

RSX1 DEC/X11 SYSTEM EXERCISER MODULE
XRSX10.P11 12-OCT-78 12:16

MACV11 3CA(1C52) 12-OCT-78 17:03 PAGE 2

1

.REM -

SEQ 0001

IDENTIFICATION

PRODUCT CODE: AC-E748I-MC
PRODUCT NAME: CXRSAIO DEC/X11 R411/RS03, RS03/LA, RS04 MODULE
DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RSA IS AN IOMDXY THAT EXERCISES RS03, RS03/LA AND RS04 DISK DRIVES ON AN RH11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE ITV.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RS03, RS03/LA AND/OR RS04'S WITH AN RH11 CONTROLLER

STORAGE:: RSA REQUIREMENTS:

- 1: DECIMAL WORDS: 1223
- 2: OCTAL WORDS: 2389
- 3: OCTAL BYTES: 4616

3. PASS DEFINITION

ONE PASS OF THE RSA MODULE CONSISTS OF 1300 CYCLES OF THE BASIC TEST SEQUENCE (WRITE -CHECK, READ, DATA-CHECK). THE TEST SEQUENCE WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS, AND DATA-CHECKS SAME.

4. EXECUTION TIME

ONE PASS OF RSA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172040, VECTOR: 204, RPI: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP AND READY

7.

MODULE OPERATION

TEST SEQUENCE:

A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
C. GET A STARTING SECTOR ADDRESS
D. GET A DRIVE ADDRESS
E. DO A WRITE -- IF ERRORS REPORT AND RETRY UP TO RETRY LIMIT
F. DO A WRITE -- CHECK -- IF ERRORS REPORT AND RETRY UP TO RETRY LIMIT
G. DO A READ -- IF ERRORS REPORT AND RETRY UP TO RETRY LIMIT
H. DO A DATA-CHECK -- IF ERRORS REPORT AND RETRY UP TO RETRY LIMIT
I. IF END OF PASS REPORT AND GO TO C
J. IF END OF DRIVES, GO TO C ELSE GO TO D

8.

OPERATION OPTIONS (SOFTWARE SWITCH)

SRI BIT0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, THE FUNCTION
IS ABORTED AND TESTING CONTINUES
SRI PIT0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, THE ERROR IS
CONSIDERED FATAL AND THE DRIVE DROPPED
SRI PIT2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND COUNT THEM
SRI PIT2 SET(1):
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT

RSX1 DEC/X11 SYSTEM EXERCISER MODULE
XRSX10.P11 12-OCT-78 12:16

MACV11 3CA(1052) 12-OCT-78 17:03 PAGE 5

SEQ 0004

9. NON-STANDARD PRINTOUTS

- A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT.
- B. ERROR MESSAGES DUMP THE CONTENTS OF THE 12 RH11/RS REGISTERS IN THE FOLLOWING ORDER:

RHCS1 RHCS2 RHCW RHRA RSDA RSDS RSER PSAS
RSLA RHDR RSMR RSDT RHRAE RHCS3

```
000000- IGWDDX <RSAL > 172040,204,5,0,0,1300,,61,BUFIN,256,,1024.  
000000- MODULE 150000,RSAL 172040,204,5,0,0,1300,,61,BUFIN,256,,1024.  
; .TITLE RSAL DEC/X11 SYSTEM EXERCISER MODULE  
; DDSCOM VERSION 6 23-NOV-78  
*****LIST BIN*****  
000000- 051522 C445C1 040 BEGIN: 150000 ;STATUS WORD.  
000000- 000000 ;MODNAM: .ASCII /RSAL / ;MODULE NAME ;USED TO KEEP TRACK OF WBUFF USAGE  
000000- 172040 ADDR: 172040+0 ;1ST DEVICE ADDR.  
000010- 000204 VECTOR: 20440 ;1ST DEVICE VECTOR.  
000015- 240 BR1: .RVTE PRTV5+0 ;1ST RR LEVEL.  
000015- 000 BR2: .RVTE PRTV0+0 ;2ND RR LEVEL.  
000014- 000001 INT01: 0+1 ;DEVICE INDICATOR 1.  
000016- 000000 SR1: OPEN ;SWITCH REGISTER 1  
000020- 000000 SR2: OPEN ;SWITCH REGISTER 2  
000022- 000000 SR3: OPEN ;SWITCH REGISTER 3  
000024- 000000 SR4: OPEN ;SWITCH REGISTER 4  
*****LIST BIN*****  
000026- 150000 STAT: 150000 ;STATUS WORD.  
000030- 000224 INT: START ;MODULE START ADDR.  
000030- 000224 SPOINT: MODSP ;MODULE STACK POINTER.  
000034- 000000 PASCNT: 0 ;PASS COUNTER.  
000036- 002424 ICONF: 1300. ;# OF ITERATIONS PER PASS=1300.  
000040- 000000 TCONF: 0 ;LOC TO COUNT ITERATIONS  
000042- 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS  
000044- 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS  
000046- 000000 SDFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS  
000048- 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS  
000050- 000000 SVSCNT: 0 ;# OF SVS ERRORS ACCUMULATED  
000054- 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED  
000056- 000000 CONFIG: 0 ;RESERVED FOR MONITOR USE  
000060- 000000 RES0: 0 ;RESERVED FOR MONITOR USE  
000062- 000000 SVR0: OPEN ;LOC TO SAVE R0.  
000064- 000000 SVR1: OPEN ;LOC TO SAVE R1.  
000066- 000000 SVR2: OPEN ;LOC TO SAVE R2.  
000070- 000000 SVR3: OPEN ;LOC TO SAVE R3.  
000072- 000000 SVR4: OPEN ;LOC TO SAVE R4.  
000074- 000000 SVR5: OPEN ;LOC TO SAVE R5.  
000076- 000000 SVR6: OPEN ;LOC TO SAVE R6.  
000100- 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.  
000102- 000000 SBADR: OPEN ;ADDR OF GOOD DATA, OR  
000104- 000000 ASB: OPEN ;CONTENTS OF CSR.  
000106- 000000 WSBADR: OPEN ;ADDR OF BAD DATA, OR  
000108- 000000 ASTAT: OPEN ;STATUS REG CONTENTS.  
000110- 000000 ERRTP: OPEN ;TYPE OF ERROR.  
000112- 000000 ASB: OPEN ;EXPECTED DATA.  
000114- 000000 AWAS: OPEN ;ACTUAL DATA.  
000116- 000000 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS  
000118- 000000 WDMTO: OPEN ;WORDS TO MEMORY PER ITERATION  
000120- 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION  
000122- 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION  
IDNUM: 61 ;MODULE IDENTIFICATION NUMBER=61
```

```
000124- 000000 RBUFVA: BUFIN ;READ BUFFER VIRTUAL ADDRESS  
000126- 000000 RBFPA: OPEN ;READ BUFFER PHYSICAL ADDRESS  
000130- 000000 RBFPSZ: 256 ;READ BUFFER EA BITS  
000132- 000400 WRBFPA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS  
000134- 000000 WBFPSZ: 256 ;WRITE BUFFER EA BITS  
000140- 002000 WBUFQ: 1024 ;WRITE BUFFER SIZE REQUESTED  
000142- 000000 WBUFQSZ: OPEN ;WRITE BUFFER SIZE AVAILABLE  
000144- 000000 CDWCT: OPEN ;CDATA/DATCK ERROR COUNT  
000146- 000000 CDWCT: OPEN ;CDATA/DATCK WORD COUNT  
000150- 000000 FREE: OPEN ;RESERVED FOR FUTURE USE  
; .RFPT SPSIZ ;MODULE STACK STARTS HERE.  
; .NLST  
; .WORD 0  
; .LIST  
; .ENDR  
000252- MODSP:  
*****LIST BIN*****
```

```

222 000252 212767 002000 177636 START: MOV #1024, WDFR ;1024 WORDS FROM MEM/ITERATION
223 000252 212767 003400 177626 MOV #256, WDTO ;256 WORDS TO MEM/ITERATION
224 000256 212767 000003 177624 MOV #3, INTR ;3 INTERRUPTS/ITERATION
225 000274 005067 002670 CLR CNT ;ZERO END OF PASS TESTER
226 000300 105067 003311 FLAG ;CLEAR FLAGS
227 000310 016767 177500 CLR DLTCNT ;CLEAR DATA LATE ERROR COUNTER
228 000316 216767 002668 MOV DVID1, DVICE ;GET DRIVE INDICATOR
229 000334 216767 002666 MOV DVICE, DRIVE ;ALSO SAVE IT IN DRIVE
230 000334 216767 000015 000041 CMQB #15, @41 ;IF THIS DEVICE IS NOT THE LOAD MEDIA THEN
231 000334 012702 000000 BNE #0, R2 ;GO TO 35
232 000340 113700 000040 MOVR #40, R0 ;LOAD R2 WITH DRIVE NUMBER
233 000344 012702 000001 MOV #1, R1 ;GET DRIVE NUMBER
234 000350 105700 002648 1S: STR R0 ;DRIVE MASK: INITIALIZE TO DRIVE 0
235 000350 001404 BEQ #0, R1 ;IF DRIVE FOUND THEN
236 000354 006301 ASL R1 ;ELSE SHIFT MASK TO NEXT DRIVE
237 000356 105300 DECB R0 ;DOWNCOUNT DRIVE #
238 000360 005202 INC R2 ;UPDATE DRIVE NUMBER
239 000362 000775 BR #1, DVICE ;CHECK AGAIN
240 000364 130167 002620 2S: BITR #1, DVICE ;IF DRIVE NOT SELECTED TO BE TESTED THEN
241 000370 001404 BEQ #0, R1 ;GO TO 35
242 000370 001404 MOV R2, DRVVE ;ELSE SET UP TO DROP DRIVE
243 000375 004767 001712 JSR PC, DROP ;AND GO DROP IT
244 000402 012767 177770 002554 MOV #8, BLK1 ;INITIALIZE BLOCK COUNTER
245 000410 004767 177777 002576 JSR #1, DRVVE ;INITIALIZE DEVICE COUNTER
246 000420 004767 003342 JSR PC, SETUP ;GENERATE REGISTER ADDRESSES
247 000425 005767 002556 JSR PC, REZET ;INITIALIZE RH REGS AND DRIVES
248 000434 001477 BFG ;DROP THE MODULE ?
249 000434 001477 BFG ;YES
250 000434 001477 BFG ;YES
251 000434 001477 BFG ;YES
252 000434 001477 BFG ;YES
253 000434 104415 000000 000124* RESTR: GETPAS, BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
254 000442 016767 177464 002550 MOV RBUFVSZ, WCNT2 ;SAVE READ BUFFER SIZE
255 000450 005467 NEG WCNT2 ;GET THE 2'S COMPLEMENT
256 000454 004767 001524* STRT: JSR PC, BLOCK ;GET NEXT BLOCK NUMBER, DISK ADDRESS
257 000460 104414 000000 000000 MOV WBUFVSZ, WCNT1 ;GET WRITE BUFFER INFORMATION
258 000464 016767 177452 002524 MOV WBUFVSZ, WCNT1 ;SAVE WRITE BUFFER SIZE
259 000472 005467 NEG WCNT1 ;GET THE 2'S COMPLEMENT
260 000472 005467 NEG WCNT1 ;GET THE 2'S COMPLEMENT
261 000472 005467 NEG WCNT1 ;GET THE 2'S COMPLEMENT
262 000472 005467 NEG WCNT1 ;GET THE 2'S COMPLEMENT
263 000472 005467 NEG WCNT1 ;GET THE 2'S COMPLEMENT
264 000472 005467 NEG WCNT1 ;GET THE 2'S COMPLEMENT
265 000510 132767 000010 004077 NEXT: JSR PC, DRVADR ;GET A DRIVE ADDRESS
266 000510 132767 000010 004077 JSR DVICE ;ANY DRIVES LEFT ?
267 000510 132767 000010 004077 BEQ #0, R1 ;NO, GO DROP THE MODULE
268 000524 000473 BITR #BIT3, FLAG ;ALL DRIVES DONE ?
269 000526 004767 STR #1, R5 ;YES, GO GET ANOTHER BLOCK
270 000532 000761 JSR R5, PEADY ;IS DRIVE READY ?
271 000532 000761 BR #1, S ;YES, CONTINUE
272 000540 105067 004052 JSR PC, NOTRDY ;NO, GO WAIT FOR IT
273 000544 004567 BR NEXT ;CONTINUE
274 000544 004567 CLR TRV3 ;ZERO RETRY COUNTERS
275 000544 004567 JSP R5, WRITE ;WRITE SOME DATA
276 000544 004567 BR RETRY1 ;IF ERRORS, TRY IT AGAIN
277 000562 142767 000004 004025 BITR #BIT2, FLAG ;DID THE DISK OVERFLOW ?
278 000562 142767 000004 004025 BEQ #0, R1 ;NO, CONTINUE
279 000562 142767 000004 004025 BICR #BIT2, FLAG ;YES, CLEAR THE OVERFLOW FLAG

```

```

278 000570 012767 177770 002366 MOV #8, PLK1 ;RESET THE BLOCK NUMBER
279 000570 004567 STR #1, R5 ;START OVER AT BEGINNING OF DISK
280 000600 004567 000226 GOA: JSP R5, WITCK ;WRITE-CHECK THE DATA
281 000604 004567 000272 GOR: JSP R5, READ ;IF ERRORS, TRY AGAIN
282 000606 004567 000272 JSP R5, READ ;READ THE DATA WRITTEN
283 000612 104415 000000 000126* JSP R5, TRV3 ;IF ERRORS, TRY AGAIN
284 000622 000624 000000 000126* CDATAS, BEGIN, RBUFA ;REQUEST FOR MONITOR TO CHECK DATA
285 000622 104413 000000* *2 ;IF ERROR, CONTINUE
286 000624 104413 000000* ENDIRS, BEGIN ;SIGNAL END OF ITERATION
287 000630 000722 BR NEXT ;MONITOR SHALL TEST END OF PASS
288 000630 000722 BR NEXT ;NO, CONTINUE

```

```

289
290
291 000632- 000000- FINI:
292 000632- 104410 000000- ; ENDS, BEGIN ; DROP THE MODULE
293 ;
294
295
296
297 000636- 105267 003750 RETRY1: INCR TRV1 ; COUNT THE RETRYS
298 000642- 122767 000003 003742 CMPR #3, TRV1 ; LIMIT EXCEEDED ?
299 000659- 001335 BNE GO ; NO, GO TRY IT AGAIN
300 000659- 104403 000000- 004544- MSGNS, BEGIN, EXCED1 ; ASCII MESSAGE CALL WITH COMMON HEADER
301 000660- 000424 BP NEXTA ; GO ON TO NEXT DRIVE
302 ;
303
304
305 000662- 105267 003725 RETRY2: INCR TRV2 ; COUNT RETRYS
306 000666- 122767 000003 003717 CMPR #3, TRV2 ; LIMIT EXCEEDED ?
307 000674- 001341 BNE GOA ; NO, TRY AGAIN
308 000676- 104403 000000- 004552- MSGNS, BEGIN, EXCED2 ; ASCII MESSAGE CALL WITH COMMON HEADER
309 000704- 000412 BP NEXTA ; GO ON TO NEXT DRIVE
310 ;
311
312
313 000706- 105267 003702 RETRY3: INCR TRV3 ; COUNT RETRYS
314 000712- 122767 000003 003674 CMPR #3, TRV3 ; LIMIT EXCEEDED ?
315 000729- 001343 BNE GOB ; NO, GO TRY AGAIN
316 000729- 104403 000000- 004560- MSGNS, BEGIN, EXCED3 ; ASCII MESSAGE CALL WITH COMMON HEADER
317 000730- 000400 BP NEXTA ; GO ON TO NEXT DRIVE
318 ;
319
320
321 000732- 032767 000001 177056 NEXTA: BIT #R10, SR1 ; DROP THE DRIVE
322 000740- 001405 BEO IS ; NO, SKIP TO NEXT DRIVE
323 000744- 004767 001346 JSP PC, DROP ; YES, DROP OFFENDING DRIVE
324 000744- 000167 177516 MSGNS, BEGIN, DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
325 1S: JMP NEXTA ; GO ON TO NEXT DRIVE
326 ;
327
328
329
330 ;
331 MACRO LINEUP EABITS ; LINE UP EA BITS FOR RHCSI
332 LINEUP EABITS ; LINE UP EA BITS FOR RHCSI
333 .MLIST
334 MOV EABITS, RO ; GET EXTENDED MEMORY BITS
335 ASL RO ; SHIFT 4 PLACES TO THE LEFT
336 ASL RO ; TO LINE UP WITH RHCSI
337 ASL RO ;
338 MOV RO, XMEM ; SAVE THE SHIFTED BITS
339 .LIST
340 .ENDM LINEUP

```

```

340
341 000760- 012767 000161 002206 WRITE: MOV #161, FUNC ; LOAD WRITE FUNCTION
342 000766- 016777 002224 003232 WCNT1, @RHWC ; LOAD WORD COUNT
343 000774- 016777 177134 003228 WPUFA, @RHBA ; LOAD BUFFER ADDRESS
344 ; LINEUP EABITS FOR RHCSI
345 001022- 016777 002136 003202 MOV R1, @RSDA ; LOAD DISK ADDRESS
346 001030- 006467 GOGO ; CONTINUE
347 001032- 012767 000151 002134 WRITCK: MOV #151, FUNC ; LOAD WRITE-CHECK FUNCTION
348 001040- 016777 002152 003160 WCNT2, @RHWC ; LOAD WORD COUNT
349 001046- 016777 177062 003154 WPUFA, @RHBA ; LOAD BUFFER ADDRESS
350 ; LINEUP EABITS FOR RHCSI
351 001074- 016777 002064 003130 MOV R1, @RSDA ; LOAD DISK ADDRESS
352 001102- 000442 BP GOGO ; CONTINUE
353 001104- 012767 000171 002062 READ: MOV #171, FUNC ; LOAD READ FUNCTION
354 001112- 016777 002102 003106 WCNT2, @RHWC ; LOAD WORD COUNT
355 001120- 016777 177002 003102 WPUFA, @RHBA ; LOAD BUFFER ADDRESS
356 ; LINEUP EABITS FOR RHCSI
357 001146- 016777 002014 003056 MOV R1, @RSDA ; LOAD DISK ADDRESS
358 001154- 000415 BP GOGO ; CONTINUE
359 ;
360
361 001156- 016777 002032 003040 CLFAR: MOV DPVVE, @RHCS2 ; LOAD UNIT ADDRESS
362 001164- 012777 000811 003030 #11, @RHCSI ; ISSUE A DRIVE CLEAR
363 001172- 012777 000377 003040 #377, @RBSA ; CLEAR ALL DISK ATTENTION SUMMARIES
364 001206- 000205 #BIT14, @RHCSI ; CLEAR ANY CONTROLLER ERRORS
365 ; RETURN
366 001210- 016777 002000 003006 GOGO: MOV DPVVE, @RHCS2 ; LOAD UNIT SELECT
367 001216- 012777 001344- 176564 #NTPUBT, @VECTOR ; SET INTERRUPT ENTRY POINTER
368 001224- 001494 #ADDR22, RES1 ; 22 BIT ADDRESSING?
369 001234- 017767 002770 001734 BIT IS ; NO
370 001242- 006267 001732 MOV @RHBA, PA18 ; GET LOWER 18 BITS
371 001246- 006267 001726 ASK XMEM ; SHIFT AE BITS INTO LOC 4+5
372 001252- 006267 001722 ASP XMEM
373 001256- 006267 001716 ASR XMEM
374 001262- 104416 000000- 003176- MAP22S, BEGIN, PA18 ; GET 22-BIT ADDR FROM 18-BIT ADDR
375 001270- 016777 001796 002732 MOV PA22, @RHBA ; LOAD BA REG
376 001276- 016777 001702 002746 MOV EA22, @RSBAE ; LOAD BA REG
377 001304- 002767 000034 001672 PIC EA22, EA22 ; CLEAR UNWANTED BITS
378 001312- 000367 001666 SWAR EA22 ; LOCATE BITS 16, 17
379 001316- 016767 001662 001642 MOV XMEM, FUNC ; TO LOAD INTO CS1
380 001324- 056767 001660 001642 BIS XMEM, FUNC ; LOAD EXTENDED MEMORY BITS
381 001332- 016777 001636 002662 1S: MOV FUNC, @RHCSI ; EXECUTE THE FUNCTION
382 001340- 104400 000000- EXITS, BEGIN ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
383 ;
384
385 001344- 000004 000000- 001352- NTRUPT: ;
386 ;
387 ;
388 001352- 004567 000272 1S: JSP RE, ERRORS ; GO CHECK FOR ERRORS
389 001356- 000204 RTS R5 ; NO ERRORS, SKIP RETRY
390 001360- 005725 TST (R5)+ ; ERRORS DETECTED, RETURN
391 001362- 000205 RTS R5 ; RETURN OK

```

391 001366 016700 001574 ROOM: MOV RLK1,R0 ;GET THE CURRENT BLOCK NUMBER
392 001370 012701 010000 MOV R4006,R1 ;LOAD THE MAX. NUMBER OF BLOCKS
393 001374 012767 000200 001570 MOV #128,,BSIZ ;SET BLOCK SIZE TO HI DENSITY
394 001402 005002 CLR R2 ;ZERO REG. 2
395 001412 016703 000001 003203 BITR #R10,FLAG ;CLEAR 32K INDICATOR
396 001412 016703 176524 MOV #R0FS,R3 ;GET THE TRANSFER SIZE
397 001416 160001 SUP R0,R1 ;GET # OF BLOCKS LEFT ON DISK
398 001420 027777 000004 002622 CMP #4,RSDT ;IS THIS A RS03/LA ?
399 001430 006267 BNE 7S ;NO
400 001434 006267 001536 ASR BSIZ ;SET SECTOR SIZE
401 001440 027701 001532 ASR BSIZ ;TO 32 WD/SECTOR
402 001444 027701 004000 CMP #2048,,P1 ;PLENTY OF ROOM LEFT?
403 001446 027701 002000 BLE 5S ;YES ERROR
404 001452 003343 RGT #1024,,R1 ;MORE THAN 32K LEFT?
405 001454 152767 000001 003133 BITR #R10,FLAG ;NO CONTINUE
406 001454 152767 002000 SUB #1024,,R1 ;YES SET THE INDICATOR
407 001462 006435 BNE 7S ;SUB 32K OF BLOCKS
408 001470 032777 000002 002552 7S: BIT #R11,RSDT ;HI DENSITY DRIVE?
409 001476 001015 BNE 1S ;YES, CONTINUE
410 001480 006267 ASR BSIZ ;NO, SET BLOCK SIZE TO 64.
411 001504 027701 002000 CMP #1024,,R1 ;LOW DENSITY. PLENTY OF ROOM LEFT?
412 001510 003451 BLE 5S ;YES, ERROR
413 001512 027701 001000 CMP #512,,R1 ;MORE THAN 32K LEFT?
414 001512 027701 001000 BNE 7S ;NO, CONTINUE
415 001520 152767 000001 003067 BITR #R10,FLAG ;YES, SET THE INDICATOR
416 001526 162701 001000 SUB #512,,R1 ;SUBTRACT 32K WORTH OF BLOCKS
417 001534 006435 BR 2S ;GO GET TOTAL NUMBER OF WORDS LEFT
418 001540 003435 CMP #12,,R1 ;HIGH DENSITY. PLENTY OF ROOM LEFT?
419 001542 027701 000400 RLF 5S ;YES, ERROR
420 001542 027701 000400 CMP #256,,R1 ;MORE THAN 32K LEFT?
421 001542 027701 000400 RGT 2S ;NO, CONTINUE
422 001550 152767 000001 003037 BITR #R10,FLAG ;YES, SET THE INDICATOR
423 001556 162701 000400 SUB #256,,R1 ;SUBTRACT 32K WORTH OF BLOCKS
424 001562 005701 R1 ;ANY BLOCKS LEFT ON DISK?
425 001562 005701 TST R1 ;NO, RETURN OK
426 001566 006762 ADD #17,P2 ;GET TOTAL # OF WORDS LEFT
427 001572 005301 DEC R1 ;ARE BLOCKS ADDED IN?
428 001574 003372 RGT 2S ;NO, KEEP ADDING
429 001576 100000 TST R2 ;REQUEST LARGER THAN 32K?
430 001576 100000 BNE 7S ;NO, GO CHECK THAT CONDITION
431 001602 042703 100000 BIT #R15,R3 ;YES, GET RID OF 32K
432 001606 132767 000001 003001 BITR #R10,FLAG ;MORE THAN 32K LEFT?
433 001612 006435 BR 0 ;NO, RETURN OK
434 001616 001404 AND 6S ;GO COMPARE
435 001620 132767 000001 002767 3S: BITR #R10,FLAG ;MORE THAN 32K LEFT?
436 001626 001002 BNE 5S ;YES, PLENTY OF ROOM, ERROR
437 001632 005403 CMP R2,P3 ;ENOUGH ROOM FOR THE TRANSFER?
438 001634 005403 RGT 6S ;RETURN OK
439 001634 005725 TST (R5)+ ;MUST BE REAL ERROR
440 001636 000205 RTS R5 ;RETURN INDICATING THE ERROR
441 001640 002767 000004 002747 6S: BITR #R12,FLAG ;SET OVERFLOW FLAG
442 001646 002767 RTS R5 ;RETURN OK

444 001650 005777 002346 ERRORS: TST @RHCS1 ; ATTENTION OR ERROR ?
445 001654 100124 001000 002352 R10 ; NO, GO ON TO NEXT FUNCTION
446 001664 0051403 001000 002352 BIT #R19,RSER ; ADDRESS OVERFLOW ?
447 001666 004567 177472 JSR #R,RGOM ; NO, CONTINUE
448 001672 0060513 000636 1S: JSR #R,RSUR1 ; YES, IS IT A REAL ERROR ?
449 001674 005777 002320 TST #RCS2 ; NO, CONTINUE
450 001674 160012 001244 R1 ; LOAD ERROR INFORMATION
451 001674 160012 000004 176076 R1 ; IS THIS A DATA LATE ERROR?
452 001676 005267 001244 INC DLTCNT ; ADD 1 TO DATA LATE ERROR COUNTER
453 001676 005267 000004 176076 BIT #R12,SR1 ; TYPE ERROR?
454 001676 001375 BNE 8S ; NO
455 001676 104403 000000 004576 MSGNS,REGIN,DLTERR ; ASCII MESSAGE CALL WITH COMMON HEADER
456 001676 004444 BR 5S ; CONT
457 001676 032777 040000 002262 11S: BIT #R14,@RHCS1 ; TRANSFER ERROR ?
458 001676 032777 020000 002252 BNE 2C ; YES
459 001676 001025 000400 002244 BIT #R13,@RHCS1 ; MASSBUS CONTROL PARITY ERROR ?
460 001676 001025 000400 002244 BNE 3S ; YES
461 001676 001025 000400 002244 BIT #R18,@RHCS2 ; MASSBUS DATA PARITY ERROR ?
462 001676 032777 040000 002244 BNE 4S ; YES
463 001676 032777 040000 002244 BIT #R14,RSDS ; ANY DRIVE ERRORS ?
464 001676 001025 002242 BNE 5S ; YES
465 001676 001025 002242 TST @RSAS ; ANY ATTENTIONS ACTIVE ?
466 001676 001025 176102 RNE 5S ; YES, CONTINUE
467 001676 005067 176102 CLR #R7,RSUR1 ; UNKNOWN ERROR
468 001676 005067 000000 004222 ***** UNKNOWN ERROR *****
469 001676 104405 000000 004222 HDRS,REGIN,TABLE ; SPECIAL CONDITION SET BUT NO REASON FOUND
470 001676 001025 001025 PR 8S ; RETURN
471 001676 001025 001025 2S: MSGNS,REGIN,TRERR ; ASCII MESSAGE CALL WITH COMMON HEADER
472 001676 001025 001025 000000 004530 BR 5S ; GO DUMP REGISTERS
473 001676 001025 001025 3S: MSGNS,REGIN,MCPERR ; ASCII MESSAGE CALL WITH COMMON HEADER
474 001676 001025 001025 000000 004534 BR 5S ; GO DUMP REGISTERS
475 001676 001025 001025 4S: MSGNS,REGIN,MCPERR ; ASCII MESSAGE CALL WITH COMMON HEADER
476 001676 001025 001025 000000 004546 TST @RSAS ; ANY ATTENTIONS ACTIVE ?
477 001676 001025 001025 5S: BNE 6S ; NO, CONTINUE
478 001676 001025 001025 6S: MOV #R0,WHO ; YES, FIND OUT WHICH DRIVE IT IS
479 001676 001025 001025 6S: MOV #R0,WHO ; SAVE ADDRESS OF DATA BUFFER
480 001676 001025 001025 6S: TSTR @PHCS2 ; CAN DATA BUFFER BE READ ?
481 001676 001025 001025 6S: BNE 7S ; YES, CONTINUE
482 001676 001025 001025 6S: MOV #R0,PHDR ; NO, LOAD ADDRESS OF ZERO
483 001676 001025 001025 6S: MOV #R0,PHDR ; ERROR DURING DATA XFER
484 001676 001025 001025 7S: ***** UNKNOWN ERROR *****
485 001676 001025 001025 7S: HDRS,REGIN,TABLE ; DUMP R11 AND RS REGISTERS
486 001676 001025 001025 7S: ***** UNKNOWN ERROR *****
487 001676 001025 001025 7S: MOV #R0,PHDR ; RESTORE DATA BUFFER ADDRESS
488 001676 001025 001025 7S: JSP #R5,CLEAR ; GO CLEAR OUT ERRORS
489 001676 001025 001025 7S: JSP #R5,CLEAR ; ERRORS DETECTED, RETURN
490 001676 001025 001025 7S: TST (R5)+ ; GO CLEAR OUT ANY ERRORS
491 001676 001025 001025 7S: RTS R5 ; NO ERRORS, SKIP RETRY
492 001676 001025 001025 7S: ; RETURN OK


```

499 002132 317701 002122 WHO: MOV #RHSAS,R1 ; GET THE ATTENTION SUMMARY
500 002136 317703 002063 MOV #RHCS2,R4 ; SAVE THE STATUS REGISTER
501 002144 005603 000001 MOV #RIT0,R2 ; SET POINTER TO DRIVE
502 002150 030201 000000 CLP R3 ; ZERO THE DRIVE COUNTER
503 002159 030201 1S: BIT R2,R1 ; IS THIS ATTENTION BIT SET ?
504 002154 004203 2S: BNE R3 ; YES, CONTINUE
505 002156 006302 3S: INC R3 ; NO, INCREMENT THE DRIVE COUNTER
506 002160 032402 000400 ASL R3 ; SET POINTER TO NEXT DRIVE
507 002160 032402 4S: BIT #RIT8,R2 ; ALL DONE ?
508 002166 004207 5S: BFC 1S ; NO, GO AGAIN
509 002166 004207 6S: RTS PC ; SOMEBODY LIED -- NO ATTENTIONS SET
510 002170 042704 000007 2S: BIC R3,R4 ; CLEAR OUT OLD UNIT NUMBER
511 002174 010477 002022 RTS R4,R4 ; LOAD THE NEW UNIT NUMBER
512 002202 000207 ; RTOS #RHCS2 ; RESTORE THE STATUS REGISTER
513 ; ; RETURN
514 002204 062767 000010 BLOCK: ADD #0,BLK1 ; STEP TO NEXT BLOCK
515 002212 022767 000777 CMP #4095,BLK1 ; ALL BLOCKS DONE ?
516 002220 100000 000736 RPL 1S ; NO, CONTINUE
517 002226 016767 000732 1S: CLR BLK1 ; YES, START OVER
518 002234 000207 2S: MOV BLK1,PLY2 ; SAME STARTING BLOCK FOR READ
519 ; ; RETURN
520 002236 005267 000752 DRVADP: INC DRVVE ; COUNT A DRIVE
521 002242 142767 000100 BICP #RIT3,FLAG ; CLEAR END OF DRIVES FLAG
522 002250 022767 000010 CMP #0,DRVVE ; ALL DRIVES CHECKED ?
523 002256 001404 000726 REQ 1S ; YES, GO FLAG END OF DRIVES
524 002260 005267 2S: INC DRVVE ; NO, IS NEXT DRIVE CHOSEN ?
525 002264 103364 3S: BIC DPVADR ; NO, GO TRY ANOTHER DRIVE
526 002266 000411 4S: RP ; NO, GO TRY ANOTHER DRIVE
527 002270 000411 1S: BISR #RIT3,FLAG ; SET END OF DRIVES FLAG
528 002274 012767 000710 MOV #0,DRVVE ; RESET DRIVE COUNTER
529 002302 016767 000700 2S: MOV DVICE,DRIVE ; RESTORE CHOSEN DRIVES
530 002312 000207 ; ; RETURN
531 002314 012701 000001 DROP: MOV #1,R1 ; INITIALIZE DROP PICKER
532 002320 016700 000070 MOV DRVVE,PC ; GET THE DRIVE NUMBER
533 002326 005301 1S: BEQ 2S ; IF DRIVE 0 GO DROP IT
534 002330 005301 2S: ASL R1 ; POINT TO NEXT DRIVE
535 002332 001375 3S: BNE R1 ; IS THIS THE ONE ?
536 002334 040167 000550 RIC R1,DVICE ; NO, LOOK AGAIN
537 ; ; DROP THE DRIVE
538 ;*****
539 ;CONVERT DRIVE TO ASCII AND
540 ;STORE AT ADRI
541 CTOAS,RFGIN,DRVVE,ADRI
542 ;*****
543 ;*****
544 ;*****
545 ;*****
546 ;*****
547 ;*****
548 ;*****
549 ;*****
550 ;*****
551 ;*****
552 ;*****
553 ;*****

```

```

554 002352 312767 177777 000634 NOTRDY: MOV #0,DRVVE ; START WITH FIRST DRIVE
555 002360 034767 000624 MOV DVICE,DRIVE ; START DRIVE SELECT
556 002370 034767 000624 1S: JSP PC,DPVADR ; GET A DRIVE ADDRESS
557 002372 132767 000610 002215 BIT #RIT3,FLAG ; ALL DRIVES CHECKED ?
558 002400 001143 000010 000552 MOV #0,CLOCK ; YES, RETURN
559 002402 012767 000010 000552 ; YES, RETURN
560 002410 312767 077777 000542 4S: MOV #77777,CLK ; TO GIVE DRIVES TIME TO GET UP TO SPEED
561 002416 004567 176534 2S: JSP PC,CLEAR ; SET THE TIMER
562 002422 004567 000126 JSR #R1,READY ; CLEAR OUT ANY ERRORS
563 002424 000757 000000 BR 1S ; IS THIS DRIVE READY ?
564 002430 104407 000000 BNE R1 ; YES, CONTINUE
565 002434 104407 000000 BREAKS,REGIN ; TEMPORARY RETURN TO MONITOR
566 002440 005367 000514 BREAKS,REGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
567 002444 005367 000514 DEC CLK ; WAIT SOME MORE ?
568 002446 005367 000514 BNE 2S ; YES
569 002452 001356 000000 DEC CLOCK ; TIME?
570 002454 004767 000056 JSR PC,ERSUP1 ; NO CONTINUE WAITING
571 002460 012767 000006 MOV #0,ERRTYP ; LOAD ERROR INFORMATION
572 002466 104406 000000 ;*****
573 ;*****
574 002466 104406 000000 004222 HRDEFS,REGIN,TABLE ;*****
575 ;*****
576 002474 004767 176514 JSR PC,DRP ; NO, DROP THE DRIVE
577 002500 104403 000000 004566 VSCNS,REGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
578 002506 000727 3S: BR 1S ; CHECK REST OF DRIVES
579 002510 000207 ; ; RETURN
580 ;*****
581 ;*****
582 002512 014167 175370 ERSUR2: MOV -(R1),ASB ; LOAD THE DATA
583 002516 014167 175362 MOV R1,RSADR ; LOAD ADDRESS OF DATA WRITTEN
584 002520 016267 175362 MOV -(R2),AWAS ; LOAD THE DATA
585 002526 016267 175352 MOV R2,ASADR ; LOAD ADDRESS OF DATA READ
586 002530 005721 587 002534 005722 TST (R1)+ ; RESET REG-1
588 002536 016767 175334 ERSUR1: MOV #RHS1,CSRA ; LOAD ADDR OF CURRENT CSR
589 002544 017767 01452 MOV #RHCS1,ACSR ; LOAD CONTENTS OF CURRENT CSR
590 002552 000207 ; ; RETURN
591 ;*****
592 ;*****
593 002554 017777 000434 RFADY: MOV DRVVE,RHCS2 ; LOAD UNIT ADDRESS
594 002560 017777 000444 TSTR #0,RSNS,PC ; SAVE STATUS IN REG. 0
595 002566 105700 596 002570 105700 000400 RPL 1S ; DRIVE READY ?
597 002572 105700 000400 BIT #RIT8,R0 ; NO
598 002574 001404 000400 BIC #RIT8,R0 ; DRIVE PRESENT ?
599 002576 032700 010000 BFC 1S ; NO
600 002578 001404 010000 BFC 1S ; MEDIUM ON-LINE ?
601 002580 001404 010000 BFC 1S ; NO
602 002582 001404 010000 1S: RTS R5 ; RETURN, READY
603 002584 000207 000207 RTS R5 ; SKIP INSTRUCTION FOLLOWING CALL
604 ; ; RETURN, NOT READY

```

```

605
606
607
608
609
610
611 002614* 016700 175166
612 002620* 010067 001378
613 002624* 005720
614 002626* 010067 001374
615 002632* 005720
616 002634* 010067 001370
617 002640* 005720
618 002642* 010067 001364
619 002646* 005720
620 002650* 010067 001350
621 002654* 005720
622 002656* 010067 001352
623 002662* 005720
624 002664* 010067 001346
625 002670* 005720
626 002672* 010067 001342
627 002676* 005720
628 002760* 010067 001336
629 002704* 005720
630 002706* 010067 001332
631 002712* 005720
632 002714* 010067 001326
633 002720* 005720
634 002722* 010067 001322
635 002726* 010067 001320 175122
636 002734* 001406
637 002736* 005720
638 002740* 010067 001306
639 002744* 005720
640 002746* 010067 001302
641
642
643 002752* 016700 175032
644 002756* 012700 000454
645 002762* 116710 175024
646
647 002766* 000207

```

```

SETUP: MOV ADDR,RO ; GET DEVICE ADDRESS
        MOV RO,RHCS1 ; GENERATE REGISTER ADDRESSES
        TST (R0)+
        MOV RO,RHWC
        TST (R0)+
        MOV RO,RHRA
        TST (R0)+
        MOV RO,PSDA
        TST (R0)+
        MOV RO,RHCS2
        TST (R0)+
        MOV RO,PSDS
        TST (R0)+
        MOV RO,PSEF
        TST (R0)+
        MOV RO,PSAS
        TST (R0)+
        MOV RO,SLA
        TST (R0)+
        MOV RO,PHDB
        TST (R0)+
        MOV RO,PSMR
        TST (R0)+
        MOV RO,PSDT ; 22 BIT ADDRESSING?
        BIT #ADDR22,RES1 ; NO
        ORG #1
        TST (R0)+
        MOV RO,PSPAE
        TST (R0)+
        MOV RO,RSCS3
        RTS PC ; RETURN

```

```

15: MOV VECTOR,RO ; GET VECTOR ADDRESS
     MOV #STP,(R0)+ ; SET POINTER JUST IN CASE
     MOVR #P1,(R0) ; SET PRIORITY

```

```

648
649
650
651 002770* 012777 000040 001226
652 002772* 012767 000040 000154
653 002774* 005777 001212
654 003004* 105777
655 003010* 105420
656 003012* 104407 000000
657 003016* 104407 000000
658 003022* 005367 000132
659 003024* 001366
660 003030* 005077 000154
661 003034* 012787 000003 175044
662
663 003042* 104405 000000 004222
664
665 003050* 000207
666
667 003052* 004767 177274
668 003054* 005767 000126
669 003062* 001434
670 003064* 004767 177146
671 003070* 032767 000010 001517
672 003074* 001026
673 003100* 016777 000110 001116
674 003104* 032777 174000 001134
675 003114* 001223
676 003116* 004567 176034
677 003122* 000760
678 003124* 004767 177164
679 003130* 012767 000006 174750
680
681 003136* 104405 000000 004222
682
683 003144* 104403 000000 004566
684 003152* 000744
685 003154* 000207
686
687
688
689

```

```

REZET: MOV #BITS,ORHCS2 ; ISSUE AN RH11 INIT
        MOV #7777,CLK ; SET THE TIMER
        TSTP #RHCS1 ; CONTROLLER READY ?
        RMT ZC ; YES, CONTINUE
        BPEAKS,BEGIN ; TEMPORARY RETURN TO MONITOR....
        BPEAKS,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
        DEC CLK ; WAIT SOME MORE ?
        BNE IS ; YES
        CLR DVICE ; NO, SET TO DROP THE MODULE
        MOV #3,ERRTVP ; CONTROLLER NOT READY
        *****
        HPEAKS,BEGIN,TABLE ; CONTROLLER NOT READY
        *****
        RTS PC ; RETURN

```

```

25: JSP PC,NOTRDY ; MAKE SURE ALL CHOSEN DRIVES ARE READY
     TST DVICE ; ANY DRIVES LEFT ?
     BEQ SS ; NO, RETURN
     JSF PC,DRVADR ; GET A DRIVE NUMBER
     BITR #P13,FLAG ; ALL DRIVES RESET ?
     RNE CS ; YES, RETURN
     MOV DPVVE,ORHCS2 ; NO, LOAD NEXT DRIVE ADDRESS
     BIT #174000,ORSDT ; IS THIS AN RS DISK ?
     BNE SS ; NO, GO DROP IT
     JSR #E,CLEAR ; GO CLEAR OUT ANY ERRORS
     RNE SS ; KEEP GOING
     JSP PC,DROP ; DROP THE NON-RS DRIVE
     MOV #6,ERRTVP ; NOT AN RS OR NOT READY
     *****
     HPEAKS,BEGIN,TABLE ; NOT AN RS DRIVE
     *****
     MSGMS,BEGIN,DPP ; ASCII MESSAGE CALL WITH COMMON HEADER
     BR CS ; MAKE SURE ALL GET RESET
     RTS PC ; RETURN

```

```

65: JSP PC,NOTRDY ; MAKE SURE ALL CHOSEN DRIVES ARE READY
     TST DVICE ; ANY DRIVES LEFT ?
     BEQ SS ; NO, RETURN
     JSF PC,DRVADR ; GET A DRIVE NUMBER
     BITR #P13,FLAG ; ALL DRIVES RESET ?
     RNE CS ; YES, RETURN
     MOV DPVVE,ORHCS2 ; NO, LOAD NEXT DRIVE ADDRESS
     BIT #174000,ORSDT ; IS THIS AN RS DISK ?
     BNE SS ; NO, GO DROP IT
     JSR #E,CLEAR ; GO CLEAR OUT ANY ERRORS
     RNE SS ; KEEP GOING
     JSP PC,DROP ; DROP THE NON-RS DRIVE
     MOV #6,ERRTVP ; NOT AN RS OR NOT READY
     *****
     HPEAKS,BEGIN,TABLE ; NOT AN RS DRIVE
     *****
     MSGMS,BEGIN,DPP ; ASCII MESSAGE CALL WITH COMMON HEADER
     BR CS ; MAKE SURE ALL GET RESET
     RTS PC ; RETURN

```

| | | | | | | | |
|-----|--------|--------|--|--|--|---------|---|
| 690 | 003156 | 000000 | | | | DLTCNT: | 0 |
| 691 | 003160 | 000000 | | | | CLK: | 0 |
| 692 | 003164 | 000000 | | | | CLOCK: | 0 |
| 693 | 003168 | 000000 | | | | BLK1: | 0 |
| 694 | 003172 | 000000 | | | | BLK2: | 0 |
| 695 | 003176 | 000000 | | | | CNT: | 0 |
| 696 | 003180 | 000000 | | | | BSIZ: | 0 |
| 697 | 003184 | 000000 | | | | FINC: | 0 |
| 698 | 003188 | 000000 | | | | DO NOT | CHANGE THE ORDER OF THE NEXT FOUR LOCATIONS |
| 699 | 003192 | 000000 | | | | PAIR: | 0 |
| 700 | 003196 | 000000 | | | | XMEM: | 0 |
| 701 | 003200 | 000000 | | | | PA2: | 0 |
| 702 | 003204 | 000000 | | | | EA22: | 0 |
| 703 | 003208 | 000000 | | | | ZERO: | 0 |
| 704 | 003212 | 000000 | | | | DRIVE: | 0 |
| 705 | 003216 | 000000 | | | | DRIVE: | 0 |
| 706 | 003220 | 000000 | | | | DRIVE: | 0 |
| 707 | 003224 | 000000 | | | | DRIVE: | 0 |
| 708 | 003228 | 000000 | | | | WCNT1: | 0 |
| 709 | 003232 | 000000 | | | | WCNT2: | 0 |
| 710 | 004224 | 000000 | | | | RUF1: | BLFW 256. |
| 711 | 004228 | 000000 | | | | TABLE: | |
| 712 | 004232 | 000000 | | | | RHCS1: | 0 |
| 713 | 004236 | 000000 | | | | RHCS2: | 0 |
| 714 | 004240 | 000000 | | | | RHCS3: | 0 |
| 715 | 004244 | 000000 | | | | RHRA: | 0 |
| 716 | 004248 | 000000 | | | | RSDS: | 0 |
| 717 | 004252 | 000000 | | | | RSDS: | 0 |
| 718 | 004256 | 000000 | | | | RSDS: | 0 |
| 719 | 004260 | 000000 | | | | RSDS: | 0 |
| 720 | 004264 | 000000 | | | | RSDS: | 0 |
| 721 | 004268 | 000000 | | | | RSDS: | 0 |
| 722 | 004272 | 000000 | | | | RSDT: | 0 |
| 723 | 004276 | 000000 | | | | RSDT: | 0 |
| 724 | 004280 | 000000 | | | | RSDT: | 0 |
| 725 | 004284 | 000000 | | | | RSDT: | 0 |
| 726 | 004288 | 000000 | | | | RSDT: | 0 |
| 727 | 004292 | 000000 | | | | RSDT: | 0 |
| 728 | 004296 | 000000 | | | | RSDT: | 0 |
| 729 | 004300 | 000000 | | | | RSDT: | 0 |
| 730 | 004304 | 000000 | | | | RSDT: | 0 |
| 731 | 004308 | 000000 | | | | RSDT: | 0 |
| 732 | 004312 | 000000 | | | | RSDT: | 0 |
| 733 | 004316 | 000000 | | | | RSDT: | 0 |
| 734 | 004320 | 000000 | | | | RSDT: | 0 |
| 735 | 004324 | 000000 | | | | RSDT: | 0 |
| 736 | 004328 | 000000 | | | | RSDT: | 0 |
| 737 | 004332 | 000000 | | | | RSDT: | 0 |
| 738 | 004336 | 000000 | | | | RSDT: | 0 |
| 739 | 004340 | 000000 | | | | RSDT: | 0 |
| 740 | 004344 | 000000 | | | | RSDT: | 0 |
| 741 | 004348 | 000000 | | | | RSDT: | 0 |
| 742 | 004352 | 000000 | | | | RSDT: | 0 |

| | | | | | | | |
|-----|--------|--------|--------|--------|---------|--------|-------------------|
| 743 | | | | | | | |
| 744 | | | | | | | |
| 745 | 004407 | 040 | 042040 | 047522 | MES5: | .ASCIZ | "DROPPED" |
| 746 | 004414 | 050120 | 042105 | 000045 | | | |
| 747 | 004421 | 051040 | 052105 | 054522 | MES6: | .ASCIZ | "RETRY EXCEEDED" |
| 748 | 004428 | 043440 | 041538 | 042505 | | | |
| 749 | 004435 | 042504 | 022504 | 000 | | | |
| 750 | 004442 | 040 | 020040 | 051127 | MES7: | .ASCIZ | "WRITE" |
| 751 | 004449 | 052111 | 000105 | | | | |
| 752 | 004456 | 020040 | 053440 | 044522 | MES8: | .ASCIZ | "WRITE-CHECK" |
| 753 | 004463 | 042524 | 041455 | 042510 | | | |
| 754 | 004470 | 042524 | 000 | | | | |
| 755 | 004477 | 040 | 020040 | 042522 | MES9: | .ASCIZ | "READ" |
| 756 | 004484 | 042111 | 000 | | | | |
| 757 | 004491 | 040 | 020040 | 042040 | MES10: | .ASCIZ | "DATA LATE ERROR" |
| 758 | 004498 | 052101 | 020101 | 040514 | | | |
| 759 | 004505 | 042524 | 042440 | 051122 | | | |
| 760 | 004512 | 051117 | 000045 | | | | |
| 761 | | | | | | | |
| 762 | 004530 | 004260 | | | TRERR: | .EVEN | |
| 763 | 004537 | 177777 | | | MES1: | 177777 | |
| 764 | 004544 | 004303 | | | MCPERR: | MES2 | 177777 |
| 765 | 004551 | 177777 | | | MDPERR: | MES3 | 177777 |
| 766 | 004558 | 043335 | | | EXCED1: | MES7 | 177777 |
| 767 | 004565 | 177777 | | | | MES1 | 177777 |
| 768 | 004572 | 004443 | | | | MES2 | 177777 |
| 769 | 004579 | 004422 | | | | MES3 | 177777 |
| 770 | 004586 | 177777 | | | | MES4 | 177777 |
| 771 | 004593 | 004454 | | | | MES5 | 177777 |
| 772 | 004600 | 043440 | | | | MES6 | 177777 |
| 773 | 004607 | 177777 | | | | MES7 | 177777 |
| 774 | 004614 | 004473 | | | | MES8 | 177777 |
| 775 | 004621 | 004422 | | | | MES9 | 177777 |
| 776 | 004628 | 177777 | | | | MES10 | 177777 |
| 777 | 004635 | 004375 | | | DPP: | | |
| 778 | 004642 | 004607 | | | | | |
| 779 | 004649 | 004407 | | | | | |
| 780 | 004656 | 177777 | | | | | |
| 781 | 004663 | 004407 | | | | | |
| 782 | 004670 | 177777 | | | DLTERR: | MES10 | 177777 |
| 783 | 004677 | 000005 | | | ADRI: | .BLKW | 0 |
| 784 | 004684 | 000 | | | NUMB: | .RYTE | 0 |
| 785 | 004691 | 000 | | | | .RYTE | 0 |
| 786 | 004698 | 004612 | | | | .EVEN | |
| 787 | 004705 | 000 | | | TRV1: | .RYTE | 0 |
| 788 | 004712 | 000 | | | TRV2: | .RYTE | 0 |
| 789 | 004719 | 000 | | | TRV3: | .RYTE | 0 |
| 790 | 004726 | 000 | | | FLAG: | .RYTE | 0 |
| 791 | | | | | | | |
| 792 | | | | | | | |
| 793 | 000001 | | | | | | |

| RSAT DEC/11 SYSTEM EXERCISER MODULE | | MACV11 30A(1052) 12-OCT-78 17:03 PAGE 21 | | | | | | | | | | | | | | SEQ 0019 | | |
|-------------------------------------|-----------|--|------|------|------|------|------|------|------|-----|------|-----|-----|------|--|----------|--|--|
| XRSAI0.P11 12-OCT-78 12:16 | | CROSS REFERENCE TABLE -- USER SYMBOLS | | | | | | | | | | | | | | | | |
| ACSR | 000102R | 193# | 589* | | | | | | | | | | | | | | | |
| ADDR | 001022R | 222# | 377* | | | | | | | | | | | | | | | |
| ADDR22= | 001006R | 222# | 377* | 635 | | | | | | | | | | | | | | |
| ADR1 | 004622R | 549# | 783# | | | | | | | | | | | | | | | |
| ASH | 000106R | 197# | 582* | | | | | | | | | | | | | | | |
| ASST | 000104R | 198# | 584* | | | | | | | | | | | | | | | |
| AWAS | 000116R | 198# | 584* | | | | | | | | | | | | | | | |
| BEGIN | 000000R | 15# | 253 | 258 | 284 | 286 | 292 | 309 | 308 | 316 | 374 | 377 | 386 | 387 | | | | |
| BIT0 | = 000001 | 456# | 470 | 474 | 480 | 491 | 492 | 549 | 565 | 568 | | | | | | | | |
| BIT1 | = 000002 | 692# | 409 | 406 | 416 | 423 | 433 | 436 | 501 | | | | | | | | | |
| BIT10 | = 002000 | 222# | 432 | | | | | | | | | | | | | | | |
| BIT11 | = 004000 | 422# | 437 | | | | | | | | | | | | | | | |
| BIT12 | = 010000 | 222# | 599 | | | | | | | | | | | | | | | |
| BIT13 | = 020000 | 222# | 460 | | | | | | | | | | | | | | | |
| BIT14 | = 040000 | 222# | 363 | 458 | 464 | | | | | | | | | | | | | |
| BIT15 | = 100000 | 222# | 432 | | | | | | | | | | | | | | | |
| BIT2 | = 000004 | 222# | 275 | 277 | 442 | 454 | | | | | | | | | | | | |
| BIT3 | = 000010 | 222# | 265 | 527 | 533 | 557 | 671 | | | | | | | | | | | |
| BIT4 | = 000020 | 222# | 473 | | | | | | | | | | | | | | | |
| BIT5 | = 000040 | 222# | 652 | | | | | | | | | | | | | | | |
| BIT6 | = 000100 | 222# | 277 | | | | | | | | | | | | | | | |
| BIT7 | = 000200 | 222# | 462 | 508 | 597 | | | | | | | | | | | | | |
| BIT8 | = 000400 | 222# | 277 | | | | | | | | | | | | | | | |
| BIT9 | = 000800 | 222# | 462 | | | | | | | | | | | | | | | |
| BLK1 | 003164R | 276# | 278* | 345 | 351 | 391 | 518* | 519 | 521* | 522 | 693# | | | | | | | |
| BLK2 | 003166R | 357# | 522* | | | | | | | | | | | | | | | |
| BLOCK | 002204R | 257# | 518# | | | | | | | | | | | | | | | |
| BRACKS | = 000014R | 222# | 644 | 566 | 656 | 657 | | | | | | | | | | | | |
| BR1 | 000012R | 161# | 644 | | | | | | | | | | | | | | | |
| BR2 | 000013R | 162# | 644 | | | | | | | | | | | | | | | |
| RSIZ | = 003172R | 393# | 400* | 401* | 411* | 427 | 696# | | | | | | | | | | | |
| RUDS | = 006000 | 222# | 277 | | | | | | | | | | | | | | | |
| BUFIN | = 003221R | 204# | 709# | | | | | | | | | | | | | | | |
| CDATA | = 104412 | 222# | 284 | | | | | | | | | | | | | | | |
| CDECT | 000144R | 214# | 277 | | | | | | | | | | | | | | | |
| CDUCT | 000146R | 214# | 277 | | | | | | | | | | | | | | | |
| CLEAR | 001156R | 360# | 494 | 496 | 562 | 676 | | | | | | | | | | | | |
| CLK | 003160R | 501# | 567* | 653* | 658* | 691# | | | | | | | | | | | | |
| CLOCK | 003162R | 503# | 569* | 692# | | | | | | | | | | | | | | |
| CNT | 003170R | 235# | 695* | | | | | | | | | | | | | | | |
| CONFIG | 000056R | 181# | 588* | | | | | | | | | | | | | | | |
| CSRA | 000160R | 191# | 588* | | | | | | | | | | | | | | | |
| DATCK | = 004411 | 222# | 284 | | | | | | | | | | | | | | | |
| DATERS | = 104412 | 222# | 284 | | | | | | | | | | | | | | | |
| DLTCNT | = 003156R | 277# | 453* | 690# | | | | | | | | | | | | | | |
| DLTCR | = 004576R | 456# | 781# | | | | | | | | | | | | | | | |
| DLTCR | = 004576R | 456# | 781# | | | | | | | | | | | | | | | |
| DRP | 004566R | 324# | 530* | 535* | 555* | 705# | | | | | | | | | | | | |
| DRPADR | 002236R | 242# | 530* | 555* | 576# | 678 | | | | | | | | | | | | |
| DRVWF | 003214R | 243# | 526# | 531 | 556 | 670 | | | | | | | | | | | | |
| DVI | 003212R | 243# | 247* | 360 | 365 | 526* | | | | | | | | | | | | |
| DVID1 | 000014R | 163# | 241 | 250 | 263 | 528 | 535 | 534* | 540 | 549 | 554* | 593 | 673 | 706# | | | | |
| EAT22 | 003254R | 376# | 377* | 378* | 379 | 702# | | | | | | | | | | | | |

| RSAT DEC/11 SYSTEM EXERCISER MODULE | | MACV11 30A(1052) 12-OCT-78 17:03 PAGE 22 | | | | | | | | | | | | | | SEQ 0020 | | | |
|-------------------------------------|-----------|--|------|------|------|------|------|------|------|-----|-----|------|------|------|--|----------|--|--|--|
| XRSAI0.P11 12-OCT-78 12:16 | | CROSS REFERENCE TABLE -- USER SYMBOLS | | | | | | | | | | | | | | | | | |
| ENDITS | = 104413 | 222# | 286 | | | | | | | | | | | | | | | | |
| ENDR | = 104450R | 222# | 292 | | | | | | | | | | | | | | | | |
| ERRDR | = 001650R | 322# | 444# | | | | | | | | | | | | | | | | |
| ERRRTP | 000106R | 196# | 468# | 488* | 572* | 661* | 679* | | | | | | | | | | | | |
| ERSUB1 | 002535R | 450# | 571 | 588# | | | | | | | | | | | | | | | |
| ERSUB2 | 002512R | 582# | 571 | | | | | | | | | | | | | | | | |
| EXCD1 | 004552R | 308# | 768# | | | | | | | | | | | | | | | | |
| EXCD2 | 004552R | 308# | 771# | | | | | | | | | | | | | | | | |
| EXCD3 | 004560R | 316# | 774# | | | | | | | | | | | | | | | | |
| EXITS | = 104400 | 222# | 382 | | | | | | | | | | | | | | | | |
| FIN | 004532R | 222# | 264 | 291# | | | | | | | | | | | | | | | |
| FLAG | 004515R | 222# | 264 | 291# | | | | | | | | | | | | | | | |
| FREE | 000150P | 657# | 671 | 700# | 277* | 395* | 406* | 416* | 423* | 433 | 436 | 442* | 527* | 533* | | | | | |
| FUNC | 003174R | 214# | 277 | | | | | | | | | | | | | | | | |
| GETPAS | = 104411 | 343# | 347* | 353* | 380* | 381 | 697# | | | | | | | | | | | | |
| GO | 005544R | 273# | 299 | | | | | | | | | | | | | | | | |
| GOA | 006600R | 276# | 280# | 307 | | | | | | | | | | | | | | | |
| GGB | 006066R | 282# | 315 | | | | | | | | | | | | | | | | |
| GOGO | 001210R | 346# | 352 | 358 | 365# | | | | | | | | | | | | | | |
| GWBUFF | = 104414 | 222# | 256 | | | | | | | | | | | | | | | | |
| HRCNT | 000044R | 178# | 470 | | | | | | | | | | | | | | | | |
| HROPS | = 104405 | 222# | 491 | 491 | 574 | 663 | 681 | | | | | | | | | | | | |
| HROPAS | 000055R | 178# | 470 | | | | | | | | | | | | | | | | |
| ICONT | 000036P | 173# | 681 | | | | | | | | | | | | | | | | |
| ICOUNT | 000046P | 174# | 681 | | | | | | | | | | | | | | | | |
| IDNUM | 000123R | 215# | 277 | | | | | | | | | | | | | | | | |
| IMODX | = 000060 | 215# | 259 | | | | | | | | | | | | | | | | |
| INIT | 000030P | 170# | 277 | | | | | | | | | | | | | | | | |
| IWR | 000120P | 202# | 224* | | | | | | | | | | | | | | | | |
| MAP22 | = 104416 | 222# | 374 | | | | | | | | | | | | | | | | |
| MCPEPR | 004534R | 477# | 764# | | | | | | | | | | | | | | | | |
| MDPEPR | 004540R | 480# | 766# | | | | | | | | | | | | | | | | |
| MES1 | 004260R | 726# | 782 | | | | | | | | | | | | | | | | |
| MES10 | 004538R | 757# | 781 | | | | | | | | | | | | | | | | |
| MES2 | 004303R | 730# | 781 | | | | | | | | | | | | | | | | |
| MES3 | 004335R | 735# | 766 | | | | | | | | | | | | | | | | |
| MES4 | 004375R | 741# | 777 | | | | | | | | | | | | | | | | |
| MES5 | 004407R | 745# | 779 | | | | | | | | | | | | | | | | |
| MES6 | 004422R | 747# | 779 | | | | | | | | | | | | | | | | |
| MES7 | 004443R | 750# | 778 | 772 | 775 | | | | | | | | | | | | | | |
| MES8 | 004454R | 752# | 771 | | | | | | | | | | | | | | | | |
| MES9 | 004475R | 755# | 774 | | | | | | | | | | | | | | | | |
| MDDNAM | 000075P | 157# | 774 | | | | | | | | | | | | | | | | |
| MODSP | 000252P | 171# | 220# | | | | | | | | | | | | | | | | |
| MSGNS | = 104403 | 222# | 300 | 308 | 316 | 324 | 456 | 474 | 477 | 480 | 577 | 683 | | | | | | | |
| MSCS | = 104402 | 222# | 300 | | | | | | | | | | | | | | | | |
| MSCS | = 104401 | 222# | 300 | | | | | | | | | | | | | | | | |
| NEXT | 000476P | 262# | 270 | 288 | 325 | | | | | | | | | | | | | | |
| NEXTA | 000722P | 301# | 309 | 317 | 321# | | | | | | | | | | | | | | |
| NOTROV | 002870R | 259# | 554# | | | | | | | | | | | | | | | | |
| NTRUPT | 002344R | 366# | 667 | | | | | | | | | | | | | | | | |
| NULL | = 000000 | 222# | | | | | | | | | | | | | | | | | |

