKT [DUPBF1] = .CST_ADDR [.DUOFF + OF_DBN, D_DBN]; ! LOAD DBN NUMBER
; 3892 2      DUPPKT [DUPBF2] = .S_PATTERN;                        ! LOAD PATTERN
; 3893 2      MSCP_PKT [.MX1, MODIFY] = 0;                          !
; 3894 2      MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD;                    ! CALL IT sequential
; 3895 2      DUPCOMMAND ();                                         ! SENDS AND RECEIVES THE COMMAND
; 3896 2
; 3897 2      IF .CST_ADDR [.DUOFF + OF_DBN, DUPERROR] EQL 1      ! status error
; 3898 2      THEN RETURN;
; 3899 2
; 3900 3      DO (MX1 = GET_PKT (.CCTLR))
; 3901 2      UNTIL (.MX1 GEQ 0);                                ! TRY TO GET AN ENVELOPE, IF FAILURE LOOP PRG ERROR
; 3902 2
; 3903 2      MSCP_PKT [.MX1, MSGLEN] = SZ_REC;                    ! PACKET SIZE                                RECEIVE DATA
; 3904 2      MSCP_PKT [.MX1, OPCODE] = OP_RCD;                    ! OPCODE = RECEIVE DATA
; 3905 2      MSCP_PKT [.MX1, BC_LO] = 4;                          ! BYTE COUNT TO BE TRANSFERED EQUALS 4
; 3906 2      MSCP_PKT [.MX1, BUF_0] = DUPPKT;                     ! LOAD DESCRIPTOR BUFFER
; 3907 2      MSCP_PKT [.MX1, MODIFY] = 0;                          !
; 3908 2      MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD;                    ! CALL IT sequential
; 3909 2      DUPCOMMAND ();                                         ! SENDS AND RECEIVES THE COMMAND
; 3910 2
; 3911 2      IF (.CST_ADDR [.DUOFF + OF_DBN, DUPERROR] EQL 1) or  !status error
; 3912 2      (.DUPPKT [DUPTYPE] NEQU 3) or                        !dup type error
; 3913 2      (.DUPPKT [DUPMSG] NEQU 3) or
; 3914 3      (.DUPPKT [DUPBF1] NEQU 0)                             !non successful write code
; 3915 2      THEN
; 3916 3      (HARD_ERROR ();
; 3917 3      CST_ADDR [.DUOFF + OF_DBN, DUPERROR] = 1;           ! SET FLAG
; 3918 2      RETURN;);                                             ! NO POINT IN CONTINUING
; 3919 2
; 3920 3      DO (MX1 = GET_PKT (.CCTLR))
; 3921 2      UNTIL (.MX1 GEQ 0);                                ! TRY TO GET AN ENVELOPE.
; 3922 2
; 3923 2      T_ADDR [T_BLK_WT] = .T_ADDR [T_BLK_WT] + 1;        !INCREMENT COUNTER IF A SUCCESS
; 3924 2
; 3925 1      END;

```

000000	010146		.SBTTL	DUPWRTDBN MULTI-DRIVE TEST ROUTINES	
			DUPWRTDBN:		
000002	013700	000000G	MOV	R1, (SP)	3831
000006	005260	000060	MOV	T_ADDR, R0	384
000012	013746	000106	INC	60(R0)	
000016	012746	000106	MOV	MX1, -(SP)	3849
000022	004737	000000G	MOV	#106, -(SP)	
000026	012760	000022 000006G	JSR	PC, BL\$MUL	
			MOV	#22, MSCP_PKT+6(R0)	

M10

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0336
Page 105
(27)

000034	112760	000003	000022G	MOVB	#3,MSCP.PKT+22(R0)	:	3850
000042	112760	000127	000026G	MOVB	#127,MSCP.PKT+26(R0)	:	3851
000050	112760	000122	000027G	MOVB	#122,MSCP.PKT+27(R0)	:	3852
000056	112760	000124	000030G	MOVB	#124,MSCP.PKT+30(R0)	:	3853
000064	112760	000104	000031G	MOVB	#104,MSCP.PKT+31(R0)	:	3854
000072	112760	000102	000032G	MOVB	#102,MSCP.PKT+32(R0)	:	3855
000100	112760	000116	000033G	MOVB	#116,MSCP.PKT+33(R0)	:	3856
000106	012760	000001	000024G	MOV	#1,MSCP.PKT+24(R0)	:	3857
000114	142760	000360	000010G	BICB	#360,MSCP.PKT+10(R0)	:	3858
000122	004737	000000V		JSR	PC,DUPCOMMAND	:	3859
000126	013700	001156'		MOV	DUOFF,R0	:	3861
000132	006300			ASL	R0	:	
000134	063700	000000G		ADD	CST.ADDR,R0	:	
000140	032760	040000	000020	BIT	#40000,20(R0)	:	
000146	001402			BEQ	1\$	:	
000150	022626			CMP	(SP)+,(SP)+	:	3831
000152	000505			BR	3\$	:	3862
000154	013716	000000G	1\$:	MOV	CCTLR,(SP)	:	3864
000160	004737	000000G		JSR	PC,GET.PKT	:	
000164	010037	000106'		MOV	R0,MX1	:	
000170	002771			BLT	1\$	:	3865
000172	010016			MOV	R0,(SP)	:	3867
000174	012746	000106		MOV	#106,-(SP)	:	
000200	004737	000000G		JSR	PC,BL\$MUL	:	
000204	012760	000034	000006G	MOV	#34,MSCP.PKT+6(R0)	:	
000212	112760	000005	000022G	MOVB	#5,MSCP.PKT+22(R0)	:	3868
000220	012760	000120	000026G	MOV	#120,MSCP.PKT+26(R0)	:	3869
000226	012760	000000G	000032G	MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3870
000234	005060	000024G		CLR	MSCP.PKT+24(R0)	:	3871
000240	142760	000360	000010G	BICB	#360,MSCP.PKT+10(R0)	:	3872
000246	152760	000020	000010G	BISB	#20,MSCP.PKT+10(R0)	:	
000254	004737	000000V		JSR	PC,DUPCOMMAND	:	3873
000260	013700	001156'		MOV	DUOFF,R0	:	3875
000264	006300			ASL	R0	:	
000266	063700	000000G		ADD	CST.ADDR,R0	:	
000272	032760	040000	000020	BIT	#40000,20(R0)	:	
000300	001016			BNE	2\$	:	
000302	013700	000000G		MOV	DUPPKT,R0	:	3876
000306	042700	007777		BIC	#7777,R0	:	
000312	020027	010000		CMP	R0,#10000	:	
000316	001007			BNE	2\$	:	
000320	013700	000000G		MOV	DUPPKT,R0	:	3877
000324	042700	170000		BIC	#170000,R0	:	
000330	020027	000006		CMP	R0,#6	:	
000334	001415			BEQ	4\$	:	
000336	004737	000000V	2\$:	JSR	PC,HARD.ERROR	:	3879
000342	013700	001156'		MOV	DUOFF,R0	:	3880
000346	006300			ASL	R0	:	
000350	063700	000000G		ADD	CST.ADDR,R0	:	
000354	052760	040000	000020	BIS	#40000,20(R0)	:	
000362	062706	000006		ADD	#6,SP	:	3881
000366	000504		3\$:	BR	5\$	:	3879
000370	013716	000000G	4\$:	MOV	CCTLR,(SP)	:	3883

N1.0

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0337
Page 106
(27)

000374	004737	000000G		JSR	PC,GET.PKT		
000400	010037	000106'		MOV	R0,MX1		
000404	002771			BLT	4\$	:	3884
000406	010016			MOV	R0,(SP)	:	3886
000410	012746	000106		MOV	#106,-(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)	:	3887
000434	012760	000006	000026G	MOV	#6,MSCP.PKT+26(R0)	:	3888
000442	012760	000000G	000032G	MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3889
000450	013701	001156'		MOV	DUOFF,R1	:	3890
000454	006301			ASL	R1		
000456	063701	000000G		ADD	CST,ADDR,R1		
000462	111137	000000G		MOVB	(R1),DUPPKT		
000466	042737	177760	000000G	BIC	#177760,DUPPKT		
000474	013701	001156'		MOV	DUOFF,R1	:	3891
000500	006301			ASL	R1		
000502	063701	000000G		ADD	CST,ADDR,R1		
000506	116137	000020	000002G	MOVB	20(R1),DUPPKT+2		
000514	105037	000003G		CLRB	DUPPKT+3		
000520	013737	000000G	000004G	MOV	S,PATTERN,DUPPKT+4	:	3892
000526	005060	000024G		CLR	MSCP.PKT+24(R0)	:	3893
000532	142760	000360	000010G	BICB	#360,MSCP.PKT+10(R0)	:	3894
000540	152760	000020	000010G	BISB	#20,MSCP.PKT+10(R0)		
000546	004737	000000V		JSR	PC,DUPCOMMAND	:	3895
000552	013700	001156'		MOV	DUOFF,R0	:	3897
000556	006300			ASL	R0		
000560	063700	000000G		ADD	CST,ADDR,R0		
000564	032760	040000	000020	BIT	#40000,20(R0)		
000572	001403			BEQ	6\$		
000574	062706	000010		ADD	#10,SP	:	3831
000600	000524			BR	10\$	:	3898
000602	013716	000000G		MOV	CCTLR,(SP)	:	3900
000606	004737	000000G		JSR	PC,GET.PKT		
000612	010037	000106'		MOV	R0,MX1		
000616	002771			BLT	6\$	:	3901
000620	010016			MOV	R0,(SP)	:	3903
000622	012746	000106		MOV	#106,-(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)	:	3904
000646	012760	000004	000026G	MOV	#4,MSCP.PKT+26(R0)	:	3905
000654	012760	000000G	000032G	MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3906
000662	005060	000024G		CLR	MSCP.PKT+24(R0)	:	3907
000666	142760	000360	000010G	BICB	#360,MSCP.PKT+10(R0)	:	3908
000674	152760	000020	000010G	BISB	#20,MSCP.PKT+10(R0)		
000702	004737	000000V		JSR	PC,DUPCOMMAND	:	3909
000706	013700	001156'		MOV	DUOFF,R0	:	3911
000712	006300			ASL	R0		
000714	063700	000000G		ADD	CST,ADDR,R0		
000720	032760	040000	000020	BIT	#40000,20(R0)		
000726	001021			BNE	7\$		
000730	013700	000000G		MOV	DUPPKT,R0	:	3912

5\$:
6\$:

1111

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11:16 V4.0 579
DISK1USER2:[POWERS]ZRQADO.BL216

SEQ 0338
Page 107
(27)

000734	042700	007777	BIC	07777,R0		
000740	020027	030000	CMF	R0,030000		
000744	001012		BNE	78		
000746	013700	000000G	MOV	DUPPKT,R0		3913
000752	042700	170000	BIC	0170000,R0		
000756	020027	000003	CMF	R0,03		
000762	001003		BNE	78		
000764	005737	000002G	TST	DUPPKT.2		3914
000770	001413		BEQ	88		
000772	004737	000000V	JSR	PC,HARD.ERROR		3915
000776	013700	001156'	MOV	DUOFF,R0		3917
001002	006300		ASL	R0		
001004	063700	000000G	ADD	CST.ADDR,R0		
001010	052760	040000 000020	BIS	040000,20(R0)		
001016	000413		BR	98		3918
001020	013716	000000G	MOV	CCTLR,(SP)		3920
001024	004737	000000G	JSR	PC,GET.PKT		
001030	010037	000106'	MOV	R0,MX1		
001034	002771		BLT	88		3921
001036	013700	000000G	MOV	T.ADDR,R0		3923
001042	005260	000056	INC	56(R0)		
001046	062706	000012	ADD	012,SP		3845
001052	012601		MOV	(SP)+,R1		3831
001054	000207		RTS	PC		

: Routine Size: 279 words, Routine Base: \$CODE\$ + 13616
: Maximum stack depth per invocation: 7 words

C11

ZRGAMS
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11 Apr-1984 11:08:35
11 Apr-1984 11:08:22VAX-11 B11:16 V4.0 5/9
DISK1USER2:[POWER5]ZRGADO.BL2,6SEQ 0339
Page 108
(28)

```

: 3926 1 GLOBAL ROUTINE DUPREDDBN : NOVALUE =
: 3927 1
: 3928 1
: 3929 1
: 3930 1 ! THIS ROUTINE IS CALLED BY DUP ROUTINE TO USE THE CONTROLLER LOCAL PROGRAM
: 3931 1 ! "REDDBN". TO USE THE PROGRAM THE OPTIONAL DUP SUB-PROTOCOL IS USED TO
: 3932 1 ! COMMUNICATE WITH THE CONTROLLER. THE PROGRAM READS A DIAGNOSTIC BLOCK (DBN)
: 3933 1 ! AND PLACES IT IN THE DUP BUFFER CALLED "DUPPKT". IF AN ERROR OCCURS WHILE
: 3934 1 ! RUNNING THE CONTROLLER LOCAL PROGRAM THE ERROR IS USUALLY REPORTED IN THE
: 3935 1 ! DUP BUFFER. (EX, ILLEGAL UNIT NUMBER, ILLEGAL BLK #, DEVICE ERROR, ZERO LENGTH MSG)
: 3936 1
: 3937 1
: 3938 1 ! IMPLICIT INPUTS:
: 3939 1 ! CST_ADDR - CONTAINS THE CURRENT CONTROLLER STATUS TABLE
: 3940 1 ! DUOFF - CURRENT OFFSET IN CST TABLE FOR PARTICULAR DRIVE
: 3941 2 BEGIN
: 3942 2 !PRINTX (DER12);
: 3943 2 T_ADDR [T_DBN_RD] = T_ADDR [T_DBN_RD] + 1; ! INCREMENT # OF READS GIVEN
: 3944 2
: 3945 2 MSCP_PKT [,MX1, MSGLEN] = SZ_ELP; ! PACKET SIZE EXECUTE REDDBN PROGRAM
: 3946 2 MSCP_PKT [,MX1, OPCODE] = OP_ELP; ! OPCODE = EXECUTE LOCAL PROGRAM
: 3947 2 MSCP_PKT [,MX1, L1] = *asci['R']; ! FILL IN PROGRAM NAME WITH ASCII LETTERS
: 3948 2 MSCP_PKT [,MX1, L2] = *asci['E'];
: 3949 2 MSCP_PKT [,MX1, L3] = *asci['D'];
: 3950 2 MSCP_PKT [,MX1, L4] = *asci['D'];
: 3951 2 MSCP_PKT [,MX1, L5] = *asci['B'];
: 3952 2 MSCP_PKT [,MX1, L6] = *asci['N'];
: 3953 2 MSCP_PKT [,MX1, MODIFY] = 1; ! STANDALONE MODIFIER
: 3954 2 MSCP_PKT [,MX1, MSGTYP] = IMM_CMD; ! CALL IT IMMEDIATE
: 3955 2 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 3956 2
: 3957 2 IF (.CST_ADDR [,DUOFF + OF_DBN, DUPERROR] EQL 1) !status error
: 3958 2 THEN RETURN;
: 3959 2
: 3960 3 DO (MX1 = GET_PKT (,CCTLR))
: 3961 2 UNTIL (,MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3962 2
: 3963 2 MSCP_PKT [,MX1, MSGLEN] = SZ_REC; ! PACKET SIZE RECEIVE DATA
: 3964 2 MSCP_PKT [,MX1, OPCODE] = OP_RCD; ! OPCODE = RECEIVE DATA
: 3965 2 MSCP_PKT [,MX1, BC_LO] = 80; ! BYTE COUNT TO BE TRANSFERED EQUALS 2 *****see pg 26 DUP sp
ec
: 3966 2 MSCP_PKT [,MX1, BUF_0] = DUPPKT; ! LOAD DESCRIPTOR BUFFER
: 3967 2 MSCP_PKT [,MX1, MODIFY] = 0;
: 3968 2 MSCP_PKT [,MX1, MSGTYP] = SEQ_CMD; ! CALL IT sequential
: 3969 2 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 3970 2
: 3971 2 IF (.CST_ADDR [,DUOFF + OF_DBN, DUPERROR] EQL 1) OR !status error
: 3972 2 (.DUPPKT [DUPTYPE] NEQU 1) OR !dup type error
: 3973 2 (.DUPPKT [DUPMSG] NEQU 5)
: 3974 2 THEN
: 3975 3 (HARD_ERROR ());
: 3976 3 CST_ADDR [,DUOFF + OF_DBN, DUPERROR] = 1; ! SET FLAG
: 3977 2 RETURN; ! NO POINT IN CONTINUING
: 3978 2

```

011

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11:16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2:5

SEQ 0340
Page 109
(28)

```

: 3979 3 DO (MX1 = GET_PKT (.CCTLR))
: 3980 2 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3981 2
: 3982 2 MSCP_PKT [.MX1, MSGLEN] = SZ_SEN; ! PACKET SIZE SEND DATA
: 3983 2 MSCP_PKT [.MX1, OPCODE] = OP_SDD; ! OPCODE = SEND DATA
: 3984 2 MSCP_PKT [.MX1, BC_LO] = 4; ! BYTE COUNT TO BE TRANSFERED EQUALS 4
: 3985 2 MSCP_PKT [.MX1, BUF_0] = DUPPKT; ! LOAD DESCRIPTOR BUFFER
: 3986 2 DUPPKT [DUPBF0] = .CST_ADDR [.DUOFF, D_DISK_NUM]; ! LOAD UNIT NUMBER (RDRX)
: 3987 2 DUPPKT [DUPBF1] = .CST_ADDR [.DUOFF + OF_DBN, D_DBN]; ! LOAD DBN NUMBER
: 3988 2 MSCP_PKT [.MX1, MODIFY] = 0; !
: 3989 2 MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD; ! CALL IT sequential
: 3990 2 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 3991 2
: 3992 2 IF .CST_ADDR [.DUOFF + OF_DBN, DUPERROR] EQL 1 !status error
: 3993 2 THEN RETURN;
: 3994 2
: 3995 3 DO (MX1 = GET_PKT (.CCTLR))
: 3996 2 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 3997 2
: 3998 2 MSCP_PKT [.MX1, MSGLEN] = SZ_REC; ! PACKET SIZE RECEIVE DATA
: 3999 2 MSCP_PKT [.MX1, OPCODE] = OP_RCD; ! OPCODE = GET DUST STATUS
: 4000 2 MSCP_PKT [.MX1, BC_LO] = 512; ! BYTE COUNT TO BE TRANSFERED EQUALS 512
: 4001 2 MSCP_PKT [.MX1, BUF_0] = DUPPKT; ! LOAD DESCRIPTOR BUFFER
: 4002 2 MSCP_PKT [.MX1, MODIFY] = 0; !
: 4003 2 MSCP_PKT [.MX1, MSGTYP] = SEQ_CMD; ! CALL IT sequential
: 4004 2 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 4005 2
: 4006 2 IF (.CST_ADDR [.DUOFF + OF_DBN, DUPERROR] EQL 1) OR !status error
: 4007 2 (.DUPPKT [DUPTYPE] NEQU 6) OR !dup type error
: 4008 3 (.DUPPKT [DUPMSG] NEQU 2)
: 4009 2 THEN
: 4010 3 (HARD_ERROR ());
: 4011 3 CST_ADDR [.DUOFF + OF_DBN, DUPERROR] = 1; ! SET FLAG
: 4012 2 RETURN;); ! NO POINT IN CONTINUING
: 4013 2
: 4014 3 DO (MX1 = GET_PKT (.CCTLR))
: 4015 2 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRG ERROR
: 4016 2
: 4017 2 T_ADDR [T_BLK_RD] = .T_ADDR [T_BLK_RD] + 1; !IF DUP NO ERROR THEN INCREMENT COUNTER
: 4018 2
: 4019 1 END;

```

000000	010146		.SBTTL	DUPREDDBN MULTI-DRIVE TEST ROUTINES	
			DUPREDDBN::		
000002	013700	000000G	MOV	R1, (SP)	3920
000006	005260	000064	MOV	T_ADDR, R0	3943
000012	013746	000106	INC	64(R0)	
000016	012746	000106	MOV	MX1, -(SP)	3945
000022	004737	000000G	MOV	0106, -(SP)	
000026	012760	000022	JSR	PC, BL\$MUL	
000034	112760	000003	MOV	022, MSCP_PKT+6(R0)	
		000022G	MOV	03, MSCP_PKT+22(R0)	3940

E11

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0341
Page 110
(28)

000042	112760	000122	000026G	MOVB	#122,MSCP,PKT+26(R0)	:	3947
000050	112760	000105	000027G	MOVB	#105,MSCP,PKT+27(R0)	:	3948
000056	112760	000104	000030G	MOVB	#104,MSCP,PKT+30(R0)	:	3949
000064	112760	000104	000031G	MOVB	#104,MSCP,PKT+31(R0)	:	3950
000072	112760	000102	000032G	MOVB	#102,MSCP,PKT+32(R0)	:	3951
000100	112760	000116	000033G	MOVB	#116,MSCP,PKT+33(R0)	:	3952
000106	012760	000001	000024G	MOV	#1,MSCP,PKT+24(R0)	:	3953
000114	142760	000360	000010G	BICB	#360,MSCP,PKT+10(R0)	:	3954
000122	004737	000000V		JSR	PC,DUPCOMMAND	:	3955
000126	013700	001156'		MOV	DUOFF,R0	:	3957
000132	006300			ASL	R0	:	
000134	063700	000000G		ADD	CST,ADDR,R0	:	
000140	032760	040000	000020	BIT	#40000,20(R0)	:	
000146	001402			BEQ	1\$	:	
000150	022626			CMP	(SP)+,(SP)+	:	3926
000152	000505			BR	3\$	:	3958
000154	013716	000000G	1\$:	MOV	CCTLR,(SP)	:	3960
000160	004737	000000G		JSR	PC,GET,PKT	:	
000164	010037	000106'		MOV	R0,MX1	:	
000170	002771			BLT	1\$	:	3961
000172	010016			MOV	R0,(SP)	:	3963
000174	012746	000106		MOV	#106,-(SP)	:	
000200	004737	000000G		JSR	PC,BL\$MUL	:	
000204	012760	000034	000006G	MOV	#34,MSCP,PKT+6(R0)	:	
000212	112760	000005	000022G	MOVB	#5,MSCP,PKT+22(R0)	:	3964
000220	012760	000120	000026G	MOV	#120,MSCP,PKT+26(R0)	:	3965
000226	012760	000000G	000032G	MOV	#DUPPKT,MSCP,PKT+32(R0)	:	3966
000234	005060	000024G		CLR	MSCP,PKT+24(R0)	:	3967
000240	142760	000360	000010G	BICB	#360,MSCP,PKT+10(R0)	:	3968
000246	152760	000020	000010G	BISB	#20,MSCP,PKT+10(R0)	:	
000254	004737	000000V		JSR	PC,DUPCOMMAND	:	3969
000260	013700	001156'		MOV	DUOFF,R0	:	3971
000264	006300			ASL	R0	:	
000266	063700	000000G		ADD	CST,ADDR,R0	:	
000272	032760	040000	000020	BIT	#40000,20(R0)	:	
000300	001016			BNE	2\$	:	
000302	013700	000000G		MOV	DUPPKT,R0	:	3972
000306	042700	007777		BIC	#7777,R0	:	
000312	020027	010000		CMP	R0,#10000	:	
000316	001007			BNE	2\$	:	
000320	013700	000000G		MOV	DUPPKT,R0	:	3973
000324	042700	170000		BIC	#170000,R0	:	
000330	020027	000005		CMP	R0,#5	:	
000334	001415			BEQ	4\$	:	
000336	004737	000000V	2\$:	JSR	PC,HARD,ERROR	:	3975
000342	013700	001156'		MOV	DUOFF,R0	:	3976
000346	006300			ASL	R0	:	
000350	063700	000000G		ADD	CST,ADDR,R0	:	
000354	052760	040000	000020	BIS	#40000,20(R0)	:	
000362	062706	000006		ADD	#6,SP	:	3977
000366	000501		3\$:	BR	5\$	:	3978
000370	013716	000000G	4\$:	MOV	CCTLR,(SP)	:	3979
000374	004737	000000G		JSR	PC,GET,PKT	:	

F11

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0342
Page 111
(28)

000400	010037	000106'	MOV	R0,MX1		
000404	002771		BLT	4\$	:	3980
000406	010016		MOV	R0,(SP)	:	3982
000410	012746	000106	MOV	#106,-(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)	:	3983
000434	012760	000004 000026G	MOV	#4,MSCP.PKT+26(R0)	:	3984
000442	012760	000000G 000032G	MOV	#DUPPKT,MSCP.PKT+32(R0)	:	3985
000450	013701	001156'	MOV	DUOFF,R1	:	3986
000454	006301		ASL	R1		
000456	063701	000000G	ADD	CST.ADDR,R1		
000462	111137	000000G	MOVB	(R1),DUPPKT		
000466	042737	177760 000000G	BIC	#177760,DUPPKT		
000474	013701	001156'	MOV	DUOFF,R1	:	3987
000500	006301		ASL	R1		
000502	063701	000000G	ADD	CST.ADDR,R1		
000506	116137	000020 000002G	MOVB	20(R1),DUPPKT+2		
000514	105037	000003G	CLRB	DUPPKT+3		
000520	005060	000024G	CLR	MSCP.PKT+24(R0)	:	3988
000524	142760	000360 000010G	BICB	#360,MSCP.PKT+10(R0)	:	3989
000532	152760	000020 000010G	BISB	#20,MSCP.PKT+10(R0)		
000540	004737	000000V	JSR	PC,DUPCOMMAND	:	3990
000544	013700	001156'	MOV	DUOFF,R0	:	3992
000550	006300		ASL	R0		
000552	063700	000000G	ADD	CST.ADDR,R0		
000556	032760	040000 000020	BIT	#40000,20(R0)		
000564	001403		BEQ	6\$		
000566	062706	000010	ADD	#10,SP	:	3926
000572	000521		BR	10\$	:	3993
000574	013716	000000G	MOV	CCTLR,(SP)	:	3995
000600	004737	000000G	JSR	PC,GET.PKT		
000604	010037	000106'	MOV	R0,MX1		
000610	002771		BLT	6\$	:	3996
000612	010016		MOV	R0,(SP)	:	3998
000614	012746	000106	MOV	#106,-(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)	:	3999
000640	012760	001002 000026G	MOV	#1002,MSCP.PKT+26(R0)	:	4000
000646	012760	000000G 000032G	MOV	#DUPPKT,MSCP.PKT+32(R0)	:	4001
000654	005060	000024G	CLR	MSCP.PKT+24(R0)	:	4002
000660	142760	000360 000010G	BICB	#360,MSCP.PKT+10(R0)	:	4003
000666	152760	000020 000010G	BISB	#20,MSCP.PKT+10(R0)		
000674	004737	000000V	JSR	PC,DUPCOMMAND	:	4004
000700	013700	001156'	MOV	DUOFF,R0	:	4006
000704	006300		ASL	R0		
000706	063700	000000G	ADD	CST.ADDR,R0		
000712	032760	040000 000020	BIT	#40000,20(R0)		
000720	001016		BNE	7\$		
000722	013700	000000G	MOV	DUPPKT,R0	:	4007
000726	042700	007777	BIC	#7777,R0		
000732	020027	060000	CMP	R0,#60000		

G11

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0343
Page 112
(28)

000736	001007		BNE	7\$		
000740	013700	000000G	MOV	DUPPKT,R0	:	4008
000744	042700	170000	BIC	#170000,R0		
000750	020027	000002	CMP	R0,#2		
000754	001413		BEQ	8\$		
000756	004737	000000V	JSR	PC,HARD.ERROR	:	4010
000762	013700	001156'	MOV	DUOFF,R0	:	4011
000766	006300		ASL	R0		
000770	063700	000000G	ADD	CST.ADDR,R0		
000774	052760	040000 000020	BJS	#40000,20(R0)		
001002	000413		BR	9\$	:	4012
001004	013716	000000G	MOV	CCTLR,(SP)	:	4014
001010	004737	000000G	JSR	PC,GET.PKT		
001014	010037	000106'	MOV	R0,MX1		
001020	002771		BLT	8\$	:	4015
001022	013700	000000G	MOV	T.ADDR,R0	:	4017
001026	005260	000062	INC	62(R0)		
001032	062706	000012	ADD	#12,SP	:	3941
001036	012601		MOV	(SP)+,R1	:	3926
001040	000207		RTS	PC		

; Routine Size: 273 words, Routine Base: \$CODE\$ + 14674
 ; Maximum stack depth per invocation: 7 words

; 4020 1

H11

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0344
Page 113
(29)

```

: 4021 1
: 4022 1 GLOBAL ROUTINE DUPCOMMAND : NOVALUE =
: 4023 1
: 4024 1 !+
: 4025 1 ! THIS ROUTINE IS CALLED BY DUP TO PROCESS COMMANDS.
: 4026 1 ! THE COMMAND ENVELOPES ARE FILLED IN DUP ROUTINES IN THE "MX1" INDEX.
: 4027 1 ! WITH THE INDEX THIS ROUTINE SENDS THE COMMAND, WAITS FOR A
: 4028 1 ! RESPONSES AND THEN PROCESSES THE RETURN PACKET.
: 4029 1 !-
: 4030 2 BEGIN
: 4031 2 !PRINTX (DER13);
: 4032 2
: 4033 2 MSCP_PKT [ ,MX1, CREDITS] = 0; ! DUP DOES NOT USE THE CREDIT SYSTEM
: 4034 2 MSCF_PKT [ ,MX1, CONNID] = CID_DUP; ! MAKE PACKAGE EQUAL A DUP COMMAND
: 4035 2 MSCP_PKT [ ,MX1, DK_NUM] = 0; ! DISK NUMBER (NOT APPLICABLE)
: 4036 2
: 4037 2 IF SEND ( ,MX1) EQLU FAILURE ! ATTEMPT SEND; IF CTRL IS OFFLINE
: 4038 2 THEN
: 4039 3 BEGIN
: 4040 3 PUT_PKT ( ,MX1);
: 4041 3 MX1 = -1; ! RETURN ENVELOPE TO POOL
: 4042 3 CST_ADDR [ ,DUOFF + OF_DBN, DUPERROR] = 1;
: 4043 3 PRINTF (DBM112); ! "DUP: PKT NOT AVAILABLE"
: 4044 3 END
: 4045 3
: 4046 2 ELSE
: 4047 2 do
: 4048 3 begin
: 4049 3 BREAK; ! BREAK FOR ACT
: 4050 3 PROC_RET_PKT (); ! PROCESS RETURN PACKET TO SEE IF OK FOR DUP
: 4051 3 end
: 4052 2 until ( ,CRN_LOW eqLU ,RP_ADDR [CRF_LO]) or ! TO ENSURE THAT ALL RETURN MESSAGES HAVE BEEN PROCESSED
: 4053 2 ( ,EOP_FLAG eq true); ! or end of pass caused by error
: 4054 1 END;

```

			.SBTTL	DUPCOMMAND MULTI-DRIVE TEST ROUTINES	
000000	013746	000106'	DUPCOMMAND:		
			MOV	MX1, -(SP)	4033
000004	012746	000106	MOV	#106, -(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)	4034
000030	005060	000016G	CLR	MSCP.PKT+16(RO)	4035
000034	013716	000106'	MOV	MX1, (SP)	4037
000040	004737	000000G	JSR	PC, SEND	
000044	005700		TST	RO	
000046	001027		BNE	1\$	
000050	013716	000106'	MOV	MX1, (SP)	4040
000054	004737	000000G	JSR	PC, PUT_PKT	
000060	012737	177777 000106'	MOV	#-1, MX1	4041
000066	013700	001156'	MOV	DUOFF, RO	4042
000072	006300		ASL	RO	

111

ZRQAM3 RD/RX EXERCISER
V01.6 MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0345
Page 114
(29)

000074	063700	000000G		ADD	CST.ADDR,R0		
000100	052760	040000	000020	BIS	#40000,20(R0)		
000106	012716	000000G		MOV	#DBM112,(SP)		
000112	012746	000001		MOV	#1,-(SP)		4043
000116	010600			MOV	SP,R0	; SP,*	
000120	104417			TRAP	17		
000122	005726			TST	(SP)+		4039
000124	000415			BR	2\$		4037
000126	104422		1\$:	TRAP	22		4048
000130	004737	000000V		JSR	PC,PROC.RETPKT		4050
000134	013700	000000G		MOV	RP,ADDR,R0		4052
000140	023760	000000G	000004	CMP	CRN.LOW,4(R0)		
000146	001404			BEQ	2\$		
000150	123727	000000G	000001	CMPB	EOP.FLAG,#1		4053
000156	001363			BNE	1\$		
000160	022626		2\$:	CMP	(SP)+,(SP)+		4030
000162	000207			RTS	PC		4022

; Routine Size: 58 words. Routine Base: \$CODE\$ + 15736
; Maximum stack depth per invocation: 5 words

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0346
Page 115
(30)

```

: 4055 1
: 4056 1 GLOBAL ROUTINE DUPIDLE : NOVALUE *
: 4057 1 !*
: 4058 1 ! THIS ROUTINE IS CALLED BY DUP ROUTINE TO INSURE THAT THE CONTROLLER
: 4059 1 ! IS NOT IN A ACTIVE STATE. IF CALLED AND THE CONTROLLER IS IN AN ACTIVE
: 4060 1 ! STATE THE CONTROLLER WILL GIVE AN ABORT COMMAND WHICH SHOULD KILL THE
: 4061 1 ! CURRENT JOB OR LOCAL PROGRAM.
: 4062 1 !-
: 4063 2 BEGIN
: 4064 2 CST_ADDR [.DUOFF + OF_DBN, DUPERROR] = 0; !CLEAR DUP ERROR FLAG;
: 4065 2
: 4066 2 MSCP_PKT [.MX1, MSGLEN] = SZ_GDS; ! PACKET SIZE GET DUST STATUS
: 4067 2 MSCP_PKT [.MX1, OPCODE] = OP_GDS; ! OPCODE = GET DUST STATUS
: 4068 2 MSCP_PKT [.MX1, MODIFY] = 0; !
: 4069 2 MSCP_PKT [.MX1, MSGTYP] = IMM_CMD; ! CALL IT IMMEDIATE
: 4070 2 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 4071 2 ! GDS ONLY RETURNS SUCCESS or it don't return
: 4072 2
: 4073 3 DO (MX1 = GET_PKT (.CCTLR))
: 4074 2 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRGRAM ERROR
: 4075 2
: 4076 2 if .CST_ADDR [.DUOFF + OF_DBN, D_ACTIVE] neq IDLE ! if not in idle state then abort the program
: 4077 2 then
: 4078 3 begin
: 4079 3 MSCP_PKT [.MX1, MSGLEN] = SZ_ABT; ! PACKET SIZE ABORT CMD
: 4080 3 MSCP_PKT [.MX1, OPCODE] = OP_ABT; ! OPCODE = ABORT PROGRAM
: 4081 3 MSCP_PKT [.MX1, MODIFY] = 0; !
: 4082 3 MSCP_PKT [.MX1, MSGTYP] = IMM_CMD; ! CALL IT IMMEDIATE
: 4083 3 DUPCOMMAND (); ! SENDS AND RECEIVES THE COMMAND
: 4084 3 !ONLY ERROR IS already in idle state
: 4085 4 DO (MX1 = GET_PKT (.CCTLR))
: 4086 3 UNTIL (.MX1 GEQ 0); ! TRY TO GET AN ENVELOPE. IF FAILURE LOOP PRGRAM ERROR
: 4087 2 end;
: 4088 1 end;

```

		.SBTTL	DUPIDLE MULTI-DRIVE TEST ROUTINES	
000000	010146	DUPIDLE::		
000002	013700	001156'	MOV R1, -(SP)	4056
000006	006300		MOV DUOFF, R0	4064
000010	063700	000000G	ASL R0	
000014	042760	040000 000020	ADD CST_ADDR, R0	
000022	013746	000106'	BIC #40000, 20(R0)	
000026	012746	000106	MOV MX1, -(SP)	4066
000032	004737	000000G	MOV #106, -(SP)	
000036	012760	000014 000006G	JSR PC, BL\$MUL	
000044	112760	000001 000022G	MOV #14, MSCP_PKT+6(R0)	4067
000052	005060	000024G	MOVB #1, MSCP_PKT+22(R0)	4068
000056	142760	000360 000010G	CLR MSCP_PKT+24(R0)	4069
000064	004737	015736'	BICB #360, MSCP_PKT+10(R0)	4070
000070	013716	000000G	JSR PC, DUPCOMMAND	4071
000074	004737	000000G	1\$: MOV CCTLR, (SP)	4073
			JSR PC, GET_PKT	

K11

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0347
Page 116
(30)

000100	010037	000106'	MOV	R0,MX1		
000104	010001		MOV	R0,R1	; MX1,*	4074
000106	002770		BLT	1\$		
000110	013700	001156'	MOV	DUOFF,R0		4076
000114	006300		ASL	R0		
000116	063700	000000G	ADD	CST.ADDR,R0		
000122	032760	020000 000020	BIT	#20000,20(R0)		
000130	001432		BEQ	3\$		
000132	010116		MOV	R1,(SP)		4079
000134	012746	000106	MOV	#106,-(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)		4080
000160	005060	000024G	CLR	MSCP.PKT+24(R0)		4081
000164	142760	000360 000010G	BICB	#360,MSCP.PKT+10(R0)		4082
000172	004737	015736'	JSR	PC,DUPCOMMAND		4083
000176	013716	000000G	MOV	CCTLR,(SP)		4085
000202	004737	000000G	JSR	PC,GET.PKT		
000206	010037	000106'	MOV	R0,MX1		
000212	002771		BLT	2\$		4086
000214	005726		TST	(SP)+		4078
000216	022626	3\$:	CMP	(SP)+,(SP)+		4063
000220	012601		MOV	(SP)+,R1		4056
000222	000207		RTS	PC		

; Routine Size: 74 words, Routine Base: \$CODE\$ + 16122
; Maximum stack depth per invocation: 5 words

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0348
Page 117
(31)

```

: 4089 1 GLOBAL ROUTINE QIO_LBN : NOVALUE =
: 4090 1
: 4091 1 !+
: 4092 1 ! THIS ROUTINE IS CALLED BY QIO_GEN TO SELECT THE LOGICAL BLOCK NUMBER TO
: 4093 1 ! BE USED FOR THE CURRENT QIO OR QIO PAIR.
: 4094 1 !
: 4095 1 ! IF THE OPERATOR CHOSE THE RANDOM BLOCK MODE OPTION, THEN THE LBN IS
: 4096 1 ! RANDOMLY CHOSEN WITHIN THE SPECIFIED LIMITS FOR THE LBN.
: 4097 1 ! OTHERWISE, THE NEXT SEQUENTIAL LBN IS DERIVED FROM THE BLOCK SEQUENCE
: 4098 1 ! TABLE (BST).
: 4099 1 !
: 4100 1 ! IMPLICIT INPUTS:
: 4101 1 !     L$LUN - CURRENT (DIAGNOSTIC SUPERVISOR) UNIT NUMBER
: 4102 1 !
: 4103 1 ! IMPLICIT OUTPUTS:
: 4104 1 !     THE LBN IS LOADED INTO ONE OR BOTH MSCP PACKETS.
: 4105 1 !-
: 4106 1
: 4107 2 begin
: 4108 2
: 4109 2 local
: 4110 2     RD_DISK : byte;      ! FLAG TO INDICATE WINCHESTER DISK SELECTED
: 4111 2
: 4112 2
: 4113 2 if .CST_ADDR [.CUOFF, D_TYPE] eq1 RX_50
: 4114 2 then
: 4115 2     RD_DISK = FALSE
: 4116 2 else
: 4117 2     RD_DISK = TRUE;
: 4118 2
: 4119 2 MAD1 [LBN_L] = .BST [.L$LUN, LO_WRD];      ! LOAD LBN INTO 1ST PACKET
: 4120 2 MAD1 [LBN_H] = .BST [.L$LUN, HI_WRD];    ! LOAD LBN INTO 1ST PACKET
: 4121 2
: 4122 2 if .MX2 geq 0      ! IF 2 QIOS
: 4123 2 then
: 4124 3     begin
: 4125 3         MAD2 [LBN_L] = .BST [.L$LUN, LO_WRD];      ! LOAD LBN INTO 2ND PACKET
: 4126 3         MAD2 [LBN_H] = .BST [.L$LUN, HI_WRD];    ! LOAD LBN INTO 2ND PACKET
: 4127 3     end;
: 4128 2
: 4129 3 if BIT_TST (SWP_FLAGS, SWF_BLK)      ! IF RANDOM BLOCK MODE
: 4130 2 then
: 4131 3     begin
: 4132 4         if NOT ((.RD_DISK) and
: 4133 4             (((.RANDOM [0] and %o'077777') mod (99)) lequ 33))
: 4134 3     then
: 4135 4         begin
: 4136 4             if .CST_ADDR [.CUOFF + 2, D_BEG1] eq1u .CST_ADDR [.CUOFF + 4, D_END1]
: 4137 4             then
: 4138 4                 BST [.L$LUN, LO_WRD] = .CST_ADDR [.CUOFF + 1, D_BEG0] +
: 4139 4                     MODULAS (.CST_ADDR [.CUOFF + 1, D_BEG0], .CST_ADDR [.CUOFF + 3, D_END0])
: 4140 4             ! select low lbn from random
: 4141 4         else
: 4142 4             ! if upper word of beg trk and end t

```

M11

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0349
Page 118
(31)

```

: 4142 5      begin
: 4143 5      BST [.L$LUN, HI_WRD] = .CST_ADDR [.CUOFF + 2, D_BEG1] +      ! select upper lbn from window
: 4144 5      MODULAS (.CST_ADDR [.CUOFF + 2, D_BEG1], .CST_ADDR [.CUOFF + 4, D_END1]);
: 4145 5
: 4146 5      if .BST [.L$LUN, HI_WRD] eql .CST_ADDR [.CUOFF + 4, D_END1]      ! IF UPPER WORD EQUALS HI LIMIT BE S
URE LOWER
: 4147 5      then .BST [.L$LUN, LO_WRD] = MODULAS (0, .CST_ADDR [.CUOFF + 3, D_END0]); ! WORD DOES NOT PASS HI LIMIT
: 4148 5
: 4149 5      if .BST [.L$LUN, HI_WRD] eql .CST_ADDR [.CUOFF + 2, D_BEG1]      ! if upper word equal to limit make
sure lower
: 4150 5      then BST [.L$LUN, LO_WRD] = %0'177777' - MODULAS (.CST_ADDR [.CUOFF + 1, D_BEG0], %0'177777');
: 4151 5      ! word is above lo limit
: 4152 5
: 4153 5      if .BST [.L$LUN, HI_WRD] gtr .CST_ADDR [.CUOFF + 2, D_BEG1] and
: 4154 5      .BST [.L$LUN, HI_WRD] lss .CST_ADDR [.CUOFF + 4, D_END1]      ! if neither of the above then any n
umber is good
: 4155 5      then BST [.L$LUN, LO_WRD] = .RANDOM [5];
: 4156 4      end;
: 4157 3      end;
: 4158 3
: 4159 2      else
: 4160 3      begin      ! ELSE - SEQUENTIAL LBN MODE
: 4161 4      if (.TRK_SGN [.L$LUN] geq 1)
: 4162 3      then      ! if positive track direction add one to multiword
: 4163 4      (if .BST [.L$LUN, LO_WRD] eqlu %0'177777'
: 4164 4      then
: 4165 5          begin
: 4166 5          BST [.L$LUN, LO_WRD] = 0;
: 4167 5          BST [.L$LUN, HI_WRD] = .BST [.L$LUN, HI_WRD] + 1;
: 4168 5          end
: 4169 4      else
: 4170 4          BST [.L$LUN, LO_WRD] = .BST [.L$LUN, LO_WRD] + 1)
: 4171 3      else      ! if negative track direction subtract one from multiword
: 4172 3      if .BST [.L$LUN, LO_WRD] eqlu %0'0'
: 4173 3      then
: 4174 4          begin
: 4175 4          BST [.L$LUN, LO_WRD] = %0'177777';
: 4176 4          BST [.L$LUN, HI_WRD] = .BST [.L$LUN, HI_WRD] - 1;
: 4177 4          end
: 4178 3      else
: 4179 3          BST [.L$LUN, LO_WRD] = .BST [.L$LUN, LO_WRD] - 1;
: 4180 3
: 4181 3
: 4182 3      if .BST [.L$LUN, LO_WRD] gequ (.CST_ADDR [.CUOFF + 3, D_END0]) and ! if hi limit then change direction
: 4183 4      .BST [.L$LUN, HI_WRD] gequ (.CST_ADDR [.CUOFF + 4, D_END1])
: 4184 3      then TRK_SGN [.L$LUN] = -1;
: 4185 3
: 4186 3      if .BST [.L$LUN, LO_WRD] lequ (.CST_ADDR [.CUOFF + 1, D_BEG0] + 1) and ! if low limit then change direction
: 4187 4      .BST [.L$LUN, HI_WRD] lequ (.CST_ADDR [.CUOFF + 2, D_BEG1])
: 4188 3      then TRK_SGN [.L$LUN] = 1;
: 4189 2      end;
: 4190 1      end;
! ROUTINE QIO_LBN

```

000000 004137 000000G

.SBTTL QIO_LBN MULTI-DRIVE TEST ROUTINES
QIO_LBN::

N11

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0350
Page 119
(31)

000004	013705	000000G		JSR	R1,\$SAVE5	:	4089
000010	013700	000000G		MOV	CST,ADDR,R5	:	4113
000014	006300			MOV	CUOFF,R0		
000016	060500			ASL	R0		
000020	132710	000020		ADD	R5,R0		
000024	001002			BITB	#20,(R0)		
000026	105004			BNE	1\$		
000030	000402			CLRB	R4	: RD.DISK	4115
000032	112704	000001	1\$:	BR	2\$	:	4113
000036	013700	000112'	2\$:	MOVB	#1,R4	: *,RD.DISK	4117
000042	013701	000000G		MOV	MAD1,R0	:	4119
000046	010103			MOV	L\$LUN,R1		
000050	006303			MOV	R1,R3		
000052	006303			ASL	R3		
000054	012702	000000G		ASL	R3		
000060	060302			MOV	#BST,R2		
000062	011260	000046		ADD	R3,R2		
000066	062703	000002G		MOV	(R2),46(R0)		
000072	011360	000050		ADD	#BST+2,R3	:	4120
000076	005737	000110'		MOV	(R3),50(R0)		
000102	002406			TST	MX2	:	4122
000104	013700	000114'		BLT	3\$		
000110	011260	000046		MOV	MAD2,R0	:	4125
000114	011360	000050		MOV	(R2),46(R0)		
000120	032737	040000 000000G	3\$:	MOV	(R3),50(R0)	:	4126
000126	001002			BIT	#40000,SWP.FLAGS	:	4129
000130	000137	017122'		BNE	4\$		
000134	006004		4\$:	JMP	10\$		
000136	103015			ROR	R4	: RD.DISK	4130
000140	013746	000000G		BCC	5\$		
000144	042716	100000		MOV	RANDOM,-(SP)	:	4133
000150	012746	000143		BIC	#100000,(SP)		
000154	004737	000000G		MOV	#143,-(SP)		
000160	022626			JSR	PC,BL\$MOD		
000162	020027	000041		CMP	(SP)+,(SP)+		
000166	101001			CMP	R0,#41		
000170	000207			BHI	5\$		
000172	013700	000000G	5\$:	RTS	PC		
000176	006300			MOV	CUOFF,R0	:	4136
000200	060500			ASL	R0		
000202	016004	000004		ADD	R5,R0		
000206	013700	000000G		MOV	4(R0),R4		
000212	006300			MOV	CUOFF,R0		
000214	060500			ASL	R0		
000216	020460	000010		ADD	R5,R0		
000222	001022			CMP	R4,10(R0)		
000224	013701	000000G		BNE	6\$		
000230	006301			MOV	CUOFF,R1	:	4138
000232	060501			ASL	R1		
000234	016146	000002		ADD	R5,R1		
000240	013700	000000G		MOV	2(R1),-(SP)	:	4139
000244	006300			MOV	CUOFF,R0		
				ASL	R0		

BT

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX 11 B11mm 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL2:6

SEQ 0351
Page 120
(31)

000246	060500		ADD	R5,R0		
000250	016046	000006	MOV	6(R0), (SP)		
000254	004737	000000G	JSR	PC,MODUL AS		
000260	066100	000002	ADD	2(R1),R0		4138
000264	010012		MOV	R0,(R2)		
000266	000530		BR	9		4139
000270	010446		MOV	R4, (SP)		4144
000272	016046	000010	MOV	10(R0), (SP)		
000276	004737	000000G	JSR	PC,MODUL AS		
000302	060400		ADD	R4,R0		4143
000304	010013		MOV	R0,(R3)		
000306	013701	000000G	MOV	L1LUN,R1		4146
000312	006301		ASL	R1		
000314	006301		ASL	R1		
000316	013700	000000G	MOV	CUOFF,R0		
000322	006300		ASL	R0		
000324	063700	000000G	ADD	CST,ADDR,R0		
000330	026160	000002G 000010	CMP	BST+2(R1),10(R0)		
000336	001015		RNE	7		
000340	005016		CLR	(SP)		4147
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,MODUL AS		
000364	010071	000000G	MOV	R0,BST(R1)		
000370	005726		TST	(SP)		
000372	013701	000000G	MOV	L1LUN,R1		4149
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		4150
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,MODUL AS		
000452	012761	177777 000000G	MOV	2+1,BIT(R1)		
000460	160061	000000G	SUB	R0,BST(R1)		
000464	005726		TST	(SP)		
000466	013700	000000G	MOV	L1LUN,R0		4153
000472	006300		ASL	R0		
000474	006300		ASL	R0		
000476	013701	000000G	MOV	CUOFF,R1		
000502	006301		ASL	R1		
000504	063701	000000G	ADD	CST,ADDR,R1		
000510	016061	000002G 000004	CMP	BST+2(R0),4(R1)		
000516	003414		BNE	9		
000520	013701	000000G	MOV	CUOFF,R1		4154

CLP

ZRQAMS
V01.6

RD/RX EXERCISER
MULTI DRIVE TEST ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX-11 B11as-16 V4.0 5/79
DISK#USER2:[POWER5]ZRQADO.BL2:6

SEQ 0352
Page 121
(31)

000524	006301		ASL	R1		
000526	063701	000000G	ADD	CST,ADDR,R1		
000532	026061	000002G 000010	CMP	HST,2(R0),10(R1)		
000540	002003		BGE	9:		
000542	013760	000012G 000000G	MOV	RANDOM,12,HST(R0)		4155
000550	022626		CMP	(SP),,(SP),		4155
000552	000207		RTS	PC		4129
000554	062701	000000G	ADD	0TRK,SGN,R1		4161
000560	105711		TSTB	(R1)		
000562	003410		BLE	12:		
000564	021227	177777	CMP	(R2),0-1		4163
000570	001003		BNE	11:		
000572	005012		CLR	(R2)		4166
000574	003213		INC	(R3)		4167
000576	000411		BR	14:		4163
000600	005212		INC	(R2)		4170
000602	000407		BR	14:		4161
000604	005712		TST	(R2)		4172
000606	001004		BNE	13:		
000610	012712	177777	MOV	0-1,(R2)		4175
000614	005313		DEC	(R3)		4176
000616	000401		BR	14:		4172
000620	005312		DEC	(R2)		4179
000622	013700	000000G	MOV	CUOFF,R0		4182
000626	006300		ASL	R0		
000630	060500		ADD	R5,R0		
000632	021260	000000	CMP	(R2),6(R0)		
000636	103411		BLO	15:		
000640	013700	000000G	MOV	CUOFF,R0		4183
000644	006300		ASL	R0		
000646	060500		ADD	R5,R0		
000650	021360	000010	CMP	(R3),10(R0)		
000654	103402		BLO	15:		
000656	112711	000377	MOVB	0377,(R1)		4184
000662	013700	000000G	MOV	CUOFF,R0		4186
000666	006300		ASL	R0		
000670	060500		ADD	R5,R0		
000672	016000	000002	MOV	2(R0),R0		
000676	005200		INC	R0		
000700	021200		CMP	(R2),R0		
000702	101011		BHI	16:		
000704	013700	000000G	MOV	CUOFF,R0		4187
000710	006300		ASL	R0		
000712	060500		ADD	R5,R0		
000714	021360	000004	CMP	(R3),4(R0)		
000720	101002		BHI	16:		
000722	112711	000001	MOVB	01,(R1)		4188
000726	000207		RTS	PC		4089

; Routine size: 236 words, Routine Base: \$CODE\$ + 16346
; Maximum stack depth per invocation: 10 words

D12

ZRQAM5
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1111 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2:6

SEQ 0353
Page 122
(32)

```

: 4191 1 !!ZZZ routine QIO_SIZE : novalue -
: 4192 1 GLOBAL ROUTINE QIO_SIZE : NOVALUE -
: 4193 1
: 4194 1 !,
: 4195 1 !
: 4196 1 ! THIS ROUTINE IS CALLED BY QIO_GEN TO SELECT THE I/O TRANSFER BYTE COUNT
: 4197 1 ! TO BE USED FOR THE CURRENT QIO OR QIO PAIR. THE BYTE COUNT IS
: 4198 1 ! DETERMINED BY A RANDOM NUMBER, AND WILL ALWAYS FALL BETWEEN 1 AND THE
: 4199 1 ! I/O BUFFER SIZE (BYTES_PER_QIO). It is assumed that BYTES_PER_QIO will
: 4200 1 ! never be larger than one binary word or 65000 bytes.
: 4201 1 !
: 4202 1 ! IMPLICIT OUTPUTS:
: 4203 1 ! THE BYTE COUNT IS LOADED INTO ONE OR BOTH MSCP PACKETS.
: 4204 1 !-
: 4205 2 begin
: 4206 2
: 4207 2 local
: 4208 2     SIZE : word,
: 4209 2     BLOCKS_LEFT : word;
: 4210 2
: 4211 2     SIZE = ((.RANDOM [4] and %o'077777') mod (.BYTES_PER_QIO + 1)) and %o'177760'; !GET BYTE COUNT FROM RANDOM NUMBER
: 4212 2
: 4213 2 if .SIZE eq 0
: 4214 2 then
: 4215 2     SIZE = 16;
: 4216 2
: 4217 2 if .CST_ADDR [.CUOFF + 4, D.END1] gtru .MAD1 [LBN.H]
: 4218 2 then BLOCKS_LEFT = %o'177777'
: 4219 2 else BLOCKS_LEFT = .CST_ADDR [.CUOFF + 3, D.END0] - .MAD1 [LBN.L] + 1;
: 4220 2
: 4221 2 if ((.SIZE + BYTES_PER_SECT - 1) / BYTES_PER_SECT) gtru .BLOCKS_LEFT
: 4222 2 then
: 4223 2     SIZE = .BLOCKS_LEFT * BYTES_PER_SECT;
: 4224 2
: 4225 2 MAD1 [BC.L0] = .SIZE;
: 4226 2
: 4227 2 if .MX2 geq 0
: 4228 2 then
: 4229 2     MAD2 [BC.L0] = .SIZE;
: 4230 2
: 4231 1 end;

```

000000	004137	000000G	.SBYTL	QIO_SIZE MULTI DRIVE TEST ROUTINES
			QIO_SIZE::	
000004	013746	000010G	JSR	R1, \$SAVE3
000010	042716	100000	MOV	RANDOM.10, -(SP)
000014	013746	000000G	BIC	%100000, (SP)
000020	005216		MOV	BYTES_PER_QIO, -(SP)
000022	004737	000000G	INC	(SP)
000026	010003		JSR	PC, BL\$MOD
000030	042703	000017	MOV	R0, R3
			BIC	%17, R3

4192
4211

E12

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0354
Page 123
(32)

000034	001002		BNE	1\$				
000036	012703	000020	MOV	#20,R3				4213
000042	013700	000000G	MOV	CUOFF,R0		; *,SIZE		4215
000046	006300		ASL	R0				4217
000050	063700	000000G	ADD	CST.ADDR,R0				
000054	013701	000112'	MOV	MAD1,R1				
000060	026061	000010 000050	CMP	10(R0),50(R1)				
000066	101403		BLOS	2\$				
000070	012702	177777	MOV	#-1,R2		; *,BLOCKS.LEFT		4218
000074	000413		BR	3\$				4217
000076	013700	000000G	MOV	CUOFF,R0				4219
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,*		4221
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,*		4223
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,*		4225
000166	005737	000110'	TST	MX2				4227
000172	002404		BLT	5\$				
000174	013700	000114'	MOV	MAD2,R0				4229
000200	010360	000026	MOV	R3,26(R0)		; SIZE,*		
000204	022626		CMP	(SP)+,(SP)+				4205
000206	000207		RTS	PC				4192

; Routine Size: 68 words, Routine Base: \$CODE\$ + 17276
 ; Maximum stack depth per invocation: 8 words

ZRQAM3
VOL.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0355
Page 124
(33)

```

: 4232 1 GLOBAL routine FILL_BUFF : novalue =
: 4233 1
: 4234 1 !+
: 4235 1 ! THIS ROUTINE IS CALLED BY QIO_GEN TO LOAD THE I/O BUFFER DESCRIBED IN
: 4236 1 ! THE FIRST MSCP PACKET WITH THE APPROPRIATE DATA PATTERN.
: 4237 1 !
: 4238 1 ! THE DATA PATTERN TO BE SELECTED IS BASED ON THE FOLLOWING ALGORITHM:
: 4239 1 !
: 4240 1 ! IF THE OPERATOR DEFINED A DATA PATTERN
: 4241 1 ! THEN
: 4242 1 ! SELECT IT
: 4243 1 ! ELSE
: 4244 1 ! GET DATA PATTERN NUMBER FROM SW P-TABLE
: 4245 1 ! IF DATA PATTERN NUMBER = 0
: 4246 1 ! THEN
: 4247 1 ! GET DATA PATTERN NUMBER FROM THE UNIT'S ENTRY
: 4248 1 ! IN THE DATA PATTERN SEQUENCE TABLE (DPST)
: 4249 1 !
: 4250 1 ! NOTE THAT PATTERN # 1 CONSISTS OF RANDOM NUMBERS, AND PATTERNS # '7 -
: 4251 1 ! 21 USE THE ACTUAL LBN OF THE WRITE REQUEST.
: 4252 1 !
: 4253 1 ! IMPLICIT INPUTS:
: 4254 1 ! L$LUN - CURRENT (DRS) UNIT NUMBER
: 4255 1 !-
: 4256 1
: 4257 2 begin
: 4258 2
: 4259 2 local
: 4260 2 DP_NUM : word,
: 4261 2 DP_ADDR,
: 4262 2 IOB_ADDR,
: 4263 2 SRC_ADDR,
: 4264 2 COUNT : word;
: 4265 2
: 4266 3 if BIT_TST (SWP_FLAGS, SWF_UDP)
: 4267 2 then
: 4268 2 DP_ADDR = SWP_UCNT
: 4269 2 else
: 4270 3 begin
: 4271 3
: 4272 3 if .SWP_DPAT neq 0
: 4273 3 then
: 4274 3 DP_NUM = .SWP_DPAT
: 4275 3 else
: 4276 4 begin
: 4277 4 DP_NUM = .DPST [.L$LUN];
: 4278 4 DPST [.L$LUN] = .DPST [.L$LUN] + 1;
: 4279 4
: 4280 4 if .DPST [.L$LUN] gtr DP_CNT
: 4281 4 then
: 4282 4 DPST [.L$LUN] = 1;
: 4283 4
: 4284 3 end;

```

! DATA PATTERN NUMBER SELECTED
! ADDR OF DATA PATTERN (LENGTH)
! I/O BUFFER ADDRESS (DESTINATION)
! WORKING SOURCE ADDRESS
! NO. OF WORDS IN DATA PATTERN
! IF USER DEFINED A DATA PATTERN
! SELECT IT
! IF USER SELECTED A PRE-DEFINED DATA PATTERN
! SELECT IT
! GET PATTERN NUMBER FROM SEQUENCE TABLE
! ADVANCE TO NEXT PATTERN NUMBER
! CHECK FOR HIGH LIMIT

ZRGAMS
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11:gs-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0356
Page 125
(33)

```

: 4285 3
: 4286 3      DP_ADDR = .DPA_1BL [.DP_NUM - 1];      ! ADDRESS OF DATA PATTERN (COUNT)
: 4287 3
: 4288 3      if .DP_NUM gequ 17
: 4289 3      then
: 4290 3
: 4291 3          if .DP_NUM
: 4292 3          then
: 4293 3              (.DP_ADDR + 2) = .MAD1 [LBN_L];    ! CHECK MACRO (IF PATTERN 17, 19, OR 21)
: 4294 3              ! LOAD LBN INTO FIRST WORD OF PATTERN
: 4295 3          else
: 4296 3              (.DP_ADDR + 4) = .MAD1 [LBN_L];    ! LOAD LBN INTO SECOND WORD OF PATTERN
: 4297 3
: 4298 3          end;
: 4299 2      IOB_ADDR = .MAD1 [BUF_0];                ! I/O BUFFER ADDRESS
: 4300 2      COUNT = .DP_ADDR;                        ! NO. OF WORDS IN DATA PATTERN
: 4301 2      SRC_ADDR = .DP_ADDR + 2;                ! START OF THE ACTUAL DATA PATTERN
: 4302 2
: 4303 2      incr N from 1 to ((.MAD1 [BC_LO] + 1) / 2) do
: 4304 3      begin
: 4305 3          .IOB_ADDR = .SRC_ADDR;                ! MOVE 1 WORD
: 4306 3          ! ADVANCE DESTINATION ADDRESS
: 4307 3          ! ADVANCE SOURCE ADDRESS
: 4308 3          ! DECREMENT COUNT
: 4309 3          COUNT = .COUNT - 1;
: 4310 3
: 4311 3          if .COUNT eq 0
: 4312 3          then
: 4313 3              begin
: 4314 3                  COUNT = .DP_ADDR;
: 4315 3                  SRC_ADDR = .DP_ADDR + 2;
: 4316 3              end;
: 4317 3          ! REPEAT DATA PATTERN
: 4318 2      end;
: 4319 1      ! WORD TRANSFER LOOP
: 4319 1      ! ROUTINE FILL_BUFF
end;

```

				.SBTTL	FILL_BUFF MULTI-DRIVE TEST ROUTINES	
000000	004137	000000G		FILL_BUFF::		
				JSR	R1,\$SAVE5	4232
000004	005746			TST	-(SP)	
000006	032737	000100	000000G	BIT	#100,SWP,FLAGS	4266
000014	001403			BEQ	1\$	
000016	012701	000000G		MOV	4SWP,UCNT,R1	4268
000022	000443			BR	5\$	4266
000024	013700	000000G	1\$:	MOV	SWP,DPAT,R0	4270
000030	001402			BEQ	2\$	
000032	010002			MOV	R0,R2	4274
000034	000414			BR	3\$	4276
000036	013700	000000G	2\$:	MOV	L\$LUN,R0	4277
000042	062700	000050'		ADD	4DPST,R0	
000046	005002			CLR	R2	
000050	151002			BISB	(R0),R2	
					! DP_NUM	
					! 4,DP_NUM	

H12

ZRGAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0357
Page 126
(33)

000052	105210			INCB	(R0)	:		4278
000054	121027	000025		CMPB	(R0),#25	:		4280
000060	101402			BLOS	3\$			
000062	112710	000001		MOVB	#1,(R0)	:		4282
000066	010200		3\$:	MOV	R2,R0	:	DP.NUM,*	4286
000070	006300			ASL	R0			
000072	016001	001072'		MOV	DPA,TBL-2(R0),R1	:	*,DP.ADDR	
000076	020227	000021		CMP	R2,#21	:	DP.NUM,*	4288
000102	103413			BLO	5\$			
000104	013700	000112'		MOV	MAD1,R0	:		4293
000110	006002			ROR	R2	:	DP.NUM	4291
000112	103004			BCC	4\$			
000114	016061	000046	000002	MOV	46(R0),2(R1)	:	*,*(DP.ADDR)	4293
000122	000403			BR	5\$	:		4291
000124	016061	000046	000004	MOV	46(R0),4(R1)	:	*,*(DP.ADDR)	4295
000132	013700	000112'		MOV	MAD1,R0	:		4299
000136	016004	000032		MOV	32(R0),R4	:	*,IOB.ADDR	
000142	011103			MOV	(R1),R3	:	DP.ADDR,COUNT	4300
000144	012705	000002		MOV	#2,R5	:		4301
000150	060105			ADD	R1,R5	:	DP.ADDR,*	
000152	010502			MOV	R5,R2	:	*,SRC.ADDR	
000154	016046	000026		MOV	26(R0),-(SP)	:		4303
000160	005216			INC	(SP)			
000162	012746	000002		MOV	#2, -(SP)			
000166	004737	000000G		JSR	PC,BL\$DIV			
000172	010066	000004		MOV	R0,4(SP)			
000176	005000			CLR	R0	:	N	
000200	000405			BR	7\$			
000202	012224		6\$:	MOV	(R2)+,(R4)+	:	SRC.ADDR,IOB.ADDR	4305
000204	005303			DEC	R3	:	COUNT	4308
000206	001002			BNE	7\$	:		4310
000210	011103			MOV	(R1),R3	:	DP.ADDR,COUNT	4313
000212	010502			MOV	R5,R2	:	*,SRC.ADDR	4314
000214	005200		7\$:	INC	R0	:	N	4303
000216	020066	000004		CMP	R0,4(SP)	:	N,*	
000222	003767			BLE	6\$			
000224	062706	000006		ADD	#6,SP	:		4232
000230	000207			RTS	PC			

; Routine Size: 77 words, Routine Base: \$CODE\$ + 17506
 ; Maximum stack depth per invocation: 10 words

; 4320 1
 ; 4321 1

ZRGAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0358
Page 127
(34)

```

: 4322 1 GLOBAL ROUTINE PROC_RETPKT : NOVALUE =
: 4323 1
: 4324 1 !,
: 4325 1 ! THIS ROUTINE IS CALLED FROM THE MULTI_DRIVE "EXECUTIVE" AND DUP_COMMAND TO CHECK FOR
: 4326 1 ! AND PROCESS ANY RETURN PACKETS THAT HAVE BEEN "SENT" BY THE "DRIVER"
: 4327 1 ! PORTION OF THE PROGRAM. THE I/O DONE QUEUE (IODQ) ACTS AS THE LINK
: 4328 1 ! BETWEEN THE TWO PROGRAM PARTS; IT HOLDS INDECES OF RETURN PACKETS WHICH
: 4329 1 ! REQUIRE PROCESSING.
: 4330 1 !
: 4331 1 ! UNDER THE MULTI-DRIVE TEST, RETURN PACKETS ORIGINATE FROM TWO SOURCES:
: 4332 1 ! 1. MSCP - THE MORE COMMON, DESCRIBING A COMPLETED I/O
: 4333 1 ! OPERATION.
: 4334 1 ! 2. DUP - THE LESS COMMON, DESCRIBING A PORTION OF I/O
: 4335 1 ! COMMUNICATIONS WITH THE CONTROLLER PROGRAM.
: 4336 1 ! 3. THE PROGRAM "DRIVER" - DESCRIBING A CONTROLLER ERROR OR
: 4337 1 ! COMMAND TIMEOUT.
: 4338 1 !-
: 4339 1
: 4340 1 while .IODQ_IN neq .IODQ_OUT do ! DO UNTIL I/O DONE QUEUE IS EMPTY
: 4341 2 begin
: 4342 2 RP_INDX = OUT_IODQ (); ! GET INDEX OF NEXT RETPKT AND ADVANCE OUT POINTER
: 4343 2 RP_ADDR = RETPKT + (.RP_INDX * RP_LEN * 2); ! CALCULATE RETPKT ADDRESS
: 4344 3 if NOT (.RP_ADDR [CONID] eal CID_DUP) ! if not DUP then
: 4345 2 then (SET_CPAR (.RP_ADDR [CTLR])); ! SET UP CURRENT CONTROLLER PARAMETERS
: 4346 2
: 4347 2 selectoneu .RP_ADDR [CONID] of ! CONNECTION ID INDICATES PACKET SOURCE
: 4348 2 set
: 4349 2
: 4350 2 [CID_MSCP] : IO_RETPKT (); ! DISK MSCP (I/O TRANSFER DONE)
: 4351 2 [CID_DUP] : DIO_RETPKT (); ! DUP (I/O TRANSFER DONE)
: 4352 2 [CID_DRIVER] : DR_RETPKT (); ! MESSAGE FROM "DRIVER"
: 4353 2
: 4354 2 [otherwise] : PRINTF (DBM12, .RP_ADDR [CONID]);!"CONN ID = XXXXX RECEIVED"
: 4355 2 tes;
: 4356 2
: 4357 1 end; ! UNTIL I/O DONE QUEUE IS EMPTY

```

				.SBTTL	PROC.RETPKT MULTI-DRIVE TEST ROUTINES	
000000	010146			PROC.RETPKT::		
				MOV	R1, -(SP)	4322
000002	023737	000000G	000000G	1\$: CMP	IODQ.IN, IODQ.OUT	4340
000010	001467			BEQ	7\$	
000012	004737	000000G		JSR	PC, OUT_IODQ	4342
000016	010037	000000G		MOV	R0, RP_INDX	
000022	010046			MOV	R0, -(SP)	4343
000024	012746	000054		MOV	#54, -(SP)	
000030	004737	000000G		JSR	PC, BL\$MUL	
000034	062700	000000G		ADD	#RETPKT, R0	
000040	010037	000000G		MOV	R0, RP_ADDR	
000044	126027	000003	000002	CMPB	3(R0), #2	4344
000052	001406			BEQ	2\$	
000054	116016	000002		MOVB	2(R0), (SP)	4345

J12

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK:USER2:[POWERS]ZRQADO.BL2;6SEQ 0359
Page 128
(34)

000060	042716	177760	BIC	#177760,(SP)		
000064	004737	000000G	JSR	PC,SET,CPAR		
000070	013700	000000G	2\$: MOV	RP,ADDR,R0	:	4347
000074	005001		CLR	R1		
000076	156001	000003	BISB	3(R0),R1		
000102	005701		TST	R1	:	4350
000104	001003		BNE	3\$		
000106	004737	000000V	JSR	PC,IO.RETPKT		
000112	000424		BR	6\$	:	4347
000114	020127	000002	3\$: CMP	R1,#2	:	4351
000120	001003		BNE	4\$		
000122	004737	000000V	JSR	PC,DIO.RETPKT		
000126	000416		BR	6\$	:	4347
000130	020127	000003	4\$: CMP	R1,#3	:	4352
000134	001003		BNE	5\$		
000136	004737	000000V	JSR	PC,DR.RETPKT		
000142	000410		BR	6\$	:	4347
000144	010116		5\$: MOV	R1,(SP)	:	4354
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)+	:		4341
000166	000705		BR	1\$	:	4340
000170	012601	7\$: MOV	(SP)+,R1	:		4322
000172	000207		RTS	PC		

; Routine Size: 62 words, Routine Base: \$CODE\$ + 17740
; Maximum stack depth per invocation: 7 words

K12

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0360
Page 129
(35)

```

: 4358 1  !!
: 4359 1  !! GLOBAL ROUTINE DIO_RETPKT : NOVALUE *
: 4360 1  !!
: 4361 1  !!
: 4362 1  !! THIS ROUTINE IS CALLED BY PROC_RETPKT TO HANDLE ALL DUP I/O TRANSFER
: 4363 1  !! RETURN PACKETS. PROCESSING OF THESE PACKETS INCLUDES DECLARING ANY
: 4364 1  !! HARD ERRORS THAT MAY HAVE OCCURRED, UPDATING THE STATISTICS.
: 4365 1  !!
: 4366 1  !! IMPLICIT INPUTS:
: 4367 1  !! RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4368 1  !! T_ADDR - ADDRESS OF THE CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 4369 1  !! CST_ADDR - ADDRESS OF THE CURRENT CONTROLLER'S CST
: 4370 1  !! DUOFF - CST OFFSET FOR THE CURRENT UNIT
: 4371 1  !! L$UN - CURRENT UNIT NUMBER
: 4372 1  !! CCTLN - CURRENT CONTROLLER NUMBER
: 4373 1  !!
: 4374 1  !! IMPLICIT OUTPUTS
: 4375 1  !! CST_ADDR [.DUOFF + OF_DBN, NODUPMEDIA] - IF THIS BIT SET NO DUP EXERCISER
: 4376 1  !!
: 4377 1  !!
: 4378 2  BEGIN
: 4379 2  LOCAL FLAG : BYTE INITIAL(BYTE(TRUE)),
: 4380 2  SUM2 : WORD,
: 4381 2  SUM : WORD;
: 4382 2  !PRINTX (DER18);
: 4383 2  ! TOTAL NUMBER OF BYTES TRANSFERRED TO/FROM A UNIT
: 4384 2  IF .RP_ADDR [STATUS] NEQU ST_SUC
: 4385 2  THEN
: 4386 2  ! IF STATUS CODE INDICATES ERROR
: 4387 3  BEGIN
: 4388 3  CST_ADDR [.DUOFF + OF_DBN, DUPERROR] = 1;
: 4389 3  ! SET DUP ERROR FLAG
: 4390 3  HARD_ERROR ();
: 4391 4  IF .RP_ADDR [ENDCOD] EQLU (OP_ELP + OP_END) OR
: 4392 4  ! IF ENDCODE IS EXECUTE LOCAL PROGRAM
: 4393 4  .RP_ADDR [ENDCOD] EQLU (OP_GDS + OP_END)
: 4394 4  ! OR GET DUST STATUS
: 4395 3  THEN BEGIN
: 4396 3  CST_ADDR [.DUOFF + OF_DBN, NODUPMEDIA] = 1;
: 4397 3  ! TURN OFF DUP EXERCISER
: 4398 3  END;
: 4399 2  ELSE
: 4400 2  ! ELSE - I/O WAS SUCCESSFUL
: 4401 3  BEGIN
: 4402 3  IF .RP_ADDR [ENDCOD] EQLU (OP_GDS + OP_END) ! IF ENDCODE IS GET DUST STATUS
: 4403 4  THEN BEGIN
: 4404 4  IF .RP_ADDR [9.11.1.0] EQL 1
: 4405 4  THEN CST_ADDR [.DUOFF + OF_DBN, D_ACTIVE] = ACTIVE
: 4406 4  ! CONTROLLER IN AN ACTIVE STATE
: 4407 4  ELSE CST_ADDR [.DUOFF + OF_DBN, D_ACTIVE] = IDLE;
: 4408 4  ! CONTROLLER IN AN IDLE STATE
: 4409 4  IF .RP_ADDR [9.9.1.0] NEQ 1 THEN
: 4410 4  ! TEST TO SEE IF CONTROLLER LOCAL PROGRAMS (PG 18 OF DUP DOC)
: 4411 5  BEGIN
: 4412 5  HARD_ERROR ();
: 4413 5  CST_ADDR [.DUOFF + OF_DBN, NODUPMEDIA] = 1;
: 4414 5  ! TURN OFF DUP EXERCISER
: 4415 5  END;
: 4416 3  END;

```

L12

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0361
Page 130
(35)

```

: 4411 3
: 4412 3
: 4413 3 IF (.RP_ADDR [ENDCOD] EQL (OP_RCD * OP_END)) AND
: 4414 3 (.DUPPKT [DUPTYPE] EQL 6) AND
: 4415 3 (.DUPPKT [DUPMSG] EQL 2) AND !IF IT IS A RECEIVE DBN COMMAND WITH TYPE 6 AND MESSAGE 2 THEN
: 4416 4 (.CST_ADDR [,DUOFF * OF_DBN, DUPWRITE] EQLU 1) ! IF WRITE FLAG SET IN CST TABLE THEN COMPARE BLOCKS
: 4417 3 THEN DUP_COMPARE ();
: 4418 3
: 4419 2 END; ! COMPARE THE FOLLOWING 512 BYTES
: 4420 2
: 4421 2 PUT_RETPKT (.RP_INDX);
: 4422 1 END; ! ROUTINE DIO_RETPKT

```

		.SBTTL	DIO.RETPKT MULTI-DRIVE TEST ROUTINES	
000000	010146	DIO.RETPKT::		
000002	112700	000001	MOV R1, -(SP)	4359
000006	013701	000000G	MOV8 #1, R0	4378
000012	005761	000016	MOV RP_ADDR, R1	4385
000016	001435		IST 16(R1)	
000020	013700	001156'	BEQ 2\$	
000024	006300		MOV DUOFF, R0	4388
000026	063700	000000G	ASL R0	
000032	052760	040000 000020	ADD CST_ADDR, R0	
000040	004737	000000V	BIS #40000, 20(R0)	
000044	013700	000000G	JSR PC, HARD_ERROR	4389
000050	126027	000014 000203	MOV RP_ADDR, R0	4390
000056	001404		CMPB 14(R0), #203	
000060	126027	000014 000201	BFC 1\$	
000066	001112		CMPB 14(R0), #201	4391
000070	013700	001156'	BNE 6\$	
000074	006300		1\$: MOV DUOFF, R0	4393
000076	063700	000000G	ASL R0	
000102	052760	100000 000020	ADD CST_ADDR, R0	
000110	000501		BIS #100000, 20(R0)	
000112	126127	000014 000201	BR 6\$	4385
000120	001036		2\$: CMPB 14(R1), #201	4399
000122	013700	001156'	BNE 5\$	
000126	006300		MOV DUOFF, R0	4403
000130	063700	000000G	ASL R0	
000134	032761	004000 000022	ADD CST_ADDR, R0	
000142	001404		BIT #4000, 22(R1)	4402
000144	052760	020000 000020	BFC 3\$	
000152	000403		BIS #20000, 20(R0)	4403
000154	042760	020000 000020	BR 4\$	4402
000162	032761	001000 000022	3\$: BIC #20000, 20(R0)	4404
000170	001012		4\$: BIT #1000, 22(R1)	4405
000172	004737	000000V	BNE 5\$	
000176	013700	001156'	JSR PC, HARD_ERROR	4407
000202	006300		MOV DUOFF, R0	4408
000204	063700	000000G	ASL R0	
000210	052760	100000 000020	ADD CST_ADDR, R0	
			BIS #100000, 20(R0)	

M12

ZRQAM3	RD/RX EXERCISER	11-Apr-1984 11:08:35	VAX-11 Bliss-16 V4.0-579	SEQ 0362
V01.6	MULTI-DRIVE TEST ROUTINES	11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2;6	Page 131 (35)

000216	013700	000000G	5\$:	MOV	RP,ADDR,R0	:	4413
000222	126027	000014 000205		CMPB	14(R0),#205		
000230	001031			BNE	6\$		
000232	013700	000000G		MOV	DUPPKT,R0	:	4414
000236	042700	007777		BIC	#7777,R0		
000242	020027	060000		CMP	R0,#60000		
000246	001022			BNE	6\$		
000250	013700	000000G		MOV	DUPPKT,R0	:	4415
000254	042700	170000		BIC	#170000,R0		
000260	020027	000002		CMP	R0,#2		
000264	001013			BNE	6\$		
000266	013700	001156'		MOV	DUOFF,R0	:	4416
000272	006300			ASL	R0		
000274	063700	000000G		ADD	CST,ADDR,R0		
000300	032760	010000 000020		BIT	#10000,20(R0)		
000306	001402			BEQ	6\$		
000310	004737	000000V		JSR	PC,DUP,COMPARE	:	4417
000314	013746	000000G	6\$:	MOV	RP,INDX,-(SP)	:	4421
000320	004737	000000G		JSR	PC,PUT,RETPKT		
000324	005726			TST	(SP)+	:	4378
000326	012601			MOV	(SP)+,R1	:	4359
000330	000207			RTS	PC		

; Routine Size: 109 words, Routine Base: \$CODE\$ + 20134
; Maximum stack depth per invocation: 3 words

; 4423 1

N12

ZRQPM3 RD/RX EXERCISER 11-Apr-1984 11:08:35 VAX-11 Bliss-16 V4.0-579 SEQ 0363
 V01.6 MULTI-DRIVE TEST ROUTINES 11-Apr-1984 11:08:22 DISK\$USER2:[POWERS]ZRQADO.BL2:6 Page 132
 (36)

```

: 4424 1 GLOBAL ROUTINE DUP_COMPARE : NOVALUE =
: 4425 1
: 4426 1
: 4427 1 !
: 4428 1 ! THIS ROUTINE IS CALLED BY DIO_RETPKT WHEN THE RECEIVE DATA COMMAND
: 4429 1 ! IS BEING PROCESSED. THIS COMMAND COMPARES THE WRITTEN BUFFER WITH
: 4430 1 ! THE PATTERN WORD GIVEN IN SEND DATA COMMAND. FOR EVERY WORD COMPARED
: 4431 1 ! THE ROUTINE INCREMENTS THE TALLY TABLE. IF THE COMPARE SHOWS AN
: 4432 1 ! ERROR, THE DBN HARD ERROR COUNTER WILL BE INCREMENTED AND THE
: 4433 1 ! THE DBN NUMBER AND BYTE COUNT WILL BE PRINTED.
: 4434 1 !
: 4435 1 ! IMPLICIT INPUTS:
: 4436 1 ! S_PATTERN ! THE SAVED PATTERN WRITTEN TO THE DBN'S
: 4437 1 ! S_DUPPKT ! THE POINTER FOR DUP BUFFER
: 4438 1 ! T_ADDR ! THE ADDRESS OF THE TALLY TABLE FOR THIS UNIT
: 4439 1 ! CST_ADDR ! THE ADDRESS OF PRESENT CONTROLLER STATUS TABLE
: 4440 1 !
: 4441 2 BEGIN
: 4442 2
: 4443 2 OWN
: 4444 2 COUNT : WORD;
: 4445 2
: 4446 2 !PRINTX (DER19);
: 4447 2 S_DUPPKT = 0;
: 4448 2 INCR COUNT FROM 1 TO 256 DO !INDEX PIONTER FOR DATA STORED IN MSCP ENV PACKET
: 4449 3 BEGIN
: 4450 3 S_DUPPKT = S_DUPPKT + 2; ! INITIALLY THIS SKIPS THE FIRST WORD OF DUPPKT
: 4451 3 IF (DUPPKT + S_DUPPKT) NEQ S_PATTERN THEN !IF THE CONTENTS OF DBN DOESN'T EQUAL PATTERN
: 4452 4 BEGIN
: 4453 4 CST_ADDR [DUOFF + OF_DBN, DUPERROR] = 1; ! SET DUP ERROR FLAG
: 4454 4 ERRHRD (46, EH_10, EMS_22); !LIST ERROR
: 4455 4 EXITLOOP;
: 4456 3 END;
: 4457 2 END;
: 4458 1 !GO THROUGH ALL DBN WORDS
!END ROUTINE DUP-COMPARE
  
```

001206 .PSECT \$GGG\$, RO
 001206 COUNT: .BLKW 1

020466 .SBTTL DUP_COMPARE MULTI-DRIVE TEST ROUTINES
 .PSECT \$CODE\$, RO

```

000000 010146 DUP_COMPARE::
000002 005037 000000G MOV R1, -(SP)
000006 012701 000400 CLR S_DUPPKT
000012 062737 000002 000000G MOV #400, R1
000020 013700 000000G 1$: ADD #2, S_DUPPKT
000024 026037 000000G 000000G MOV S_DUPPKT, R0
000032 001415 CMP DUPPKT(R0), S_PATTERN
BEQ 2$
  
```

[3]

ZRQAMS
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Release 16 V4.0 5/9
DISK1USER2:[POWER5]ZRQADO.BL2;6

SEQ 0364
Page 134
(36)

000034	013700	001156	MOV	DUOFF,R0			
000040	006300		ASL	R0			4453
000042	063700	000000G	ADD	CST,ADDR,R0			
000046	052760	040000 000020	BJS	#40000,20(R0)			
000054	104456		TRAP	56			4454
000056	000056		WORD	56			
000060	000000G		WORD	EH,10			
000062	000000G		WORD	EH5,22			
000064	000402		BR	33			4455
000066	005301	24:	DEC	R1		COUNT	4448
000070	001350		BFE	13			
000072	012601	33:	MOV	(SP)+,R1			4424
000074	000207		RTS	PC			

; Routine Size: 31 words, Routine Base: \$CODE + 20466
; Maximum stack depth per invocation: 3 words

; 4459 1
; 4460 1
; 4461 1

C13

ZRGAMS
V01.6RD-RX EXERCISER
MULTI DRIVE TEST ROUTINES11 Apr 1984 11:08:35
11-Apr-1984 11:08:22VAX 11 B11a 16 V4.0 5/9
DISK:USER2:[POWER5]ZRGADO.BL2;6SEQ 0365
Page 134
(37)

```

: 4462 1 GLOBAL routine IO RETPKT : novalue =
: 4463 1
: 4464 1
: 4465 1
: 4466 1 THIS ROUTINE IS CALLED BY PROC RETPKT TO HANDLE ALL I/O TRANSFER
: 4467 1 RETURN PACKETS. PROCESSING OF THESE PACKETS INCLUDES DECLARING ANY
: 4468 1 HARD ERRORS THAT MAY HAVE OCCURRED, UPDATING THE STATISTICS, AND
: 4469 1 PERFORMING HOST WRITE-COMPARES IF REQUIRED.
: 4470 1
: 4471 1 IMPLICIT INPUTS:
: 4472 1 CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 4473 1 RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
: 4474 1 T_ADDR - ADDRESS OF CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 4475 1 CCTLN - CURRENT CONTROLLER NUMBER
: 4476 1 LUN - CURRENT UNIT NUMBER
: 4477 1
: 4478 2 begin
: 4479 2
: 4480 2 local
: 4481 2 FLAG : byte initial (byte (TRUE));
: 4482 2
: 4483 2 FSET UPAR ();
: 4484 2 ST_CODE = .RP_ADDR [STSCOD];
: 4485 2 SB_CODE = .RP_ADDR [SUBCOD];
: 4486 2
: 4487 3 if (.ST_CODE neq ST_SUC)
: 4488 2 then
: 4489 3 begin
: 4490 3 HARD_ERROR ();
: 4491 3 COMPARE_DATA = FALSE;
: 4492 3
: 4493 3 if (.ST_CODE neq ST_OFL) and
: 4494 3 (.ST_CODE neq ST_AVL) and
: 4495 3 (.T_ADDR [ERR_HARD] geq .SWP_ERROR)
: 4496 3 then
: 4497 4 begin
: 4498 4 DOR [LUN] = DU_HERR;
: 4499 4 DODU (LUN);
: 4500 3 end;
: 4501 3
: 4502 3 end
: 4503 2 else
: 4504 3 begin
: 4505 3 UPD_IO_TALLY ();
: 4506 3
: 4507 4 if (.RP_ADDR [ENOCOD] eq (OP_WRT or OP_END))
: 4508 3 then
: 4509 3 COMPARE_DATA = TRUE;
: 4510 3
: 4511 3 if (BIT_IST (SWP_FLAGS, SWP_HWC)) and
: 4512 4 (.COMPARE_DATA)
: 4513 3 then
: 4514 3 FLAG = HOST_WRT_CHK ();

```

```

: FIND UNIT'S ENTRY IS CST AND SET UP UNIT-RELATED DATA
: GET STATUS CODE FROM RETPKT
: GET SUB CODE, IF ANY
: IF STATUS CODE INDICATES ERROR
: UPDATE ERROR COUNT
: NO POINT IN DOING HOST COMPARES ON ERRORS
: DROP UNIT IF ERROR COUNTS EXCEEDS LIMIT
: LOAD REASON FOR DROPPING UNIT
: DROP UNIT
: IF I/O WAS SUCCESSFUL
: UPDATE I/O TALLY (STATISTICS)
: HOST COMPARES MAY BE ALLOWED IF NO FURTHER ERRORS
: IF HOST IS DOING WRITE COMPARES
: SAVE I/O PACKET OR DO WRITE-CHECK

```


013

ZRQAMS
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWER5]ZRQADO.BL2;6

SEQ 0366
Page 135
(37)

```

: 4515 3
: 4516 2
: 4517 2
: 4518 2
: 4519 2
: 4520 2
: 4521 2
: 4522 2
: 4523 1

      end;

      IF .FLAG
      then
        SWEEP ();

      QIO (.CCTLR) = .QIO (.CCTLR) - 1;
      end;

```

```

! 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::		
			JSR	R1,SAVE2	4462
000004	112701	000001	MOV	#1,R1	4478
000010	004737	000000V	JSR	PC,FSET,UPAR	4483
000014	013700	000000G	MOV	RP,ADDR,R0	4484
000020	116037	000016 000000G	MOV	16(R0),ST.CODE	
000026	042737	177740 000000G	BIC	#177740,ST.CODE	
000034	016002	000016	MOV	16(R0),R2	4485
000040	006202		ASR	R2	
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	IST	ST.CODE	4487
000066	001431		BEQ	1\$	
000070	004737	000000V	JSR	PC,HARD.ERROR	4490
000074	105937	001170'	CLRB	COMPARE.DATA	4491
000100	023727	000000G 000003	CMP	ST.CODE,#3	4493
000106	001447		BEQ	3\$	
000110	023727	000000G 000004	CMP	ST.CODE,#4	4494
000116	001443		BEQ	3\$	
000120	013700	000000G	MOV	T,ADDR,R0	4495
000124	026037	000014 000000G	CMP	14(R0),SWP.ERROR	
000132	103435		BLO	3\$	
000134	013700	000000G	MOV	1\$UN,R0	4498
000140	112760	000004 000000G	MOV	#4,DUR(R0)	
000146	104451		TRAP	51	4499
000150	000426		BR	3\$	4487
000152	004737	000000V	JSR	PC,UPD,IO,TALLY	4505
000156	013700	000000G	MOV	RP,ADDR,R0	4507
000162	126027	000014 000242	CMPB	14(R0),#242	
000170	001003		BNE	2\$	
000172	112737	000001 001170'	MOV	#1,COMPARE.DATA	4509
000200	032737	000040 000000G	BIT	#40,SWP.FLAGS	4511
000206	001407		BEQ	3\$	
000210	032737	000001 001170'	BIT	#1,COMPARE.DATA	4512
000216	001403		BEQ	3\$	
000220	004737	000000V	JSR	PC,HST,WRT,CHK	4514
000224	110001		MOV	R0,R1	
				#,FLAG	

F13

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0367
Page 136
(37)

000226	006001		3\$:	ROR	R1			
000230	103002			BCC	4\$		; FLAG	4518
000232	004737	000000V		JSR	PC,SWEEP			4520
000236	013700	000000G	4\$:	MOV	CCTL,R0			4522
000242	105360	000000G		DECB	QIO(R0)			
000246	000207			RTS	PC			4462

; Routine Size: 84 words, Routine Base: \$CODE\$ + 20564
; Maximum stack depth per invocation: 5 words

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0368
Page 137
(38)

```

: 4524 1 GLOBAL routine FSET_UPAR : novalue *
: 4525 1
: 4526 1 !,
: 4527 1 ! THIS ROUTINE IS CALLED BY IO_RETPKT AND OTHERS TO SEARCH THE CURRENT
: 4528 1 ! CONTROLLER STATUS TABLE (CST) FOR THE DISK ADDRESS WHICH IS
: 4529 1 ! CONTAINED IN THE CURRENT RETURN PACKET. WHEN FOUND, THE OFFSET INTO THE
: 4530 1 ! CST IS USED AS INPUT TO SET_UPAR, WHICH SETS UP CURRENT UNIT-RELATED
: 4531 1 ! DATA PARAMETERS.
: 4532 1 !
: 4533 1 ! IMPLICIT INPUTS:
: 4534 1 ! RP_ADDR - ADDRESS OF CURRENT RETURN PACKET
: 4535 1 ! CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 4536 1 !-
: 4537 1
: 4538 2 begin
: 4539 2
: 4540 2 incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do ! FOR EACH UNIT
: 4541 2
: 4542 2 if .CST_ADDR [.OFFSET + OF_DATA, D_DISK_NUM] eq1 .RP_ADDR [DISK] ! IF RETPKT UNIT MATCHES CST ENTRY
: 4543 2 then
: 4544 3 beg
: 4545 3 SET_UPAR (.OFFSET); ! SET UP UNIT-RELATED DATA
: 4546 3 return; ! DONE
: 4547 2 end;
: 4548 2
: 4549 2 PRINTF (DBM19, .RP_ADDR [DISK], .CCTLR); ! "CAN'T FIND DISK XXX IN CST X"
: 4550 1 end; ! ROUTINE FSET_UPAR

```

			.SHTTL FSET_UPAR MULTI-DRIVE TEST ROUTINES	
000000	004137	000000G	FSET_UPAR::	
			JSR R1, \$SAVE4	4524
000004	012702	000003	MOV #3, R2	4540
000010	010201		1\$: MOV R2, R1	4542
000012	006301		ASL R1	
000014	063701	000000G	ADD CST_ADDR, R1	
000020	013700	000000G	MOV RP_ADDR, R0	
000024	016004	000010	MOV 10(R0), R4	
000030	111103		MOVB (R1), R3	
000032	042703	177760	BIC #177760, R3	
000036	020304		CMP R3, R4	
000040	001005		BNE 2\$	
000042	010206		MOV R2, (SP)	4545
000044	004737	000000G	JSR PC, SET_UPAR	
000050	005726		IST (SP)	4546
000052	000207		RTS PC	4544
000054	062702	000012	2\$: ADD #12, R2	4540
000060	020227	000041	CMP R2, #41	
000064	003751		BLE 1\$	
000066	013746	000000G	MOV CCTLR, (SP)	4549
000072	013700	000000G	MOV RP_ADDR, R0	
000076	016046	000010	MOV 10(R0), (SP)	
000102	012746	000000G	MOV #DBM19, (SP)	

613

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0369
Page 138
(38)

000106	012746	000003	MOV	#3,(SP)
000112	010600		MOV	SP,R0
000114	104417		TRAP	17
000116	062706	000010	ADD	#10,SP
000122	000207		RTS	PC

; SP,4

;
;

4538
4524

; Routine Size: 42 words, Routine Base: \$CODE\$ + 21034
; Maximum stack depth per invocation: 11 words

ZRDAMS
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11gs-16 V4.0-579
DISK\$USER2:[POWERS]ZRDADO.BL2;6SEQ 0370
Page 139
(39)

```

: 4551 1 GLOBAL routine HARD_ERROR : novalue =
: 4552 1
: 4553 1 !
: 4554 1 !
: 4555 1 ! THIS ROUTINE IS CALLED BY IO_RETPKT AND OTHERS TO INCREMENT THE HARD
: 4556 1 ! ERROR STATISTIC FIELD FOR THE CURRENT UNIT. IF THE HARD ERROR COUNT
: 4557 1 ! HAS EXCEEDED THE OPERATOR-SPECIFIED LIMIT, THEN THE UNIT IS DROPPED
: 4558 1 ! FROM TESTING.
: 4559 1 !
: 4560 1 ! IMPLICIT INPUTS:
: 4561 1 ! L$LUN - CURRENT UNIT NUMBER
: 4562 1 ! CST_ADDR - ADDRESS OF CURRENT CONTROLLER'S CST
: 4563 1 ! CUOFF - CST OFFSET FOR CURRENT UNIT
: 4564 1 ! T_ADDR - ADDRESS OF CURRENT UNIT'S STATISTICS BLOCK (TALLY)
: 4565 1 !-
: 4566 2 begin
: 4567 2 T_ADDR [ERR_HARD] = .T_ADDR [ERR_HARD] + 1; ! INCREMENT UNIT'S HARD ERROR COUNT
: 4568 2
: 4569 2 selectoneu .ST_CODE of
: 4570 2 set
: 4571 2
: 4572 2 [ST_SUC]: if .SB_CODE neq 0 ! SUCCESS WITH NON-ZERO SUB-CODE
: 4573 2 then
: 4574 3 begin
: 4575 3
: 4576 3 if .SB_CODE eq 4
: 4577 3 then
: 4578 3 T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1
: 4579 3 else
: 4580 3 T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4581 3
: 4582 3 if .APT_MODE
: 4583 3 then
: 4584 3 ERR_HRD_RTNE_APT (44)
: 4585 3 else
: 4586 3 ERR_HRD_RTNE (44);
: 4587 3
: 4588 2 end;
: 4589 2
: 4590 2 [ST_CMD]: begin ! INVALID COMMAND
: 4591 3 T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4592 3
: 4593 3 if .APT_MODE
: 4594 3 then
: 4595 3 ERR_HRD_RTNE_APT (31)
: 4596 3 else
: 4597 3 ERR_HRD_RTNE (31);
: 4598 3
: 4599 2 end;
: 4600 2
: 4601 3 [ST_ABO]: begin ! COMMAND ABORTED
: 4602 3 T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4603 3

```

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0371
Page 140
(39)

```

: 4604 3      if .APT_MODE
: 4605 3      then
: 4606 3          ERR_HRD_RTNE_APT (32)
: 4607 3      else
: 4608 3          ERR_HRD_RTNE (32);
: 4609 3
: 4610 2      end;
: 4611 2
: 4612 3      [ST_DFL] :      begin                                ! OFFLINE
: 4613 3          T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4614 3
: 4615 3          if .APT_MODE
: 4616 3          then
: 4617 4              begin
: 4618 4                  .MAIL_BOX_TESTNUM = 1;
: 4619 4                  .MAIL_BOX_SUBTST = .CST_ADDR [.CUOFF + OF_DATA, D_DISK_NUM];
: 4620 3              end;
: 4621 3
: 4622 3          ERRDF (18, EGD_18, EMS_18);                                !
: 4623 3          DUR [.L$LUN] = DU_DFATAL;                                ! DEVICE FATAL ERROR
: 4624 3          DODU (.L$LUN);                                            ! DROP UNIT
: 4625 2      end;
: 4626 2
: 4627 3      [ST_AVL]:      begin                                ! WENT TO AVAILABLE STATE
: 4628 3          T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4629 3
: 4630 3          if .APT_MODE
: 4631 3          then
: 4632 4              begin
: 4633 4                  .MAIL_BOX_TESTNUM = 1;
: 4634 4                  .MAIL_BOX_SUBTST = .CST_ADDR [.CUOFF + OF_DATA, D_DISK_NUM];
: 4635 3              end;
: 4636 3
: 4637 3          ERRDF (24, EGD_18, EMS_24);                                !
: 4638 3          DUR [.L$LUN] = DU_DFATAL;                                ! DEVICE FATAL ERROR
: 4639 3          DODU (.L$LUN);                                            ! DROP UNIT
: 4640 2      end;
: 4641 2
: 4642 3      [ST_MFE]:      begin                                ! MEDIA FORMAT ERROR
: 4643 3          T_ADDR [ERR_HRD_SEK] = .T_ADDR [ERR_HRD_SEK] + 1;
: 4644 3
: 4645 3          if .APT_MODE
: 4646 3          then
: 4647 3              ERR_HRD_RTNE_APT (35)
: 4648 3          else
: 4649 3              ERR_HRD_RTNE (35);
: 4650 3
: 4651 2      end;
: 4652 2
: 4653 3      [ST_WPT]:      begin                                ! DEVICE WRITE PROTECTED
: 4654 3
: 4655 3          if .SB_CODE eq 128
: 4656 3          then

```

ZRQAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0572
Page 141
(39)

```

: 4657 3      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1
: 4658 3      else
: 4659 3      T_ADDR [ERR_HRD_DRY] = .T_ADDR [ERR_HRD_DRY] + 1;
: 4660 3
: 4661 3      if .APT_MODE
: 4662 3      then
: 4663 3      ERR_HRD_RTNE_APT (36)
: 4664 3      else
: 4665 3      ERR_HRD_RTNE (36);
: 4666 3
: 4667 2      end;
: 4668 2
: 4669 3      [ST_CMP]:      begin                                ! COMPARE ERROR
: 4670 3      T_ADDR [ERR_HRD_DAT] = .T_ADDR [ERR_HRD_DAT] + 1;
: 4671 3
: 4672 3      if .APT_MODE
: 4673 3      then
: 4674 3      ERR_HRD_RTNE_APT (37)
: 4675 3      else
: 4676 3      ERR_HRD_RTNE (37);
: 4677 3
: 4678 2      end;
: 4679 2
: 4680 3      [ST_DAT]:      begin                                ! DATA ERROR
: 4681 3
: 4682 3      if .SB_CODE eq 2
: 4683 3      then
: 4684 3      T_ADDR [ERR_HRD_SEK] = .T_ADDR [ERR_HRD_SEK] + 1
: 4685 3      else
: 4686 3      T_ADDR [ERR_HRD_DAT] = .T_ADDR [ERR_HRD_DAT] + 1;
: 4687 3
: 4688 3      if (.SB_CODE eq 0) and
: 4689 3      (not .FORCED_ERROR) and
: 4690 4      (BIT_TST (SWP_FLAGS, SWF_FER))
: 4691 3      then
: 4692 4      begin
: 4693 4      FORCED_ERROR = TRUE;                                ! BLOCK WITH "FORCED ERROR" FOUND
: 4694 4      FER_LBN = .RP_ADDR [LBN_LO];
: 4695 4      FER_BC = .RP_ADDR [CBCNT_LO];
: 4696 3      end;
: 4697 3
: 4698 3      if .APT_MODE
: 4699 3      then
: 4700 3      ERR_HRD_RTNE_APT (38)
: 4701 3      else
: 4702 3      ERR_HRD_RTNE (38);
: 4703 3
: 4704 2      end;
: 4705 2
: 4706 3      [ST_HST]:      begin                                ! HOST ACCESS ERROR
: 4707 3      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
: 4708 3
: 4709 3      if .APT_MODE

```

K13

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0373
Page 142
(39)

```

: 4710 3      then
: 4711 3      ERR_HRD_RTNE_APT (39)
: 4712 3      else
: 4713 3      ERR_HRD_RTNE (39);
: 4714 3
: 4715 2      end;
: 4716 2
: 4717 3      [ST_CNT]: begin
: 4718 3      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;      ! CONTROLLER ERROR
: 4719 3
: 4720 3      if .APT_MODE
: 4721 3      then
: 4722 3      ERR_HRD_RTNE_APT (40)
: 4723 3      else
: 4724 3      ERR_HRD_RTNE (40);
: 4725 3
: 4726 2      end;
: 4727 2
: 4728 3      [ST_DRV]: begin      ! DRIVE ERROR
: 4729 3
: 4730 3      if .SB_CODE eq1 3
: 4731 3      then
: 4732 3      T_ADDR [ERR_HRD_SEK] = .T_ADDR [ERR_HRD_SEK] + 1
: 4733 3      else
: 4734 3      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4735 3
: 4736 3      if .APT_MODE
: 4737 3      then
: 4738 3      ERR_HRD_RTNE_APT (41)
: 4739 3      else
: 4740 3      ERR_HRD_RTNE (41);
: 4741 3
: 4742 2      end;
: 4743 2
: 4744 3      [ST_D1A]: begin      ! MESSAGE FROM INTERNAL DIAGNOSTICS
: 4745 3      T_ADDR [ERR_HRD_DRV] = .T_ADDR [ERR_HRD_DRV] + 1;
: 4746 3
: 4747 3      if .APT_MODE
: 4748 3      then
: 4749 3      ERR_HRD_RTNE_APT (43)
: 4750 3      else
: 4751 3      ERR_HRD_RTNE (43);
: 4752 3
: 4753 2      end;
: 4754 2
: 4755 3      [otherwise]: begin      ! PRINT STATUS CODE IF NO MATCH
: 4756 3      C_ERR_TBL [.CCTLR, C_ERR_HRD] = .C_ERR_TBL [.CCTLR, C_ERR_HRD] + 1;
: 4757 3
: 4758 3      if .APT_MODE
: 4759 3      then
: 4760 3      ERR_HRD_RTNE_APT (45)
: 4761 3      else
: 4762 3      ERR_HRD_RTNE (45);

```


L13

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0374
Page 143
(39)

```

; 4763 3
; 4764 2
; 4765 2
; 4766 2
; 4767 2
; 4768 1
end;
tes;
end;

```

! ROUTINE HARD_ERROR

```

000000 004137 000000G .SBTTL HARD.ERROR MULTI-DRIVE TEST ROUTINES
HARD.ERROR::
000004 013701 000000G JSR R1,$SAVE3 ; 4551
000010 005261 000014 MOV T,ADDR,R1 ; 4567
000014 013702 000000G INC 14(R1)
000020 001030 MOV ST.CODE,R2 ; 4569
000022 013703 000000G BNE 5$ ; 4572
000026 001001 MOV SB.CODE,R3
000030 000207 BNE 1$
000032 012700 000050 RTS PC
000036 060100 1$: MOV #50,R0 ; 4578
000040 020327 000004 ADD R1,R0
000044 001002 CMP R3,#4 ; 4576
000046 105210 BNE 2$
000050 000402 INCB (R0) ; 4578
000052 105260 000001 BR 3$ ; 4576
000056 032737 000001 001162' 2$: INCB 1(R0) ; 4580
000064 001403 3$: BIT #1,APT.MODE ; 4582
000066 012746 000054 BEQ 4$
000072 000556 MOV #54,-(SP) ; 4584
000074 012746 000054 4$: MOV #54,-(SP) ; 4586
000100 000556 BR 16$
000102 020227 000001 5$: CMP R2,#1 ; 4590
000106 001014 BNE 7$
000110 105261 000051 INCB 51(R1) ; 4591
000114 032737 000001 001162' BIT #1,APT.MODE ; 4593
000122 001403 BEQ 6$
000124 012746 000037 MOV #37,-(SP) ; 4595
000130 000567 BR 20$
000132 012746 000037 6$: MOV #37,-(SP) ; 4597
000136 000567 BR 22$
000140 020227 000002 7$: CMP R2,#2 ; 4601
000144 001014 BNE 9$
000146 105261 000050 INCB 50(R1) ; 4602
000152 032737 000001 001162' BIT #1,APT.MODE ; 4604
000160 001403 BEQ 8$
000162 012746 000040 MOV #40,-(SP) ; 4606
000166 000567 BR 24$
000170 012746 000040 8$: MOV #40,-(SP) ; 4608
000174 000567 BR 26$
000176 020227 000003 9$: CMP R2,#3 ; 4610
000202 001036 BNE 11$
000204 105261 000050 INCB 50(R1) ; 4612
000210 032737 000001 001162' BIT #1,APT.MODE ; 4615

```

M13

ZRQAM3	RD/RX EXERCISER	11-Apr-1984 11:08:35	VAX-11 B1:ss-16 V4.0-579	SEQ 03/5
V01.6	MULTI-DRIVE TEST ROUTINES	11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2;6	Page 144 (39)
000216	001415	BEQ	10\$	
000220	012777 000001 001164'	MOV	#1, @MAIL.BOX.TESTNUM	4618
000226	013700 000000G	MOV	CUOFF, R0	4619
000232	006300	ASL	R0	
000234	063700 000000G	ADD	CST, ADDR, R0	
000240	111077 001166'	MOVB	(R0), @MAIL.BOX.SUBTST	
000244	042777 177760 001166'	BIC	#177760, @MAIL.BOX.SUBTST	
000252	104455	TRAP	55	4622
000254	000022	.WORD	22	
000256	000000G	.WORD	EGD.18	
000260	000000G	.WORD	EMS.18	
000262	013700 000000G	MOV	L\$LUN, R0	4623
000266	112760 000005 000000G	MOVB	#5, DUR(R0)	
000274	104451	TRAP	51	4624
000276	000207	RTS	PC	4569
000300	020227 000004	CMP	R2, #4	4627
000304	001036	BNE	15\$	
000306	105261 000050	INCB	50(R1)	4628
000312	032737 000001 001162'	BIT	#1, APT.MODE	4630
000320	001415	BEQ	12\$	
000322	012777 000001 001164'	MOV	#1, @MAIL.BOX.TESTNUM	4633
000330	013700 000000G	MOV	CUOFF, R0	4634
000334	006300	ASL	R0	
000336	063700 000000G	ADD	CST, ADDR, R0	
000342	111077 001166'	MOVB	(R0), @MAIL.BOX.SUBTST	
000346	042777 177760 001166'	BIC	#177760, @MAIL.BOX.SUBTST	
000354	104455	TRAP	55	4637
000356	000030	.WORD	30	
000360	000000G	.WORD	EGD.18	
000362	000000G	.WORD	EMS.24	
000364	013700 000000G	MOV	L\$LUN, R0	4638
000370	112760 000005 000000G	MOVB	#5, DUR(R0)	
000376	104451	TRAP	51	4639
000400	000207	RTS	PC	4569
000402	020227 000005	CMP	R2, #5	4642
000406	001014	BNE	17\$	
000410	105261 000046	INCB	46(R1)	4643
000414	032737 000001 001162'	BIT	#1, APT.MODE	4645
000422	001403	BEQ	15\$	
000424	012746 000043	MOV	#43, -(SP)	4647
000430	000562	BR	35\$	
000432	012746 000043	MOV	#43, -(SP)	4649
000436	000562	BR	37\$	
000440	020227 000006	CMP	R2, #6	4653
000444	001025	BNE	23\$	
000446	012700 000050	MOV	#50, R0	4657
000452	050100	ADD	R1, R0	
000454	023727 000000G 000200	CMP	SB.CODE, #200	4655
000462	001003	CNE	18\$	
000464	105260 000001	INCB	1(R0)	4657
000470	000401	BIT	19\$	4659
000472	105210	INCB	(R0)	4659
000474	032737 000001 001162'	BIT	#1, APT.MODE	4661

N13

ZRQAM3		RD/RX EXERCISER		11-Apr-1984 11:08:35		VAX-11 Bliss-16 V4.0-579		SEQ 0376
V01.6		MULTI-DRIVE TEST ROUTINES		11-Apr-1984 11:08:22		DISK\$USER2:[POWERS]ZRQADO.BL2;6		Page 145 (39)
000502	001403				BEQ	21\$		
000504	012746	000044			MOV	044, -(SP)	:	4663
000510	000577		20\$:		BR	43\$		
000512	012746	000044		21\$:	MOV	044, -(SP)	:	4665
000516	000577		22\$:		BR	45\$		
000520	020227	000007		23\$:	CMP	R2, #7	:	4669
000524	001014				BNE	27\$		
000526	105261	000047			INCB	47(R1)	:	4670
000532	032737	000001	001162'		BIT	01, APT.MODE	:	4672
000540	001403				BEQ	25\$		
000542	012746	000045			MOV	045, -(SP)	:	4674
000546	000577		24\$:		BR	47\$		
000550	012746	000045		25\$:	MOV	045, -(SP)	:	4676
000554	000560			26\$:	BR	45\$		
000556	020227	000010		27\$:	CMP	R2, #10	:	4680
000562	001053				BNE	32\$		
000564	012700	000046			MOV	046, R0	:	4684
000570	060100				ADD	R1, R0		
000572	023727	000000G	000002		CMP	SB.CODE, #2	:	4682
000600	001002				BNE	28\$		
000602	105210				INCB	(R0)	:	4684
000604	000402				BR	29\$	:	4682
000606	105260	000001		28\$:	INCB	1(R0)	:	4686
000612	005737	000000G		29\$:	IST	SB.CODE	:	4688
000616	001023				BNE	30\$		
000620	152737	0.0001	000000G		BITB	01, FORCED.ERROR	:	4689
000626	001017				BNE	30\$		
000630	052737	004000	000000G		BIT	04000, SWP.FLAGS	:	4690
000636	001413				BEQ	30\$		
000640	112737	000001	000000G		MOV8	01, FORCED.ERROR	:	4693
000646	013700	000000G			MOV	RP, ADDR, R0	:	4694
000652	016037	000050	000000G		MOV	50(R0), FER.LBN		
000660	016037	000044	000000G		MOV	44(R0), FER.BC	:	4695
000666	032737	000001	001162'	30\$:	BIT	01, APT.MODE	:	4698
000674	001403				BEQ	31\$		
000676	012746	000046			MOV	046, -(SP)	:	4700
000702	000521				BR	47\$		
000704	012746	000046		31\$:	MOV	046, -(SP)	:	4702
000710	000523				BR	49\$		
000712	020227	000011		32\$:	CMP	R2, #11	:	4706
000716	001014				BNE	34\$		
000720	105261	000051			INCB	51(R1)	:	4707
000724	032737	000001	001162'		BIT	01, APT.MODE	:	4709
000732	001403				BEQ	33\$		
000734	012746	000047			MOV	047, -(SP)	:	4711
000740	000502				BR	47\$		
000742	012746	000047		33\$:	MOV	047, -(SP)	:	4713
000746	000504				BR	49\$		
000750	020227	000012		34\$:	CMP	R2, #12	:	4717
000754	001014				BNE	38\$		
000756	105261	000050			INCB	50(R1)	:	4718
000762	032737	000001	001162'		BIT	01, APT.MODE	:	4720
000770	001403				BEQ	36\$		

1314

JRQAM3
VOL.6

RD-RX EXERCISER
MULTI DRIVE TEST ROUTINES

11 Apr 1984 11:08:35
11 Apr 1984 11:08:22

VAX 11 B1155 14 V4.0 5/79
DISK1USER2:[POWER5]ZRWADO.B1216

SEQ 0377
Page 146
(39)

000772	012746	000050		MOV	050, (SP)		4722
000776	000463		354:	BR	474		
001000	012746	000050	364:	MOV	050, (SP)		4724
001004	000463		374:	BR	494		
001008	020227	000013	384:	CMP	R2,013		4728
001012	001023			BNE	424		
001014	023727	000000; 000003		CMP	SB.CODE,03		4730
001022	001003			BNE	394		
001024	105261	000046		INCB	46(R1)		4732
001030	000402			BR	404		4730
001032	105261	000050	394:	INCB	50(R1)		4734
001036	032737	000001 001162'	404:	BIT	01,APT.MODE		4736
001044	001403			BEQ	414		
001046	012746	000051		MOV	051, (SP)		4738
001052	000433			BR	474		
001054	012746	000051	414:	MOV	051, (SP)		4740
001060	000437			BR	494		
001062	020227	000037	424:	CMP	R2,037		4744
001066	001014			BNE	464		
001070	105261	000050		INCB	50(R1)		4746
001074	032737	000001 001162'		BIT	01,APT.MODE		4748
001102	001403			BEQ	444		
001104	012746	000053		MOV	053, (SP)		4750
001110	000416		434:	BR	474		
001112	012746	000053	444:	MOV	053, (SP)		4752
001116	000420		454:	BR	494		
001120	013700	000000;	464:	MOV	CCTR,RO		4754
001124	006300			ASL	RO		
001126	105260	000000;		INCB	C.ERR,1BL(RO)		
001132	032737	000001 001162'		BIT	01,APT.MODE		4758
001140	001403			BEQ	484		
001142	012746	000055		MOV	055, (SP)		4760
001146	004737	000000V	474:	JSR	PC,ERR,HRD,RTNE,APT		
001152	000404			BR	504		4758
001154	012746	000055	484:	MOV	055, (SP)		4762
001160	004737	000000V	494:	JSR	PC,ERR,HRD,RTNL		
001164	005726		504:	TST	(SP)		4766
001166	000207			RTS	PC		4768

; Routine size: 516 words, Routine Base: 1CODE1 + 21160
; Maximum stack depth per invocation: 6 words

C14

ZRQANS
VOL.6

RD/RX EXERCISER
MULTI DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX 11-B1159-16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL216

SEQ 0378
Page 147
(40)

```

4769 1 GLOBAL routine UPD_IO_TALLY : novalue *
4770 1
4771 1
4772 1
4773 1
4774 1
4775 1
4776 1
4777 1
4778 1
4779 1
4780 1
4781 1
4782 1
4783 1
4784 1
4785 1
4786 2
4787 2
4788 2
4789 2
4790 2
4791 2
4792 3
4793 2
4794 3
4795 3
4796 3
4797 3
4798 3
4799 3
4800 3
4801 3
4802 2
4803 2
4804 3
4805 2
4806 3
4807 3
4808 3
4809 3
4810 3
4811 3
4812 3
4813 2
4814 2
4815 2
4816 3
4817 2
4818 3
4819 3
4820 3
4821 3

GLOBAL routine UPD_IO_TALLY : novalue *

THIS ROUTINE IS CALLED FROM IO RETPKT FOR ALL I/O TRANSFER RETURN
PACKETS WITH "SUCCESS" STATUS CODES. ITS PURPOSE IS TO UPDATE ALL THE
APPROPRIATE STATISTICAL FIELDS FOR THE CURRENT UNIT. A CHECK IS ALSO
MADE ON THE TOTAL NUMBER OF BYTES TRANSFERRED THUS FAR, IF THE
OPERATOR-SPECIFIED LIMIT HAS BEEN REACHED, THEN THE UNIT IS DROPPED.

IMPLICIT INPUTS:
RP_ADDR - ADDRESS OF THE CURRENT RETURN PACKET
T_ADDR - ADDRESS OF THE CURRENT UNIT'S STATISTICS BLOCK (TALLY)
CST_ADDR - ADDRESS OF THE CURRENT CONTROLLER'S CST
COFF - CST OFFSET FOR THE CURRENT UNIT
LUN - CURRENT UNIT NUMBER

begin
local
    THOUSANDS : word,
    MILLIONS : word;

! TOTAL NO. OF BYTES XFERRED TO/FROM A UNIT

! IF ENCODE IS READ
if ,RP_ADDR [ENCODE] eq (OP_RD or OP_END)
then
begin
    T_ADDR [TOT_READS_LO] = ,T_ADDR [TOT_READS_LO] + 1;
    T_ADDR [BYTES_READ_LO] = ,T_ADDR [BYTES_READ_LO] + ,RP_ADDR [BCNT_LO];
    T_ADDR [TOT_BYT_READ_LO] = ,T_ADDR [TOT_BYT_READ_LO] + ,RP_ADDR [BCNT_LO];
    OVF_CHK (T_ADDR [TOT_READS_LO]);
    OVF_CHK (T_ADDR [BYTES_READ_LO]);
    OVF_CHK (T_ADDR [TOT_BYT_READ_LO]);
end
else
    if ,RP_ADDR [ENCODE] eq (OP_WRT or OP_END)
    then
        begin
            T_ADDR [TOT_WRITES_LO] = ,T_ADDR [TOT_WRITES_LO] + 1;
            T_ADDR [BYTES_WRT_LO] = ,T_ADDR [BYTES_WRT_LO] + ,RP_ADDR [BCNT_LO];
            T_ADDR [TOT_BYT_WRT_LO] = ,T_ADDR [TOT_BYT_WRT_LO] + ,RP_ADDR [BCNT_LO];
            OVF_CHK (T_ADDR [TOT_WRITES_LO]);
            OVF_CHK (T_ADDR [BYTES_WRT_LO]);
            OVF_CHK (T_ADDR [TOT_BYT_WRT_LO]);
        end;

! IF ENCODE IS WRITE

! INCREMENT NO. OF WRITES, ADD BYTE COUNT
! CHECK FOR FIELD OVERFLOW

! IF (,RP_ADDR [ENCODE] eq (OP_RD or OP_END)) or
! (,RP_ADDR [ENCODE] eq (OP_WRT or OP_END))
then
begin
    MILLIONS = ,T_ADDR [BYTES_READ_HI] + ,T_ADDR [BYTES_WRT_HI];
    THOUSANDS = ,T_ADDR [BYTES_READ_LO] + ,T_ADDR [BYTES_WRT_LO];
    ! TOTAL BYTES TRANSFERRED

```

014

ZRQAMS
VOL.6

RD-RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX 11 B1155 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL2:6

SEQ.0379
Page 148
(40)

```

: 4822 3      IF ,THOUSANDS gequ 1000
: 4823 3      then
: 4824 4          begin
: 4825 4              MILLIONS = ,MILLIONS + 1;
: 4826 4              THOUSANDS = ,THOUSANDS + 1000;
: 4827 3          end;
: 4828 3
: 4829 3      !
: 4830 3      ! THIS ADDED BECAUSE IT WILL TAKE FOREVER TO TRANSFER ON THE ORDER OF A MEGABYTE TO A FLOPPY
: 4831 3      ! BUT IT IS A MUCH MORE REASONABLE MEASURE FOR THE RD51/52 WINCHESTER. THE QUESTION NOW REFERS TO
: 4832 3      ! THE TOTAL DATA TRANSFER TO THE CONTROLLER AND THIS IS PRETTY CLOSE SINCE THE FLOPPIES GET
: 4833 3      ! ABOUT 1/1000 THE DATA THE HARD DISK(S) GET.
: 4834 3      !
: 4835 3
: 4836 3      IF ,SWP_XFER eq 0
: 4837 3      then
: 4838 4          begin
: 4839 4
: 4840 4              IF ,THOUSANDS gtru 100
: 4841 4              then
: 4842 4                  EOP_FLAG = TRUE;
: 4843 4
: 4844 4              end
: 4845 3          else
: 4846 3
: 4847 3              IF ,MILLIONS gequ ,SWP_XFER
: 4848 3              then
: 4849 3                  EOP_FLAG = TRUE;
: 4850 3
: 4851 2          end;
: 4852 2      ROUND_OUTPUT ();
: 4853 2
: 4854 1      end;

```

! COUNT THE LOWER OVERFLOW TOO!

! IF THERE IS A TRANSFER LIMIT

! SET END-OF-PASS FLAG

! IF TRANSFER LIMIT IS REACHED

! SET END OF PASS FLAG

! IF UNIT IS STILL ALIVE

! ROUND TOTALS TO FIT PRINT POSITIONS

! ROUTINE UPD_IO_TALLY

			UPD_IO_TALLY	UPD_IO_TALLY MULTI DRIVE TEST ROUTINES	
000000	004137	000000G	UPD_IO_TALLY::		
000004	013701	000000G	JSR	R1,\$SAVEP	4'6'
000010	126127	000014 000241	MOV	RP,ADDR,R1	4'9'
000016	001027		CMPB	14(R1),0241	
000020	013700	000000G	BNE	11	
000024	005260	000016	MOV	T,ADDR,R0	4'9'
000030	066110	000020	INC	16(R0)	
000034	066160	000020 000032	ADD	20(R1),(R0)	4'9'
000042	012746	000016	ADD	20(R1),32(R0)	4'9'
000046	060016		MOV	016,(SP)	4'98
000050	004737	000000V	ADD	R0,(SP)	
000054	013716	000000G	JSR	PC,OVE_CHK	
000060	004737	000000V	MOV	T,ADDR,(SP)	4'9'
000064	013716	000000G	JSR	PC,OVE_CHK	
000070	062716	000032	MOV	T,ADDR,(SP)	4800
000074	000435		ADD	052,(SP)	
			BR	21	

ZRQAM3 V01.6	RD/RX EXERCISER MULTI-DRIVE TEST ROUTINES	11-Apr-1984 11:08:55 11-Apr-1984 11:08:22	VAX-11 B1195 15 V4.0 579 DISK\$USER2:[POWERS]ZRQADO.BL2:6	SEQ 0380 Page 149 (40)
000076 126127	000014 000242	1\$: CMPB 14(R1),0242	:	4804
000104 001034		BNE 3:	:	
000106 013700	000000G	MOV T,ADDR,RO	:	4807
000112 005260	000024	INC 24(RO)	:	
000116 066160	000020 000006	ADD 20(R1),6(RO)	:	4808
000124 066160	000020 000040	ADD 20(R1),40(RO)	:	4809
000132 012746	000024	MOV 024, -(SP)	:	4810
000136 060016		ADD RO, (SP)	:	
000140 004737	000000V	JSR PC,OVF,CHK	:	
000144 013716	000000G	MOV T,ADDR,(SP)	:	4811
000150 062716	000006	ADD 06, (SP)	:	
000154 004737	000000V	JSR PC,OVF,CHK	:	
000160 013716	000000G	MOV T,ADDR,(SP)	:	4812
000164 062716	000040	ADD 040, (SP)	:	
000170 004737	000000V	JSR PC,OVF,CHK	:	
000174 005726		TST (SP),	:	4806
000176 013700	000000G	MOV RP,ADDR,RO	:	4815
000202 126027	000014 000241	CMPB 14(RO),0241	:	
000210 001404		BEQ 4:	:	
000212 126027	000014 000242	CMPB 14(RO),0242	:	4816
000220 001034		BNE 8:	:	
000222 013700	000000G	MOV T,ADDR,RO	:	4817
000226 016002	000004	MOV 4(RO),R2	:	
000232 066002	000012	ADD 12(RO),R2	:	
000236 016001	000002	MOV 2(RO),R1	:	4820
000242 066001	000010	ADD 10(RO),R1	:	
000246 020127	001750	CMP R1,01750	:	4822
000252 103403		BLO 5:	:	
000254 005202		INC R2	:	
000256 162701	001750	SUB 01750,R1	:	4825
000262 013700	000000G	MOV SWP,XFER,RO	:	4826
000266 001004		BNE 6:	:	4836
000270 020127	000144	CMP R1,0144	:	
000274 101406		BLOS 8:	:	4840
000276 000402		BR 7:	:	
000300 020200		CMP R2,RO	:	4842
000302 103403		BLO 8:	:	4847
000304 112737	000001 000000G	MOV8 01,EUP,FLAG	:	
000312 004737	000000V	JSR PC,ROUND,OUTPUT	:	4849
000316 000207		RTS PC	:	4769

; Routine Size: 104 words, Routine Base: \$CODE\$ + 22350
; Maximum stack depth per invocation: 5 words

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 81199 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0381
Page 150
(41)

```

: 4855 1 GLOBAL routine OVF_CHK (ADDR) : novalue =
: 4856 1 !+
: 4857 1 ! THIS ROUTINE IS CALLED FROM UPD_IO_TALLY TO CHECK FOR OVERFLOW IN
: 4858 1 ! CERTAIN STATISTICAL FIELDS OF THE CURRENT UNIT. SPECIFICALLY, THE
: 4859 1 ! LOW-ORDER FIELD OF THE NUMBER OF BYTES READ OR WRITTEN IS CHECKED FOR
: 4860 1 ! EXCEEDING 1000. IF TRUE, THEN THE HIGH-ORDER COUNT IS INCREMENTED. IF
: 4861 1 ! THAT EXCEEDS 1000, THEN THE MEGABYTE COUNT IS INCREMENTED.
: 4862 1 !
: 4863 1 ! INPUTS:
: 4864 1 ! ADDR - ADDRESS OF THE BYTES_READ_LO OR BYTES_WRITE_LO FIELD FOR
: 4865 1 ! THE CURRENT UNIT (SEE STATISTIC TABLE (TALLY) LAYOUT)
: 4866 1 !-
: 4867 2 begin
: 4868 2
: 4869 2 while ..ADDR gequ 1000 do ! IF LO-ORDER OVERFLOW
: 4870 3 begin
: 4871 3 .ADDR = ..ADDR - 1000; ! SUBTRACT 1000
: 4872 3 (.ADDR + 2) = .(.ADDR + 2) + 1; ! INCR HI-ORDER
: 4873 2 end;
: 4874 2
: 4875 2 if .(.ADDR + 2) gequ 1000 ! IF HI-ORDER OVERFLOW
: 4876 2 then
: 4877 3 begin
: 4878 3 (.ADDR + 2) = .(.ADDR + 2) - 1000; ! SUBTRACT 1000
: 4879 3 (.ADDR + 4) = .(.ADDR + 4) + 1; ! INCREMENT MBYTES
: 4880 2 end;
: 4881 2
: 4882 1 end; ! ROUTINE OVF_CHK

```

		.SBTTL	OVF_CHK MULTI-DRIVE TEST ROUTINES	
000000	010146	OVF_CHK:		
000002	016600	000004	MOV R1, (SP)	4855
000006	012701	000002	MOV 4(SP), R0	4859
000012	060001		MOV R2, R1	4860
000014	021027	001750	ADD R0, R1	
000020	103404		1\$: CMP (R0), #1750	4869
000022	162710	001750	BLU 2\$	
000026	005211		SUB #1750, (R0)	4871
000030	000771		INC (R1)	4872
000032	021127	001750	BR 1\$	4873
000036	103404		2\$: CMP (R1), #1750	4874
000040	162711	001750	BLU 3\$	
000044	005260	000004	SUB #1750, (R1)	4878
000050	012601		INC 4(R0)	4879
000052	000207		3\$: MOV (SP)+, R1	4880
			RTS PC	4882

; Routine Size: 22 words, Routine Base: \$CODE\$ + 22670
; Maximum stack depth per invocation: 2 words

ZRGAM3
V01.6RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0382
Page 151
(42)

```

: 4883 1 GLOBAL routine ROUND_OUTPUT : novalue =
: 4884 1
: 4885 1 !-
: 4886 1 !- THIS ROUTINE ROUNDS THE TOTALS TO FIT PRINT POSITIONS.
: 4887 1 !-
: 4888 1
: 4889 2 begin
: 4890 2
: 4891 2 if .T_ADDR [TOT_READS_HI] gtru 9999
: 4892 2 then
: 4893 3 begin
: 4894 3
: 4895 3 if .T_ADDR [TOT_READS_LO] lssu 999
: 4896 3 then
: 4897 4 begin
: 4898 4 T_ADDR [TOT_READS_HI] = .T_ADDR [TOT_READS_HI] - 1;
: 4899 4 T_ADDR [TOT_READS_LO] = .T_ADDR [TOT_READS_LO] + 1000;
: 4900 3 end;
: 4901 3
: 4902 3 T_ADDR [TOT_READS_LO] = .T_ADDR [TOT_READS_LO] - 999;
: 4903 3 T_ADDR [TOT_READS_HI] = .T_ADDR [TOT_READS_HI] - 9999;
: 4904 2 end;
: 4905 2
: 4906 2 if .T_ADDR [TOT_WRITES_HI] gtru 9999
: 4907 2 then
: 4908 3 begin
: 4909 3
: 4910 3 if .T_ADDR [TOT_WRITES_LO] lssu 999
: 4911 3 then
: 4912 4 begin
: 4913 4 T_ADDR [TOT_WRITES_HI] = .T_ADDR [TOT_WRITES_HI] - 1;
: 4914 4 T_ADDR [TOT_WRITES_LO] = .T_ADDR [TOT_WRITES_LO] + 1000;
: 4915 3 end;
: 4916 3
: 4917 3 T_ADDR [TOT_WRITES_LO] = .T_ADDR [TOT_WRITES_LO] - 999;
: 4918 3 T_ADDR [TOT_WRITES_HI] = .T_ADDR [TOT_WRITES_HI] - 9999;
: 4919 2 end;
: 4920 2
: 4921 2 if .T_ADDR [MTOT_BYT_RED] gtru 999
: 4922 2 then
: 4923 3 begin
: 4924 3
: 4925 3 if .T_ADDR [TOT_BYT_RED_HI] lssu 999
: 4926 3 then
: 4927 4 begin
: 4928 4 T_ADDR [MTOT_BYT_RED] = .T_ADDR [MTOT_BYT_RED] - 1;
: 4929 4 T_ADDR [TOT_BYT_RED_HI] = .T_ADDR [TOT_BYT_RED_HI] + 1000;
: 4930 3 end;
: 4931 3
: 4932 3 if .T_ADDR [TOT_BYT_RED_LO] lssu 999
: 4933 3 then
: 4934 4 begin
: 4935 4 T_ADDR [TOT_BYT_RED_HI] = .T_ADDR [TOT_BYT_RED_HI] - 1;

```

H14

ZRQAN3
VOL.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX 11 B1199-16 V4.0-5/9
DISK:USER2:[POWERS]ZRQADO.BL2;6

SEQ 0383
Page 152
(42)

```

: 4936 4      T_ADDR [TOT_BYT_RED_LO] = .T_ADDR [TOT_BYT_RED_LO] + 1000;
: 4937 4
: 4938 4      if .T_ADDR [TOT_BYT_RED_HI] lssu 999
: 4939 4      then
: 4940 5          begin
: 4941 5              T_ADDR [MTOT_BYT_RED] = .T_ADDR [MTOT_BYT_RED] - 1;
: 4942 5              T_ADDR [TOT_BYT_RED_HI] = .T_ADDR [TOT_BYT_RED_HI] + 1000;
: 4943 4          end;
: 4944 3      end;
: 4945 3
: 4946 3      T_ADDR [TOT_BYT_RED_LO] = .T_ADDR [TOT_BYT_RED_LO] - 999;
: 4947 3      T_ADDR [TOT_BYT_RED_HI] = .T_ADDR [TOT_BYT_RED_HI] - 999;
: 4948 3      T_ADDR [MTOT_BYT_RED] = .T_ADDR [MTOT_BYT_RED] - 999;
: 4949 2      end;
: 4950 2
: 4951 2      if .T_ADDR [MTOT_BYT_WRT] gtru 999
: 4952 2      then
: 4953 3          begin
: 4954 3
: 4955 3              if .T_ADDR [TOT_BYT_WRT_HI] lssu 999
: 4956 3              then
: 4957 4                  begin
: 4958 4                      T_ADDR [MTOT_BYT_WRT] = .T_ADDR [MTOT_BYT_WRT] - 1;
: 4959 4                      T_ADDR [TOT_BYT_WRT_HI] = .T_ADDR [TOT_BYT_WRT_HI] + 1000;
: 4960 3                  end;
: 4961 3
: 4962 3              if .T_ADDR [TOT_BYT_WRT_LO] lssu 999
: 4963 3              then
: 4964 4                  begin
: 4965 4                      T_ADDR [TOT_BYT_WRT_HI] = .T_ADDR [TOT_BYT_WRT_HI] - 1;
: 4966 4                      T_ADDR [TOT_BYT_WRT_LO] = .T_ADDR [TOT_BYT_WRT_LO] + 1000;
: 4967 4
: 4968 4                      if .T_ADDR [TOT_BYT_WRT_HI] lssu 999
: 4969 4                      then
: 4970 5                          begin
: 4971 5                              T_ADDR [MTOT_BYT_WRT] = .T_ADDR [MTOT_BYT_WRT] - 1;
: 4972 5                              T_ADDR [TOT_BYT_WRT_HI] = .T_ADDR [TOT_BYT_WRT_HI] + 1000;
: 4973 4                          end;
: 4974 3                      end;
: 4975 3
: 4976 3                      T_ADDR [TOT_BYT_WRT_LO] = .T_ADDR [TOT_BYT_WRT_LO] - 999;
: 4977 3                      T_ADDR [TOT_BYT_WRT_HI] = .T_ADDR [TOT_BYT_WRT_HI] - 999;
: 4978 3                      T_ADDR [MTOT_BYT_WRT] = .T_ADDR [MTOT_BYT_WRT] - 999;
: 4979 2                  end;
: 4980 2
: 4981 1      end;

```

```

000000 004137 000000G      ,SBITU ROUND,OUTPUT MULTI-DRIVE TEST ROUTINES
                                ROUND,OUTPUT;;
000004 013700 000000G      JSR      R1,SAVE 3
000010 012702 000020      MOV      T_ADDR,R0
                                MOV      #20,R2

```

4883
4891

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1155-16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO,BL2;6

SEQ 0384
Page 153
(42)

000014	060002		ADD	R0,R2		
000016	021227	023417	CMP	(R2),#23417		
000022	101415		BLOS	2\$		
000024	012701	000016	MOV	#16,R1	:	4895
000030	060001		ADD	R0,R1		
000032	021127	001747	CMP	(R1),#1747		
000036	103003		BHIS	1\$		
000040	005312		DEC	(R2)	:	4898
000042	062711	001750	ADD	#1750,(R1)	:	4899
000046	162711	001747	1\$: SUB	#1747,(R1)	:	4902
000052	162712	023417	SUB	#23417,(R2)	:	4903
000056	012702	000026	2\$: MOV	#26,R2	:	4906
000062	060002		ADD	R0,R2		
000064	021227	023417	CMP	(R2),#23417		
000070	101415		BLOS	4\$		
000072	012701	000024	MOV	#24,R1	:	4910
000076	060001		ADD	R0,R1		
000100	021127	001747	CMP	(R1),#1747		
000104	103003		BHIS	3\$		
000106	005312		DEC	(R2)	:	4913
000110	062711	001750	ADD	#1750,(R1)	:	4914
000114	162711	001747	3\$: SUB	#1747,(R1)	:	4917
000120	162712	023417	SUB	#23417,(R2)	:	4918
000124	012703	000036	4\$: MOV	#36,R3	:	4921
000130	060003		ADD	R0,R3		
000132	021327	001747	CMP	(R3),#1747		
000136	101436		BLOS	7\$		
000140	012701	000034	MOV	#34,R1	:	4925
000144	060001		ADD	R0,R1		
000146	021127	001747	CMP	(R1),#1747		
000152	103003		BHIS	5\$		
000154	005313		DEC	(R3)	:	4928
000156	062711	001750	ADD	#1750,(R1)	:	4929
000162	012702	000032	5\$: MOV	#32,R2	:	4932
000166	060002		ADD	R0,R2		
000170	021227	001747	CMP	(R2),#1747		
000174	103011		BHIS	6\$		
000176	005311		DEC	(R1)	:	4935
000200	062712	001750	ADD	#1750,(R2)	:	4936
000204	021127	001747	CMP	(R1),#1747	:	4938
000210	103003		BHIS	6\$		
000212	005313		DEC	(R3)	:	4941
000214	062711	001750	ADD	#1750,(R1)	:	4942
000220	162712	001747	6\$: SUB	#1747,(R2)	:	4946
000224	162711	001747	SUB	#1747,(R1)	:	4947
000230	162713	001747	SUB	#1747,(R3)	:	4948
000234	012702	000044	7\$: MOV	#44,R2	:	4951
000240	060002		ADD	R0,R2		
000242	021227	001747	CMP	(R2),#1747		
000246	101435		BLOS	10\$		
000250	012701	000042	MOV	#42,R1	:	4955
000254	060001		ADD	R0,R1		
000256	021127	001747	CMP	(R1),#1747		

J14

ZRQAM3	RD/RX EXERCISER	11-Apr-1984 11:08:35	VAX-11 B1196-16 V4.0 5/9	SEQ 0381
V01.6	MULTI-DRIVE TEST ROUTINES	11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQAD0.BL2;6	Page 154
				(42)

000262	103003		BHIS	8\$		
000264	005312		DEC	(R2)	:	4958
000266	062711	001750	ADD	#1750,(R1)	:	4959
000272	062700	000040	ADD	#40,R0	:	4962
000276	021027	001747	CMP	(R0),#1747	:	
000302	103011		BHIS	9\$		
000304	005311		DEC	(R1)	:	4965
000306	062710	001750	ADD	#1750,(R0)	:	4966
000312	021127	001747	CMP	(R1),#1747	:	4968
000316	103003		BHIS	9\$		
000320	005312		DEC	(R2)	:	4971
000322	062711	001750	ADD	#1750,(R1)	:	4972
000326	162710	001747	SUB	#1747,(R0)	:	4976
000332	162711	001747	SUB	#1747,(R1)	:	4977
000336	162712	001747	SUB	#1747,(R2)	:	4978
000342	000207	10\$:	RTS	PC	:	4883

; Routine Size: 114 words, Routine Base: \$CODE\$ + 22744
; Maximum stack depth per invocation: 5 words


```

ZRDQAM3      RD/RX EXERCISER      11-Apr-1984 11:08:35      VAX-11 Bliss-16 V4.0-579      SEQ 0387
VOL16        MULTI-DRIVE TEST ROUTINES 11-Apr-1984 11:08:22      DISK$USER2;[POWERS]ZRDQAD0.BL2:6      Page 156
                                                (43)

```

```

5035 4      T_ADDR [ERR_HARD] = .T_ADDR [ERR_HARD] + 1;
5036 4      T_ADDR [ERR_HRD_HST] = .T_ADDR [ERR_HRD_HST] + 1;
5037 4
5038 4      IF .APT_MODE
5039 4      then
5040 4          ERR_HRD_RTNE_APT (42)                                ! I/O REQUEST FAILED
5041 4      else
5042 4          ERR_HRD_RTNE (42);
5043 4
5044 4      EMS_CMP (RETPKT + (.index * RP_LEN + 2));
5045 4
5046 4      IF .T_ADDR [ERR_HARD] gequ .SWP_ERROR
5047 4      then
5048 5          begin
5049 5              DDU (.L$LUN) = DU_HERR;                                ! IF ERROR COUNT EXCEEDED
5050 5              DDU (.L$LUN);                                ! DROP UNIT
5051 4          end;
5052 4
5053 4      exitloop;                                ! NO NEED TO CONTINUE
5054 3      end;                                ! IF COMPARE ERROR
5055 3
5056 2      end;                                ! IF ASSOCIATED WRITE RETPKT WAS FOUND
5057 2
5058 2      return (.FLAG);
5059 1      end;                                ! ROUTINE HOST WRT CHK

```

ADDRESS	DATA	COMMENT	ROUTINE
000000	004137	000000G	HOST.WRT.CHK::
000004	005746		JSR R1,\$SAVE5
000006	112705	000001	IST -(SP)
000012	013700	000000G	MOVB #1,R5
000016	126027	000014 000242	MOV RP,ADDR,R0
000024	001002		CMPB 14(R0),#242
000026	105005		BNE 1\$
000030	000511		CLRB R5
000032	126027	000014 000241	BR 8\$
000040	001105		1\$: CMPB 14(R0),#241
000042	004737	000000V	BNE 8\$
000046	005700		JSR PC,RPS,REM
000050	002501		TST R0
000052	010046		BLT 8\$
000054	012746	000054	MOV R0,(SP)
000060	004737	000000G	MOV #54, -(SP)
000064	010066	000004	JSR PC,HL,\$MUL
000070	062700	000024G	MOV R0,4(SP)
000074	011001		ADD #RETPKT+24,R0
000076	013700	000000G	MOV (R0),R1
000102	016002	000024	MOV RP,ADDR,R0
000106	016004	000020	MOV 24(R0),R2
000112	005003		MOV 20(R0),R4
000114	000453		CLR R3
			BR 6\$

M14

ZRQAMS	RD/RX EXERCISER	11-Apr-1984 11:08:35	VAX-11 Bliss-16 V4.0-579	SEQ 0388
V01.6	MULTI-DRIVE TEST ROUTINES	11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2;6	Page 157 (43)

000116	121112		2\$:	CMPB	(R1),(R2)		; BUFF1,BUFF2	5027
000120	001003			BNE	3\$			
000122	005201			INC	R1		; BUFF1	5030
000124	005202			INC	R2		; BUFF2	5031
000126	000446			BR	6\$			5027
000130	013700	000000G	3\$:	MOV	T,ADDR,RO			5035
000134	005260	000014		INC	14(RO)			
000140	105260	000051		INCB	51(RO)			5036
000144	032737	000001 001162'		BIT	01,APT,MODE			5038
000152	001405			BEQ	4\$			
000154	012716	000052		MOV	052,(SP)			5040
000160	004737	000000V		JSR	PC,ERR,HRD,RTNE,APT			
000164	000404			BR	5\$			5038
000166	012716	000052	4\$:	MOV	052,(SP)			5042
000172	004737	000000V		JSR	PC,ERR,HRD,RTNE			
000176	016616	000004	5\$:	MOV	4(SP),(SP)			5044
000202	062716	000000G		ADD	0RETPKT,(SP)			
000206	004737	000000G		JSR	PC,EMS,CMP			
000212	013700	000000G		MOV	T,ADDR,RO			5046
000216	026037	000014 000000G		CMP	14(RO),SWP.ERROR			
000224	103412			BLO	7\$			
000226	013700	000000G		MOV	L\$LUN,RO			5049
000232	112760	000004 000000G		MOVB	04,DUR(RO)			
000240	104451			TRAP	51			5050
000242	000403			BR	7\$			5034
000244	005203		6\$:	INC	R3		; I	5025
000246	020304			CMP	R3,R4		; I,COUNT	
000250	003722			BLE	2\$			
000252	022626		7\$:	CMP	(SP)+,(SP)+			5019
000254	005000		8\$:	CLR	RO			5058
000256	150500			BISB	R5,RO		; FLAG,+	
000260	005726			IST	(SP)+			4982
000262	000207			RTS	PC			

; Routine Size: 90 words, Routine Base: \$CODE\$ + 23310
; Maximum stack depth per invocation: 11 words

N14

ZRQAM3
V01.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1195-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0389
Page 158
(44)

```

: 5060 1 GLOBAL routine SWEEP : novalue =
: 5061 1
: 5062 1 !
: 5063 1 ! THIS ROUTINE IS CALLED FROM IO_RETPKT AND OTHERS TO DEALLOCATE THE
: 5064 1 ! RESOURCES ASSOCIATED WITH THE CURRENT RETURN PACKET. THIS INCLUDES THE
: 5065 1 ! PACKET ITSELF AND THE I/O BUFFER. IN ADDITION, IF THE HOST IS
: 5066 1 ! PERFORMING WRITE-COMPARES, AND IF THE CURRENT RETURN PACKET IS A READ
: 5067 1 ! FUNCTION, THEN THE CURRENT CONTROLLER'S RP_SAVE AREA IS SEARCHED FOR
: 5068 1 ! THE ASSOCIATED WRITE RETPKT SO THAT ITS RESOURCES CAN ALSO BE
: 5069 1 ! DEALLOCATED.
: 5070 1 !
: 5071 1 ! IMPLICIT INPUTS:
: 5072 1 ! RP_ADDR - ADDRESS OF CURRENT RETURN PACKET
: 5073 1 ! RP_INDX - INDEX OF CURRENT RETURN PACKET
: 5074 1 !-
: 5075 1
: 5076 2 begin
: 5077 2
: 5078 2 local
: 5079 2 index : signed word;
: 5080 2
: 5081 2 if (.RP_ADDR [ENDCOD] and OP_MSK) eq1 OP_RD ! IF READ OPCODE OR ENDCODE
: 5082 2 then
: 5083 2
: 5084 3 if BIT_TST (SWP_FLAGS, SWF_HWC) ! IF HOST IS DOING WRITE-COMPARES
: 5085 2 then
: 5086 2
: 5087 2 if (index = RPS_REM ()) geq 0 ! IF ASSOCIATED WRITE RETPKT IS FOUND
: 5088 2 then
: 5089 3 begin
: 5090 3 PUT_IO_BUFF (RETPKT (.index, BUFF_0)); ! RETURN WRITE I/O BUFFER TO POOL
: 5091 3 PUT_RETPKT (.index); ! RETURN WRITE PACKET TO POOL
: 5092 2 end;
: 5093 2
: 5094 2 PUT_IO_BUFF (RP_ADDR [BUFF_0]); ! RETURN CURRENT I/O BUFFER TO POOL
: 5095 2 PUT_RETPKT (.RP_INDX); ! RETURN CURRENT RETPKT TO POOL
: 5096 1 end; ! ROUTINE SWEEP

```

000000	010146		.SBTTL	SWEEP MULTI-DRIVE TEST ROUTINES	
000002	013700	000000G	SWEEP::	MOV R1, (SP)	5060
000006	116000	000014		MOV RP_ADDR, R0	5081
000012	042700	177600		MOVB 14(R0), R0	
000016	020027	000041		BIC #177600, R0	
000022	001026			CMP R0, #41	
000024	032737	000040 000000G		BNE 1\$	
000032	001422			BIT #40, SWP_FLAGS	5084
000034	004737	000000V		BEG 1\$	
000040	010001			JSR PC, RPS_REM	5087
000042	002416			MOV R0, R1	
000044	010146			BIT 1\$	
000046	012746	000054		MOV R1, (SP)	5090
				MOV #54, -(SP)	
					INDEX, *

BT

ZRQAMS
VOL.6

RD-RX EXERCISER
MULTI DRIVE TEST ROUTINES

11 Apr 1984 11:08:35
11 Apr 1984 11:08:22

VAX 11-811-14 V4.0 5/79
DISKUSER2:[POWER5]ZRQADO.BL2:6

SEQ 0390
Page 159
(44)

000052	004737	0000000	JSR	PC,BLIMUL		
000056	062700	0000240	ADD	#RETPKT,24,R0		
000060	010016		MOV	R0,(SP)		
000064	004737	0000000	JSR	PC,PUT_IO_BUF		
000070	010116		MOV	R1,(SP)	INDEX,*	5091
000074	004737	0000000	JSR	PC,PUT_RETPKT		
000078	022626		CMP	(SP), (SP)		5089
000100	013746	0000000	MOV	RP,ADDR, (SP)		5094
000104	062716	000024	ADD	#24,(SP)		
000110	004737	0000000	JSR	PC,PUT_IO_BUF		
000114	013716	0000000	MOV	RP,INDX, (SP)		5095
000120	004737	0000000	JSR	PC,PUT_RETPKT		
000124	005726		TEST	(SP)		5096
000128	012601		MOV	(SP),R1		5060
000130	000707		RTN	PC		

! Routine size: 45 words, Routine Base: \$CODE\$ + 23574
! Maximum stack depth per invocation: 4 words

C15

ZRQAMS
Vol.6

RD RX EXERCISER
MULTI DRIVE TEST ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX 11 B11: 16 V4.0 5/9
DISK#USER2:[POWERS]ZRQADO.B1 2:6

SEQ 0391
Page 160
(45)

```

: 5097 1 GLOBAL routine RPS_REM *
: 5098 1
: 5099 1
: 5100 1 THIS ROUTINE SEARCHES THE CURRENT CONTROLLER'S RP SAVE AREA FOR A
: 5101 1 RETURN PACKET WHOSE COMMAND REFERENCE NUMBER (CRN) IS ONE LESS THAN THE
: 5102 1 CRN OF THE CURRENT RETURN PACKET (I.E., SEARCHING FOR THE SAVED WRITE
: 5103 1 OPERATION ASSOCIATED WITH THE CURRENT READ OPERATION). IF FOUND, THE
: 5104 1 RP SAVE ENTRY IS CLEARED (TO 1) AND THE RETPKT INDEX OF THE WRITE
: 5105 1 OPERATION IS RETURNED TO THE CALLER.
: 5106 1
: 5107 1 IMPLICIT INPUTS:
: 5108 1 RP_ADDR ADDRESS OF THE CURRENT RETURN PACKET
: 5109 1
: 5110 1 OUTPUTS:
: 5111 1 INDEX (VALUE OF THIS ROUTINE) - INDEX OF THE RETPKT CONTAINING
: 5112 1 A CRN WHICH IS ONE LESS THAN THE CURRENT
: 5113 1
: 5114 1
: 5115 2 begin
: 5116 2
: 5117 2 local
: 5118 2 index : signed word initial (-1);
: 5119 2
: 5120 2 incr COUNT from 0 to RP CNT 1 do
: 5121 2
: 5122 2 if (.RP_USE [.COUNT] eal .CCTLR) and
: 5123 2 (.RETPKT [.COUNT, ENCOD] eal (OP_WRT or OP_END))
: 5124 2 then
: 5125 2
: 5126 2 if ((.RETPKT [.COUNT, CRF_LO] eal (.RP_ADDR [CRF_LO] - 1)) and
: 5127 2 (.RETPKT [.COUNT, CRF_HI] eal .RP_ADDR [CRF_HI])) or
: 5128 2 ((.RETPKT [.COUNT, CRF_HI] eal (.RP_ADDR [CRF_HI] - 1)) and
: 5129 2 (.RETPKT [.COUNT, CRF_LO] eal #0'177777')) and
: 5130 2 (.RP_ADDR [CRF_LO] eal 0)
: 5131 2 then
: 5132 2 begin
: 5133 2 index = .COUNT;
: 5134 2 exitloop;
: 5135 2 end;
: 5136 2
: 5137 2 return .index;
: 5138 1 end;

```

000000	004157	000000G			
			RPS_REM:	1: 177777	
000004	012704	177777			
000010	005003				
000017	116300	000000G	13:		
000016	020037	000000G			
000017	001053				
000024	010346				

[] 1

JRQAM3
V01.6

RD/RX EXERCISER
MULTI DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11 Apr 1984 11:08:22

VAX-11 B1199 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 039:
Page 161
(45)

000026	012746	000054	MOV	054, (SP)		
000032	004737	0000006	JSR	PC, BL \$MU		
000036	022626		CMP	(SP)+, (SP)+		
000040	126027	0000146 000242	CMPH	RE TPKT+14(R0), 0242		
000046	001041		BNE	41		
000050	010346		MOV	R3, (SP)	: COUNT, +	5126
000052	012746	000054	MOV	054, -(SP)		
000056	004737	0000006	JSR	PC, BL \$MU		
000062	022626		CMP	(SP)+, (SP)+		
000064	013701	0000006	MOV	RP, ADDR, R1		
000070	016102	000004	MOV	4(R1), R2		
000074	005302		DEC	R2		
000076	026002	0000046	CMP	RE TPKT+4(R0), R2		
000102	001004		BNE	21		
000104	026061	0000066 000006	CMP	RE TPKT+6(R0), 6(R1)	:	5127
000112	001415		BEQ	31		
000114	016102	000006	MOV	6(R1), R2	:	5128
000120	005302		DEC	R2		
000122	026002	0000066	CMP	RE TPKT+6(R0), R2		
000126	001011		BNE	41		
000130	026027	0000046 177777	CMP	RE TPKT+4(R0), 0-1	:	5129
000136	001005		BNE	41		
000140	005761	000004	TST	4(R1)	:	5130
000144	001002		BNE	41		
000146	010304	31:	MOV	R3, R4	: COUNT, INDEX	5133
000150	000404		BR	51	:	5134
000152	005203	41:	INC	R3	: COUNT	5130
000154	020327	000007	CMP	R3, 07	: COUNT, +	
000160	003714		BLE	11		
000162	010400	51:	MOV	R4, R0	: INDEX, +	5115
000164	000207		RTS	PC	:	5097

: Routine Size: 59 words, Routine Base: \$CODE\$ + 23726
: Maximum stack depth per invocation: 8 words

PROGRAMS
VOL.6

RD/RX EXERCISER
MULTI-DRIVE TEST ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX 11 Release 16 V4.0 5/9
DISK\$USER2:[POWER5]ZRDAD0.BL2;6

SEQ 0593
Page 162
(46)

```

: 5139 1 GLOBAL routine DR RETPKT : novalue
: 5140 1
: 5141 1 !
: 5142 1 ! THIS ROUTINE IS CALLED BY PROC RETPKT FOR ALL PACKETS ORIGINATING AT
: 5143 1 ! THE "DRIVER" PORTION OF THE PROGRAM. THIS INCLUDES PACKETS DESCRIBING
: 5144 1 ! FATAL DEVICE ERRORS.
: 5145 1 !
: 5146 1 ! FOR FATAL DEVICE ERRORS, THIS ROUTINE RELEASES ALL RESOURCES HELD BY
: 5147 1 ! THE CONTROLLER. THE CONTROLLER IS MARKED OFFLINE IN ITS CST, AND ALL
: 5148 1 ! UNITS ATTACHED TO THE CONTROLLER ARE DROPPED.
: 5149 1 !
: 5150 1 ! IMPLICIT INPUTS:
: 5151 1 ! RP_INDEX INDEX OF THE CURRENT RETURN PACKET
: 5152 1 ! RP_ADDR ADDRESS OF THE CURRENT RETURN PACKET
: 5153 1 ! CST_ADDR ADDRESS OF THE CURRENT CONTROLLER'S CST
: 5154 1 ! CCTLN - CURRENT CONTROLLER NUMBER
: 5155 1 !-
: 5156 1
: 5157 2 begin
: 5158 2
: 5159 2
: 5160 2 PUTA BUFF (0); ! RELEASE ALL I/O BUFFERS HELD BY CONTROLLER
: 5161 2
: 5162 2 incr index from 0 to RP_CNT - 1 do ! FOR EACH ENTRY IN CONTROLLER'S RP SAVE
: 5163 2
: 5164 2 IF (RP_USE (.index) eq) .CCTLN ! IF VALID RETPKT INDEX
: 5165 2 then
: 5166 2 PUT RETPKT (.index); ! RETURN RETPKT TO POOL
: 5167 2
: 5168 2 QIO (.CCTLN) = 0; ! CLEAR NO. OF OUTSTANDING QIOs
: 5169 2 CST_ADDR (STATE) = OFFLINE; ! MARK CST OFFLINE
: 5170 2 DROP CCTLN (.CCTLN, DU CFATAL); ! DROP CONTROLLER'S UNITS
: 5171 2 PUT RETPKT (.RP_INDEX); ! PUT BACK RETPKT
: 5172 1 end; ! ROUTINE DR RETPKT

```

			SBTTL	DR.RETPKT MULTI-DRIVE TEST ROUTINES	
000000	010146		DR.RETPKT::		
000002	004737	000000G	MOV	R1, (SP)	5139
000006	005001		JSR	PC,PUTA,BUFF	5140
000010	116100	000000G	CLR	R1	5141
000014	020037	000000G	13: MOV8	RP,USE(R1),RO	5142
000020	001004		CMP	RO,CCTLN	5143
000022	010146		BNE	23	5144
000024	004737	000000G	MOV	R1, (SP)	5145
000030	005726		JSR	PC,PUT,RETPKT	5146
000032	005201		IST	(SP)	5147
000034	020127	000007	23: INC	R1	5148
000040	003763		CMP	R1,27	5149
000042	013701	000000G	BLE	13	5150
000046	105061	000000G	MOV	CCTLN,R1	5151
000052	013700	000000G	CLRB	QIO(R1)	5152
			MOV	CST.ADDR,RO	5153

115

ZRQAMS
V01.6

RD/RX EXERCISER
MULTI DRIVE TEST ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX 11 Bitbus 16 V4.0 579
DISK\$USER2:[POWERS]ZRQAD0.BL2:6

SEQ 0394
Page 163
(46)

000056	042760	100000	000002	BIC	#100000,2(R0)		
000064	010146			MOV	R1,-(SP)	:	5170
000066	012746	000006		MOV	#6,-(SP)		
000072	004737	000000G		JSR	PC,DROP.CILR		
000076	013716	000000G		MOV	RP,INDX,(SP)	:	5171
000102	004737	000000G		JSR	PC,PUT.RETPKT		
000106	022626			CMP	(SP)+,(SP)+	:	5157
000110	012601			MOV	(SP)+,R1	:	5139
000112	000207			RTS	PC		

; Routine Size: 58 words, Routine Base: \$CODE\$ + 24114
; Maximum stack depth per invocation: 4 words

015

ZRQAM3 RD/RX EXERCISER 11-Apr-1984 11:08:35 VAX 11 Bliss 16 V4.0 579 SEQ 0395
 V01.6 RDRX INTERRUPT SERVICE ROUTINES 11-Apr-1984 11:08:22 DISK\$USER2:[POWERS]ZRQADO.BL2;6 Page 164
 (47)

```

: 5173 1 *sbttl 'RDRX INTERRUPT SERVICE ROUTINES'
: 5174 1
: 5175 1 !
: 5176 1 ! THERE EXISTS AN RDRX INTERRUPT SERVICE ROUTINE FOR EACH DEVICE
: 5177 1 ! CONTROLLER. EACH SERVICE ROUTINE BEGINS BY SIMPLY SETTING THE
: 5178 1 ! APPROPRIATE CONTROLLER NUMBER INTO "ICTLR". ALL SERVICE ROUTINES THEN
: 5179 1 ! BRANCH TO A COMMON INTERRUPT PROCESSING ROUTINE.
: 5180 1 !
: 5181 1
: 5182 2 BGN$RV (AZINT0);
: 5183 2 ICTLR = 0;
: 5184 2 AZINT ();
: 5185 1 ENDSRV;

```

```

000000 010046          ,SBTTL AZINT0 RDRX INTERRUPT SERVICE ROUTINES
000002 005037 000104' AZINT0:;MOV    R0, -(SP)
000006 004737 000000V    CLR      ICTLR
000012 012600          JSR      PC,AZINT
000014 000002          MOV      (SP)+,R0
                                RTI

```

; Routine Size: 7 words, Routine Base: \$CODE\$ + 24230
 ; Maximum stack depth per invocation: 2 words

H15

ZRQAMS
V01.6

RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX 11 B11aa 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL216

SEQ 0796
Page 155
(48)

```

: 5186 1 GLOBAL routine AZINT : novalue =
: 5187 1
: 5188 1 !
: 5189 1 ! THIS IS THE COMMON INTERRUPT SERVICE ROUTINE FOR ALL RDRX CONTROLLERS.
: 5190 1 ! AFTER CALCULATING THE DCT ADDRESS FOR THE INTERRUPTING DEVICE, THIS
: 5191 1 ! ROUTINE WILL SAVE THE CURRENT CONTENTS OF THE SA REGISTER IN THE DCT.
: 5192 1 ! THEN, IF THE "IGNORE INTERRUPT" BIT IS SET, NO FURTHER ACTION IS TAKEN.
: 5193 1 ! OTHERWISE, THE SA VALUE IS CHECKED FOR A FATAL ERROR, AND THE COMMAND
: 5194 1 ! AND RESPONSE RINGS ARE POLLED.
: 5195 1 !
: 5196 1 !
: 5197 2 begin
: 5198 2 IDCT_ADDR = DCT + (.ICTLR * DCT_LEN * 2); ! GET DCT ADDRESS
: 5199 2 ICST_ADDR = CST + (.ICTLR * CST_LEN * 2); ! GET CST ADDRESS
: 5200 2 IRDRX_ADDR = ICST_ADDR [IP_ADDR]; ! GET RDRX ADDRESS
: 5201 2 ICOM_ADDR = COMM_AREA + (.ICTLR * COMM_LEN * 2); ! GET COMM AREA ADDR
: 5202 2 IDCT_ADDR [SA_SAVE] = IRDRX_ADDR [RCSA, RC_ALL]; ! SAVE SA REGISTER
: 5203 2
: 5204 2 if .IDCT_ADDR [IG_INT] ! IGNORE INTERRUPT?
: 5205 2 then
: 5206 2 return; ! RETURN IF INTERRUPTS IGNORED
: 5207 2
: 5208 3 if BIT_TST (IDCT_ADDR [SA_SAVE], SA_ERR) ! IF FATAL ERROR
: 5209 2 then
: 5210 2 FATAL_ERROR ()
: 5211 2 else
: 5212 3 begin
: 5213 3 POLL_CRING (); ! POLL COMMAND RING
: 5214 3 POLL_RRING (); ! POLL RESPONSE RING
: 5215 2 end;
: 5216 2
: 5217 1 end;

```

000000	010146		.SBTL	AZINT RDRX INTERRUPT SERVICE ROUTINES	
000002	005746		AZINT::	MOV R1, (SP)	5186
000004	013701	000104		IST (SP)	
000010	010146			MOV ICTLR, R1	5188
000012	012746	000022		MOV R1, (SP)	
000016	004737	000000G		MOV #22, (CP)	
000022	062700	000000G		JSR PC, BL \$MUL	
000026	010037	000100		ADD #DCT, R0	
000032	010116			MOV R0, IDCT_ADDR	
000034	012746	000126		MOV R1, (SP)	5198
000040	004737	000000G		MOV #126, (CP)	
000044	062700	000000G		JSR PC, BL \$MUL	
000050	010037	000076		ADD #CST, R0	
000054	011037	000000G		MOV R0, ICST_ADDR	
000060	010116			MOV (R0), IRDRX_ADDR	5199
000062	012746	000050		MOV R1, (SP)	
000066	004737	000000G		MOV #50, (CP)	
000072	062700	000000		JSR PC, BL \$MUL	
				ADD #COMM_AREA, R0	5201

ZRQAM5	RD/RX EXERCISER	11-Apr-1984 11:08:35	VAX-11 B1196 16 V4.0 579	SEQ 0597
V01.6	RDRX INTERRUPT SERVICE ROUTINES	11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2:6	Page 166 (48)

000076	010037	0000747	MOV	RO,ICOM.ADDR		
000102	013701	0001007	MOV	IDCT.ADDR,R1	:	5202
000106	013700	000000G	MOV	IRDRX.ADDR,RO	:	
000112	016066	000202 000010	MOV	2(RO),10(SP)	: *,RC.REG	
000120	016661	000010 000002	MOV	10(SP),2(R1)	: RC.REG,*	
000126	032711	040000	BIT	040000,(R1)	: *,IDCT.ADDR	5204
000132	001016		BNE	2*	:	5186
000134	016601	000010	MOV	10(SP),R1	:	5208
000140	042701	077777	RIC	077777,R1	:	
000144	020127	100000	CMP	R1,0-100000	:	
000150	001003		BNE	1*	:	
000152	004737	000000V	JSR	PC,FATAL.ERROR	:	5210
000156	000404		BR	2*	:	5208
000160	004737	000000V	JSR	PC,POLL.CRING	:	5214
000164	004737	000000V	JSR	PC,POLL.HRRING	:	5214
000170	062706	000012	ADD	012,SP	:	5186
000174	012601		MOV	(SP)+,R1	:	
000176	000207		RTS	PC	:	

: Routine Size: 64 words, Routine Base: \$CODE\$ + 24246
: Maximum stack depth per invocation: 7 words

: 5218 1

ZQAMS
VOL.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:55
11-Apr-1984 11:08:22VAX-11 B11sg 16 V4.0-579
DISK\$USER2:[POWERS]ZQADO.BL2:6SEQ 0398
Page 167
(49)

```

: 5219 1
: 5220 1
: 5221 1  !+
: 5222 1 GLOBAL ROUTINE DUP_RSP : NOVALUE +
: 5223 1
: 5224 1  !+
: 5225 1 ! THIS ROUTINE IS CALLED BY POLL RING FOR EACH DUP RESPONSE.
: 5226 1 ! ITS GENERAL PURPOSE IS TO ACT ON A DATAGRAM OR SEQUENTIAL MESSAGE.
: 5227 1 ! IF THE MESSAGE TYPE IS SEQUENTIAL, THE ROUTINE COPIES THE
: 5228 1 ! CONTENTS OF THE MESSAGE ENVELOPE INTO A RETURN PACKET SO THAT THE
: 5229 1 ! ENVELOPE CAN BE RETURNED TO THE CONTROLLER.
: 5230 1 !
: 5231 1 ! IMPLICIT INPUTS:
: 5232 1 ! ICTLR - INTERRUPTING CONTROLLER NUMBER
: 5233 1 ! IPKT_ADDR - ADDRESS OF MSCP ENVELOPE CONTAINING RESPONSE
: 5234 2
: 5235 2 begin
: 5236 2 local
: 5237 2   R_INDEX : signed word,
: 5238 2   SRC_ADDR,
: 5239 2   DST_ADDR,
: 5240 2   R_ADDR : ref block [RP_LEN, word] field (RP_FIELDS);
: 5241 2 !PRINTX (DER34);
: 5242 2
: 5243 2 incr COUNT from 0 to PKT_CNT - 1 do
: 5244 2
: 5245 2   if (.MSCP_PKT [.COUNT, CRN_LO] eq .IPKT_ADDR [CRN_LO]) and      ! IF THIS IS THE ASSOC CMD
: 5246 2   (.MSCP_PKT [.COUNT, CRN_HI] eq .IPKT_ADDR [CRN_HI]) and
: 5247 2   (.MSCP_PKT [.COUNT, PKT_LO] neq .IPKT_ADDR [PKT_LO]) and
: 5248 2   ((.MSCP_PKT [.COUNT, OPCODE] and OP_END) neq OP_END) and
: 5249 2   (.MSCP_PKT [.COUNT, CONNID] eq CID_DUP) and
: 5250 2   ((.IPKT_ADDR [OPCODE] and OP_END) eq OP_END)
: 5251 2 then
: 5252 2   begin
: 5253 2     P_INDEX = .COUNT;          ! SET PKT NUMBER
: 5254 2     exitloop;
: 5255 2   end;
: 5256 2
: 5257 2 if .P_INDEX lss 0
: 5258 2 then
: 5259 2   begin
: 5260 2     PRINTX (DBM108, .IPKT_ADDR [CRN_LO]); ! UNKNOWN COMMAND REF. NUMBER
: 5261 2     return;
: 5262 2   end;
: 5263 2
: 5264 2 if (R_INDEX = GET_RET_PKT (.ICTLR)) lss 0 ! IF RETPKT IS NOT AVAILABLE
: 5265 2 then
: 5266 2   PRINTX (DBM112) ! "DUP_RSP: RETPKT NOT AVAILABLE"
: 5267 2 else
: 5268 2   begin
: 5269 2     SRC_ADDR = .IPKT_ADDR + 6; ! SET UP COPY (SKIP OVER PKT DESC)
: 5270 2     R_ADDR = DST_ADDR = RETPKT + (.R_INDEX * RP_LEN * 2); ! START OF ALLOCATED RETPKT
: 5271 2

```

K15

ZRQAMS RDRX EXERCISER 11-Apr-1984 11:08:35 VAX-11 B11ss 16 V4.0 579 SEQ 0399
 V01.6 RDRX INTERRUPT SERVICE ROUTINES 11-Apr-1984 11:08:22 DISK\$USER2:[POWERS]ZRQADO.BL2;6 Page 168
 (49)

```

: 5272 3      incr COUNT from 1 to RP_LEN do
: 5273 4      begin
: 5274 4          DST_ADDR = ., SRC_ADDR;
: 5275 4          DST_ADDR = .DST_ADDR + 2;
: 5276 4          SRC_ADDR = .SRC_ADDR + 2;
: 5277 3      end;
: 5278 3
: 5279 3      IN_IODQ (.R_INDEX);
: 5280 2      end;
: 5281 2
: 5282 2
: 5283 2      if .P_INDEX geq 0
: 5284 2      then
: 5285 2          PUT_PKT (.P_INDEX);
: 5286 2
: 5287 1      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:		
000004	013701	000000G	JSR	R1,\$SAVE3	5221
000010	005002		MOV	IPKT.ADDR,R1	5245
000012	010246		CLR	R2	5243
000014	012746	000106	1\$: MOV	R2,(SP)	5245
000020	004737	000000G	MOV	#106,-(SP)	
000024	022626		JSR	PC,BL\$MUL	
000026	026061	000012G 000012	CMP	(SP)+,(SP)+	
000034	001024		CMP	MSCP.PKT+12(R0),12(R1)	
000036	026061	000014G 000014	BNE	2\$	
000044	001020		CMP	MSCP.PKT+14(R0),14(R1)	5246
000046	026011	000000G	BNE	2\$	
000052	001415		CMP	MSCP.PKT(R0),(R1)	5247
000054	105760	000022G	BEQ	2\$	
000060	100412		TSTB	MSCP.PKT+22(R0)	5248
000062	126027	000011G 000002	BMI	2\$	
000070	001006		CMPB	MSCP.PKT+11(R0),#2	5249
000072	105761	000022	BNE	2\$	
000076	100003		TSTB	22(R1)	5250
000100	010237	000000G	BPL	2\$	
000104	000406		MOV	R2,P_INDEX	5253
000106	005202		BR	3\$	5252
000110	020227	000013	2\$: INC	R2	5243
000114	003736		CMP	R2,#13	
000116	005737	000000G	BLE	1\$	
000122	002013		TST	P_INDEX	5251
000124	016146	000012	3\$: BGE	4\$	
000130	012746	000000G	MOV	12(R1),-(SP)	5260
000134	012746	000002	MOV	#DBM108,-(SP)	
000140	010600		MOV	#2,-(SP)	
000142	104417		MOV	SP,R0	5254
000144	062706	000006	TRAP	17	
000150	000207		ADD	#6,SP	5251
			RTS	PC	5259

ZRQAM3	RD/RX EXERCISER	11-Apr-1984 11:08:35	VAX-11 H11sg-16 V4.0-579	SEQ 0400
V01.6	RDRX INTERRUPT SERVICE ROUTINES	11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2;6	Page 169 (49)

000152	013746	000104	4\$:	MOV	ICTLR, -(SP)	:	5264
000156	004737	000000G		JSR	PC, GET, RETPKT	:	
000162	010001			MOV	R0, R1	: *, R. INDEX	
000164	005726			TST	(SP)+	:	
000166	005701			TST	R1	: R. INDEX	
000170	002007			BGE	5\$	:	
000172	012746	000000G		MOV	#DBM112, -(SP)	:	5266
000176	012746	000001		MOV	#1, -(SP)	:	
000202	010600			MOV	SP, R0	: SP, *	
000204	104417			TRAP	17	:	
000206	000424			BR	7\$	:	5264
000210	013702	000000G	5\$:	MOV	IPK1, ADDR, R2	: *, SRC. ADDR	5269
000214	062702	000006		ADD	#6, R2	: *, SRC. ADDR	
000220	010146			MOV	R1, -(SP)	: R. INDEX, *	5270
000222	012746	000054		MOV	#54, -(SP)	:	
000226	004737	000000G		JSR	PC, BL\$MUL	:	
000232	062700	000000G		ADD	#RETPKT, R0	:	
000236	010003			MOV	R0, R3	: *, DST. ADDR	
000240	012700	000026		MOV	#26, R0	: *, COUNT	5272
000244	012223		6\$:	MOV	(R2)+, (R3)+	: SRC. ADDR, DST. ADDR	5274
000246	005300			DEC	R0	: COUNT	5272
000250	001375			BNE	6\$	:	
000252	010116			MOV	R1, (SP)	: R. INDEX, *	5279
000254	004737	000000G		JSR	PC, IN. I00Q	:	
000260	013700	000000G	7\$:	MOV	P. INDEX, R0	:	5283
000264	002403			BLT	8\$	:	
000266	010016			MOV	R0, (SP)	:	5285
000270	004737	000000G		JSR	PC, PUT, PKT	:	
000274	022626		8\$:	CMP	(SP)+, (SP)+	:	5284
000276	000207			RTS	PC	:	5221

: Routine Size: 96 words, Routine Base: \$CODE\$ + 24446
 : Maximum stack depth per invocation: 9 words

: 5288 1

M15

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0401
Page 170
(50)

```

: 5289 1 GLOBAL routine FATAL_ERROR : novalue =
: 5290 1
: 5291 1 !
: 5292 1 ! THIS ROUTINE IS CALLED BY THE INTERRUPT SERVICE ROUTINE (AZINT) UPON
: 5293 1 ! DETECTING AN UNRECOVERABLE ERROR THROUGH THE DEVICE'S SA REGISTER.
: 5294 1 ! ITS PURPOSE IS TO CLEAN UP DEVICE DATA IN THE "DRIVER" PORTION OF
: 5295 1 ! THE EXERCISER, AND TO INFORM THE "PROGRAM" PORTION OF THE EVENT VIA
: 5296 1 ! RETURN PACKET.
: 5297 1 !
: 5298 1 ! IMPLICIT INPUTS:
: 5299 1 ! ICTLR - INTERRUPTING CONTROLLER NUMBER
: 5300 1 ! IDCT_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 5301 1 ! ICST_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S CST
: 5302 1 ! IRDRX_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S IP REGISTER
: 5303 1 !-
: 5304 1
: 5305 2 begin
: 5306 2
: 5307 2 local
: 5308 2     index : signed word,
: 5309 2     U_SAVE : word;
: 5310 2
: 5311 2 SA_REG = .IDCT_ADDR [SA_SAVE];
: 5312 2 U_SAVE = .L$LUN;
: 5313 2 C_ERR_TBL [.ICTLR, C_ERR_HRD] = .C_ERR_TBL [.ICTLR, C_ERR_HRD] + 1;
: 5314 2
: 5315 2 if .APT_MODE
: 5316 2 then
: 5317 3     begin
: 5318 3         .MAIL_BOX_TESTNUM = 1;
: 5319 3         .MAIL_BOX_SUBST = 0;
: 5320 3     end;
: 5321 2
: 5322 2 L$LUN = .ICST_ADDR [OF_LUN + OF_DATA, D_UNIT];
: 5323 2 ERRDEF (14, EGD_14, EMS_14);
: 5324 2 L$LUN = .U_SAVE;
: 5325 2 DRV_CTLERR (.ICTLR);
: 5326 2
: 5327 2 if (index = GET RETPKT (.ICTLR)) lss 0
: 5328 2 then
: 5329 3     PRINTF (DBM18)
: 5330 2 else
: 5331 3     begin
: 5332 3         RETPKT [.index, CONID] = CID_DRIVER;
: 5333 3         RETPKT [.index, MESTYP] = MT_FATAL;
: 5334 3         RETPKT [.index, CTLR] = .ICTLR;
: 5335 3         IN_IODQ (.index);
: 5336 3     end;
: 5337 2
: 5338 1 end;

```

```

! SET CURRENT UNIT TO FIRST IN CONTROLLER
! FATAL CONTROLLER ERROR
! RESTORE PRE-INTERRUPT CURRENT UNIT
! CLEAN UP DRIVER DATA FOR CONTROLLER
! TRY TO GET A RETPKT; IF FAILURE
! "FATAL_ERROR: RETPKT NOT AVAILABLE"
! IF RETPKT WAS ALLOCATED
! SET CONNECTION ID TO "DRIVER"
! FATAL ERROR
! CONTROLLER NUMBER
! LOAD RETPKT INDEX INTO IODQ
! IF RETPKT WAS ALLOCATED
! ROUTINE FATAL_ERR

```

.SBTTL FATAL_ERROR RDRX INTERRUPT SERVICE ROUTINES

N15

ZRQAM3	RD/RX EXERCISER	11-Apr-1984 11:08:35	VAX-11 Bliss-16 V4.0-579	SEQ 0402
V01.6	RDRX INTERRUPT SERVICE ROUTINES	11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2;6	Page 1/1
				(50)

000000	004137	000000G	FATAL.ERROR::		
			JSR	R1,\$SAVE2	
000004	013700	000100'	MOV	IDCT,ADDR,R0	
000010	016037	000002' 000000G	MOV	2(R0),SA,REG	
000016	013701	000000G	MOV	L\$LUN,R1	; *,U.SAVE
000022	013700	000104'	MOV	ICTLR,R0	
000026	006300		ASL	R0	
000030	105260	000000G	INCB	C.ERR,TBL(R0)	
000034	032737	000001' 001162'	BIT	#1,APT,MODE	
000042	001405		BEQ	1\$	
000044	012777	000001' 001164'	MOV	#1,@MAIL,BOX,TESTNUM	
000052	005077	001166'	CLR	@MAIL,BOX,SUBTST	
000056	013700	000076'	1\$: MOV	ICST,ADDR,R0	
000062	016002	000006	MOV	6(R0),R2	
000066	000302		SWAB	R2	
000070	042702	177760	BIC	#177760,R2	
000074	010237	000000G	MOV	R2,L\$LUN	
000100	104455		TRAP	55	
000102	000016		.WORD	16	
000104	000000G		.WORD	EGD,14	
000106	000000G		.WORD	EMS,14	
000110	010137	000000G	MOV	R1,L\$LUN	; U.SAVE,*
000114	013746	000104'	MOV	ICTLR,-(SP)	
000120	004737	000000G	JSR	PC,DRV,CTLERR	
000124	013716	000104'	MOV	ICTLR,(SP)	
000130	004737	000000G	JSR	PC,GET,RETPKT	
000134	010001		MOV	R0,R1	; *,INDEX
000136	002007		BGE	2\$	
000140	012716	000000G	MOV	@DBM18,(SP)	
000144	012746	000001	MOV	#1,(SP)	
000150	010600		MOV	SP,R0	; SP,*
000152	104417		TRAP	17	
000154	000424		BR	3\$	
000156	010116		2\$: MOV	R1,(SP)	; INDEX,*
000160	012746	000054	MOV	#54,-(SP)	
000164	004737	000000G	JSR	PC,BL\$MUL	
000170	062700	000002G	ADD	@RETPKT,2,R0	
000174	112760	000003' 000001	MOVB	#3,1(R0)	
000202	013702	000104'	MOV	ICTLR,R2	
000206	042702	177760	BIC	#177760,R2	
000212	112710	000060	MOVB	#60,(R0)	
000216	150210		BISL	R2,(R0)	
000220	010116		MOV	R1,(SP)	; INDEX,*
000222	004737	000000G	JSR	PC,IN,1000	
000226	022626		3\$: CMP	(SP)+,(SP)+	
000230	000207		RTS	PC	

; Routine Size: 77 words, Routine Base: \$CODE\$ + 24746
; Maximum stack depth per invocation: 7 words

PROGRAMS
VOL.6

RDX EXERCISER
RDX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX 11-31199 16 V4.0 5/79
DISK10SER2:(POWER5)ZQAD0.BL 216

SEQ 0405
Page 172
(51)

```

: 5539 1 GLOBAL routine POLL CRING : novalue -
: 5540 1
: 5541 1
: 5542 1
: 5543 1 THIS ROUTINE IS CALLED BY THE RDX INTERRUPT SERVICE ROUTINE (AZINT)
: 5544 1 FOR EACH DEVICE INTERRUPT EXCEPT DURING INITIALIZATION OR FATAL ERROR.
: 5545 1 ITS PURPOSE IS TO SCAN THE DEVICE'S COMMAND RING AND CHECK FOR ANY
: 5546 1 COMMAND SLOTS THAT HAVE BEEN "TAKEN" BY THE CONTROLLER. SUCH SLOTS
: 5547 1 HAVE BEEN RETURNED TO THE HOST, INDICATED BY A ZERO OWNERSHIP BIT. FOR
: 5548 1 EACH SLOT THAT HAS BEEN RETURNED TO THE HOST, THE CRING COUNT IS
: 5549 1 DECREMENTED, AND THE CR POLL ADDRESS IS ADVANCED TO THE NEXT SLOT IN
: 5550 1 THE COMMAND RING.
: 5551 1
: 5552 1 IMPLICIT INPUTS:
: 5553 1 ICTLR - INTERRUPTING CONTROLLER NUMBER
: 5554 1 IDCT_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 5555 1 ICOM_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S COMM AREA
: 5556 1
: 5557 2 begin
: 5558 2
: 5559 3 while ((IDCT_ADDR [CRING CNT] gtr 0) and
: 5560 3 not (BIT TST ((IDCT_ADDR [CR POLL] + 2), ED OWN))) do
: 5561 3 begin
: 5562 3 IDCT_ADDR [CRING CNT] = IDCT_ADDR [CRING CNT] - 1
: 5563 3 IDCT_ADDR [CR POLL] = IDCT_ADDR [CR POLL] + 4
: 5564 3
: 5565 3 if IDCT_ADDR [CR POLL] gtr IDCT_ADDR [CR END]
: 5566 3 then
: 5567 3 IDCT_ADDR [CR POLL] = IDCT_ADDR [CR BEG]
: 5568 3
: 5569 2 end
: 5570 2
: 5571 2 ICOM_ADDR [CMD INT] = 0
: 5572 1 end

```

000000	004157	000000G	15BIT	POLL CRING RDX INTERRUPT SERVICE ROUTINES		
000004	015701	000100		POLL CRING:		5539
000010	012702	000016		JSR	R1, HAVE	5540
000014	060102			MOV	IDCT_ADDR, R1	5541
000016	105711			MOV	#16, R2	5542
000020	001422			ADD	R1, R2	
000022	016100	000016	14:	BEQ	#1	5543
000026	016000	000002		MOV	16(R1), R0	5544
000032	042700	011111		MOV	2(R0), R0	
000036	070027	100000		BIC	#11111, R0	
000042	001411			CMR	R0, #100000	
000044	105711			BEQ	#1	
000046	062711	000004		DECH	(R1)	5545
000052	021761	000012		ADD	#4, (R1)	5546
000056	101757			CMR	(R2), 12(R1)	5547
				BLOS	11	5548

C16

ZRQANS
VOL.6

R0 RX EXERCISE
R0RX INTERRUPT SERVICE ROUTINES

11 Apr 1984 11:08:55
11 Apr 1984 11:08:22

VAX 11 R1100 16 V4.0 5/79
[15K\$USER2:[POWERS]ZRQADO.BL216

SEQ 0404
Page 175
(51)

000060	01611:	000010	MOV	10(R1),(R1)	:	5367
000064	000754		BR	15	:	5359
000066	015700	000014	MOV	ICOM,ADDR,R0	:	5371
000070	005060	000004	CLR	4(R0)	:	
000076	000707		RTS	PC	:	5339

; Routine size: 32 words, Routine Base: \$CODE1 + 25200
; Maximum stack depth per invocation: 4 words

[D] 11

ZRQAM3
V01.6

RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX-11 B1159 16 V4.0 5/9
DISK\$USER2:[POWER5]ZRQADO.BL2;6

SEQ 0405
Page 1/4
(52)

```

: 5373 1 GLOBAL routine POLL_RING : novalue *
: 5374 1
: 5375 1
: 5376 1
: 5377 1
: 5378 1 THIS ROUTINE IS CALLED BY THE RDRX INTERRUPT SERVICE ROUTINE (AZINT)
: 5379 1 FOR EACH DEVICE INTERRUPT EXCEPT DURING INITIALIZATION OR FATAL ERROR.
: 5380 1 ITS PURPOSE IS TO SCAN THE DEVICE'S RESPONSE RING AND CHECK FOR ANY
: 5381 1 SLOTS WHICH HAVE BEEN RETURNED TO THE HOST (OWNERSHIP BIT = 0). FOR
: 5382 1 EACH SUCH SLOT, THE ASSOCIATED MESSAGE IS PROCESSED BASED ON ITS
: 5383 1 CONNECTION ID (DISK OR DUP). AFTER PROCESSING, THE MESSAGE PACKET
: 5384 1 IS RE-INITIALIZED AND RETURNED TO THE CONTROLLER (OWNERSHIP BIT SET
: 5385 1 TO 1).
: 5386 1
: 5387 1 IMPLICIT INPUTS:
: 5388 1 ICTLR - NUMBER OF INTERRUPTING CONTROLLER
: 5389 1 IDCT_ADDR - ADDRESS OF INTERRUPTING CONTROLLER'S DCT
: 5390 1
: 5391 1
: 5392 2 begin
: 5393 2
: 5394 2
: 5395 2 while not (BIT_TEST (.IDCT_ADDR [RR_POLL] + 2), ED_0WN)) do ! WHILE 0 = 0
: 5396 3 begin
: 5397 3 IPKT_ADDR = .IDCT_ADDR [RR_POLL] + 10; ! ADDRESS OF RESPONSE PACKET
: 5398 3
: 5399 3
: 5400 4 IF NOT (.IPKT_ADDR [CONNID] EQL CID_DUP) !
: 5401 3 THEN !
: 5402 3 (.CREDIT_BAL = .CREDIT_BAL + .IPKT_ADDR [CREDITS]); !
: 5403 3 ! IT WAS NOTICE THAT DUP WAS SENDING BACK CREDITS WHICH IT SHOULD NOT. !
: 5404 3 selectone .IPKT_ADDR [CONNID] of !
: 5405 3 set !
: 5406 3
: 5407 3 [CID_DISK] : DISK_RSP ();
: 5408 3
: 5409 3 [CID_DUP] : DUP_RSP (); !
: 5410 3
: 5411 3 [otherwise] : PRINTF (DBMPO, .IPKT_ADDR [CONNID], .IRDRX_ADDR);
: 5412 3 ! "BAD CONN ID = XXXXX FROM XXXXX"
: 5413 3
: 5414 3
: 5415 3
: 5416 3 IPKT_ADDR [MSGLEN] = MSG_LEN + 2; ! RE-INIT PKT FIELDS; MESSAGE LENGTH
: 5417 3 IDCT_ADDR [RR_POLL] = .IDCT_ADDR [RR_POLL] + 2; ! ADVANCE TO NEXT WORD OF RING SLOT
: 5418 3 .IDCT_ADDR [RR_POLL] = .IPKT_ADDR [PKT_HI]; ! RETURN SLOT TO CONTROLLER
: 5419 3 .IDCT_ADDR [RR_POLL] = .IDCT_ADDR [RR_POLL] or ED_0WN or ED_FLAG; ! OWNERSHIP TO 0
: 5420 3 IDCT_ADDR [RR_POLL] = .IDCT_ADDR [RR_POLL] + 2; ! ADVANCE TO NEXT RING SLOT
: 5421 3
: 5422 3 IF .IDCT_ADDR [RR_POLL] gtr .IDCT_ADDR [RR_END] ! IF BEYOND END OF RING
: 5423 3 then
: 5424 3 IDCT_ADDR [RR_POLL] = .IDCT_ADDR [RR_BEG]; ! CYCLE TO TOP OF RING
: 5425 3

```


PROGRAMS
V01.6

RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX-11 B1155 16 V4.0 579
DISK4USER2:[POWERS]ZRDADO.BL2;6

SEQ 0406
Page 175
(52)

5426 3
5427 2
5428 2
5429 2
5430 1

end;
ICOM_ADDR (RSP INT) = 0;
end;

! WHILE LOOP
! CLR RESPONSE INTERRUPT WRD IN RING HEADER

			TITLE POLL RING RDRX INTERRUPT SERVICE ROUTINES		
000000	004157	000000G	POLL_RING::		
			JSR	R1, \$SAVE 3	5373
000004	015701	000100	MOV	IOCT, ADDR, R1	5395
000010	062701	000014	ADD	#14, R1	
000014	011100		1\$: MOV	(R1), R0	
000016	016000	000002	MOV	2(R0), R0	
000022	042700	077777	BIC	#77777, R0	
000026	020027	100000	CMP	R0, # 100000	
000032	001504		BEQ	6\$	
000034	017137	000000	MOV	#0(R1), IPKT, ADDR	5397
000042	162737	000012	SUB	#12, IPKT, ADDR	
000050	013700	000000G	MOV	IPKT, ADDR, R0	5400
000054	005002		CLR	R2	
000056	156002	000011	BISB	11(R0), R2	
000062	020227	000002	CMP	R2, #2	
000066	001406		BEQ	2\$	
000070	116003	000010	MOVH	10(R0), R3	5402
000074	042703	177760	BIC	#177760, R3	
000100	060337	000000G	ADD	R3, CREDIT, BAL	
000104	005702		2\$: TST	R2	5407
000106	001003		BNE	5\$	
000110	004757	000000V	JSR	PC, DISK, RSP	
000114	000421		BR	5\$	5404
000116	020227	000002	3\$: CMP	R2, #2	5409
000122	001003		BNE	4\$	
000124	004757	024446	JSR	PC, DUP, RSP	
000130	000413		BR	5\$	5404
000132	015746	000000G	4\$: MOV	IRDRX, ADDR, (SP)	5411
000136	010246		MOV	R2, (SP)	
000140	012746	000000G	MOV	#0(R2), (SP)	
000144	012746	000003	MOV	#3, (SP)	
000150	010600		MOV	SP, R0	! SP, *
000152	104417		TRAP	17	
000154	062706	000010	ADD	#10, SP	
000160	015700	000000G	5\$: MOV	IPKT, ADDR, R0	5416
000164	012760	000074	MOV	#74, 6(R0)	
000172	015702	000100	MOV	IOCT, ADDR, R2	5417
000176	010201		MOV	R2, R1	
000200	062701	000014	ADD	#14, R1	
000204	062711	000002	ADD	#2, (R1)	
000210	016071	000002	MOV	2(R0), #0(R1)	5418
000216	052771	140000	BIC	#40000, #0(R1)	5419
000224	062711	000002	ADD	#2, (R1)	5420
000230	021162	000006	CMP	(R1), 6(R2)	5422
000234	101667		BLUS	1\$	

F 16

TRQAMS
V01.6 RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX 11-B1166 16 V4.0 5/9
DISK\$USER2:[POWER5]ZROADO,BL 2,6

SEQ 0407
Page 1/6
(52)

000236	016211	000004	MOV	4(R2),(R1)	:	5425
000242	000664		BR	1:	:	5395
000244	013700	000074	MOV	ICOM,ADDR,RO	:	5429
000250	005060	000006	CLR	6(RO)		
000254	000207		RTS	PC	:	5373

; Routine Size: 87 words, Routine Base: \$CODE\$ + 25300
; Maximum stack depth per invocation: 10 words

; 5431 1

016

TRUAM3
VOL.6

RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1155 16 V4.0 5/79
DISK\$USER2:[POWERS]ZRUADO.B1.2;6

SEQ 0408
Page 177
(53)

```

: 5432 1
: 5433 1
: 5434 1 GLOBAL routine DISK RSP : novalue
: 5435 1
: 5436 1
: 5437 1
: 5438 1
: 5439 1
: 5440 1
: 5441 1
: 5442 1
: 5443 1
: 5444 1
: 5445 1
: 5446 1
: 5447 1
: 5448 1
: 5449 1
: 5450 1
: 5451 1
: 5452 1
: 5453 1
: 5454 1
: 5455 1
: 5456 1
: 5457 1
: 5458 1
: 5459 1
: 5460 1
: 5461 1

THIS ROUTINE IS CALLED BY POLL RING FOR EACH RESPONSE MESSAGE
WHICH HAS A CONNECTION ID INDICATING A DISK MSCP ORIGINATOR
(I.E., ALL EXCEPT DUP RESPONSES). ITS PURPOSE IS TO PASS
CONTROL TO THE APPROPRIATE ROUTINE BASED ON THE MESSAGE TYPE
FIELD (SEQUENTIAL, DATAGRAM, OR CREDIT NOTIFICATION).

IMPLICIT INPUTS:
    IPKT_ADDR - ADDRESS OF MSCP PACKET CONTAINING RESPONSE
                MESSAGE

selectoneu .IPKT_ADDR [MSGTYP] of

    set
    [MT SEQ] : SEQUEN ();
    [MT DG] : DATAGM ();
    [otherwise] : PRINTF (DBM21, .IPKT_ADDR [MSGTYP]); ! "MESSAGE TYPE XX RECEIVED"
    tes;

```

			.SBTIL DISK,RSP RDRX INTERRUPT SERVICE ROUTINES	
000000	010146		DISK,RSP::	
000002	013700	000000G	MOV R1, (SP)	5434
000006	116001	000010	MOV IPKT_ADDR,R0	5451
000012	006201		MOVH 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\$	5436
000030	004737	000000V	JSR PC,SEQUEN	
000034	000417		BR 3\$	5451
000036	020127	000001	1\$: CMP R1,#1	5453
000042	001003		BNE 1\$	
000044	004737	000000V	JSR PC,DATAGM	
000050	000417		BR 3\$	5451
000052	010146		2\$: MOV R1, (SP)	5460
000054	012746	000000G	MOV DBM21, (SP)	
000060	012746	000002	MOV #2, (SP)	
000064	010600		MOV SP,R0	1 SP,+

H16

ZRQAM5 RD RX EXERCISER
V01.6 RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:55
11-Apr-1984 11:08:22

VAX 11 B1166 16 V4.0 579
DISK\$USER2:[POWERS]ZRQADO.BL2:6

SEQ 0409
Page 1/8
(55)

000066	104417		TRAP	17
000070	062706	000006	ADD	#6,SP
000074	012601	33:	MOV	(SP)+,R1
000076	000207		RTS	PC

5434

; Routine Size: 32 words, Routine Base: \$CODE\$ + 25556
; Maximum stack depth per invocation: 6 words

; 5460 1
; 5462 1
; 5464 1

ZRRQANS
VOL.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11 Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX 11 B1144 14 V4.0 5/9
DISK\$USER2:[POWERS]ZRRQADO.BL2:6SEQ 0410
Page 1/9
(54)

```

: 5465 1 GLOBAL routine SEQUEN : novalue *
: 5466 1
: 5467 1
: 5468 1
: 5469 1
: 5470 1
: 5471 1
: 5472 1
: 5473 1
: 5474 1
: 5475 1
: 5476 1
: 5477 1
: 5478 1
: 5479 1
: 5480 2 begin
: 5481 2
: 5482 2 local
: 5483 2 P_INDEX : signed word initial (-1),
: 5484 2 R_INDEX : signed word,
: 5485 2 SRC_ADDR,
: 5486 2 DST_ADDR,
: 5487 2 R_ADDR : ref block [RP_LEN, word] field (RP_FIELDS);
: 5488 2
: 5489 2 incr COUNT from 0 to PKT_CNT 1 do
: 5490 2
: 5491 2 if (.MSCP_PKT [.COUNT, CRN_LO] eql .IPKT_ADDR [CRN_LO]) and
: 5492 2 (.MSCP_PKT [.COUNT, CRN_HI] eql .IPKT_ADDR [CRN_HI]) and
: 5493 2 (.MSCP_PKT [.COUNT, PKT_LO] neq .IPKT_ADDR [PKT_LO]) and
: 5494 2 ((.MSCP_PKT [.COUNT, OPCODE] and OP_END) neq OP_END) and
: 5495 2 (.MSCP_PKT [.COUNT, MSGTYP] eql MT_SEQ) and
: 5496 2 ((.IPKT_ADDR [OPCODE] and OP_END) eql OP_END) and
: 5497 2 (.PKT_USE [.COUNT] eql .ICTLR)
: 5498 2 then
: 5499 2 begin
: 5500 2 P_INDEX = .COUNT;
: 5501 2 exitloop;
: 5502 2 end;
: 5503 2
: 5504 2 if .P_INDEX lss 0
: 5505 2 then
: 5506 2 begin
: 5507 2 PRINTF (DBM108, .IPKT_ADDR [CRN_HI], .IPKT_ADDR [CRN_LO]);
: 5508 2 return;
: 5509 2 end;
: 5510 2
: 5511 2 if .MSCP_PKT [.P_INDEX, OPCODE] neq (.IPKT_ADDR [OPCODE] and (not OP_END))
: 5512 2 then
: 5513 2 PRINTF (DBM111, .MSCP_PKT [.P_INDEX, OPCODE], .IPKT_ADDR [OPCODE], .IPKT_ADDR [CRN_HI], .IPKT_ADDR [CRN_LO]);
: 5514 2
: 5515 2 if ((.IPKT_ADDR [OPCODE] eql (OP_RD or OP_END)) or
: 5516 2 (.IPKT_ADDR [OPCODE] eql (OP_WRT or OP_END))) and
: 5517 2 ((.IPKT_ADDR [STATUS_CODE] eql ST_SUC) and

```

! ASSUME NO ASSOCIATED COMMAND PKT

! IF THIS IS THE ASSOC CMD

! SET PKT NUMBER

! IF COMMAND NOT FOUND

! UNKNOWN COMMAND REF. NUMBER

! IF OPCODE MISMATCH

ZRGAMS
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX 11 B11ss 16 V4.0 5/9
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0411
Page 180
(54)

```

: 5518 2      (,IPKT_ADDR [STATUS SUBCODE] eq1 0)) and
: 5519 3      ((,MSCP_PKT [,P_INDEX, BC_LO] neq ,IPKT_ADDR [BC_LO]) or
: 5520 3      (,MSCP_PKT [,P_INDEX, BC_HI] neq ,IPKT_ADDR [BC_HI]))
: 5521 2      then
: P 5522 2      PRINTF (DBM112,
: P 5523 2      ,MSCP_PKT [,P_INDEX, BC_HI], ,MSCP_PKT [,P_INDEX, BC_LO], ,IPKT_ADDR [BC_HI], ,IPKT_ADDR [BC_LO],
: 5524 2      ,IPKT_ADDR [CRN_HI], ,IPKT_ADDR [CRN_LO]);
: 5525 2
: 5526 2      IF ,MSCP_PKT [,P_INDEX, RSP RECEIVED]
: 5527 2      then
: 5528 3          begin
: 5529 3              PRINTF (DBM120, ,MSCP_PKT [,P_INDEX, CRN_HI], ,MSCP_PKT [,P_INDEX, CRN_LO]);
: 5530 3              PUT_PKT (,P_INDEX);
: 5531 3              return;
: 5532 3          end
: 5533 2      else
: 5534 2          ,MSCP_PKT [,P_INDEX, RSP RECEIVED] = TRUE;
: 5535 2
: 5536 2      IF (R_INDEX = GET_RETPKT (,ICTLR)) lss 0
: 5537 2      then
: 5538 3          begin
: 5539 3              PRINTF (DBM220);
: 5540 3              PUT_PKT (,P_INDEX);
: 5541 3              return;
: 5542 3          end
: 5543 2      else
: 5544 3          begin
: 5545 3              SRC_ADDR = ,IPKT_ADDR + 6;
: 5546 3              R_ADDR = DST_ADDR + RETPKT + (,R_INDEX * RP_LEN + 2);
: 5547 3
: 5548 3              INCR COUNT FROM 1 TO RP_LEN DO
: 5549 4                  begin
: 5550 4                      ,DST_ADDR = ,SRC_ADDR;
: 5551 4                      ,DST_ADDR = ,DST_ADDR + 2;
: 5552 4                      ,SRC_ADDR = ,SRC_ADDR + 2;
: 5553 4
: 5554 4                      IF ,IPKT_ADDR [OPCODE] eq1 (OP_ONL or OP_END)
: 5555 4                      then
: 5556 4                          IF ,COUNT eq1 10
: 5557 4                          then
: 5558 4                              ,SRC_ADDR = ,SRC_ADDR + 4;
: 5559 3                      end;
: 5560 3
: 5561 3              R_ADDR [CTLR] = ,ICTLR;
: 5562 3
: 5563 3              IF ,P_INDEX geq 0
: 5564 3              then
: 5565 3
: 5566 3                  IF (,IPKT_ADDR [OPCODE] eq1 (OP_RD or OP_END)) or
: 5567 3                  (,IPKT_ADDR [OPCODE] eq1 (OP_WRT or OP_END)) or
: 5568 4                  (,IPKT_ADDR [OPCODE] eq1 (OP_ACC or OP_END))
: 5569 4                  then
: 5570 4                      begin

```

! MARK RESPONSE RECEIVED

! IF RETPKT IS NOT AVAILABLE

! "SEQUEN: RETPKT NOT AVAILABLE"

! SET UP COPY (SKIP OVER PKT DESC)
! START OF ALLOCATED RETPKT! COPY 1 WORD
! ADVANCE DESTINATION ADDR
! ADVANCE SOURCE ADDR

! IF THIS IS THE ONLINE END MESSAGE

! SKIP OVER RESERVED WORDS

! IN ONLINE END MESSAGE
! COPY LOOP

! LOAD CONTROLLER NUMBER INTO PKT

! IF ASSOC. CMD PKT WAS FOUND

! IF END MESSAGE IS
! READ, WRITE, OR
! ACCESS

K16

ZRGAMS
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0412
Page 181
(54)

```

: 5571 4      R_ADDR [CMDMOD] = .MSCP_PKT [.P_INDEX, MODIFY];
: 5572 4      R_ADDR [CBCNT_LO] = .MSCP_PKT [.P_INDEX, BC_LO];
: 5573 4      R_ADDR [CBCNT_HI] = .MSCP_PKT [.P_INDEX, BC_HI];
: 5574 4      R_ADDR [LBN_LO] = .MSCP_PKT [.P_INDEX, LBN_L];
: 5575 4      R_ADDR [LBN_HI] = .MSCP_PKT [.P_INDEX, LBN_H];
: 5576 4      R_ADDR [BUF_0] = .MSCP_PKT [.P_INDEX, BUF_0];
: 5577 4      R_ADDR [BUF_1] = .MSCP_PKT [.P_INDEX, BUF_1];
: 5578 3      end;
: 5579 3
: 5580 3      IN_IODQ (.R_INDEX);
: 5581 2      end;
: 5582 2
: 5583 2      IF (.IPKT_ADDR [STATUS_CODE] neq ST_SUC) or
: 5584 3      (.IPKT_ADDR [STATUS_SUBCODE] neq 0)
: 5585 2      then
: 5586 2          LAST_PKT [.ICTLR, LAST_HRD_ERR] = HRD_OCCURED;
: 5587 2      else
: 5588 2          LAST_PKT [.ICTLR, LAST_HRD_ERR] = HRD_NOT_OCCURED;
: 5589 2
: 5590 2      LAST_PKT [.ICTLR, LAST_CRN_LO] = .IPKT_ADDR [CRN_LO];
: 5591 2      LAST_PKT [.ICTLR, LAST_CRN_HI] = .IPKT_ADDR [CRN_HI];
: 5592 2      SCAN_ERRLOG ();
: 5593 2
: 5594 2      IF .P_INDEX geq 0
: 5595 2      then
: 5596 2          PUT_PKT (.P_INDEX);
: 5597 2
: 5598 1      end;

```

! COPY
! RELEVANT
! FIELDS
! FROM
! COMMAND
! PACKET
! TO RETPKT
! IF ENCODED WAS READ/WRITE/ACCESS

! PUT RETPKT INDEX INTO IODQ
! IF RETPKT WAS ALLOCATED

! SAVE ERROR CONDITION
!
! SAVE COMMAND REFERENCE NUMBER
!
! PRINT ANY ASSOCIATED ERROR LOGS
!
! IF ASSOC CMD PKT WAS FOUND
!
! RETURN COMMAND PACKET TO POOL
!
! ROUTINE DISK_RSP

			.SBTTL	SEQUEN RDRX INTERRUPT SERVICE ROUTINES	
000000	004137	000000G	SEQUEN: JSR	R1, \$SAVES	5465
000004	005746		TST	-(SP)	
000006	012746	177777	MOV	0-1, -(SP)	; *, P_INDEX
000012	013701	000000G	MOV	IPKT_ADDR, R1	; COUNT
000016	005002		CLR	R2	; COUNT, *
000020	010246		MOV	R2, -(SP)	
000022	012746	000106	MOV	0106, (SP)	
000026	004737	000000G	JSR	PC, BL \$MUL	
000032	022623		CMP	(SP)+, (SP)+	
000034	026061	000012G 000012	CMP	MSCP_PKT+12(R0), 12(R1)	
000042	001030		BNE	2\$	
000044	026061	000014G 000014	CMP	MSCP_PKT+14(R0), 14(R1)	; 5492
000052	001024		BNE	2\$	
000054	026011	000000G	CMP	MSCP_PKT(R0), (R1)	; 5493
000060	001421		BEQ	2\$	
000062	105760	000022G	TSTB	MSCP_PKT+22(R0)	; 5494
000066	100416		BMI	2\$	
000070	132760	000360 000010G	BITB	0360, MSCP_PKT+10(R0)	; 5495
000076	001012		BNE	2\$	
000100	105761	000022	TSTB	22(R1)	; 5496
000104	100007		BPL	2\$	
000106	116200	000000G	MOVW	PKT_USE(R2), R0	; *(COUNT), *

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0413
Page 182
(54)

000112	020037	000104'	CMP	R0,ICTLR		
000116	001002		BNE	2\$		
000120	010216		MOV	R2,(SP)	; COUNT,P.INDEX	5500
000122	000405		BR	3\$		5499
000124	005202		INC	R2	; COUNT	5489
000126	020227	000013	CMP	R2,#13	; COUNT,*	
000132	003732		BLE	1\$		
000134	005716		TST	(SP)	; P.INDEX	5504
000136	002013		BGE	4\$		
000140	016146	000012	MOV	12(R1),-(SP)		5507
000144	016146	000014	MOV	14(R1),-(SP)		
000150	012746	000000G	MOV	#DBM108, -(SP)		
000154	012746	000003	MOV	#3, -(SP)		
000160	010600		MOV	SP,R0	; SP,*	
000162	104417		TRAP	17		
000164	000545		BR	9\$		
000166	011646		MOV	(SP), -(SP)	; P.INDEX,*	5508
000170	012746	000106	MOV	#106, -(SP)		5511
000174	004737	000000G	JSR	PC,BL\$MUL		
000200	010001		MOV	R0,R1		
000202	022626		CMP	(SP)*, (SP)*		
000204	013700	000000G	MOV	IPKT,ADDR,R0		
000210	116003	000022	MOVB	22(R0),R3		
000214	042703	177600	BIC	#177600,R3		
000220	005002		CLR	R2		
000222	156102	000022G	BISB	MSCP,PKT+22(R1),R2		
000226	020203		CMP	R2,R3		
000230	001422		BEQ	5\$		
000232	016046	000012	MOV	12(R0), -(SP)		5513
000236	016046	000014	MOV	14(R0), -(SP)		
000242	005046		CLR	-(SP)		
000244	116016	000022	MOVB	22(R0),(SP)		
000250	005046		CLR	-(SP)		
000252	116116	000022G	MOVB	MSCP,PKT+22(R1),(SP)		
000256	012746	000000G	MOV	#DBM111, -(SP)		
000262	012746	000005	MOV	#5, -(SP)		
000266	010600		MOV	SP,R0	; SP,*	
000270	104417		TRAP	17		
000272	062706	000014	ADD	#14,SP		
000276	013700	000000G	MOV	IPKT,ADDR,R0		5515
000302	126027	000022	CMPB	22(R0),#241		
000310	001404		BEQ	6\$		
000312	126027	000022	CMPB	22(R0),#242		5516
000320	001045		BNE	8\$		
000322	012702	000024	MOV	#24,R2		5517
000326	060002		ADD	R0,R2		
000330	132712	000037	BITS	#37,(R2)		
000334	001037		BNE	8\$		
000336	032712	177740	BIT	#177740,(R2)		5518
000342	001034		BNE	8\$		
000344	026160	000026G	CMP	MSCP,PKT+26(R1),26(R0)		5519
000352	001004		BNE	7\$		
000354	026160	000030G	CMP	MSCP,PKT+30(R1),30(R0)		5520

M16

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0414
Page 183
(54)

000362	001424			BEQ	8\$		
000364	016046	0C0012	7\$:	MOV	12(R0), -(SP)	:	5524
000370	016046	000014		MOV	14(R0), -(SP)		
000374	016046	000026		MOV	26(R0), -(SP)		
000400	016046	000030		MOV	30(R0), -(SP)		
000404	016146	000026G		MOV	MSCP, PKT+26(R1), -(SP)		
000410	016146	000030G		MOV	MSCP, PKT+30(R1), -(SP)		
000414	012746	000000G		MOV	QDBM112, -(SP)		
000420	012746	000007		MOV	Q7, -(SP)		
000424	010600			MOV	SP, R0	: SP, *	
000426	104417			TRAP	17		
000430	062706	000020		ADD	Q20, SP		
000434	132761	000400 000005G	8\$:	BITB	Q400, MSCP, PKT+5(R1)	:	5526
000442	001422			BEQ	10\$		
000444	016146	000012G		MOV	MSCP, PKT+12(R1), -(SP)	:	5529
000450	016146	000014G		MOV	MSCP, PKT+14(R1), -(SP)		
000454	012746	000000G		MOV	QDBM120, -(SP)		
000460	012746	000003		MOV	Q3, -(SP)		
000464	010600			MOV	SP, R0	: SP, *	
000466	104417			TRAP	17		
000470	016616	000010		MOV	10(SP), (SP)	: P, INDEX, *	5530
000474	004737	000000G		JSR	PC, PUT, PKT		
000500	062706	000010	9\$:	ADD	Q10, SP	:	5531
000504	000137	027016'		JMP	21\$	:	5528
000510	112761	000001 000005G	10\$:	MOV8	Q1, MSCP, PKT+5(R1)	:	5534
000516	013746	000104'		MOV	ICTLR, -(SP)	:	5536
000522	004737	000000G		JSR	PC, GET, RETPKT		
000526	010066	000004		MOV	R0, 4(SP)	: *, R, INDEX	
000532	005726			TST	(SP),		
000534	005766	000002		TST	2(SP)	: R, INDEX	
000540	002007			BGE	11\$		
000542	012746	000000G		MOV	QDBM22, -(SP)	:	5539
000546	012746	000001		MOV	Q1, -(SP)		
000552	010600			MOV	SP, R0	: SP, *	
000554	104417			TRAP	17		
000556	000563			BR	19\$	:	5540
000560	013704	000000G	11\$:	MOV	IPKT, ADDR, R4	: *, SRC, ADDR	5545
000564	062704	000006		ADD	Q6, R4	: *, SRC, ADDR	
000570	016646	000002		MOV	2(SP), -(SP)	: R, INDEX, *	5546
000574	012746	000054		MOV	Q54, -(SP)		
000600	004737	000000G		JSR	PC, RL \$MUL		
000604	062700	000000G		ADD	QRETPKT, R0		
000610	010005			MOV	R0, R5	: *, DST, ADDR	
000612	013702	000000G		MOV	IPKT, ADDR, R2	:	5554
000616	012703	000001		MOV	Q1, R3	: *, COUNT	5548
000622	012425		12\$:	MOV	(R4), (R5),	: SRC, ADDR, DST, ADDR	5550
000624	126227	000022 000211		CMPB	22(R2), Q211	:	5554
000632	001005			BNE	13\$		
000634	020327	000012		CMP	R3, Q12	: COUNT, *	5556
000640	001002			BNE	13\$		
000642	062704	000004		ADD	Q4, R4	: *, SRC, ADDR	5558
000646	005203		13\$:	INC	R3	: COUNT	5548
000650	020327	000026		CMP	R3, Q26	: COUNT, *	

ZRQAM2	RD/R....B1	ZRQAM3	RD/R....B5	ZRQAM3	RD/R....B9	ZRQAM3	RD/R....B13
ZRQAM2	RD/R....C1	ZRQAM3	RD/R....C5	ZRQAM3	RD/R....C9	ZRQAM3	RD/R....C13
ZRQAM2	RD/R....D1	ZRQAM3	RD/R....D5	ZRQAM3	RD/R....D9	ZRQAM3	RD/R....D13
ZRQAM2	RD/R....E1	ZRQAM3	RD/R....E5	ZRQAM3	RD/R....E9	ZRQAM3	RD/R....E13
ZRQAM2	RD/R....F1	ZRQAM3	RD/R....F5	ZRQAM3	RD/R....F9	ZRQAM3	RD/R....F13
ZRQAM2	RD/R....G1	ZRQAM3	RD/R....G5	ZRQAM3	RD/R....G9	ZRQAM3	RD/R....G13
ZRQAM2	RD/R....H1	ZRQAM3	RD/R....H5	ZRQAM3	RD/R....H9	ZRQAM3	RD/R....H13
ZRQAM2	RD/R....I1	ZRQAM3	RD/R....I5	ZRQAM3	RD/R....I9	ZRQAM3	RD/R....I13
ZRQAM2	RD/R....J1	ZRQAM3	RD/R....J5	ZRQAM3	RD/R....J9	ZRQAM3	RD/R....J13
ZRQAM2	RD/R....K1	ZRQAM3	RD/R....K5	ZRQAM3	RD/R....K9	ZRQAM3	RD/R....K13
ZRQAM2	RD/R....L1	ZRQAM3	RD/R....L5	ZRQAM3	RD/R....L9	ZRQAM3	RD/R....L13
ZRQAM2	RD/R....M1	ZRQAM3	RD/R....M5	ZRQAM3	RD/R....M9	ZRQAM3	RD/R....M13
ZRQAM2	RD/R....N1	ZRQAM3	RD/R....N5	ZRQAM3	RD/R....N9	ZRQAM3	RD/R....N13
ZRQAM2	RD/R....B2	ZRQAM3	RD/R....B6	ZRQAM3	RD/R....B10	ZRQAM3	RD/R....B14
ZRQAM2	RD/R....C2	ZRQAM3	RD/R....C6	ZRQAM3	RD/R....C10	ZRQAM3	RD/R....C14
ZRQAM2	RD/R....D2	ZRQAM3	RD/R....D6	ZRQAM3	RD/R....D10	ZRQAM3	RD/R....D14
ZRQAM2	RD/R....E2	ZRQAM3	RD/R....E6	ZRQAM3	RD/R....E10	ZRQAM3	RD/R....E14
ZRQAM2	RD/R....F2	ZRQAM3	RD/R....F6	ZRQAM3	RD/R....F10	ZRQAM3	RD/R....F14
ZRQAM2	RD/R....G2	ZRQAM3	RD/R....G6	ZRQAM3	RD/R....G10	ZRQAM3	RD/R....G14
ZRQAM2	RD/R....H2	ZRQAM3	RD/R....H6	ZRQAM3	RD/R....H10	ZRQAM3	RD/R....H14
ZRQAM2	RD/R....I2	ZRQAM3	RD/R....I6	ZRQAM3	RD/R....I10	ZRQAM3	RD/R....I14
ZRQAM2	RD/R....J2	ZRQAM3	RD/R....J6	ZRQAM3	RD/R....J10	ZRQAM3	RD/R....J14
ZRQAM2	RD/R....K2	ZRQAM3	RD/R....K6	ZRQAM3	RD/R....K10	ZRQAM3	RD/R....K14
ZRQAM2	RD/R....L2	ZRQAM3	RD/R....L6	ZRQAM3	RD/R....L10	ZRQAM3	RD/R....L14
ZRQAM3	RD/R....M2	ZRQAM3	RD/R....M6	ZRQAM3	RD/R....M10	ZRQAM3	RD/R....M14
ZRQAM3	RD/R....N2	ZRQAM3	RD/R....N6	ZRQAM3	RD/R....N10	ZRQAM3	RD/R....N14
ZRQAM3	RD/R....B3	ZRQAM3	RD/R....B7	ZRQAM3	RD/R....B11	ZRQAM3	RD/R....B15
ZRQAM3	RD/R....C3	ZRQAM3	RD/R....C7	ZRQAM3	RD/R....C11	ZRQAM3	RD/R....C15
ZRQAM3	RD/R....D3	ZRQAM3	RD/R....D7	ZRQAM3	RD/R....D11	ZRQAM3	RD/R....D15
ZRQAM3	RD/R....E3	ZRQAM3	RD/R....E7	ZRQAM3	RD/R....E11	ZRQAM3	RD/R....E15
ZRQAM3	RD/R....F3	ZRQAM3	RD/R....F7	ZRQAM3	RD/R....F11	ZRQAM3	RD/R....F15
ZRQAM3	RD/R....G3	ZRQAM3	RD/R....G7	ZRQAM3	RD/R....G11	ZRQAM3	RD/R....G15
ZRQAM3	RD/R....H3	ZRQAM3	RD/R....H7	ZRQAM3	RD/R....H11	ZRQAM3	RD/R....H15
ZRQAM3	RD/R....I3	ZRQAM3	RD/R....I7	ZRQAM3	RD/R....I11	ZRQAM3	RD/R....I15
ZRQAM3	RD/R....J3	ZRQAM3	RD/R....J7	ZRQAM3	RD/R....J11	ZRQAM3	RD/R....J15
ZRQAM3	RD/R....K3	ZRQAM3	RD/R....K7	ZRQAM3	RD/R....K11	ZRQAM3	RD/R....K15
ZRQAM3	RD/R....L3	ZRQAM3	RD/R....L7	ZRQAM3	RD/R....L11	ZRQAM3	RD/R....L15
ZRQAM3	RD/R....M3	ZRQAM3	RD/R....M7	ZRQAM3	RD/R....M11	ZRQAM3	RD/R....M15
ZRQAM3	RD/R....N3	ZRQAM3	RD/R....N7	ZRQAM3	RD/R....N11	ZRQAM3	RD/R....N15
ZRQAM3	RD/R....B4	ZRQAM3	RD/R....B8	ZRQAM3	RD/R....B12	ZRQAM3	RD/R....B16
ZRQAM3	RD/R....C4	ZRQAM3	RD/R....C8	ZRQAM3	RD/R....C12	ZRQAM3	RD/R....C16
ZRQAM3	RD/R....D4	ZRQAM3	RD/R....D8	ZRQAM3	RD/R....D12	ZRQAM3	RD/R....D16
ZRQAM3	RD/R....E4	ZRQAM3	RD/R....E8	ZRQAM3	RD/R....E12	ZRQAM3	RD/R....E16
ZRQAM3	RD/R....F4	ZRQAM3	RD/R....F8	ZRQAM3	RD/R....F12	ZRQAM3	RD/R....F16
ZRQAM3	RD/R....G4	ZRQAM3	RD/R....G8	ZRQAM3	RD/R....G12	ZRQAM3	RD/R....G16
ZRQAM3	RD/R....H4	ZRQAM3	RD/R....H8	ZRQAM3	RD/R....H12	ZRQAM3	RD/R....H16
ZRQAM3	RD/R....I4	ZRQAM3	RD/R....I8	ZRQAM3	RD/R....I12	ZRQAM3	RD/R....I16
ZRQAM3	RD/R....J4	ZRQAM3	RD/R....J8	ZRQAM3	RD/R....J12	ZRQAM3	RD/R....J16
ZRQAM3	RD/R....K4	ZRQAM3	RD/R....K8	ZRQAM3	RD/R....K12	ZRQAM3	RD/R....K16
ZRQAM3	RD/R....L4	ZRQAM3	RD/R....L8	ZRQAM3	RD/R....L12	ZRQAM3	RD/R....L16
ZRQAM3	RD/R....M4	ZRQAM3	RD/R....M8	ZRQAM3	RD/R....M12	ZRQAM3	RD/R....M16
ZRQAM3	RD/R....N4	ZRQAM3	RD/R....N8	ZRQAM3	RD/R....N12	ZRQAM3	RD/R....N16

B1

ZRQAMS V01.6	RD/RX EXERCISER RDRX INTERRUPT SERVICE ROUTINES	11-Apr-1984 11:08:35 11-Apr-1984 11:08:22	VAX-11 B1100-1, V4.0-579 DISK#USER2:[POWERS]ZRQADO.BL2;6	SEQ 0415 Page 184 (54)
000654 003762		BLE 121		
000656 013703 000104'		MOV ICTLR,R3		5561
000662 042703 177760		BIC #177760,R3		
000666 142760 000017 000002		BICB #17,2(R0)	: *,*(R.ADDR)	
000674 150360 000702		BISB R3,2(R0)	: *,*(R.ADDR)	
000700 005766 000004		TST 4(SP)	: P.INDEX	5563
000704 002441		BLT 151		
000706 005003		CLR R3		5566
000710 156203 000022		BISB 22(R2),R3		
000714 020327 000241		CMP R3,0241		
000720 001406		BEQ 141		
000722 020327 000242		CMP R3,0242		5567
000726 001403		BEQ 141		
000730 020327 000220		CMP R3,0220		5568
000734 001025		BNE 151		
000736 016160 000024G 000012	141:	MOV MSCP,PKT+24(R1),12(R0)	: *,*(R.ADDR)	5571
000744 016160 000026G 000044		MOV MSCP,PKT+26(R1),44(R0)	: *,*(R.ADDR)	5572
000752 016160 000030G 000046		MOV MSCP,PKT+30(R1),46(R0)	: *,*(R.ADDR)	5573
000760 016160 000046G 000050		MOV MSCP,PKT+46(R1),50(R0)	: *,*(R.ADDR)	5574
000766 016160 000050G 000052		MOV MSCP,PKT+50(R1),52(R0)	: *,*(R.ADDR)	5575
000774 016160 000032G 000024		MOV MSCP,PKT+32(R1),24(R0)	: *,*(R.ADDR)	5576
001002 016160 000034G 000026		MOV MSCP,PKT+34(R1),26(R0)	: *,*(R.ADDR)	5577
001010 016616 000006	151:	MOV 6(SP),(SP)	: R.INDEX,+	5580
001014 004737 000000G		JSR PC,IN.IODQ		
001020 005726		TST (SP),		5544
001022 013716 000104'		MOV ICTLR,(SP)		5586
001026 012746 000006		MOV #6,-(SP)		
001032 004737 000000G		JSR PC,BL#MUL		
001036 013701 000000G		MOV IPKT,ADDR,R1		5583
001042 012703 000024		MOV #24,R3		
001046 060103		ADD R1,R3		
001050 132713 000037		BITB #37,(R3)		
001054 001003		BNE 161		
001056 032713 177740		BIT #177740,(R3)		5584
001062 001404		BEQ 171		
001064 012760 000001 000116'	161:	MOV #1,LAST.PKT(R0)		5586
001072 000402		BR 181		5583
001074 005060 000116'	171:	CLR LAST.PKT(R0)		5588
001100 016160 000012 000120'	181:	MOV 12(R1),LAST.PKT+2(R0)		5590
001106 016160 000014 000122'		MOV 14(R1),LAST.PKT+4(R0)		5591
001114 004737 000000V		JSR PC,SCAN.ERRLOG		5592
001120 005766 000004		TST 4(SP)	: P.INDEX	5594
001124 002404		BLT 201		
001126 016616 000004	191:	MOV 4(SP),(SP)	: P.INDEX,+	5596
001132 004737 000000G		JSR PC,PUT.PKT		
001136 022626	201:	CMP (SP),(SP),		5480
001140 022626	211:	CMP (SP),(SP),		5465
001142 000207		RTS PC		

: Routine Size: 306 words, Routine Base: \$CODE\$ + 25656
: Maximum stack depth per invocation: 18 words

C1

ZRGAMS
VOL.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRGADO.BL2;6SEQ 0416
Page 185
(55)

```

: 5599 1 GLOBAL routine SCAN_ERRLOG : novalue =
: 5600 1
: 5601 1
: 5602 1 !! THIS ROUTINE SCANS THE ERROR-LOG SAVE AREA AND PRINTS ANY ERROR-LOGS RECEIVED FOR THE ASSOCIATED RESPONSE
: 5603 1 !!
: 5604 1
: 5605 2 begin
: 5606 2
: 5607 2 local
: 5608 2 TEMP_UNIT,
: 5609 2 SFT_ERR_PRINTED : byte initial (byte (FALSE));
: 5610 2
: 5611 2 incr index from 0 to EP_CNT do
: 5612 3 begin
: 5613 3
: 5614 3 if (.ELOG_PKT [.index, EL_CNTR] eq1 .ICTLR) and
: 5615 3 (.ELOG_PKT [.index, EL_CRN_LO] eq1 .IPKT_ADDR [CRN_LO]) and
: 5616 3 (.ELOG_PKT [.index, EL_CRN_HI] eq1 .IPKT_ADDR [CRN_HI]) and
: 5617 4 (.ELOG_PKT [.index, EL_CONTENTS] eq1 FULL)
: 5618 3 then
: 5619 4 begin
: 5620 4 ! ERROR-LOG PENDING THIS RESPONSE
: 5621 4 if .LAST_PKT [.ICTLR, LAST_HRD_ERR] eq1 HRD_NOT_OCCURED
: 5622 4 ! IF SOFT ERROR OCCURED
: 5623 4 then
: 5624 4 if .ELOG_PKT [.index, EL_FORMAT] lequ 4
: 5625 4 then
: 5626 5 begin
: 5627 5 SOFT_ERROR (.index);
: 5628 5 ! UPDATE SOFT ERROR COUNT
: 5629 5 TEMP_UNIT = .L$UN;
: 5630 5 ! SAVE UNIT NUMBER AS KNOWN TO DRS
: 5631 5 incr OFFSET from (0 + OF_UN) to ((UNITS_PER_CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5632 5 if (.ICST_ADDR [.OFFSET + OF_DATA, D_DISK_NUM] eq1 .ELOG_PKT [.index, EL_DK_NUM]) and
: 5633 6 (.ICST_ADDR [.OFFSET + OF_DATA, D_PRESENT] eq1 PRESENT)
: 5634 5 then
: 5635 6 begin
: 5636 6 L$UN = .ICST_ADDR [.OFFSET + OF_DATA, D_UNIT];
: 5637 6 ! CORRECT UNIT NO. FOR ERROR MESSAGE
: 5638 5 exitloop;
: 5639 5 end;
: 5640 5 case .ELOG_PKT [.index, EL_FORMAT] from 0 to 4 of
: 5641 5 set
: 5642 5
: 5643 5 [0]: if .APT_MODE
: 5644 5 ! CONTROLLER ERROR
: 5645 5 then
: 5646 5 ERR_SOFT_RTNE_APT (50, .index)
: 5647 5 else
: 5648 5 ERR_SOFT_RTNE (50);
: 5649 5
: 5650 5 [1]: if .APT_MODE
: 5651 5 ! HOST MEMORY ACCESS ERROR
: 5651 5 then
: 5651 5 ERR_SOFT_RTNE_APT (51, .index)

```

01

ZRQAMS
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK#USER2:[POWERS]ZRQADC.BL2;6

SEQ 0417
Page 186
(55)

```

: 5652 5
: 5653 5
: 5654 5
: 5655 5
: 5656 5
: 5657 5
: 5658 5
: 5659 5
: 5660 5
: 5661 5
: 5662 5
: 5663 5
: 5664 5
: 5665 5
: 5666 5
: 5667 5
: 5668 5
: 5669 5
: 5670 5
: 5671 5
: 5672 5
: 5673 5
: 5674 5
: 5675 5
: 5676 5
: 5677 4
: 5678 4
: 5679 4
: 5680 5
: 5681 4
: 5682 4
: 5683 4
: 5684 4
: 5685 4
: 5686 3
: 5687 3
: 5688 3
: 5689 4
: 5690 5
: 5691 3
: 5692 4
: 5693 3
: 5694 4
: 5695 4
: 5696 4
: 5697 3
: 5698 3
: 5699 2
: 5700 2
: 5701 1

else
    ERR_SOFT_RTNE (51);

[2]: if .APT_MODE
then
    ERR_SOFT_RTNE_APT (52, .index)
else
    ERR_SOFT_RTNE (52);

[3]: if .APT_MODE
then
    ERR_SOFT_RTNE_APT (53, .index)
else
    ERR_SOFT_RTNE (53);

[4]: if .APT_MODE
then
    ERR_SOFT_RTNE_APT (54, .index)
else
    ERR_SOFT_RTNE (54);

tes;

L$LUN = .TEMP_UNIT;
SFT_ERR_PRINTED = TRUE;
end
else
    PRINTF (DBM109, .ELOG_PKT (.index, EL_FORMAT));

if not (.SFT_ERR_PRINTED)
then
    PRINTB (CRLF);

EMS_EL (.index);
end
else
    if (.ELOG_PKT (.index, EL_NTR) eq1 .IC1LR) and
        ((.ELOG_PKT (.index, EL_CRN_HI) lssu .IPKT_ADDR [CRN_HI]) or
        ((.ELOG_PKT (.index, EL_CRN_HI) eq1 .IPKT_ADDR [CRN_HI]) and
        (.ELOG_PKT (.index, EL_CRN_LO) lssu .IPKT_ADDR [CRN_LO])) and
        (.ELOG_PKT (.index, EL_CONTENTS) eq1 FULL)
    then
        begin
            PRINTB (CRLF);
            EMS_EL (.index);
        end;

end;

end;

end;


```

! DISK TRANSFER ERROR

! SOI ERROR

! SMALL DISK ERROR

! RESTORE UNIT NUMBER
! SOFT ERROR PRINTOUT OCCURED

! UNKNOWN ERROR-LOG FORMAT

! EXTRA CARRIAGE-RETURN/LINE-FEED

! PRINT ERROR-LOG CONTENTS

! CARRIAGE-RETURN/LINE-FEED
! PRINT ERROR-LOG CONTENTS

! ERROR-LOG SAVE AREA SCAN

.SBTTL SCAN.ERRLOG RDRX INTERRUPT SERVICE ROUTINES

F 1

ZRQAM3 V01.6	RD/RX EXERCISER RDRX INTERRUPT SERVICE ROUTINES	11-Apr-1984 11:08:35 11-Apr-1984 11:08:22	VAX-11 B1199-16 V4.0-579 DISK\$USER2:[POWERS]ZRQADO.BL2;6	SEQ 0418 Page 187 (55)
000000	004137 000000G	SCAN.ERRLOG: :		
000004	005746	JSR R1,\$SAVE5	:	5599
000006	105005	TST -(SP)		
000010	005002	CLRB R5	: SFT.ERR.PRINTED	5605
000012	010246	CLR R2	: INDEX	5611
000014	012746 000102	1\$: MOV R2,-(SP)	: INDEX,*	5614
000020	004737 000000G	MOV #102,-(SP)		
000024	010001	JSR PC,BL\$MUL		
000026	022626	MOV R0,R1		
000030	012703 000000G	CMP (SP)*,(SP)*		
000034	060103	MOV #ELOG,PKT,R3		
000036	005004	ADD R1,R3		
000040	005000	CLR R4		
000042	151300	CLR R0		
000044	020037 000104'	BISB (R3),R0		
000050	001016	CMP R0,ICTLR		
000052	005204	BNE 2\$		
000054	013700 000000G	INC R4		
000060	026160 000006G 000012	MOV IPKT,ADDR,R0	:	5615
000066	001007	CMP ELOG,PKT+6(R1),12(R0)		
000070	026160 000010G 000014	BNE 2\$		
000076	001003	CMP ELOG,PKT+10(R1),14(R0)	:	5616
000100	126327 000001 000001	BNE 2\$		
000106	001402	CMPB 1(R3),#1	:	5617
000110	000137 027562'	2\$: BEQ 3\$		
000114	013746 000104'	JMP 25\$		
000120	012746 000006	3\$: MOV ICTLR,-(SP)	:	5621
000124	004737 000000G	MOV #6,-(SP)		
000130	022626	JSR PC,BL\$MUL		
000132	005760 000116'	CMP (SP)*,(SP)*		
000136	001161	TST LAST.PKT(R0)		
000140	126127 000016G 000004	BNE 23\$		
000146	101142	CMPB ELOG,PKT+16(R1),#4	:	5624
000150	010246	BHI 21\$		
000152	004737 000000V	MOV R2,-(SP)	: INDEX,*	5627
000156	013766 000000G 000002	JSR PC,SOFT.ERROR		
000164	012703 000006	MOV L\$LUN,2(SP)	: *,TEMP.UNIT	5628
000170	010300	MOV #6,R3	: *,OFFSET	5630
000172	063700 000076'	4\$: MOV R3,R0	: OFFSET,*	5632
000176	016146 000012G	ADD ICST,ADDR,R0		
000202	111004	MOV ELOG,PKT+12(R1),-(SP)		
000204	042704 177760	MOVB (R0),R4		
000210	020426	BIC #177760,R4		
000212	001012	CMP R4,(SP)*		
000214	032710 040000	BNE 5\$		
000220	001407	BIT #40000,(R0)	:	5633
000222	011004	BEQ 5\$		
000224	000304	MOV (R0),R4	:	5636
000226	042704 177760	SWAB R4		
000232	010437 000000G	BIC #177760,R4		
000236	000405	MOV R4,L\$LUN		
000240	062703 000024	BR 6\$	:	5635
		5\$: ADD #24,R3	: *,OFFSET	5630

F1

ZRQAM3	RD/RX EXERCISER			11-Apr-1984 11:08:35	VAX-11 Bliss-16 V4.0-579	SEQ 0419
V01.6	RDRX INTERRUPT SERVICE ROUTINES			11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2;6	Page 188
						(55)
000244	020327	000102		CMP	R3,#102	
000250	003747			BLE	4\$	; OFFSET,*
000252	003000		6\$:	CLR	R0	
000254	153700	001162		BISB	APT,MODE,R0	5643
000260	116101	000016G		MOVB	ELOG,PKT+16(R1),R1	
000264	042701	177400		BIC	#177400,R1	5640
000270	006301			ASL	R1	
000272	066107	000000		ADD	P,AAA(R1),PC	; Case dispatch
000276	032700	000001	8\$:	BIT	#1,R0	
000302	001403			BEQ	9\$	5643
000304	012716	000062		MOV	#62,(SP)	
000310	000442			BR	17\$	5645
000312	012716	000062	9\$:	MOV	#62,(SP)	
000316	000446			BR	19\$	5647
000320	032700	000001	10\$:	BIT	#1,R0	
000324	001403			BEQ	11\$	5649
000326	012716	000063		MOV	#63,(SP)	
000332	000431			BR	17\$	5651
000334	012716	000063	11\$:	MOV	#63,(SP)	
000340	000435			BR	19\$	5653
000342	032700	000001	12\$:	BIT	#1,R0	
000346	001403			BEQ	13\$	5655
000350	012716	000064		MOV	#64,(SP)	
000354	000420			BR	17\$	5657
000356	012716	000064	13\$:	MOV	#64,(SP)	
000362	000424			BR	19\$	5659
000364	032700	000001	14\$:	BIT	#1,R0	
000370	001403			BEQ	15\$	5661
000372	012716	000065		MOV	#65,(SP)	
000376	000407			BR	17\$	5663
000400	012716	000065	15\$:	MOV	#65,(SP)	
000404	000413			BR	19\$	5665
000406	006000		16\$:	ROR	R0	
000410	103007			BCC	18\$	5667
000412	012716	000066		MOV	#66,(SP)	
000416	010216		17\$:	MOV	R2,-(SP)	; INDEX,*
000420	004737	000000V		JSR	PC,ERR,SOFT,RTNE,APT	
000424	005726			TST	(SP),	
000426	000404			BR	20\$	
000430	012716	000066	18\$:	MOV	#66,(SP)	5667
000434	004737	000000V	19\$:	JSR	PC,ERR,SOFT,RTNE	5671
000440	016637	000002	20\$:	MOV	2(SP),L\$LUN	
000446	112705	000001		MOVB	#1,R5	; TEMP,UNIT,*
000452	000412			BR	22\$	; *,SFT,ERR,PRINTED
000454	005046		21\$:	CLR	(SP)	
000456	116116	000016G		MOVB	ELOG,PKT+16(R1),(SP)	
000462	012746	000000G		MOV	#DBM109,-(SP)	
000466	012746	000002		MOV	#2,(SP)	
000472	010600			MOV	SP,R0	; SP,*
000474	104417			TRAP	17	
000476	022626			CMP	(SP),(SP),	
000500	005726		22\$:	TST	(SP),	
000502	032705	000001	23\$:	BIT	#1,R5	; *,SFT,ERR,PRINTED

61

ZRQAMS V01.6	RD/RX EXERCISER RDRX INTERRUPT SERVICE ROUTINES	11-Apr-1984 11:08:35 11-Apr-1984 11:08:22	VAX-11 B1199-16 V4.0-579 DISK\$USER2:[POWERS]ZRQADO.BL2;6	SEQ 0420 Page 189 (55)

000506	001007		BNF	24:			
000510	012746	000000G	MOV	0CRLF, (SP)	:		5682
000514	012746	000001	MOV	01, -(SP)	:		
000520	010600		MOV	SP, R0	:	SP, *	
000522	104414		TRAP	14	:		
000524	022626		CMP	(SP)+, (SP)+	:		
000526	010246		MOV	R2, -(SP)	:	INDEX, *	5684
000530	004737	000000G	JSR	PC, EMS.EL	:		
000534	005726		TST	(SP)+	:		5619
000536	000433		BR	27:	:		5614
000540	006004		ROR	R4	:		5688
000542	103031		BCC	27:	:		
000544	013700	000000G	MOV	IPKT.ADDR, R0	:		5689
000550	026160	000010G 000014	CMP	ELOG.PKT+10(R1), 14(R0)	:		
000556	103405		BLO	26:	:		
000560	001022		BNE	27:	:		5690
000562	026160	000006G 000012	CMP	ELOG.PKT+6(R1), 12(R0)	:		5691
000570	103016		BHIS	27:	:		
000572	126327	000001 000001	26:	CMPB	1(R3), 01	:	5692
000600	001012		BNE	27:	:		5695
000602	012746	000000G	MOV	0CRLF, -(SP)	:		
000606	012746	000001	MOV	01, -(SP)	:		
000612	010600		MOV	SP, R0	:	SP, *	
000614	104414		TRAP	14	:		
000616	010216		MOV	R2, (SP)	:	INDEX, *	5696
000620	004737	000000G	JSR	PC, EMS.EL	:		
000624	022626		CMP	(SP)+, (SP)+	:		5694
000626	005202		INC	R2	:	INDEX	5611
000630	020227	000014	CMP	R2, 014	:	INDEX, *	
000634	003002		RGT	28:	:		
000636	000137	027034	JMF	1:	:		
000642	005726		28:	TST	(SP)+	:	5599
000644	000207		RTS	PC	:		

; Routine Size: 211 words, Routine Base: \$CODE\$ + 27022
; Maximum stack depth per invocation: 12 words

000000			.PSECT	\$PLIT\$, R0, D		

000000	000700	P.AAA:			; CASE Table for SCAN.ERRLOG+0272	5640
000002	000022	7:	.WORD	0	; [8:]	
000004	000044		.WORD	22	; [10:]	
000006	000066		.WORD	44	; [12:]	
000010	000110		.WORD	66	; [14:]	
			.WORD	110	; [16:]	

H1

ZRQAM3
V01.6

RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK:USER2:[POWERS]ZRQADO.BL2;6

SEQ 0421
Page 190
(56)

```

: 5702 1 GLOBAL routine DATAGM : novalue *
: 5703 1
: 5704 1
: 5705 1
: 5706 1
: 5707 1
: 5708 1
: 5709 1
: 5710 1
: 5711 1
: 5712 1
: 5713 1
: 5714 2
: 5715 2
: 5716 2
: 5717 2
: 5718 2
: 5719 2
: 5720 2
: 5721 2
: 5722 2
: 5723 2
: 5724 2
: 5725 2
: 5726 2
: 5727 2
: 5728 2
: 5729 2
: 5730 2
: 5731 2
: 5732 2
: 5733 3
: 5734 3
: 5735 3
: 5736 2
: 5737 2
: 5738 2
: 5739 2
: 5740 2
: 5741 2
: 5742 2
: 5743 2
: 5744 2
: 5745 2
: 5746 2
: 5747 2
: 5748 2
: 5749 2
: 5750 2
: 5751 2
: 5752 2
: 5753 2
: 5754 2

!
! THIS ROUTINE HANDLES ALL DATAGRAM (ERROR LOG) MESSAGES RECEIVED FROM
! THE RDRX
!
! IMPLICIT INPUTS:
!   IPKT_ADDR - ADDRESS OF MSCP PACKET CONTAINING ERROR LOG
!               MESSAGE
!   ICST_ADDR - ADDRESS OF THE INTERRUPTING CONTROLLER'S CST
!
begin
  local
    index : signed word initial (-1),
    SAVE_ADDR : ref block [EP_LEN, word] field (EP_FIELDS),
    SRC_ADDR,
    DST_ADDR,
    TEMP_UNIT,
    SET_ERR_PRINTED : byte initial (byte (FALSE)),
    PACKET_LEN : word;

! FIND AN EMPTY SLOT IN THE ERROR-LOG PACKET SAVE AREA

  incr COUNT from 0 to EP_CNT - 1 do
    if .ELOG_PKT [.COUNT, EL_CONTENTS] eq1 EMPTY
    then
      begin
        index = .COUNT;
        exitloop;
      end;

  if .index lss 0
  then
    index = EP_CNT;

! IF EMPTY SLOT FOUND
! SAVE INDEX INTO THE SAVE AREA
! IF NO SLOT FOUND, USE LAST SPARE SLOT

! SAVE THE PACKET CONTENTS

  SAVE_ADDR = ELOG_PKT + (.index * EP_LEN + 2);
  SAVE_ADDR [EL_CONTENTS] = FULL;
  SAVE_ADDR [EL_CNTR] = .ICSTR;
  SRC_ADDR = .IPKT_ADDR + 6;
  DST_ADDR = .SAVE_ADDR + 2;
  PACKET_LEN = ((.IPKT_ADDR [MSCP_LEN] + 1) / 2) * 2;

! ADDRESS OF THE SAVE AREA
! MARK IT FULL
! OWNERSHIP
! SETUP COPY ADDRESSES
! LENGTH OF ERROR-LOG INCLUDING ENVELOPE

  if .PACKET_LEN gtru EP_LEN - 1
  then

```

ZRQAM5
V01.6

RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0422
Page 191
(56)

```

: 5755 2      PACKET_LEN = 2P_LEN 1;
: 5756 2
: 5757 2      incr COUNT from 1 to .PACKET_LEN do
: 5758 3      begin
: 5759 3          .DST_ADDR = .SRC_ADDR;
: 5760 3          SRC_ADDR = .SRC_ADDR + 2;
: 5761 3          DST_ADDR = .DST_ADDR + 2;
: 5762 3      end;
: 5763 2
: 5764 2
: 5765 2      ! CHECK IF THE CORRESPONDING RESPONSE HAS ALREADY BEEN RECEIVED
: 5766 2
: 5767 2
: 5768 2      if (.SAVE_ADDR [EL_CRN_LO] eq1 .LAST_PKT [.ICTLR, LAST_CRN_LO]) and
: 5769 3      (.SAVE_ADDR [EL_CRN_HI] eq1 .LAST_PKT [.ICTLR, LAST_CRN_HI])
: 5770 2      then
: 5771 3      begin
: 5772 3          ! LOG REFERS TO THE LAST RESPONSE RECEIVED
: 5773 3          if .LAST_PKT [.ICTLR, LAST_HRD_ERR] eq1 HRD_NOT_OCCURED
: 5774 3          then
: 5775 3              ! IF SOFT ERROR HAD OCCURED
: 5776 3              if .SAVE_ADDR [EL_FORMAT] lequ 4
: 5777 3              then
: 5778 4              begin
: 5779 4                  SOFT_ERROR (.index);
: 5780 4                  TEMP_UNIT = .L$UN;
: 5781 4                  ! UPDATE SOFT ERROR COUNT
: 5782 4                  ! SAVE UNIT NUMBER AS KNOWN TO DRS
: 5783 4                  incr OFFSET from (0 + OF_UN) to ((UNITS PER CNTR - 1) * UNIT_SIZE + OF_UN) by UNIT_SIZE do
: 5784 4                      if (.ICST_ADDR [.OFFSET + OF_DATA, D_DISK_NUM] eq1 .SAVE_ADDR [EL_DK_NUM]) and
: 5785 5                      (.ICST_ADDR [.OFFSET + OF_DATA, D_PRESENT] eq1 PRESENT)
: 5786 4                      then
: 5787 5                      begin
: 5788 5                          L$UN = .ICST_ADDR [.OFFSET + OF_DATA, D_UNIT]; ! CORRECT UNIT NUMBER FOR ERROR MESSAGE
: 5789 5                      end;
: 5790 4                      exitloop;
: 5791 4                      end;
: 5792 4
: 5793 4      case .SAVE_ADDR [EL_FORMAT] from 0 to 4 of
: 5794 4      set
: 5795 4
: 5796 4      [0] :      if .APT_MODE
: 5797 4      then
: 5798 4          ERR_SOFT_RTNE_APT (50, .index)
: 5799 4      else
: 5800 4          ERR_SOFT_RTNE (50);
: 5801 4
: 5802 4      [1] :      if .APT_MODE
: 5803 4      then
: 5804 4          ERR_SOFT_RTNE_APT (51, .index)
: 5805 4      else
: 5806 4          ERR_SOFT_RTNE (51);
: 5807 4

```


11

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0 579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0423
Page 192
(56)

```

: 5808 4          [2] : if .APT_MODE          ! DISK TRANSFER ERROR
: 5809 4          then
: 5810 4              ERR_SOFT_RTNE_APT (52, .index)
: 5811 4          else
: 5812 4              ERR_SOFT_RTNE (52);
: 5813 4
: 5814 4          [3] : if .APT_MODE          ! SOI ERROR
: 5815 4          then
: 5816 4              ERR_SOFT_RTNE_APT (53, .index)
: 5817 4          else
: 5818 4              ERR_SOFT_RTNE (53);
: 5819 4
: 5820 4          [4] : if .APT_MODE          ! SMALL DISK ERROR
: 5821 4          then
: 5822 4              ERR_SOFT_RTNE_APT (54, .index)
: 5823 4          else
: 5824 4              ERR_SOFT_RTNE (54);
: 5825 4          tes;
: 5826 4
: 5827 4
: 5828 4          L$LUN = .TEMP_UNIT;          ! RESTORE UNIT NUMBER
: 5829 4          SFT_ERR_PRINTED = TRUE;      ! SOFT ERROR PRINTOUT OCCURED
: 5830 4          end
: 5831 4
: 5832 3          else
: 5833 3              PRINTF (DBM109, .SAVE_ADDR [EL_FORMAT]); ! ERROR LOG FORMAT UNKNOWN
: 5834 3
: 5835 4          if not (.SFT_ERR_PRINTED)
: 5836 4
: 5837 3          then
: 5838 3              PRINTB (CRLF);              ! EXTRA CARRIEGE-RETURN/LINE-FEED
: 5839 3
: 5840 3          EMS_EL (.index);              ! PRINT PACKET CONTENTS
: 5841 3          end                          ! CORRESPONDING RESPONSE RECEIVED
: 5842 3
: 5843 2          else
: 5844 2
: 5845 2          if (.SAVE_ADDR [EL_CRN_HI] lssu .LAST_PKT [.ICTLR, LAST_CRN_HI]) or
: 5846 3              ((.SAVE_ADDR [EL_CRN_HI] eq1 .LAST_PKT [.ICTLR, LAST_CRN_HI]) and
: 5847 3              (.SAVE_ADDR [EL_CRN_LO] lssu .LAST_PKT [.ICTLR, LAST_CRN_LO]))
: 5848 3
: 5849 2          then
: 5850 3              begin
: 5851 3                  PRINTB (CRLF);              ! LOG REFERS TO SOME PREVIOUS RESPONSE
: 5852 3                  EMS_EL (.index);          ! CARRIAGE-RETURN/LINE-FEED
: 5853 3                  end;                      ! PRINT PACKET CONTENTS
: 5854 2
: 5855 1          end;

```

027670

.SBITI DATAQM RDRX INTERRUPT SERVICE ROUTINES
.PSECT \$CODE\$, RO

K1

ZRQAM3 V01.6	RD/RX EXERCISER RDRX INTERRUPT SERVICE ROUTINES	11-Apr-1984 11:08:35 11-Apr-1984 11:08:22	VAX-11 Blues-16 V4.0-579 DISK\$USER2:[POWERS]ZRQAD0.BL2;6	SEQ 0424 Page 193 (56)
000000	004137 000000G	DATAGM: JSR	R1,\$SAVE5	
000004	012704 177777	MOV	0-1,R4	
000010	105046	CLRR	-(SP)	
000012	005001	CLR	R1	
000014	010146	1\$: MOV	R1, -(SP)	
000016	012746 000102	MOV	0102, -(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	010104	MOV	R1,R4	: COUNT, INDEX
000040	000405	BR	3\$	
000042	005201	2\$: INC	R1	: COUNT
000044	020127 000013	CMP	R1, 013	: COUNT, *
000050	003761	BLE	1\$	
000052	005704	TST	R4	: INDEX
000054	002002	3\$: BGE	4\$	
000056	012704 000014	MOV	014,R4	: *, INDEX
000062	010446	4\$: MOV	R4, -(SP)	: INDEX, *
000064	012746 000102	MOV	0102, -(SP)	
000070	004737 000000G	JSR	PC,BL\$MUL	
000074	052700 000000G	ADD	0ELOG,PKT,R0	
000100	010001	MOV	R0,R1	: *,SAVE, ADDR
000102	111761 000001	MOVB	(PC), 1(R1)	: *,*(SAVE, ADDR)
000106	115711 000104	MOVB	ICTLR, (R1)	: *,SAVE, ADDR
000112	013700 000000G	MOV	IPKT, ADDR, R0	
000116	012705 000006	MOV	06,R5	: *,SRC, ADDR
000122	060005	ADD	R0,R5	: *,SRC, ADDR
000124	012703 000002	MOV	02,R5	: *,DST, ADDR
000130	060103	ADD	R1,R3	: SAVE, ADDR, DST, ADDR
000132	016016 000006	MOV	6(R0), (SP)	
000136	005216	INC	(SP)	
000140	012746 000002	MOV	02, -(SP)	
000144	004737 000000G	JSR	PC,BL\$DIV	
000150	062700 000002	ADD	02,R0	
000154	020027 000040	CMP	R0, 040	: PACKET, LEN, *
000160	101402	BLOS	5\$	
000162	012700 000040	MOV	040,R0	: *,PACKET, LEN
000166	005002	5\$: CLR	R2	: COUNT
000170	000401	BR	7\$	
000172	012523	6\$: MOV	(R5)+, (R3)+	: SRC, ADDR, DST, ADDR
000174	005202	7\$: INC	R2	: COUNT
000176	020200	CMP	R2,R0	: COUNT, PACKET, LEN
000200	003774	BLE	6\$	
000202	013716 000104	MOV	ICTLR, (SP)	
000206	012746 000006	MOV	06, (SP)	
000212	004737 000000G	JSR	PC,BL\$MUL	
000216	005726	TST	(SF)+	
000220	026160 000006 000120	CMP	6(R1), LAST, PKT+2(R0)	: *(SAVE, ADDR), *
000226	001402	BEQ	8\$	
000230	000137 030532	JMP	30\$	
000234	026160 000010 000122	8\$: CMP	10(R1), LAST, PKT+4(R0)	: *(SAVE, ADDR), *
000242	001402	BEQ	9\$	

ZLJAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0425
Page 194
(56)

000244	000137	030540'		JMP	31\$			
000250	005760	000116'	9\$:	TST	LAST.PKT(RO)	:		5773
000254	001153			BNE	28\$	:		
000256	005003			CLR	R3	:		5776
000260	156103	000016		BISB	16(R1),R3	:	*(SAVE,ADDR),*	
000264	020327	000004		CMP	R3,#4	:		
000270	101135			BHI	27\$	:		
000272	010416			MOV	R4,(SP)	:	INDEX,*	5779
000274	004737	000000V		JSR	PC,SOFT.ERROR	:		
000300	013705	000000G		MOV	L\$LUN,R5	:	*,TEMP.UNIT	5780
000304	012702	000006		MOV	#6,R2	:	*,OFFSET	5782
000310	010200		10\$:	MOV	R2,R0	:	OFFSET,*	5784
000312	063700	000076'		ADD	ICST.ADDR,RO	:		
000316	016146	000012		MOV	12(R1),-(SP)	:	*(SAVE,ADDR),*	
000322	111046			MOV8	(R0),-(SP)	:		
000324	042716	177760		BIC	#177760,(SP)	:		
000330	022626			CMP	(SP)+,(SP)+	:		
000332	001012			BNE	11\$	:		
000334	032710	040000		BIT	#40000,(R0)	:		5785
000340	001407			BEQ	11\$	:		
000342	011046			MOV	(R0),-(SP)	:		5788
000344	000316			SWAB	(SP)	:		
000346	042716	177760		BIC	#177760,(SP)	:		
000352	012637	000000G		MOV	(SP)+,L\$LUN	:		
000356	000405			BR	12\$	:		5787
000360	062702	000024	11\$:	ADD	#24,R2	:	*,OFFSET	5782
000364	020227	000102		CMP	R2,#102	:	OFFSET,*	
000370	003747			BLE	10\$	:		
000372	003000		12\$:	CLR	R0	:		5796
000374	153700	001162'		BISB	APT.MODE,R0	:		
000400	006303			ASL	R3	:		5792
000402	066307	000012'		ADD	P.AAB(R3),PC	:	Case dispatch	
000406	032700	000001	14\$:	BIT	#1,R0	:		5796
000412	001403			BEQ	15\$	:		
000414	012716	000062		MOV	#62,(SP)	:		5798
000420	000442			BR	23\$	:		
000422	012716	000032	15\$:	MOV	#62,(SP)	:		5800
000426	000446			BR	25\$	:		
000430	032700	000001	16\$:	BIT	#1,R0	:		5802
000434	001403			BEQ	17\$	:		
000436	012716	000063		MOV	#63,(SP)	:		5804
000442	000431			BR	23\$	:		
000444	012716	000063	17\$:	MOV	#63,(SP)	:		5806
000450	000435			BR	25\$	:		
000452	032700	000001	18\$:	BIT	#1,R0	:		5808
000456	001403			BEQ	19\$	:		
000460	012716	000064		MOV	#64,(SP)	:		5810
000464	000420			BR	23\$	:		
000466	012716	000064	19\$:	MOV	#64,(SP)	:		5812
000472	000424			BR	25\$	:		
000474	032700	000001	20\$:	BIT	#1,R0	:		5814
000500	001403			BEQ	21\$	:		
000502	012716	000065		MOV	#65,(SP)	:		5816

M1

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0426
Page 195
(56)

000506	000407			BH	23:				
000510	012716	000165		MOV		065,(SP)	:		5818
000514	000413			BR		25:			
000516	006000			RJR	22:	R0	:		5820
000520	103007			BCC		24:			
000522	012716	000066		MOV		066,(SP)	:		5822
000526	010416			MOV	23:	R4,-(SP)	:	INDEX, *	
000530	004737	000000V		JSR		PC,ERR.SOF1.RTNE.APT			
000534	005726			TST		(SP)+			
000536	000404			BR		26:			5820
000540	012716	000066		MOV	24:	066,(SP)	:		5824
000544	004737	000000V		JSR	25:	PC,ERR.SOFT.RTNE			
000550	010537	000000G		MOV	26:	R5,L\$LUN	:	TEMP.UNIT, *	5828
000554	112766	000001 000006		MOVB		01,6(SP)	:	*,SFT.ERR.PRINTED	5829
000562	000410			BR		28:			5776
000564	010316			MOV	27:	R3,(SP)	:		5833
000566	012746	000000G		MOV		0DBM109,-(SP)			
000572	012746	000002		MOV		02,-(SP)			
000576	010600			MOV		SP,R0	:	SP, *	
000600	104417			TRAP		17			
000602	022626			CMP		(SP)+,(SP)+			
000604	032766	000001 000006		BIT	28:	01,6(SP)	:	*,SFT.ERR.PRINTED	5835
000612	001007			BNE		29:			
000614	012716	000000G		MOV		0CRLF,(SP)	:		5838
000620	012746	000001		MOV		01,-(SP)			
000624	010600			MOV		SP,R0	:	SP, *	
000626	104414			TRAP		14			
000630	005726			TST		(SP)+			
000632	010416			MOV	29:	R4,(SP)	:	INDEX, *	5840
000634	004737	000000G		JSR		PC,EMS.EL			
000640	000426			BR		33:			5768
000642	026160	000010 000122'		CMP	30:	10(R1),LAST.PKT+4(R0)	:	*(SAVE.ADDR), *	5845
000650	103410			BLO	31:	32:			
000652	026160	000010 000122'		CMP		10(R1),LAST.PKT+4(R0)	:	*(SAVE.ADDR), *	5846
000660	001016			BNE		33:			
000662	026160	000006 000120'		CMP		6(R1),LAST.PKT+2(R0)	:	*(SAVE.ADDR), *	5847
000670	103012			BHIS		33:			
000672	012716	000000G		MOV	32:	0CRLF,(SP)	:		5851
000676	012746	000001		MOV		01,-(SP)			
000702	010600			MOV		SP,R0	:	SP, *	
000704	104414			TRAP		14			
000706	010416			MOV		R4,(SP)	:	INDEX, *	5852
000710	004737	000000G		JSR		PC,EMS.EL			
000714	005726			TST		(SP)+			5850
000716	062700	000010		ADD	33:	010,SP	:		5702
000722	000207			RTS		PC			

; Routine Size: 234 words, Routine Base: \$CODE\$ + 27670
 ; Maximum stack depth per invocation: 14 words

000012

.PSECT \$PLIT\$, R0, D

N1

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss 16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0427
Page 196
(56)

000012	000000	P.AAB:		
000014	000022	13\$:	.WORD	0
000016	000044		.WORD	22
000020	000066		.WORD	44
000022	000110		.WORD	66
			.WORD	110

; CASE Table for DATAGM+0402
; [14\$]
; [16\$]
; [18\$]
; [20\$]
; [22\$]

5792

; 5856 1
; 5857 1
; 5858 1

B2

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11s-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0428
Page 197
(57)

```

: 5859 1 GLOBAL routine SOFT_ERROR (index) : novalue *
: 5860 1
: 5861 1 !
: 5862 1 ! THIS ROUTINE UPDATES THE SOFT ERROR COUNT IN THE TALLY TABLE FOR EACH
: 5863 1 ! ERROR LOG MESSAGE RECEIVED
: 5864 1 !
: 5865 1 ! IMPLICIT INPUTS:
: 5866 1 ! ICST_ADDR - ADDRESS OF THE INTERRUPTING CONTROLLER'S CST
: 5867 1 !
: 5868 1
: 5869 2 begin
: 5870 2
: 5871 2 local
: 5872 2 FOUND: byte initial (byte (FALSE)),
: 5873 2 SOFT_OCCURED : byte initial (byte (FALSE)),
: 5874 2 UNIT: word,
: 5875 2 ERROR_CODE : byte,
: 5876 2 ERROR_SUB : word,
: 5877 2 RETRIES : word,
: 5878 2 TALLY_ADDR : ref block [TALLY_LEN, word] field (T_FIELDS),
: 5879 2 ELOG_ADDR : ref block [EP_LEN, word] field (EP_FIELDS);
: 5880 2
: 5881 2 ELOG_ADDR = ELOG_PKT + (.index * EP_LEN * 2);
: 5882 2 ERROR_CODE = .ELOG_ADDR [EL_CODE];
: 5883 2 ERROR_SUB = .ELOG_ADDR [EL_SUBCODE];
: 5884 2
: 5885 2 if (BIT_TST (SWP_FLAGS, SWF_TRY)) and
: 5886 3 (.ELOG_ADDR [EL_FORMAT] eq 2)
: 5887 2 then
: 5888 2 RETRIES = .ELOG_ADDR [EL_RETRY]
: 5889 2 else
: 5890 2 RETRIES = 1;
: 5891 2
: 5892 2 if .RETRIES eq 0
: 5893 2 then
: 5894 2 RETRIES = 1;
: 5895 2
: 5896 2 incr OFFSET from (0 * OF_UN) to ((UNITS PER CNTR - 1) * UNIT_SIZE * OF_UN) by UNIT_SIZE do
: 5897 2
: 5898 2 if (.ICST_ADDR [.OFFSET * OF_DATA, D_PRESENT] eq PRESENT) and
: 5899 3 (.ICST_ADDR [.OFFSET * OF_DATA, D_DISK_NUM] eq .ELOG_ADDR [EL_DK_NUM])
: 5900 2 then
: 5901 3 begin
: 5902 3 FOUND = TRUE;
: 5903 3 UNIT = .ICST_ADDR [.OFFSET * OF_DATA, C_UNIT];
: 5904 3 exitloop;
: 5905 2 end;
: 5906 2
: 5907 2 ! if (.ELOG_ADDR [EL_SUCCESS]) or
: 5908 2 ! (.ELOG_ADDR [EL_CONTINUE])
: 5909 2 ! then
: 5910 2 SOFT_OCCURED = TRUE;
: 5911 2

```

! ADDR OF ERROR PKT
! ERROR CODE
! ERROR SUBCODE

! COUNT EACH RETRY

! IGNORE RETRIES

! IN CASE OF A BUG

! DISK TO UNIT NO.

! DISK'S UNIT NO.

! SOFT ERROR FLAG

C2

ZRQAM5
VOL.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0429
Page 198
(57)

```

: 5912 2      if .FOUND
: 5913 2      then
: 5914 3          begin
: 5915 3              TALLY_ADDR = TALLY + (.UNIT * TALLY_LEN * 2);
: 5916 3
: 5917 3              if .SOFT_OCCURED
: 5918 3              then
: 5919 3                  select new .ERROR_CODE of
: 5920 3                      set
: 5921 3
: 5922 3                      [ST_MFE]:      TALLY_ADDR [ERR_SFT_SEK] = .TALLY_ADDR [ERR_SFT_SEK] + .RETRIES; ! SOFT-MEDIA FORMAT
: 5923 3
: 5924 3                      [ST_DAT]:      if .ERROR_SUB eal 2
: 5925 3                      then
: 5926 3                          TALLY_ADDR [ERR_SFT_SEK] = .TALLY_ADDR [ERR_SFT_SEK] + .RETRIES
: 5927 3                      else
: 5928 3                          TALLY_ADDR [ERR_SFT_DAT] = .TALLY_ADDR [ERR_SFT_DAT] + .RETRIES;
: 5929 3
: 5930 3                      [ST_HST]:      TALLY_ADDR [ERR_SFT_HST] = .TALLY_ADDR [ERR_SFT_HST] + .RETRIES; ! SOFT-HOST ACCESS
: 5931 3                      [ST_CNT]:      C_ERR_TBL [.,ICTLR, C_ERR_SFT] = .C_ERR_TBL [.,ICTLR, C_ERR_SFT] + .RETRIES;
: 5932 3
: 5933 3
: 5934 3                      ! SOFT-CONTROLLER
: 5935 3                      [ST_DRV]:      if .ERROR_SUB eal 3
: 5936 3                      then
: 5937 3                          TALLY_ADDR [ERR_SFT_SEK] = .TALLY_ADDR [ERR_SFT_SEK] + .RETRIES
: 5938 3                      else
: 5939 3                          TALLY_ADDR [ERR_SFT_DRV] = .TALLY_ADDR [ERR_SFT_DRV] + .RETRIES;
: 5940 3
: 5941 3                  tes
: 5942 3      else
: 5943 3          if (.ELOG_ADDR [EL_CRN_LO] eal 0) and
: 5944 4          (.ELOG_ADDR [EL_CRN_HI] eal 0)
: 5945 3          then
: 5946 3              select new .ERROR_CODE of
: 5947 3                  set
: 5948 3
: 5949 3                  [ST_MFE]:      TALLY_ADDR [ERR_HRD_SEK] = .TALLY_ADDR [ERR_HRD_SEK] + 1;
: 5950 3
: 5951 3                  [ST_DAT]:      if .ERROR_SUB eal 2
: 5952 3                  then
: 5953 3                      TALLY_ADDR [ERR_HRD_SEK] = .TALLY_ADDR [ERR_HRD_SEK] + 1
: 5954 3                  else
: 5955 3                      TALLY_ADDR [ERR_HRD_DAT] = .TALLY_ADDR [ERR_HRD_DAT] + 1;
: 5956 3
: 5957 3                  [ST_HST]:      TALLY_ADDR [ERR_HRD_HST] = .TALLY_ADDR [ERR_HRD_HST] + 1;
: 5958 3
: 5959 3                  [ST_CNT]:      C_ERR_TBL [.,ICTLR, C_ERR_HRD] = .C_ERR_TBL [.,ICTLR, C_ERR_HRD] + 1;
: 5960 3
: 5961 3
: 5962 3                  ! HARD-CONTROLLER
: 5963 3                  [ST_DRV]:      if .ERROR_SUB eal 3
: 5964 3                  then
: 5965 3                      TALLY_ADDR [ERR_HRD_SEK] = .TALLY_ADDR [ERR_HRD_SEK] + 1

```

86
5)

ZRQAM3
V01.6

RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK\$USER2:([POWERS])ZRQAD0.BL2:6

SEQ 0430
Page 199
(57)

```

; 5965 3
; 5966 3
; 5967 3
; 5968 3
; 5969 3
; 5970 2
; 5971 2
; 5972 2
; 5973 2
; 5974 2
; 5975 2
; 5976 2
; 5977 2
; 5978 1

else
    TALLY_ADDR [ERR_HRD_DRV] = ,TALLY_ADDR [ERR_HRD_DRV] + 1;
tes;
end
else
    IF ,SOFT_OCCURED
    then
        C_ERR_TBL [,ICTLR, C_ERR_SFT] = ,C_ERR_TBL [,ICTLR, C_ERR_SFT] + 1
    else
        C_ERR_TBL [,ICTLR, C_ERR_HRD] = ,C_ERR_TBL [,ICTLR, C_ERR_HRD] + 1;
    end;
end;

! UNIT NOT FOUND
! RTNE SOFT_ERROR

```

```

030614
.SBTL SOFT.ERROR RDRX INTERRUPT SERVICE ROUTINES
.PSECT $CODE$, RO

000000 004137 000000G SOFT.ERROR::
000004 005746 JSR R1,$SAVE5 ; 5859
000006 105046 TST -(SP) ;
000010 105046 CLRB -(SP) ; FOUND 5869
000012 016646 000024 CLRB -(SP) ; SOFT_OCCURED
000016 012746 000102 MOV 24(SP),-(SP) ; INDEX,* 5881
000022 004737 000000G MOV #102, -(SP)
000026 062700 000000G JSR PC,BL$MUL
000032 010001 ADD #ELOG.PKT,RO
000034 116100 000020 MOV R0,R1 ; *,ELOG.ADDR
000040 042700 177740 MOVB 20(R1),R0 ; *(ELOG.ADDR),* 5882
000044 105003 BIC #177740,R0
000046 050003 CLRB R3 ; ERROR.CODE
000050 016105 000020 BIS R0,R3 ; *,ERROR.CODE
000054 006205 MOV 20(R1),R5 ; *(ELOG.ADDR),ERROR.SUB 5883
000056 006205 ASR R5 ; ERROR.SUB
000060 006205 ASR R5 ; ERROR.SUB
000062 006205 ASR R5 ; ERROR.SUB
000064 006205 ASR R5 ; ERROR.SUB
000066 042705 174000 BIC #174000,R5 ; *,ERROR.SUB
000072 013700 000000G MOV SWP.FLAGS,RO ;
000076 042700 077777 BIC #77777,R0 ; 5885
000102 020027 100000 CMP R0,#-100000
000106 001010 BNE 1$
000110 126127 000016 000002? CMPB 16(R1),#2 ; *(ELOG.ADDR),* 5886
000116 001004 BNE 1$
000120 005004 CLR R4 ; RETRIES 5888
000122 156104 000051 BISB 51(R1),R4 ; *(ELOG.ADDR),RETRIES
000126 000402 BR 2$ ;
000130 012704 000001 1$: MOV #1,R4 ; *,RETRIES 5885
000134 005704 2$: TST R4 ; RETRIES 5890
000136 001002 BNE 3$ ; RETRIES 5892
000140 012704 000001 MOV #1,R4 ; *,RETRIES 5894

```


E2

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliiss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0431
Page 200
(57)

000144	012702	000006	3\$:	MOV	#6,R2			
000150	010200		4\$:	MOV	R2,R0		; *,OFFSET	5896
000152	063700	000076		ADD	ICST,ADDR,R0		; OFFSET, *	5898
000156	032710	040000		BIT	#40000,(R0)			
000162	001421			BEQ	5\$			
000164	016146	000012		MOV	12(R1), -(SP)		; *(ELOG,ADDR), *	5899
000170	111046			MOVB	(R0), -(SP)			
000172	042716	177760		BIC	#177760,(SP)			
000176	022626			CMP	(SP)+,(SP)+			
000200	001012			BNE	5\$			
000202	112766	000001 000006		MOVB	#1,6(SP)		; *,FOUND	5902
000210	011046			MOV	(R0), -(SP)			5903
000212	000316			SWAB	(SP)			
000214	042716	177760		BIC	#177760,(SP)			
000220	012666	000010		MOV	(SP)+,10(SP)		; *,UNIT	
000224	000405			BR	6\$			5901
000226	062702	000024	5\$:	ADD	#24,R2		; *,OFFSET	5896
000232	020227	000102		CMP	R2,#102		; OFFSET, *	
000236	003744			BLE	4\$			
000240	112766	000001 000004	6\$:	MOVB	#1,4(SP)		; *,SOFT.OCCURED	5910
000246	032766	000001 000006		BIT	#1,6(SP)		; *,FOUND	5912
000254	001002			BNE	7\$			
000256	000137	031502		JMP	22\$			
000262	016616	000010	7\$:	MOV	10(SP), (SP)		; UNIT, *	5915
000266	012746	000066		MOV	#66, -(SP)			
000272	004737	000000G		JSR	PC,BL\$MUL			
000276	062700	000000G		ADD	#TALLY,R0			
000302	032766	000001 000006		BIT	#1,6(SP)		; *,SOFT.OCCURED	5917
000310	001503			BEQ	14\$			
000312	120327	000005		CMPB	R3,#5		; ERROR.CODE, *	5922
000316	001462			BEQ	12\$			
000320	120327	000010		CMPB	R3,#10		; ERROR.CODE, *	5924
000324	001022			BNE	9\$			
000326	012702	000052		MOV	#52,R2			5926
000332	060002			ADD	R0,R2		; TALLY,ADDR, *	
000334	020527	000002		CMP	R5,#2		; ERROR.SUB, *	5924
000340	001005			BNE	8\$			
000342	005001			CLR	R1			5926
000344	151201			BISB	(R2),R1			
000346	060401			ADD	R4,R1		; RETRIES, *	
000350	110112			MOVB	R1,(R2)			
000352	000543			BR	21\$			5924
000354	005001		8\$:	CLR	R1			5928
000356	156201	000001		BISB	1(R2),R1			
000362	060401			ADD	R4,R1		; RETRIES, *	
000364	110162	000001		MOVB	R1,1(R2)			
000370	000534			BR	21\$			5914
000372	120327	000011	9\$:	CMPB	R3,#11		; ERROR.CODE, *	5930
000376	001007			BNE	10\$			
000400	005001			CLR	R1			
000402	156001	000055		BISB	55(R0),R1		; *(TALLY,ADDR), *	
000406	060401			ADD	R4,R1		; RETRIES, *	
000410	110160	000055		MOVB	R1,55(R0)		; *,*(TALLY,ADDR)	

F2

ZRQAM3	RD/RX EXERCISER		11-Apr-1984 11:08:35	VAX-11 B1199-16 V4.0-579	SEQ 0432
V01.6	RDRX INTERRUPT SERVICE ROUTINES		11-Apr-1984 11:08:22	DISK\$USER2:[POWERS]ZRQADO.BL2;6	Page 201 (57)
000414	000522		BR 21\$		
000416	120327	000012	10\$: CMPB R3,012	; ERROR.CODE,*	5919
000422	001012		BNE 11\$		5932
000424	013702	000104'	MOV ICTLR,R2		
000430	006302		ASL R2		
000432	005001		CLR R1		
000434	156201	000001G	BISB C.ERR.TBL+1(R2),R1		
000440	060401		ADD R4,R1	; RETRIES,*	
000442	110162	000001G	MOV8 R1,C.ERR.TBL+1(R2)		
000446	000505		BR 21\$		
000450	120327	000013	11\$: CMPB R3,013	; ERROR.CODE,*	5919
000454	001102		BNE 21\$		5935
000456	020527	000003	CMP R5,03	; ERROR.SUB,*	
000462	001007		BNE 13\$		
000464	005001		12\$: CLR R1		5937
000466	156001	000052	BISB 52(R0),R1	; *(TALLY.ADDR),*	
000472	060401		ADD R4,R1	; RETRIES,*	
000474	110160	000052	MOV8 R1,52(R0)	; *,*(TALLY.ADDR)	
000500	000470		BR 21\$		5935
000502	005001		13\$: CLR R1		5939
000504	156001	000054	BISB 54(R0),R1	; *(TALLY.ADDR),*	
000510	060401		ADD R4,R1	; RETRIES,*	
000512	110160	000054	MOV8 R1,54(R0)	; *,*(TALLY.ADDR)	
000516	000461		BR 21\$		5919
000520	005761	000006	14\$: TST 6(R1)	; *(ELOG.ADDR)	5943
000524	001056		BNE 21\$		
000526	005761	000010	TST 10(R1)	; *(ELOG.ADDR)	5944
000532	001053		BNE 21\$		
000534	120327	000005	CMPB R3,05	; ERROR.CODE,*	5949
000540	001443		BEQ 19\$		
000542	120327	000010	CMPB R3,010	; ERROR.CODE,*	5951
000546	001013		BNE 16\$		
000550	012704	000046	MOV 046,R4		5953
000554	060004		ADD R0,R4	; TALLY.ADDR,*	
000556	020527	000002	CMP R5,02	; ERROR.SUB,*	5951
000562	001002		BNE 15\$		
000564	105214		INCB (R4)		5953
000566	000435		BR 21\$		5951
000570	105264	000001	15\$: INCB 1(R4)		5955
000574	000432		BR 21\$		5946
000576	120327	000011	16\$: CMPB R3,011	; ERROR.CODE,*	5957
000602	001003		BNE 17\$		
000604	105260	000051	INCB 51(R0)	; *(TALLY.ADDR)	
000610	000424		BR 21\$		5946
000612	120327	000012	17\$: CMPB R3,012	; ERROR.CODE,*	5959
000616	001006		BNE 18\$		
000620	013702	000104'	MOV ICTLR,R2		
000624	006302		ASL R2		
000626	105262	000000G	INCB C.ERR.TBL(R2)		
000632	000413		BR 21\$		5946
000634	120327	000013	18\$: CMPB R3,013	; ERROR.CODE,*	5962
000640	001010		BNE 21\$		
000642	020527	000003	CMP R5,03	; ERROR.SUB,*	

62

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0433
Page 202
(57)

000646	001003			BNE	20\$			
000650	105260	000046	19\$:	INCB	46(R0)		; *(TALLY,ADDR)	5964
000654	000402			BR	21\$			5962
000656	105260	000050	20\$:	INCB	50(R0)		; *(TALLY,ADDR)	5966
000662	005726		21\$:	TST	(SP)+			5914
000664	000415			BR	24\$			5912
000666	013700	000104'	22\$:	MOV	ICTLR,R0			5974
000672	006300			ASL	R0			
000674	062700	000000G		ADD	0C.ERR.TBL,R0			
000700	032766	000001 000004		BIT	01.4(SP)		; *,SOFT.OCCURED	5972
000706	001403			BEQ	23\$			
000710	105260	000001		INCB	1(R0)			5974
000714	000401			BR	24\$			5972
000716	105210		23\$:	INCB	(R0)			5976
000720	062706	000012	24\$:	ADD	012,SP			5859
000724	000207			RTS	PC			

; Routine Size: 235 words, Routine Base: \$CODE\$ + 30614
; Maximum stack depth per invocation: 13 words

H2

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0434
Page 203
(58)

```

: 5979 1 routine ERR_HRD_RTNE (ERRNUM) : novalue *
: 5980 1
: 5981 1 !
: 5982 1 ! THIS ROUTINE DECIDES WHETHER TO ISSUE AN 'ERRHRD' MACRO CALL TO DRS OR TO FAKE
: 5983 1 ! THE SAME EFFECT WITHOUT ISSUING THE CALL
: 5984 1 !-
: 5985 1
: 5986 2 begin
: 5987 2
: 5988 2 local
: 5989 2 CUR_PRIORITY : word;
: 5990 2
: 5991 2 builtin
: 5992 2 PC;
: 5993 2
: 5994 2 GETPRI (CUR_PRIORITY);
: 5995 2 SETPRI (PRI04);
: 5996 2
: 5997 2 if (.ERRNUM lequ 34) or
: 5998 2 (.ERRNUM gtru 38) or
: 5999 2 (.ERRNUM eal 36) or
: 6000 3 (.ERRNUM eal 37)
: 6001 2 then
: 6002 2
: 6003 3 if BIT_TST (SWP_FLAGS, SWF_HRD)
: 6004 2 then
: 6005 2
: 6006 2 case .ERRNUM from 31 to 45 of
: 6007 2 set
: 6008 2
: 6009 2 [31]: ERRHRD (31, EGH_30, EMS_30); ! INVALID COMMAND
: 6010 2
: 6011 2 [32]: ERRHRD (32, EGH_30, EMS_30); ! COMMAND ABORTED
: 6012 2
: 6013 2 [33]: ; !
: 6014 2
: 6015 2 [34]: ; !
: 6016 2
: 6017 2 [35]: ; ! MEDIA FORMAT ERROR
: 6018 2
: 6019 2 [36]: ERRHRD (36, EGH_30, EMS_30); ! WRITE PROTECTED
: 6020 2
: 6021 2 [37]: ERRHRD (37, EGH_30, EMS_30); ! COMPARE ERROR
: 6022 2
: 6023 2 [38]: ; ! DATA ERROR
: 6024 2
: 6025 2 [39]: ERRHRD (39, EGH_30, EMS_30); ! HOST BUFFER ACCESS ERROR
: 6026 2
: 6027 2 [40]: ERRHRD (40, EGH_30, EMS_30); ! CONTROLLER ERROR
: 6028 2
: 6029 2 [41]: ERRHRD (41, EGH_30, EMS_30); ! DRIVE ERROR
: 6030 2
: 6031 2 [42]: ERRHRD (42, EGH_30, 0); ! MUST WRITE COMPARE ERROR

```

ZRQAM3
V01.6RDRX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0435
Page 204
(58)

```

: 6032 2
: 6033 2
: 6034 2
: 6035 2
: 6036 2
: 6037 2
: 6038 2
: 6039 2
: 6040 3
: 6041 3
: 6042 3
: 6043 3
: 6044 3
: 6045 3
: 6046 4
: 6047 4
: 6048 4
: 6049 3
: 6050 2
: 6051 2
: 6052 2
: 6053 3
: 6054 2
: 6055 2
: 6056 3
: 6057 2
: 6058 2
: 6059 2
: 6060 2
: 6061 2
: 6062 2
: 6063 2
: 6064 2
: 6065 2
: 6066 2
: 6067 3
: 6068 3
: 6069 3
: 6070 3
: 6071 3
: 6072 2
: 6073 2
: 6074 2
: 6075 2
: 6076 1

[43]: ERRHRD (43, EGH_30, EMS_30); ! MESSAGE FROM INTERNAL DIAGNOSTICS
[44]: ERRHRD (44, EGH_30, EMS_30); ! DUPLICATE UNIT NUMBER
[45]: ERRHRD (45, EGH_30, EMS_30); ! INVALID END CODE
tes
else
begin
!****increment error count
PRINTB (HRD_MSG, .ERRNUM, .L$LUN, .PC); ! INCREMENT TOTAL ERROR COUNT
! PRINT ERROR MESSAGE JUST LIKE DRS
if .ERRNUM neq 42
then
begin
PRINTB (HRD_SUB); ! NEXT LINE FOR NON-HOST COMPARE ERRORS
EMS_ERR (); ! PRINT REST OF THE INFORMATION
end;
end;
if (.ERRNUM eq 35) or
(.ERRNUM eq 38)
then
! FOR BAD-BLOCK TYPE ERRORS
if BIT_TST (SWP_FLAGS, SWF_BLK)
then
! IF ERRORS TO BE TREATED NORMALLY
select neu .ERRNUM of
set
[35]: ERRHRD (35, EGH_30, EMS_30); ! MEDIA FORMAT ERROR
[38]: ERRHRD (38, EGH_30, EMS_30); ! DATA ERROR
tes
else
begin
!****increment error count
PRINTB (HRD_MSG, .ERRNUM, .L$LUN, .PC); ! INCREMENT TOTAL ERROR COUNT
PRINTB (HRD_SUB); ! PRINT ERROR LINE JUST LIKE DRS
EMS_ERR (); ! PRINT NEXT LINE TOO
end;
! PRINT REST OF THE INFORMATION
end;
SETPRI (.CUR_PRIORITY); ! PRIORITY BACK TO NORMAL
end;

```

```

000000 004137 000000G      .SBYTL ERR.HRD.RTNE RDRX INTERRUPT SERVICE ROUTINES
ERR.HRD.RTNE:
JSR      R1,$SAVE2
TRAP     40
MOV      R0,R2
MOV      #200,R0

```

5970
5994
5995

J2

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0436
Page 205
(58)

000014	104441			TRAP	41		
000016	016601	000010		MOV	10(SP),R1	; ERRNUM,*	5997
000022	020127	000042		CMP	R1,#42		
000026	101111			BLOS	1#		
000030	020127	000046		CMP	R1,#46		5998
000034	101006			BHI	1#		
000036	020127	000044		CMP	R1,#44		5999
000042	001403			BEQ	1#		
000044	020127	000045		CMP	R1,#45		6000
000050	001131			BNE	16#		
000052	032737	010000	000000G	1#:	BIT	#10000,SWP.FLAGS	6003
000060	001475			BEQ	14#		
000062	010100			MOV	R1,R0		6006
000064	162700	000037		SUB	#37,R0		
000070	006300			ASL	R0		
000072	066007	000024'		ADD	P,AAC(R0),PC	; Case dispatch	
000076	104456		3#:	TRAP	56		6009
000100	000037			.WORD	37		
000102	000000G			.WORD	EGH.30		
000104	000000G			.WORD	EMS.30		
000106	000512			BR	16#		6006
000110	104456		4#:	TRAP	56		6011
000112	000040			.WORD	40		
000114	000000G			.WORD	EGH.30		
000116	000000G			.WORD	EMS.30		
000120	000505			BR	16#		6006
000122	104456		5#:	TRAP	56		6019
000124	000044			.WORD	44		
000126	000000G			.WORD	EGH.30		
000130	000000G			.WORD	EMS.30		
000132	000500			BR	16#		6006
000134	104456		6#:	TRAP	56		6021
000136	000045			.WORD	45		
000140	000000G			.WORD	EGH.30		
000142	000000G			.WORD	EMS.30		
000144	000473			BR	16#		6006
000146	104456		7#:	TRAP	56		6025
000150	000047			.WORD	47		
000152	000000G			.WORD	EGH.30		
000154	000000G			.WORD	EMS.30		
000156	000466			BR	16#		6006
000160	104456		8#:	TRAP	56		6027
000162	000050			.WORD	50		
000164	000000G			.WORD	EGH.30		
000166	000000G			.WORD	EMS.30		
000170	000461			BR	16#		6006
000172	104456		9#:	TRAP	56		6029
000174	000051			.WORD	51		
000176	000000G			.WORD	EGH.30		
000200	000000G			.WORD	EMS.30		
000202	000454			BR	16#		6006
000204	104456		10#:	TRAP	56		6031
000206	000052			.WORD	52		

K2

ZRQAM3
V01.6RD/RX EXERCISER
RURX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0437
Page 206
(58)

000210	000000G		.WORD	EGH.30		
000212	000000		.WORD	0		
000214	000447		BR	16\$		
000216	104456	11\$:	TRAP	56		6006
000220	000053		.WORD	53		6033
000222	000000G		.WORD	EGH.30		
000224	000000G		.WORD	EMS.30		
000226	000442		BR	16\$		
000230	104456	12\$:	TRAP	56		6006
000232	000054		.WORD	54		6035
000234	000000G		.WORD	EGH.30		
000236	000000G		.WORD	EMS.30		
000240	000435		BR	16\$		
000242	104456	13\$:	TRAP	56		6006
000244	000055		.WORD	55		6037
000246	000000G		.WORD	EGH.30		
000250	000000G		.WORD	EMS.30		
000252	000430		BR	16\$		
000254	010746	14\$:	MOV	PC, -(SP)		6003
000256	013746	000000G	MOV	L\$LUN, -(SP)		6042
000262	010146		MOV	R1, -(SP)		
000264	012746	000000G	MOV	\$HRD.MSG, -(SP)		
000270	012746	000004	MOV	\$4, -(SP)		
000274	010600		MOV	SP, R0		
000276	104414		TRAP	14		
000300	020127	000052	CMP	R1, \$52		
000304	001411		BEQ	15\$		6044
000306	012716	000000G	MOV	\$HRD.SUB, (SP)		
000312	012746	000001	MOV	\$1, -(SP)		6047
000316	010600		MOV	SP, R0		
000320	104414		TRAP	14		
000322	004737	000000G	JSR	PC, EMS.ERR		
000326	005726		TST	(SP),		6048
000330	062706	000012	ADD	\$12, SP		6046
000334	020127	000043	15\$:	CMP		6040
000340	001403		16\$:	R1, \$43		6052
000342	020127	000046	BEQ	17\$		
000346	001050		CMP	R1, \$46		6053
000350	032737	040000	BNE	20\$		
000356	001420	000000G	17\$:	BIT		6056
000360	020127	000043	BEQ	\$40000, SWP.FLAGS		
000364	001005		CMP	R1, \$43		6062
000366	104456		BNE	18\$		
000370	000043		TRAP	56		
000372	000000G		.WORD	43		
000374	000000G		.WORD	EGH.30		
000376	000434		.WORD	EMS.30		
000400	020127	000046	BR	20\$		
000404	001031	18\$:	CMP	R1, \$46		6054
000406	104456		BNE	20\$		6064
000410	000046		TRAP	56		
000412	000000G		.WORD	46		
000414	000000G		.WORD	EGH.30		
			.WORD	EMS.30		

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0,BL2;6SEQ 0438
Page 207
(58)

000416	000424		BR	20:			
000420	010746		MOV	PC, -(SP)			6056
000422	013746	000000G	19:	MOV	L\$LUN, (SP)	; PC, *	6069
000426	010146		MOV	R1, -(SP)			
000430	012746	000000G	MOV	HRD.MSG, -(SP)			
000434	012746	000004	MOV	4, -(SP)			
000440	010600		MOV	SP, R0		; SP, *	
000442	104414		TRAP	14			
000444	012716	000000G	MOV	HRD.SUB, (SP)			6070
000450	012746	000001	MOV	1, -(SP)			
000454	010600		MOV	SP, R0		; SP, *	
000456	104414		TRAP	14			
000460	004737	000000G	JSR	PC, EMS.ERR			6071
000464	062706	000014	ADD	14, SP			6067
000470	010200		20:	MOV	R2, R0	; CUR.PRIORITY, *	6074
000472	104441		TRAP	41			
000474	000207		RTS	PC			5979

; Routine Size: 159 words, Routine Base: \$CODE\$ + 31542
 ; Maximum stack depth per invocation: 11 words

000024

.PSECT \$PLIT\$, R0, D

P.AAC:
2:

000024	000000		.WORD	0		; CASE Table for ERR.HRD.RTNE+0072	6006
000026	000012		.WORD	12		; [3]	
000030	000236		.WORD	236		; [4]	
000032	000236		.WORD	236		; [16]	
000034	000236		.WORD	236		; [16]	
000036	000024		.WORD	24		; [16]	
000040	000036		.WORD	36		; [5]	
000042	000236		.WORD	236		; [6]	
000044	000050		.WORD	50		; [16]	
000046	000062		.WORD	62		; [7]	
000050	000074		.WORD	74		; [8]	
000052	000106		.WORD	106		; [9]	
000054	000120		.WORD	120		; [10]	
000056	000132		.WORD	132		; [11]	
000060	000144		.WORD	144		; [12]	
						; [13]	

M2

SEQ 0439
Page 208
(59)ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

```

: 6077 1 routine ERR_SOFT_RTNE (ERRNUM) : novalue =
: 6078 1
: 6079 1
: 6080 1 ! THIS ROUTINE DECIDES WHETHER TO ISSUE AN 'ERRSOFT' MACRO CALL TO DRS OR TO FAKE
: 6081 1 ! THE SAME EFFECT WITHOUT ISSUING THE CALL
: 6082 1 !-
: 6083 1
: 6084 2 begin
: 6085 2
: 6086 2 builtin
: 6087 2 PC;
: 6088 2
: 6089 3 if BIT_TST (SWP_FLAGS, SWF_SFT) ! IF SOFT ERRORS TO BE TREATED LIKE OTHER ERRORS
: 6090 2 then
: 6091 2
: 6092 2 case .ERRNUM from 50 to 54 of
: 6093 2 set
: 6094 2
: 6095 2 [50]: ERRSOFT (50, 0, 0); ! CONTROLLER ERROR
: 6096 2
: 6097 2 [51]: ERRSOFT (51, 0, 0); ! HOST MEMORY ACCESS ERROR
: 6098 2
: 6099 2 [52]: ERRSOFT (52, 0, 0); ! DISK TRANSFER ERROR
: 6100 2
: 6101 2 [53]: ERRSOFT (53, 0, 0); ! SDI ERROR
: 6102 2
: 6103 2 [54]: ERRSOFT (54, 0, 0); ! SMALL DISK ERROR
: 6104 2 tes
: 6105 2 else
: 6106 3 begin
: 6107 3 !****increment error count ! INCREMENT TOTAL ERROR COUNT
: 6108 3 PRINTB (SFT_MSG, .ERRNUM, .L$LUN, .PC); ! PRINT ERROR LINE JUST LIKE DRS
: 6109 2 end;
: 6110 2
: 6111 1 end;

```

032240

.SBTTL ERR_SOFT_RTNE RDRX INTERRUPT SERVICE ROUTINES
.PSECT \$CODE\$, RO

000000	032737	020000	000000G	ERR_SOFT_RTNE:			
000006	001440			BIT	#20000, SWP_FLAGS	:	6089
000010	016600	000002		BEQ	7:		
000014	162700	000062		MOV	2(SP), R0	: ERRNUM, *	6092
000020	006300			SUB	#62, R0		
000022	066007	000062'		ASL	R0		
000026	104457			ADD	P, AAD(R0), PC	: Case dispatch	
000030	000062			2\$: TRAP	57	:	6095
000032	000000			.WORD	62		
000034	000000			.WORD	0		
000036	000207			.WORD	0		
000040	104457			RTS	PC	:	6097
				3\$: TRAP	57	:	6097

N2

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6

SEQ 0440
Page 209
(59)

000042	000063		.WORD	63		
000044	000000		.WORD	0		
000046	000000		.WORD	0		
000050	000207		RTS	PC		
000052	104457	4\$:	TRAP	57		6092
000054	000064		.WORD	64		6099
000056	000000		.WORD	0		
000060	000000		.WORD	0		
000062	000207		RTS	PC		
000064	104457	5\$:	TRAP	57		6092
000066	000065		.WORD	65		6101
000070	000000		.WORD	0		
000072	000000		.WORD	0		
000074	000207		RTS	PC		
000076	104457	6\$:	TRAP	57		6092
000100	000066		.WORD	66		6103
000102	000000		.WORD	0		
000104	000000		.WORD	0		
000106	000207		RTS	PC		
000110	010746	7\$:	MOV	PC, -(SP)	; PC, *	6089
000112	013746	000000G	MOV	L\$LUN, -(SP)		6108
000116	016646	000006	MOV	6(SP), -(SP)	; ERRNUM, *	
000122	012746	000000G	MOV	\$SFT.MSG, -(SP)		
000126	012746	000004	MOV	\$4, -(SP)		
000132	010600		MOV	SP, R0	; SP, *	
000134	104414		TRAP	14		
000136	062706	000012	ADD	\$12, SP		6106
000142	000207		RTS	PC		6077

; Routine Size: 50 words, Routine Base: \$CODE\$ + 32240
; Maximum stack depth per invocation: 7 words

000062 .PSECT \$PLIT\$, R0, D

000062	000000	P.AAD:			; CASE Table for ERR.SOFT.RTNE+0022	6092
000064	000012	1\$:	.WORD	0	; [2\$]	
000066	000024		.WORD	12	; [3\$]	
000070	000036		.WORD	24	; [4\$]	
000072	000050		.WORD	36	; [5\$]	
			.WORD	50	; [6\$]	

B3

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B11:16 V4.0-579
DISK1USER2:[POWERS]ZRQADO.BL2;6

SEQ 0441
Page 210
(60)

```

: 6112 1 routine ERR_HRD_RTNE_APT (ERRNUM) : novalue =
: 6113 1
: 6114 1
: 6115 1
: 6116 1 ! THIS ROUTINE DECIDES WHETHER TO ISSUE AN 'ERRHRD' MACRO CALL TO DRS OR TO FAKE
: 6117 1 ! THE SAME EFFECT WITHOUT ISSUING THE CALL
: 6118 1 !
: 6119 1
: 6120 1
: 6121 2 begin
: 6122 2
: 6123 2
: 6124 2 local
: 6125 2 CUR_PRIORITY;
: 6126 2
: 6127 2 builtin
: 6128 2 PC;
: 6129 2
: 6130 2 GETPRI (CUR_PRIORITY);
: 6131 2 SETPRI (PRI04);
: 6132 2 ! DON'T ALLOW SOFT_ERROR MESSAGES TO COME IN NOW
: 6133 2
: 6134 2 if .APT_MODE
: 6135 2 then
: 6136 2
: 6137 3 begin
: 6138 3 .MAIL_BOX_TESTNUM = .RP_ADDR (LBN_L0);
: 6139 3 .MAIL_BOX_SUBTST = .RP_ADDR (DISK);
: 6140 3 end;
: 6141 2
: 6142 2
: 6143 2 if (.ERRNUM lequ 34) or
: 6144 2 (.ERRNUM gtru 38) or
: 6145 2 (.ERRNUM eq 36) or
: 6146 3 (.ERRNUM eq 37)
: 6147 3
: 6148 2 then
: 6149 2
: 6150 3 if BIT_1ST (SWP_FLAGS, SWP_HRD)
: 6151 2 then
: 6152 2
: 6153 2 case .ERRNUM from 31 to 45 of
: 6154 2 set
: 6155 2
: 6156 2 [31]: ERRDF (31, EGH_30, EMS_30); ! INVALID COMMAND
: 6157 2
: 6158 2 [32]: ERRDF (32, EGH_30, EMS_30); ! COMMAND ABORTED
: 6159 2
: 6160 2 [33]: ;
: 6161 2
: 6162 2 [34]: ;
: 6163 2
: 6164 2 [35]: ; ! MEDIA FORMAT ERROR

```

C3

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0442
Page 211
(60)

```

: 6165 2
: 6166 2
: 6167 2
: 6168 2
: 6169 2
: 6170 2
: 6171 2
: 6172 2
: 6173 2
: 6174 2
: 6175 2
: 6176 2
: 6177 2
: 6178 2
: 6179 2
: 6180 2
: 6181 2
: 6182 2
: 6183 2
: 6184 2
: 6185 2
: 6186 2
: 6187 2
: 6188 2
: 6189 3
: 6190 3
: 6191 3
: 6192 3
: 6193 3
: 6194 3
: 6195 3
: 6196 3
: 6197 4
: 6198 4
: 6199 4
: 6200 3
: 6201 2
: 6202 2
: 6203 2
: 6204 3
: 6205 3
: 6206 2
: 6207 2
: 6208 3
: 6209 2
: 6210 2
: 6211 2
: 6212 2
: 6213 2
: 6214 2
: 6215 2
: 6216 2
: 6217 2

[36]: ERRDF (36, EGH_30, EMS_30); ! WRITE PROTECTED
[37]: ERRDF (37, EGH_30, EMS_30); ! COMPARE ERROR
[38]: ; ! DATA ERROR
[39]: ERRDF (39, EGH_30, EMS_30); ! HOST BUFFER ACCESS ERROR
[40]: ERRDF (40, EGH_30, EMS_30); ! CONTROLLER ERROR
[41]: ERRDF (41, EGH_30, EMS_30); ! DRIVE ERROR
[42]: ERRDF (42, EGH_30, 0); ! HOST WRITE COMPARE ERROR
[43]: ERRDF (43, EGH_30, EMS_30); ! MESSAGE FROM INTERNAL DIAGNOSTICS
[44]: ERRDF (44, EGH_30, EMS_30); ! DUPLICATE UNIT NUMBER
[45]: ERRDF (45, EGH_30, EMS_30); ! INVALID END CODE
tes

else
begin
!****increment error count ! INCREMENT TOTAL ERROR COUNT
PRINTB (DF_MSG, .ERRNUM, .L$LUN, .PC); ! PRINT ERROR MESSAGE JUST LIKE DRS

if .ERRNUM neq 42
then
begin
PRINTB (HRD_SUB); ! NEXT LINE FOR NON-HOST COMPARE ERRORS
EMS_ERR (); ! PRINT REST OF THE INFORMATION
end;
end;

if (.ERRNUM eq 35) or
(.ERRNUM eq 38)
then
if BIT_TST (SWP_FLAGS, SWF_BLK)
then
select one of
get
[35]: ERRDF (35, EGH_30, EMS_30); ! MEDIA FORMAT ERROR
[38]: ERRDF (38, EGH_30, EMS_30); ! DATA ERROR
tes

```

D3

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1111-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0443
Page 212
(60)

```

: 6218 2
: 6219 2
: 6220 2
: 6221 3
: 6222 3
: 6223 3
: 6224 3
: 6225 3
: 6226 2
: 6227 2
: 6228 2
: 6229 2
: 6230 2
: 6231 2
: 6232 1

                else
                    begin
!****increment error count
PRINTB (DF_MSG, .ERRNUM, .L$LUN, .PC);
PRINTB (HRD_SUB);
EMS_ERR ();
end;
                    ! INCREMENT TOTAL ERROR COUNT
                    ! PRINT ERROR LINE JUST LIKE DRS
                    ! PRINT NEXT LINE TOO
                    ! PRINT REST OF THE INFORMATION

                SETPRI (.CUR_PRIORITY);
                    ! PRIORITY BACK TO NORMAL

            end;

```

032404

.SBTTL ERR.HRD.RTNE.APT RDRX INTERRUPT SERVICE ROUTINES
.PSECT \$CODE\$, RO

```

000000 004137 000000G      ENR.HRD.RTNE.APT:
000004 104440      JSR      R1,$SAV.2
000006 010002      TRAP     40
000010 012700 000200      MOV      R0,R2
000014 104441      MOV      @200,R0
000016 032737 000001 001162'  TRAP     41
000024 001412      BIT      @1,APT.MODE
000026 013700 000000G      BEQ      1$
000032 016077 000050 001164'  MOV      RP,ADDR,R0
000040 013700 000000G      MOV      50(R0),@MAIL.BOX.TESTNUM
000044 016077 000010 001166'  MOV      RP,ADDR,R0
000052 016601 000010      MOV      10(R0),@MAIL.BOX.SUBTST
000056 020127 000042 1$:      MOV      10(SP),R1
000062 101411      CMP      R1,@42
000064 020127 000046      BLOS     2$
000070 101006      CMP      R1,@46
000072 020127 000044      BHI      2$
000076 001403      CMP      R1,@44
000100 020127 000045      BEQ      2$
000104 001131      CMP      R1,@45
000106 032737 010000 000000G  BNE      17$
000114 001475 2$:      BIT      @10000,SWP.FLAGS
000116 010100      BEQ      15$
000120 162700 000037      MOV      R1,R0
000124 006300      SUB      @37,R0
000126 066007 000074'      ASL      R0
000132 104455 4$:      ADD      P,AAE(R0),PC
000134 000037      TRAP     55
000136 000000G      .WORD     37
000140 000000G      .WORD     ECH.30
000142 000512      .WORD     EMS.30
000144 104455 5$:      BR       17$
                        TRAP     55

```

F 3

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B11ss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ 0444
Page 213
(60)

000146	000040		.WORD	40		
000150	000000G		.WORD	EGH.30		
000152	000000G		.WORD	EMS.30		
000154	000505		BR	17\$		
000156	104455	6\$:	TRAP	55	:	6153
000160	000044		.WORD	44	:	6166
000162	000000G		.WORD	EGH.30		
000164	000000G		.WORD	EMS.30		
000166	000500		BR	17\$		
000170	104455	7\$:	TRAP	55	:	6153
000172	000045		.WORD	45	:	6168
000174	000000G		.WORD	EGH.30		
000176	000000G		.WORD	EMS.30		
000200	000473		BR	17\$		
000202	104455	8\$:	TRAP	55	:	6153
000204	000047		.WORD	47	:	6172
000206	000000G		.WORD	EGH.30		
000210	000000G		.WORD	EMS.30		
000212	000466		BR	17\$		
000214	104455	9\$:	TRAP	55	:	6153
000216	000050		.WORD	50	:	6174
000220	000000G		.WORD	EGH.30		
000222	000000G		.WORD	EMS.30		
000224	000461		BR	17\$		
000226	104455	10\$:	TRAP	55	:	6153
000230	000051		.WORD	51	:	6176
000232	000000G		.WORD	EGH.30		
000234	000000G		.WORD	EMS.30		
000236	000454		BR	17\$		
000240	104455	11\$:	TRAP	55	:	6153
000242	000052		.WORD	52	:	6178
000244	000000G		.WORD	EGH.30		
000246	000000		.WORD	0		
000250	000447		BR	17\$		
000252	104455	12\$:	TRAP	55	:	6153
000254	000053		.WORD	53	:	6180
000256	000000G		.WORD	EGH.30		
000260	000000G		.WORD	EMS.30		
000262	000442		BR	17\$		
000264	104455	13\$:	TRAP	55	:	6153
000266	000054		.WORD	54	:	6182
000270	000000G		.WORD	EGH.30		
000272	000000G		.WORD	EMS.30		
000274	000435		BR	17\$		
000276	104455	14\$:	TRAP	55	:	6153
000300	000055		.WORD	55	:	6184
000302	000000G		.WORD	EGH.30		
000304	000000G		.WORD	EMS.30		
000306	000430		BR	17\$		
000310	010746	15\$:	MOV	PC, -(SP)	:	6150
000312	013746	000000G	MOV	L\$UN, -(SP)	:	6191
000316	010146		MOV	R1, -(SP)		
000320	012746	000000G	MOV	#DF.MSG, -(SP)		

F3

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6SEQ.0445
Page 214
(60)

000324	012746	000004		MOV	04, -(SP)		
000330	010600			MOV	SP, R0	; SP, *	
000332	104414			TRAP	14		
000334	020127	000052		CMP	R1, 052		6194
000340	001411			BEQ	16\$		
000342	012716	000000G		MOV	0HRD, SUB, (SP)		6198
000346	012746	000001		MOV	01, -(SP)		
000352	010600			MOV	SP, R0	; SP, *	
000354	104414			TRAP	14		
000356	004737	000000G		JSR	PC, EMS, ERR		6199
000362	005726			YST	(SP),		6197
000364	062706	000012	16\$:	ADD	012, SP		6189
000370	020127	000043	17\$:	CMP	R1, 043		6203
000374	001403			BEQ	18\$		
000376	020127	000046		CMP	R1, 046		6204
000402	001050			BNE	21\$		
000404	032737	040000 000000G	18\$:	BIT	040000, SWP, FLAGS		6208
000412	001420			BEQ	20\$		
000414	020127	000043		CMP	R1, 043		6214
000420	001005			BNE	19\$		
000422	104455			TRAP	55		
000424	000043			.WORD	43		
000426	000000G			.WORD	EGH, 30		
000430	000000G			.WORD	EMS, 30		
000432	000434			BR	21\$		6211
000434	020127	000046	19\$:	CMP	R1, 046		6216
000440	001031			BNE	21\$		
000442	104455			TRAP	55		
000444	000046			.WORD	46		
000446	000000G			.WORD	EGH, 30		
000450	000000G			.WORD	EMS, 30		
000452	000424			BR	21\$		6208
000454	010746		20\$:	MOV	PC, -(SP)	; PC, *	6223
000455	013746	000000G		MOV	L\$LUN, (SP)		
000456	010146			MOV	R1, -(SP)		
000464	012746	000000G		MOV	0DF, MSG, -(SP)		
000470	012746	000004		MOV	04, -(SP)		
000474	010600			MOV	SP, R0	; SP, *	
000476	104414			TRAP	14		
000500	012716	000000G		MOV	0HRD, SUB, (SP)		6224
000504	012746	000001		MOV	01, -(SP)		
000510	010600			MOV	SP, R0	; SP, *	
000512	104414			TRAP	14		
000514	004737	000000G		JSR	PC, EMS, ERR		6205
000520	062706	000014		ADD	014, SP		6221
000524	010200		21\$:	MOV	R2, R0	; CUR.PRIORITY, *	6229
000526	104441			TRAP	41		
000530	000207			RTS	PC		6112

; Routine Size: 173 words, Routine Base: \$CODE\$ + 32404
; Maximum stack depth per invocation: 11 words

63

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0446
Page 215
(60)

000074

.PSECT \$PLIT\$, RD , D

P.AAE:
3\$:

000074 000000
000076 000012
000100 000236
000102 000236
000104 000236
000106 000024
000110 000036
000112 000236
000114 000050
000116 000062
000120 000074
000122 000106
000124 000120
000126 000132
000130 000144

.WORD 0
.WORD 12
.WORD 236
.WORD 236
.WORD 236
.WORD 24
.WORD 36
.WORD 236
.WORD 50
.WORD 62
.WORD 74
.WORD 106
.WORD 120
.WORD 132
.WORD 144

; CASE Table for ERR.HRD.RTNE.AP+0126 6153
; [4\$]
; [5\$]
; [17\$]
; [17\$]
; [17\$]
; [6\$]
; [7\$]
; [17\$]
; [8\$]
; [9\$]
; [10\$]
; [11\$]
; [12\$]
; [13\$]
; [14\$]

; 6233 1
; 6234 1

H3

SEQ 044 /
Page 216
(61)ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

```

: 6235 1 routine ERR_SOFT_RTNE_APT (ERRNUM, index) : novalue =
: 6236 1
: 6237 1 !
: 6238 1 ! THIS ROUTINE DECIDES WHETHER TO ISSUE AN 'ERRSOFT' MACRO CALL TO DRS OR TO FAKE
: 6239 1 ! THE SAME EFFECT WITHOUT ISSUING THE CALL
: 6240 1 !-
: 6241 1
: 6242 2 begin
: 6243 2
: 6244 2 local
: 6245 2 ELOG_ADDR : ref block [EP_LEN, word] field (EP_FIELDS);
: 6246 2
: 6247 2 builtin
: 6248 2 PC;
: 6249 2
: 6250 2 ELOG_ADDR = ELOG_PKT + (.index * EP_LEN * 2); ! ADDRESS OF THE SAVED ERROR-LOG INFORMATION
: 6251 2
: 6252 2 if .APT_MODE
: 6253 2 then
: 6254 3 begin
: 6255 3 .MAIL_BOX_TESTNUM = .ELOG_ADDR [EL_BLOCK]; ! CHANGE TEST NUMBER TO SHOW LBN UNDER APT ONLY
: 6256 3 .MAIL_BOX_SUBST = .ELOG_ADDR [EL_DK_NUM]; ! CHANGE SUB-TEST NUMBER TO SHOW DISK NUMBER IN APT ONLY
: 6257 2 end;
: 6258 2
: 6259 3 if BIT_TST (SWP_FLAGS, SWF_SFT) ! IF SOFT ERRORS TO BE TREATED LIKE OTHER ERRORS
: 6260 2 then
: 6261 2
: 6262 2 case .ERRNUM from 50 to 54 of
: 6263 2 set
: 6264 2
: 6265 2 [50]: ERRDF (50, 0, 0); ! (ONTROLLER ERROR
: 6266 2
: 6267 2 [51]: ERRDF (51, 0, 0); ! HOST MEMORY ACCESS ERROR
: 6268 2
: 6269 2 [52]: ERRDF (52, 0, 0); ! DISK TRANSFER ERROR
: 6270 2
: 6271 2 [53]: ERRDF (53, 0, 0); ! SDI ERROR
: 6272 2
: 6273 2 [54]: ERRDF (54, 0, 0); ! SMALL DISK ERROR
: 6274 2 tes
: 6275 2 else
: 6276 3 begin
: 6277 3 !****increment error count
: 6278 3 PRINTB (DF_MSG, .ERRNUM, .L$LUN, .PC); ! INCREMENT TOTAL ERROR COUNT
: 6279 2 end; ! PRINT ERROR LINE JUST LIKE DRS
: 6280 2
: 6281 1 end;

```

033136

.SBTTL ERR_SOFT_RTNE_APT RDRX INTERRUPT SERVICE ROUTINES
.PSECT \$CODE\$, RO

000000 016646 000002

ERR_SOFT_RTNE_APT;

ZRQAM3
V01.6RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQAD0.BL2;6SEQ 0448
Page 217
(61)

000004	012746	000102		MOV	2(SP), -(SP)	; INDEX, *	6250
000010	004737	000000G		MOV	#102, -(SP)		
000014	062700	000000G		JSR	PC, BL\$MUL		
000020	032737	000001	001162'	ADD	#ELOG.PKT, R0		
000026	001406			BIT	#1, APT.MODE		6252
000030	016077	000056	001164'	BEQ	1\$		
000036	016077	000012	001166'	MOV	56(R0), #MAIL.BOX.TESTNUM	; *(ELOG.ADDR), *	6255
000044	032737	020000	000000G	MOV	12(R0), #MAIL.BOX.SUBTST	; *(ELOG.ADDR), *	6256
000052	001110			BIT	#20000, SWP.FLAGS		6259
000054	016600	000010		BEQ	8\$		
000060	162700	000062		MOV	10(SP), R0	; ERRNUM, *	6262
000064	006300			SUB	#62, R0		
000066	066007	000132'		ASL	R0		
000072	104455			ADD	P.AAF(R0), PC	; Case dispatch	
000074	000062			TRAP	55		6265
000076	000000			.WORD	62		
000100	000000			.WORD	0		
000102	000441			.WORD	0		
000104	104455			BR	9\$		6262
000106	000063			TRAP	55		6267
000110	000000			.WORD	63		
000112	000000			.WORD	0		
000114	000434			.WORD	0		
000116	104455			BR	9\$		6262
000120	000064			TRAP	55		6269
000122	000000			.WORD	64		
000124	000000			.WORD	0		
000126	000427			.WORD	0		
000130	104455			BR	9\$		6262
000132	000065			TRAP	55		6271
000134	000000			.WORD	65		
000136	000000			.WORD	0		
000140	000422			.WORD	0		
000142	104455			BR	9\$		6262
000144	000066			TRAP	55		6273
000146	000000			.WORD	66		
000150	000000			.WORD	0		
000152	000415			.WORD	0		
000154	010716			BR	9\$		6259
000156	013746	000000G		MOV	PC, (SP)	; PC, *	6278
000162	016646	000012		MOV	L\$LUN, -(SP)		
000166	012746	000000G		MOV	12(SP), -(SP)	; ERRNUM, *	
000172	012746	000004		MOV	#DF.MSG, -(SP)		
000176	010600			MOV	#4, (SP)		
000200	104414			MOV	SP, R0	; SP, *	
000202	062706	000010		TRAP	14		
000206	022626			ADD	#10, SP		6276
000210	000207			CMP	(SP), (SP)+		6242
				RTS	PC		6235

; Routine Size: 69 words, Routine Base: \$CODE\$ + 33136
 ; Maximum stack depth per invocation: 8 words

J3

ZRQAM3
V01.6

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0449
Page 218
(61)

000132

.PSECT \$PLIT\$, RO, D

000132 000000
000134 000012
000136 000024
000140 000036
000142 000050

P.AAF:
2\$: .WORD 0
.WORD 12
.WORD 24
.WORD 36
.WORD 50

; CASE Table for ERR.SOFT.RTNE.A+0066 6262
; [3\$]
; [4\$]
; [5\$]
; [6\$]
; [7\$]

; 6282 1
; 6283 1
; 6284 1 end
; 6285 1
; 6286 0 eludom

OTS external references

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

PSECT SUMMARY

Psect Name	Words	Attributes
\$GGG\$	324	RO, I, LCL, REL, CON
\$CODE\$	7028	RO, I, LCL, REL, CON
\$PLIT\$	50	RO, D, LCL, REL, CON

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
DISK\$USER2:[POWERS]ZRQADO.L16;3	404	333	82	21	00:00.1

COMMAND QUALIFIERS

BLISS/PDP11 ZRQADO.BL2/LIST*ZRQADO.L52/OBJECT*ZRQADO.OB2/SOURCE*PAGE:53

K3

ZRQAM4

RD/RX EXERCISER
RDRX INTERRUPT SERVICE ROUTINES11-Apr-1984 11:08:35
11-Apr-1984 11:08:22VAX 11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZROAD0.BL2;6SEQ 0450
Page 219
(62)

```

: 6287 0 module ZRQAM4 (
: 6288 0
: 6289 0 *title 'RD/RX EXERCISER'
: 6290 0 ident = 'V01.6',
: 6291 0 addressing_mode (absolute),
: 6292 0 environment (noeig)
: 6293 0 )
: 6294 0
: 6295 1 begin
: 6296 1
: 6297 1 *sbttl 'LASTAD AND SETUP'
: 6298 1
: 6299 1 library 'ZRQADO.L16';
: 6300 1
: 6301 1 require 'BLSMAC.REQ';
: 7792 1 ! DIAGNOSTIC SUPERVISOR LIBRARY
: 7793 2 LASTAD
: 7794 2
: 7795 2 BGNSETUP (4)
: 7796 2 !ZZZ
: P 7797 2 BGNPTAB
: P 7798 2 INIT_IP_ADDR, INIT_INTR_VECT, INIT_BR_LEVEL, %0'100060', 0, 0, RD52_MAX_LBN, 0 !ZZZ
: P 7799 2 ! IP, VECTOR, BR, DISK ADDR, START BLOCK, END BLOCK
: P 7800 2
: 7801 2 ENDP TAB
: 7802 2
: P 7803 2 BGNPTAB
: P 7804 2 INIT_IP_ADDR, INIT_INTR_VECT, INIT_BR_LEVEL, %0'100001', 0, 0, RX50_MAX_LBN, 0 !ZZZ
: P 7805 2 ! IP, VECTOR, BR, DISK ADDR, START BLOCK, END BLOCK
: 7806 2 ENDP TAB
: 7807 2
: P 7808 2 BGNPTAB
: P 7809 2 INIT_IP_ADDR, INIT_INTR_VECT, INIT_BR_LEVEL, %0'100002', 0, 0, RX50_MAX_LBN, 0 !ZZZ
: P 7810 2 ! IP, VECTOR, BR, DISK ADDR, START BLOCK, END BLOCK
: 7811 2 ENDP TAB
: P 7812 2 BGNPTAB
: P 7813 2 INIT_IP_ADDR, INIT_INTR_VECT, INIT_BR_LEVEL, %0'100003', 0, 0, RX50_MAX_LBN, 0 !ZZZ
: P 7814 2 !HERE'S ONE FOR THE 4TH DRIVE
: 7815 2 ENDP TAB
: 7816 2 !ZZZ
: 7817 1 ENDSETUP

```

```

.TITLE ZRQAM4 RD/RX EXERCISER
.IDENT /V01.6/
.ENABL AMA

```

```

000000
000000 000124'
000002 000000C
000004 000034'
000006 000010

```

```

.PSECT $XYZ$, RO
BL$LAS: .WORD T$FREE
        .WORD <<T$FREE <BL$LAS+4>>/2>
P.AAA: .WORD 1$LAST+30
        .WORD 10

```

; Plit count word

13

ZRQAM4
V01.6

RD/RX EXERCISER
LASTAD AND SETUP

11-Apr-1984 11:08:35
11-Apr-1984 11:08:22

VAX-11 B1199-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0451
Page 220
(62)

000010 172150
000012 000154
000014 000004
000016 100060
000020 000000
000022 000000
000024 150477
000026 000000
000030 000060
000032 000010
000034 172150
000036 000154
000040 000004
000042 100001
000044 000000
000046 000000
000050 001437
000052 000000
000054 000104
000056 000010
000060 172150
000062 000154
000064 000004
000066 100002
000070 000000
000072 000000
000074 001437
000076 000000
000100 000000
000102 000010
000104 172150
000106 000154
000110 000004
000112 100003
000114 000000
000116 000000
000120 001437
000122 000000
000124 000000

P.AAB: .WORD -5630
.WORD 154
.WORD 4
.WORD -77720
.WORD 0
.WORD 0
.WORD -27301
.WORD 0
P.AAC: .WORD L\$LAST+54
.WORD 10
P.AAD: .WORD -5630
.WORD 154
.WORD 4
.WORD -77777
.WORD 0
.WORD 0
.WORD 1437
.WORD 0
P.AAE: .WORD L\$LAST+100
.WORD 10
P.AAF: .WORD -5630
.WORD 154
.WORD 4
.WORD -77776
.WORD 0
.WORD 0
.WORD 1437
.WORD 0
P.AAG: .WORD 0
.WORD 10
P.AAH: .WORD -5630
.WORD 154
.WORD 4
.WORD -77775
.WORD 0
.WORD 0
.WORD 1437
.WORD 0
T\$FREE: .WORD 0

; Plit count word

; Plit count word

; Plit count word

000004
000004
000004
000010
000030
000034
000054
000060
000100
000104

L\$LAST+
T\$PTHV+
\$LAS5+
\$REM5+
\$LAS4+
\$REM4+
\$LAS3+
\$REM3+
\$LAS1+
\$REM2+
BL\$LAS+4
4
P.AAA
P.AAB
P.AAC
P.AAD
P.AAE
P.AAF
P.AAG
P.AAH

M3

ZRQAM4
V01.6

RD/RX EXERCISER
LASTAD AND SETUP

11-Apr-1964 11:08:35
11-Apr-1984 11:08:22

VAX 11 Bliss-16 V4.0-579
DISK\$USER2:[POWERS]ZRQADO.BL2;6

SEQ 0452
Page 221
(62)

000000 000207

.SBTTL \$END.LINK LASTAD AND SETUP
\$END.LINK:::
RTS PC

7791

; Routine Size: 1 word, Routine Base: \$XYZ\$ + 0126
; Maximum stack depth per invocation: 0 words

; 7818 1 end
; 7819 1
; 7820 0 eludom

PSECT SUMMARY

; Psect Name Words Attributes
; \$XYZ\$ 44 R0 , I , LCL, REL, CON

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
DISK\$USER2:[POWERS]ZRQADO.L16;3	404	7	1	21	00:00.2

COMMAND QUALIFIERS

; BLISS/PDP11 ZRQADO.BL2/LIST=ZRQADO.LS2/OBJECT=ZRQADO.OB2/SOURCE=PAGE;53

; Size: 7029 code + 417 data words
; Run Time: 03:40.4
; Elapsed Time: 05:47.2
; Lines/CPU Min: 2129
; Lexemes/CPU Min: 20098
; Memory Used: 477 pages
; Compilation Complete

ZRQAM3	RD/R....B1
ZRQAM3	RD/R....C1
ZRQAM3	RD/R....D1
ZRQAM3	RD/R....E1
ZRQAM3	RD/R....F1
ZRQAM3	RD/R....G1
ZRQAM3	RD/R....H1
ZRQAM3	RD/R....I1
ZRQAM3	RD/R....J1
ZRQAM3	RD/R....K1
ZRQAM3	RD/R....L1
ZRQAM3	RD/P....M1
ZRQAM3	RD/R....N1

ZRQAM3	RD/R....B2
ZRQAM3	RD/R....C2
ZRQAM3	RD/R....D2
ZRQAM3	RD/R....E2
ZRQAM3	RD/R....F2
ZRQAM3	RD/R....G2
ZRQAM3	RD/R....H2
ZRQAM3	RD/R....I2
ZRQAM3	RD/R....J2
ZRQAM3	RD/R....K2
ZRQAM3	RD/R....L2
ZRQAM3	RD/R....M2
ZRQAM3	RD/R....N2

ZRQAM3	RD/R....B3
ZRQAM3	RD/R....C3
ZRQAM3	RD/R....D3
ZRQAM3	RD/R....E3
ZRQAM3	RD/R....F3
ZRQAM3	RD/R....G3
ZRQAM3	RD/R....H3
ZRQAM3	RD/R....I3
ZRQAM3	RD/R....J3
ZRQAM4	RD/R....K3
ZRQAM4	RD/R....L3
ZRQAM4	RD/R....M3