

.REM 8

IDENTIFICATION  
-----

PRODUCT CODE: AC-E968C-MC  
PRODUCT NAME: CXRMBCO RH11/RM03 DUAL PORT MOD  
PRODUCT DATE: SEPTEMBER 1978  
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS 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) 1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RMB IS AN IOMOD THAT EXERCISES RM03/RM02 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 TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RM03/RM02'S WITH TWO RH11 CONTROLLERS

STORAGE: RMB REQUIRES:

- 1. DECIMAL WORDS: 1793
- 2. OCTAL WORDS: 03401
- 3. OCTAL BYTES: 7002

3. PASS DEFINITION

ONE PASS OF THE RMB MODULE CONSISTS OF 300 CYCLES OF THE BASIC TEST SEQUENCE. A-PORT DOES A BIT-FLIP CHECK ON THE DATA COMPARE IN THE CURRENT SECTION (CALLED THROUGHOUT PROGRAM) AND THEN WRITES BLK 0 WITH THE FLAG WORD (THE FIRST WORD OF BLOCK ZERO), AND THE A-PORT JUST NEXT. IT WRITES BLK 0 TO A-PORT THEN SITS IN A LOOP PERIODICALLY READING BLK 0 TO SEE IF A-PORT HAS MODIFIED THE FLAG. B-PORT SITS IN A LOOP WAITING FOR THE FLAG IN BLK ZERO TO BE WRITTEN WITH DATA FROM B-PORT. WHEN IT DOES, B-PORT READS THE DATA THAT A-PORT JUST WROTE. IT THEN B-PORT WRITES THE DATA BACK AND A-PORT SEEKS FOR THE FLAG IN BLK 0. IF THE DATA AFTER READING BLK 0 AND SEEING THE FLAG REVERSED, B-PORT THEN RE-READS THE DATA AND RE-WRITES IT. THIS VERIFICATION ARBITRARILY WRITES BLK 0 TO A-PORT AND B-PORT. B-PORT DOES NOT DO ANY IN-CORE COMPARISONS.

B-PORT SHOULD NEVER ENCOUNTER A BAD BLK BECAUSE IT ONLY USES BLK'S WHICH A-PORT HAS ALREADY SUCCESSFULLY USED. THEREFORE, WHEN A-PORT GETS 4 ERRORS WHILE ATTEMPTING A-PORT ON ONE BLK, A-PORT MOVES TO THE NEXT BLK. NEVER GETTING A-PORT THAT DRIVE IS PREPARED BECAUSE IT IS MOST LIKELY A FAULTY SINCE A-PORT WAS JUST ABLE TO SUCCESSFULLY USE THIS BLK. LOCATION 206 THROUGH 244 CONTAIN PCOM FOR 16 BAD BLK'S. IN ERROR TYPEOUTS WHICH DUMP THE REGISTER, THE LAST ITEM TYPED IN THE TABLE IS THE CURRENT BLOCK NUMBER. ENTER THIS INTO THE BAD BLK TABLE TO AVOID ERRORS FROM UNKNOWN MEDIA BAD SPOTS. THIS TABLE ONLY VALID FOR A-PORT SINCE B-PORT

ALWAYS GETS ITS BLK ADDRESSES FROM A-PORT. IF YOU MODIFY  
THE WRITE BUFFER SIZE YOU MUST ADD SECTORS TO THE BAD BLK  
TABLE TO AVOID THE ERRORS. FOR EXAMPLE, IF BLK 3474 IS BAD  
AND YOU DOUBLE THE WRITE TRANSFER SIZE TO 1000 OCTAL BYTES  
YOU MUST ADD BLK 3473 TO THE TABLE SO THE PROGRAM DOES NOT START  
A TRANSFER THAT WILL EXTEND ON INTO THE KNOWN BAD BLK.

B-PORT MUST HAVE SRI BIT 4 SET. \*\*\*\*\* SRI = 2C (OCTAL) \*\*\*\*\*  
THIS MODULE IS A DUAL-PORT  
MODULE TEST, AND ONLY USES THE FIRST 7777 BLKS  
OF THE PACK. IF YOU WANT TO VERIFY THE COMPLETE MEDIUM, YOU  
SHOULD RUN RMA MODULE, THE NORMAL SINGLE PORT TEST.

4. EXECUTION TIME  
-----

ONE PASS OF RMB RUNNING ALONE ON A PDP-11/70 TAKES APPROXIMATELY ONE MINUTE.

5. CONFIGURATION REQUIREMENTS  
-----

DEFAULT PARAMETERS:

DEVADR: 176700, VECTOR: 254, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP  
-----

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY  
THIS MODULE ALSO SUPPORTS RP04/5/6 ON THE SAME MASSBUS CONTROLLER.  
HOWEVER, THIS MODULE IS NOT USED TO EXERCISE RP04/5/6 ALONE.

7. OPERATION OPTIONS  
-----

SRI BIT2 SET(1):  
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT

SRI BIT2 CLEAR(0):  
TYPE OUT DATA LATE ERRORS AND COUNT THEM

SRI BIT4 SET (1) ;-R-PORT PROGRAM \*\*\*\*\*SRI = 20 (OCTAL) \*\*\*\*\*

SRI BIT15 SET (1):  
32 REGISTER OPTION CN RH70

SRI BIT15 CLEAR (0):  
22 REGISTER OPTION CN RH70

8. NON-STANDARD PRINTOUTS  
-----

- A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT
- B. ERROR MESSAGES DUMP THE CONTENTS OF THE 20 RH11/RM03 REGISTERS IN THE FOLLOWING ORDER:

RMCS1 RMCW RMR1 RMDA RMC2 RMOF RMR1 RMA8  
RMLA RMDH RMR2 RMR1 RMDA RMC2 RMOF RMR1 RMA8  
RMR2 RMR2 RMR1 RMDA RMC2 RMOF RMR1 RMA8  
RMR2 RMR2 RMR1 RMDA RMC2 RMOF RMR1 RMA8

9. DUAL PORT SETUP:

TO RUN A DUAL PORT SYSTEM, SRI HAS TO BE MODIFIED TO INDICATE TO THE MODULE, WHICH PORT THE MODULE IS LOCATED ON. SEE SECTION 7. FOR SRI OPTIONS. THE CONTROLLER SELECT SWITCH ON THE RM03 MUST BE IN THE A/B POSITION. THIS SWITCH IS ACTIVATED WHEN THE DRIVE IS CYCLED UP. IF SWITCH WAS NOT IN THIS POSITION WHEN DRIVE WAS POWERED UP, THE FOLLOWING STEPS MUST BE TAKEN. PLACE THE SWITCH IN THE A/B POSITION, DISABLE THE DRIVE, (USING THE DISABLE SWITCH ON THE DRIVE), THEN ENABLE THE DRIVE WITH THE SAME SWITCH. THIS WILL PUT THE RM03 IN THE DUAL PORT MODE.

10. BAD SPOT FILE  
-----

TOTAL 16 BAD BLOCKS CAN BE RETRIEVED FROM PORT A. THE TABLE LABELED "BADSPOT" IS SET UP TO RETRIEVE ALL BAD SPOT FILES FROM ALL DRIVES ASSIGNED IN THE BADMAP.DIVISION (CYLINDER 822, TRACK 4, SECTOR 0 - MANUFACTURER BAD SPOT FILE) (CYLINDER 822, TRACK 4, SECTOR 12 - USER PAD SPOT FILE) LOCATION 1722, MUST BE SET TO 10 TO SEARCH USER BAD SPOT FILE



```

310 000422 000000
311 000424 000000
312 000426 000000
313 000430 000000
314 000432 000000
315 000436 000000
316 000440 000000
317 000444 000000
318 000448 000000
319 000444 000000
320 001444 000000
321 001444 000000
322 001446 000000
323 001450 000000
324 001452 000000
325 001456 000000
326 001460 000000
327 001464 000000
328 001468 000000
329 001472 000000
330 001476 000000
331 001480 000000
332 001484 000000
333 001488 000000
334 001492 000000
335 001496 000000
336 001500 000000
337 001504 000000
338 001508 000000
339 001512 000000
340 001516 000000
341 001520 000000
342 001524 000000
343 001528 000000
344 001532 000000
345 001536 000000
346 001540 000000
347 001544 000000
348 001548 000000
349 001552 000000
350 001556 000000
351 001560 000000
352 001564 000000
353 001568 000000
354 001572 000000
355 001576 000000

```

```

DTCNT: 0
DTC: 000000
TIMER: 000000
ZERAD: 000000
FERADR: 000000
CNT: 000000
MIXDR: 000000
MIXDRW: 000000
RABSEC: 000000
RBUF: .BLKW
MIX DRIVE FLAG = 1 IF NOT RM03 RM02
SHAREWARE DETECT PAD SECTOR
256.

```

```

TABLE:
RMCS1: 177777
RMBC: 000000
RMBA: 000000
RMCA: 000000
RMCS2: 000000
RMDS: 000000
RMER: 000000
RMAS: 000000
RMFA: 000000
RMDB: 000000
RMRR: 000000
RMDT: 000000
RMSW: 000000
RMDF: 000000
RMDC: 000000
RMHR: 000000
RMND: 000000
RMED: 000000
RMCC: 000000
RMBA: 000000
RMCS3: 0

```

```

356 001574 000432
357 001576 000436
358 001600 000376
359 001622 177777

```

```

IFERAD: FERADP
IFERCT: CNT
XRLE: BLK
177777

```

```

370 001614 0 2767 000400 176302 START: MOV #256,WDTO ;256 WORDS TO MEM/ITERATION
371 001612 0 2767 000400 176276 MOV #256,WDFR ;256 WORDS FROM MEM/ITERATION
372 001620 0 2767 000000 176272 MOV #3,INTR ;3 INTERRUPTS/ITERATION
373 001624 0 2767 178162 176554 MOV DVID,DVICE ;GET DRIVES TO TEST
374 001634 0 2767 000000 176530 MOV SPOIN,R6 ;START AT TRACK POINT
375 001640 0 2767 000001 176530 MOV #1,BLK ;SET 1ST TIME FLAG BIT
376 001646 0 2767 000001 176532 MOV #1,FLAG ;INITIATE UNIT NUMBER
377 001654 0 2767 177777 176532 MOV #1,UNITNO ;INITIATE UNIT
378 001662 0 2767 004103 JSR PC,SETUP ;SETUP PSEL BIT
379 001670 0 2767 002543 JSR PC,PICKOR ;CLEAR THE RH
380 001672 0 2767 000240 JSP PC,SETUP2 ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
381 001674 0 2767 004332 JSP PC,REZET ;RESET END-OF-PASS COUNTER
382 001676 0 2767 003744 JSP PC,REZET ;SETUP PSEL BIT
383 001678 0 2767 000000 000124 GETPAS,REGIN,RBUFVA ;CLEAR THE RH
384 001712 0 2767 176500 176072 CLR CNT ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
385 001716 0 2767 000020 DEC #2,SR1 ;RESET END-OF-PASS COUNTER
386 001724 0 2767 011161 BNE SR1 ;BRANCH IF NOT DONE
387 001726 0 2767 177777 176460 MOV #1,UNITNO ;THEY DON'T HAVE TO RETRIEVE THE BAD SPOT FILE
388 001734 0 2767 000252 MOV #RADSPT,R1 ;RESET UNIT NUMBER
389 001744 0 2767 000820 1S: CLR #16,D0 ;TABLE ENTRY
390 001746 0 2767 005345 CLR #16,D0 ;16 WORDS
391 001752 0 2767 002454 RES2X: JSR PC,PICKOR ;CLEAR THE TABLE
392 001754 0 2767 000520 BR 10$ ;DECREMENT ONE WORD
393 001756 0 2767 176452 CLR TRV ;PICK UP THE FIRST ON LINE DRIVE
394 001760 0 2767 176414 176414 MOV #770,FLAG ;BRANCH IF ALL DRIVES ARE ACCESSED
395 001762 0 2767 001466 176402 MOV #2,CYL ;SPECIAL FLAG TAG FOR RETRIEVE BAD SPOT FILE
396 001764 0 2767 000004 176374 MOV #4,DSKADR+1 ;CYL 622
397 001766 0 2767 000000 176364 MOV #0,DSKADR ;TRACK 4
398 001768 0 2767 003440 JSR #5,READY ;SEC 0 (SEC 12 USER)
399 001770 0 2767 003174 BR #6,READY ;ACCESS THE DRIVE
400 001772 0 2767 176404 JSR PC,NOTRDY ;BRANCH IF READY
401 001774 0 2767 100471 JST #1,XDV ;WAIT IF NOT READY
402 001776 0 2767 001222 JSR #2,RE ;EXIT IF NOT RMO3 OR RMO2
403 001778 0 2767 000444 JST #2,RE ;READ THE BAD SPOT FILE
404 001780 0 2767 000000 MOV #RBUF,R1 ;INPUT BUFFER
405 001782 0 2767 001000 MOV #R1,R2 ;END ADDRESS
406 001784 0 2767 000010 ADD #10,R2 ;LAST LOCATION OF THE INPUT BUFFER
407 001786 0 2767 000252 MOV #RADSPT,R3 ;LAST SPOT FILE STARTS FROM 5TH WORDS
408 001788 0 2767 000040 MOV #R3,R4 ;RADSPT RELOCATABLE TABLE ADDRESS
409 001790 0 2767 000532 1S: ADD #2,R4 ;LAST LOCATION OF THE TABLE
410 001792 0 2767 000040 CMP #10,(R1) ;16 WORDS
411 001794 0 2767 000532 BLOS #6,CYL ;BLOCK OVER CYL 410, THEN DON'T WORRY
412 001796 0 2767 176274 MOV #R1,RO ;INCREMENT THE POINTER
413 001798 0 2767 001116 MOV #R1,RO ;CYLINDER ADDRESS
414 001800 0 2767 001406 BR 3$ ;BRANCH IF CYL 0
415 001802 0 2767 000240 2S: ADD #32,*5,PLK ;BLOCK # FOR ONE CYLINDER
416 001804 0 2767 001356 DEC RO ;DECREMENT TEMP COUNT
417 001806 0 2767 000003 3S: MOV #R1,RO ;LOOP IF NOT ZERO
418 001808 0 2767 000240 4S: REQ #32,.BLK ;TRACK NUMBER
419 001810 0 2767 176242 4S: ADD #32,.BLK ;BRANCH IF TRACK 0
420 001812 0 2767 000360 DEC RO ;ADJUST BLOCK FOR ONE TRACK
421 001814 0 2767 000374 BNE 4$ ;BRANCH IF NOT DONE

```

```

422 001816 0 2767 000002 5S: MOVR 2(R1),PC ;SECTOR NUMBER
423 001818 0 2767 176226 ADD #R1,BLK ;UPDATE SECTOR # FOR BLOCK COUNT
424 001820 0 2767 000713 TST #3 ;ENTRY FOR THE TABLE ?
425 001822 0 2767 000166 BNE #5 ;BRANCH IF NOT
426 001824 0 2767 176216 MOV #R3,BLK ;LOAD INTO TABLE
427 001826 0 2767 000004 6S: ADD #2,R3 ;UPDATE TABLE POINTER
428 001828 0 2767 000004 7S: CMP #R1,R2 ;UPDATE THE BAD SPOT FILE POINTER
429 001830 0 2767 000004 BHS #5 ;END OF FILE
430 001832 0 2767 000004 CME #R1,R4 ;BRANCH IF IT IS
431 001834 0 2767 000004 BLS #5 ;END OF TABLE ?
432 001836 0 2767 000004 BLS #5 ;BRANCH IF NOT
433 001838 0 2767 000004 MSENS,MF #5,BEGIN ;DON'T DO ANYTHING
434 001840 0 2767 000002 8S: ADD #2,R3 ;SUPDTE THE TABLE ENTRY
435 001842 0 2767 000002 BR #5 ;CHECK THE TABLE AND FILE ENTRY
436 001844 0 2767 000001 9S: BR RES2X ;TO NEXT DRIVE
437 001846 0 2767 000001 10S: MOV #1,BLK ;RESET BLOCK NUMBER
438 001848 0 2767 176150 MOV #1,FLAG ;RESTORE FIRST TIME FLAG
439 001850 0 2767 176152 RR RSTR1 ;CONTINUE
440 001852 0 2767 176156 RR RSTR1 ;+ SUPPORT - DT03
441 001854 0 2767 176146 RESTRT: TST CNT1 ;+ SUPPORT
442 001856 0 2767 177326 BNE RSTR1 ;+ FOR
443 001858 0 2767 000004 TST CNT ;+ DT03
444 001860 0 2767 177326 BNE RSTR1 ;+ BUS
445 001862 0 2767 000004 JMP START ;+ SWITCH
446 001864 0 2767 000000 RSTR1: ;+
447 001866 0 2767 000000 GETPAS,REGIN,RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
448 001868 0 2767 176128 CLR CNT ;CLEAR THE LOCATE COUNTER
449 001870 0 2767 177174 MOV #1,UNITNO ;PRE-SET UNIT NUMBER
450 001872 0 2767 000000 176116 LOOP1: GETPAS,REGIN ;GET WRITE BUFFER INFORMATION
451 001874 0 2767 002244 JSR PC,PICKOR ;GO PICK A BLK
452 001876 0 2767 002122 LOOP2: JSP PC,PICKOR ;GO PICK A DRIVE
453 001878 0 2767 176116 BR 1$ ;RETURNS HERE IF ALL DRIVES DONE
454 001880 0 2767 000044 JSP PC,CYCL ;DESE CLR RE TRY COUNT
455 001882 0 2767 000000 BR LOOP2 ;GO DO A CYCLE ON THIS DRIVE
456 001884 0 2767 000001 1S: RLC #1,FLAG ;DO IT TO NEXT DRIVE
457 001886 0 2767 176052 1S: TST DVICE ;CLEAR FIRST TIME FLAG
458 001888 0 2767 000000 176050 BNE #2 ;ANYBODY LEFT TO CHECK?
459 001890 0 2767 000000 2S: ENDS,REGIN ;RR IF YES
460 001892 0 2767 176046 INC CNT1 ;+BUMP COUNTER FOR DT03
461 001894 0 2767 000000 ENDS,REGIN ;SIGNAL END OF ITERATION
462 001896 0 2767 000000 RR LOOP1 ;MONITOR SHALL TEST END CF PASS

```



```

473 002360 002767 000020 175430 CYCLE: BIT #20,SRI ;B-PORT?
474 002366 001114 ;BNE IF SC, ELSE DO A-PORT
475 002370 004567 003064 JSR R5,READY ;READY?
476 002374 000402 ;YES
477 002376 004767 002620 BR CVA ;NO
478 002378 004767 001636 JSR R5,NOTRDY ;SET THE A-PORT FLAG
479 002410 004767 001636 JSR R5,BLKADR ;CONVERT BLK TO DISK ADDR
480 002414 004567 000452 JSR R5,WRITE ;GO WRITE A BLOCK
481 002420 004767 176416 JSR R5,WRITE ;FOUND FAT SLOT?
482 002424 004767 176416 JSR R5,WRITE ;BRANCH IF NOT?
483 002426 004767 176416 JSR R5,WRITE ;OTHERWISE EXIT
484 002428 004767 176416 JSR R5,WRITE ;GO DO WRITE CHECK
485 002430 004567 000532 JSR R5,WRITE ;GO READ A BLOCK
486 002434 004567 000235 JSR R5,WRITE ;REQUEST FOR MONITOR TO CHECK DATA
487 002440 104412 000060 CDATAS,BEGIN,RBUFA ;IF ERROR, CONTINUE
488 002446 002450 +2
489
490
491
492 002450 004767 002154 ;NOW UPDATE BLOCK 0 FOR B-PORT'S INFORMATION
493 002454 004767 175726 JSR PC,CLRRR ;CLEAR THE READ BUFFER
494 002462 004767 175710 MOV FLAG,RBUF ;PUT INFO IN FIRST WORD
495 002470 004567 001859 MOV BLK,RBUF+2 ;PUT CURRENT ADDR IN NXT
496 002474 001277 000000 JSR R5,WRTOO ;GO UPDATE BLOCK 0
497 002502 001277 000913 MOV #0,TIMER ;INIT WAIT LOOP
498 002510 004567 175712 JSR R5,WRMCS1 ;GIVE THE DRIVE TO R
499 002514 001066 4S: DEC R0
500 002516 004767 002126 JSR PC,DROP ;TEMPORARY RETURN TO MONITOR
501 002520 004767 000000 MCON,S,BEGIN,PORTHG ;THEN CONTINUE AT NEXT INSTRUCTION.
502 002524 004767 000000 RTS PC
503 002532 012700 000002 MOV #2,R0
504 002536 104407 000000 BREAKS,BEGIN
505 002542 004407 000000 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
506 002546 004407 000000 DEC R0 ;THEN CONTINUE AT NEXT INSTRUCTION.
507 002550 001372 4S: BNE R4S,READY
508 002552 004567 002702 JSR R5,READY ;READY?
509 002556 004767 002436 JSR PC,NOTRDY ;YES
510 002560 004567 001952 JSR R5,RD00 ;NO
511 002564 004567 000002 NOP ;GO READ BLOCK 0
512 002568 004767 175644 BR R5,RD00 ;CHANGE TO BR IF ONLY SINGLE PORT
513 002572 001740 000002 MOV #2,RBUF ;HAS B UPDATED BLK 0 YET?
514 002576 004567 000454 JSR R5,READ ;BR BACK IF NOT
515 002580 004567 000600 CDATAS,BEGIN,RRUFA ;GO READ THE DATA B WROTE
516 002584 004567 000600 RTS PC ;REQUEST FOR MONITOR TO CHECK DATA
517 002588 004567 000600 ;IF ERROR, CONTINUE

```

```

520 002620 002767 000000 175600 CYCLER: MOV #0,TIMER
521 002624 004567 002620 JSR R5,READY ;READY?
522 002632 004767 000402 BR CVA ;YES
523 002634 004767 002362 JSR R5,NOTRDY ;NO
524 002640 004567 004776 JSR R5,RD00 ;GO SEE IF A HAS DONE YET
525 002644 004767 000001 CYR: BIT #1,FLAG ;IS THIS THE FIRST TIME SINCE START?
526 002648 004767 000001 BEQ R1,PC ;BR IF NC, SKIP THIS CHECK
527 002652 004767 000001 BEQ R4,RBUF ;HAS A WRITTEN THIS BLOCK?
528 002654 002767 000004 BEQ R4,RBUF ;BR IF NO
529 002658 001410 000001 BEQ R1,PC ;HAS A SET THE FIRST TIME FLAG?
530 002664 002767 000001 BEQ R1,PC ;BR IF NOT MUST WAIT
531 002670 001404 000002 BIT #2,RBUF ;HAS A WRITTEN THIS BLOCK SINCE B DID?
532 002674 004767 000002 BEQ R2,PC ;BR IF SC, ELSE
533 002678 001432 000013 MOV #13,RRMCS1 ;GIVE PORT TO A
534 002704 001277 000013 MOV #2,R0
535 002710 001270 000002
536 002714 104407 000000 BREAKS,BEGIN
537 002718 004407 000000 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
538 002722 004407 000000 DEC R0 ;THEN CONTINUE AT NEXT INSTRUCTION.
539 002726 005300 4S: BNE R4S,TIMER
540 002730 001372 175470 JSR R5,TIMER
541 002734 005300 001704 BNE R4S,PC
542 002738 001704 000000 JSR PC,DROP ;ASCII MESSAGE CALL WITH COMMON HEADER
543 002742 004767 000000 MCON,S,BEGIN,PORTHG
544 002746 004567 002500 RTS PC ;READY?
545 002750 004567 002224 JSR R5,READY ;YES
546 002754 004767 002224 JSR R5,NOTRDY ;NO
547 002758 004767 002224 BR CVA ;TRY AGAIN
548 002762 004767 175452 3S: MOV RBUF+2,BLK
549 002766 001256 JSR PC,BLKADR ;GET THE CURRENT BLK
550 002770 004567 001256 JSR R5,READ ;GENERATE DISK ADDR FROM IT
551 002774 004567 000254 JSR R5,WRITE ;GO READ WHAT I WROTE
552 002778 004567 000344 JSR R5,WRITEB ;GO WRITE IT BACK OUT
553 002782 004567 000434 JSR R5,WRITECB ;GO WRITE CHECK IT
554 002786 004767 000602 JSR PC,CLRRR ;CLEAR BUFFER
555 002790 004767 175356 JSR R2,FLAG ;SET BIT SAYING B'S DONE
556 002794 004567 175400 MOV FLAG,RBUF ;PUT INFO INTO RBUF
557 002798 004767 000013 JSR R5,WRTOO ;GO WRITE IT FOR A TO SEE
558 002802 001270 000002 MOV #13,RRMCS1 ;GIVE DRIVE TO A
559 002806 001270 4S: MOV #2,R0
560 002810 104407 000000 BREAKS,BEGIN
561 002814 004407 000000 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
562 002818 004407 000000 DEC R0 ;THEN CONTINUE AT NEXT INSTRUCTION.
563 002822 005300 4S: BNE R4S,PC
564 002826 001372 175470 JSR R5,PC
565 002830 005300 001372
566 002834 004767 000600
567 002838 004767 000600

```

```

568 ; MACRO LINEUP EABITS ; LINE UP EA BITS FOR RHCSI
569 LINEUP EABITS ; LINE UP EA BITS FOR RHCSI
570 MOV EABITS,RO ; GET EXTENDED MEMORY BITS
571 ASL RO ; SHIFT 4 PLACES TO THE LEFT
572 ASL RO ; TO LINE UP WITH RHCSI
573 ASL RO
574 MOV RO,XMEM ; SAVE THE SHIFTED BITS
575 ENDM LINEUP
576
577
578
579
580 003072 012767 000161- 175324 WRITE: MOV #161,FUNC ; LOAD WRITE FUNCTION
581 003106 012767 003072- 175324 MOV #WRITE,FERADR ; SAVE WHERE WE WERE
582 003112 005416 175030 MOV #RBUF57,-(SP) ; GET WRITE SIZE
583 003114 012777 176402 NEG (SP) ; NEGATE IT
584 003120 012777 175010 MOV (SP)+,RRMWC ; LOAD WORD COUNT
585 003134 012777 175246 MOV #RUFFP,RRMBA ; LOAD BUFFER ADDRESS
586 003134 012777 176412 MOV #DSKADR,RRMDA ; LOAD DISK ADDRESS
587 003142 012700 174776 MOV #CYL,RRMDC ; LOAD CYLINDER ADDRESS
588 003146 006300 MOV #RBUFEA,RO ; LINE UP EA BITS FOR RHCSI
589 003150 006300 ASL RO ; GET EXTENDED MEMORY BITS
590 003150 006300 ASL RO ; SHIFT 4 PLACES TO THE LEFT
591 003150 006300 ASL RO ; TO LINE UP WITH RHCSI
592 003150 006300 ASL RO
593 003162 000167 175206 MOV RO,XMEM ; SAVE THE SHIFTED BITS
594 003166 000167 000556 JMP GO ; CONTINUE
595 003174 012767 003166- 175230 WRITCK: MOV #151,FUNC ; LOAD WRITE-CHECK FUNCTION
596 003174 012767 003166- 175230 MOV #WRITCK,FERADR ; SAVE WHERE WE WERE
597 003206 005416 174734 MOV #RBUF57,-(SP) ; GET WRITE SIZE
598 003210 012677 176314 NEG (SP) ; NEGATE IT
599 003214 012777 174714 MOV (SP)+,RRMWC ; LOAD WORD COUNT
600 003222 012777 175152 MOV #RUFFP,RRMBA ; LOAD BUFFER ADDRESS
601 003222 012777 176276 MOV #DSKADR,RRMDA ; LOAD DISK ADDRESS
602 003230 012777 175146 MOV #CYL,RRMDC ; LOAD CYLINDER ADDRESS
603 003236 012777 176316 LINEUP ; LINE UP EA BITS FOR RHCSI
604 003242 006300 MOV #RBUFEA,RO ; GET EXTENDED MEMORY BITS
605 003244 006300 ASL RO ; SHIFT 4 PLACES TO THE LEFT
606 003246 006300 ASL RO ; TO LINE UP WITH RHCSI
607 003246 006300 ASL RO
608 003252 000167 175112 MOV RO,XMEM ; SAVE THE SHIFTED BITS
609 003256 000167 000462 JMP GO ; CONTINUE
610 003262 012767 000171- 175134 READ: MOV #171,FUNC ; LOAD READ FUNCTION
611 003262 012767 003262- 175134 MOV #READ,FERADR ; SAVE WHERE WE WERE
612 003306 005416 174634 MOV #RBUF57,-(SP) ; GET READ SIZE
613 003310 012677 176212 NEG (SP) ; NEGATE IT
614 003310 012777 174712 MOV (SP)+,RRMWC ; LOAD WORD COUNT
615 003316 012777 176206 MOV #RUFFP,RRMBA ; LOAD BUFFER ADDRESS
616 003316 012777 176206 MOV #DSKADR,RRMDA ; LOAD DISK ADDRESS
617 003324 012777 176222 MOV #CYL,RRMDC ; LOAD CYLINDER ADDRESS
618 003330 012700 174572 LINEUP ; LINE UP EA BITS FOR RHCSI
619 003336 006300 MOV #RBUFEA,RO ; GET EXTENDED MEMORY BITS
620 003340 006300 ASL RO ; SHIFT 4 PLACES TO THE LEFT
621 003342 006300 ASL RO ; TO LINE UP WITH RHCSI
622 003342 006300 ASL RO
623

```

```

624 003344 006300 ASL RO ; TO LINE UP WITH RHCSI
625 003344 006300 ASL RO
626 003344 006300 ASL RO
627 003344 006300 ASL RO
628 003344 006300 ASL RO
629 003344 006300 ASL RO
630 003344 006300 ASL RO
631 003344 006300 ASL RO
632 003344 006300 ASL RO
633 003344 006300 ASL RO
634 003344 006300 ASL RO
635 003344 006300 ASL RO
636 003344 006300 ASL RO
637 003344 006300 ASL RO
638 003344 006300 ASL RO
639 003344 006300 ASL RO
640 003344 006300 ASL RO
641 003344 006300 ASL RO
642 003344 006300 ASL RO
643 003344 006300 ASL RO
644 003344 006300 ASL RO
645 003344 006300 ASL RO
646 003344 006300 ASL RO
647 003344 006300 ASL RO
648 003344 006300 ASL RO
649 003344 006300 ASL RO
650 003344 006300 ASL RO
651 003344 006300 ASL RO
652 003344 006300 ASL RO
653 003344 006300 ASL RO
654 003344 006300 ASL RO
655 003344 006300 ASL RO
656 003344 006300 ASL RO
657 003344 006300 ASL RO
658 003344 006300 ASL RO
659 003344 006300 ASL RO
660 003344 006300 ASL RO
661 003344 006300 ASL RO
662 003344 006300 ASL RO
663 003344 006300 ASL RO
664 003344 006300 ASL RO
665 003344 006300 ASL RO
666 003344 006300 ASL RO
667 003344 006300 ASL RO
668 003344 006300 ASL RO
669 003344 006300 ASL RO
670 003344 006300 ASL RO
671 003344 006300 ASL RO
672 003344 006300 ASL RO
673 003344 006300 ASL RO
674 003344 006300 ASL RO
675 003344 006300 ASL RO
676 003344 006300 ASL RO
677 003344 006300 ASL RO
678 003344 006300 ASL RO
679 003660 012777 174530 175642 CLEAR: MOV UNITNC,RRMCS2 ; LOAD UNIT ADDRESS

```

680 003666 012777 000011 175624 MOV #11,ARMCS1 ;ISSUE A DRIVE CLEAR
681 003674 012777 000000 ;WAIT
682 003674 012777 000000 ;FOR DRIVE CLEAR TO FINISH
683 003700 012777 000023 175612 1S: MOV #23,ARMCS1 ;ISSUE A PACK ACK
684 003706 012777 000000 ;
685 003712 012777 000000 ;
686 003712 012777 000000 ;
687 003716 012777 175614 2S: BR #23,ARMCS1 ;NO WAIT TILL DONE
688 003722 012777 175610 ;CLEAR AS BIT
689 003726 012777 175564 ;CLEAR ANY CONTROLLER ERRORS
690 003734 012777 175610 ;SET BIT FOR 11 FORMAT
691 003742 012777 000205 ;RETURN
692 ;
693 003744 012777 174444 60: MOV UNITNO,ARMCS2 ;LOAD UNIT SELECT
694 003752 012777 174076 ;11770 MONITOR?
695 003760 012777 174376 ;NO
696 003762 012777 175536 ;GET 18 BIT ADDR
697 003770 012777 174374 ;SHIFT EA BITS TO POSITION 4,5
698 003774 012777 174364 ;
699 004000 000267 174364 ;
700 004004 000267 174360 ;
701 004010 000267 000000 ;
702 004016 012777 174350 ;
703 004022 012777 175500 ;
704 004028 012777 175536 ;
705 004034 012777 174330 ;
706 004040 000367 174330 ;
707 004044 012777 174316 ;
708 004052 012777 174312 ;
709 004060 012777 174344 1S: MOV XMEM,FUNC ;LOAD EXTENDED MEMORY BITS
710 004068 012777 175432 ;EXECUTE THE FUNCTION
711 ;
712 NTRUPT: EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
713 MOV R0,(SP) ;SAVE R0
714 MOV BITTAB(R0),R0 ;DRIVE #
715 MOV ARMAS-(SP),R0 ;DRIVE INDICATOR
716 BIC R0,(SP) ;CLEAR CCRESPONDING
717 MOV #20,ARMAS ;ATTENTION BIT
718 MOV #20,ARMAS ;IN THE ATTENTION SUMMARY REGISTER
719 MOV #20,ARMAS ;RESTORE R0
720 ;
721 ;-----
722 ;
723 ;
724 ;
725 ;
726 ;
727 ;
728 ;
729 ;
730 ;
731 ;
732 ;
733 ;
734 ;
735 ;
736 004120 000064 000000 004120 ;
737 ;
738 ;
739 ;
740 ;
741 ;
742 ;
743 ;
744 ;
745 ;
746 ;
747 ;
748 ;
749 ;
750 ;
751 ;
752 ;
753 ;
754 ;
755 ;
756 ;
757 ;
758 ;
759 ;
760 ;
761 ;
762 ;
763 ;
764 ;
765 ;
766 ;
767 ;
768 ;
769 ;
770 ;
771 ;
772 ;
773 ;
774 ;
775 ;
776 ;
777 ;
778 ;
779 ;
780 ;
781 ;
782 ;
783 ;
784 ;
785 ;
786 ;
787 ;
788 ;
789 ;
790 ;
791 ;

736 004214 012767 000776 174164 3S: CMP #770,FLAG ;IN PROCESS OF RETRIVING RAD SPOT
737 004222 012767 000000 ;BRANCH IF NOT
738 004230 012767 000316 ;IF SO, THEN BRANCH BACK
739 004238 012767 174154 ;TRY A DIFFERENT BLOCK
740 004246 012767 174042 ;WAIT TO RE-DO SAME DRIVE WERE ON
741 004254 012767 000004 ;DO DO IT
742 ;
743 ;
744 ;
745 ;
746 ;
747 ;
748 ;
749 ;
750 ;
751 ;
752 ;
753 ;
754 ;
755 ;
756 ;
757 ;
758 ;
759 ;
760 ;
761 ;
762 ;
763 ;
764 ;
765 ;
766 ;
767 ;
768 ;
769 ;
770 ;
771 ;
772 ;
773 ;
774 ;
775 ;
776 ;
777 ;
778 ;
779 ;
780 ;
781 ;
782 ;
783 ;
784 ;
785 ;
786 ;
787 ;
788 ;
789 ;
790 ;
791 ;

```

792 004500 004567 000754 JSR R5,READV ;SEE IF DRIVE IS READY
793 004504 004567 000754 BR R5 ;IF IT WAS READY
794 004504 004567 000510 JSR R5,NOTRDY ;ELSE GO CLEAR IT AND CHECK AGAIN
795 004512 022777 024024 4S: CMP #24024,RMDT ;RM02?
796 004529 001407 024024 175016 BEQ #24024,RMDT ;EXIT IF SO
797 004530 001403 024024 175016 CMP #24024,RMDT ;RM03?
798 004533 0012767 177777 173700 BEQ #24024,RMDT ;BRANCH IF SO
799 004540 0006267 177777 173700 MOV #1,MIXDV ;OTHERWISE SET THE FLAG
800 004544 0016661 0006267 177777 PC ;RETURN WITH A UNITNO READY
801 004544 0016661 0006267 177777 BITTAB: 1001
802 004544 0016661 0006267 177777 4004
803 004544 0016661 0006267 177777 20026
804 004544 0016661 0006267 177777 100100
805
806
807 004552 062767 000001 17361E PICKBK: ADD #1,BLK ;DO NEXT BLOCK(SECTOR)
808 004560 026777 17361E 077777 CMP BLK,#77777
809 004566 102403 000001 17360C BLD 1S ;
810 004570 012403 000001 17360C MOV #1,BLK ;GO BACK TO BLOCK 1
811 004576 005767 17363F 1S: RST MIXDV ;NOT RM03/RM02?
812 004602 100411 000252* BRANCH IF NOT
813 004604 012700 000252* MOV #BADSPTR,RO ;BAD SPOT TABLE
814 004610 012700 000252* MOV #16,R1 ;LOOK FOR 16 ENTRIES
815 004614 026777 17355F 2S: CMP BLK,(R0)+ ;IS THIS A BAD BLK?
816 004620 017554 026777 17355F BEQ #1,R1 ;IF YES, GO PICK A NEW ONE
817 004622 013301 026777 17355F DEC R0 ;COUNT A TABLE LOOK-UP
818 004624 013373 026777 17355F RNE 2S ;BR BACK IF MORE TO GO
819 004626 01207 026777 17355F PC 3S:
820
821 004630 01700 000444* CLRRB: MOV #RBUF,RO ;CLEAR RBUF BUFFER
822 004634 01701 173272 000444* MOV #RBUF52,R1 ;GET ITS ADDR AND SIZE
823 004640 00020 000444* CLRCOM: CLR (R0)+ ;CLEAR ANOTHER
824 004642 000301 000444* DEC R1 ;CLEAR ANOTHER
825 004644 000379 000444* BNE CLRCOM ;BR BACK TILL DONE
826 004646 000207 000444* RTS PC
827
828 004650 01701 000001 DROP: MOV #1,R1 ;INITIALIZE DROP PICKER
829 004654 01700 173534 000001 MOV UNITNO,RO ;GET THE DRIVE NUMBER
830 004660 001403 000001 1S: BEQ #2S ;IF DRIVE 0 GO DROP IT
831 004662 000301 000001 ASL R1 ;POINT TO NEXT DRIVE
832 004664 000379 000001 DEC R0 ;IS THIS THE ONE?
833 004666 000167 173514 2S: BNE 1S ;NO, LOCK AGAIN
834 004670 000167 173514 PC ;DROP THE DRIVE
835
836
837
838 004674 014426 000000* 000414* OTOAS,BEGIN,UNITNO,ADR1 ;CONVERT UNITNO TO ASCII AND
839 004676 014426 000000* 000414* STORE AT ADR1
840
841 004678 014426 000000* 000414* ;*****
842 004680 014426 000000* 000414* PC ; RETURN
843
844
845
846
847 004706 005067 173530 ERRORS: CLR BADSEC ;
848 004712 005777 174622 RST #RMCS1 ; ATTENTION OR ERROR?
849 004716 100402 174622 RST #R5 ; BR IF YES
850 004720 005728 174622 RST #R5+ ; BUMP FOR GOOD RETURN

```

```

849 004722 005265 174566 1S: RTS R5 ;AND RETURN
850 004724 015900 174566 1S: CLR RO
851 004732 012160 000312* 22S: MOV #RMCS1,R1 ;ADDRESS OF RMCS1
852 004736 005770 000312* TST (R0),S(R0) ;READ AND STORE
853 004740 002770 000344 22S: CMP #46,R0 ;ARE CONTROLLER AND RM03
854 004744 001372 000344 BNE #22S ;REGISTERS
855 004746 005767 000666* JSR PC,ERRSUB1 ; LOAD ERROR INFORMATION
856 004750 005767 17334F 000666* BIT #40 ; IS THIS A DATA LATE ERROR?
857 004756 100412 000666* BPL #40 ; NO
858 004760 005267 000666* 173024* BNC DTCNT ;ADD 1 TO DATA LATE COUNTER
859 004764 005767 000666* 173024* BIT #BIT2,R1 ;TYPE ERROR AND COUNT IT?
860 004772 001163 000000* 006730* BNE HS ;NO
861 004776 104403 000000* 006730* MSGNS,BEGIN,DI,ERP ;ASCII MESSAGE CALL WITH COMMON HEADER
862 004780 005767 000000* 006730* BR 11S ;CONT
863 004784 005767 17330C 11S: BNE #BIT13,C ;MABUSBUS CONTROL PARITY ERROR?
864 004788 001035 17330C 11S: BNE #BIT8,S+10 ;YES
865 004792 002767 17330C 11S: BIT #BIT8,S+10 ;MABUSBUS DATA PARITY ERROR?
866 004796 001035 17326 11S: BNE #BIT14,S ;YES
867 004800 005767 17326 11S: BIT #BIT14,S ;TRANSFER ERROR?
868 004804 001035 17326 11S: BNE #BIT14,S+12 ;YES
869 004808 005767 17326 11S: BIT #BIT14,S+12 ;ANY DRIVE ERRORS?
870 004812 001035 17326 11S: BNE #BIT14,S+12 ;YES
871 004816 005767 17326 11S: TST #S+16 ;ANY ATTENTIONS ACTIVE?
872 004820 001035 17326 11S: BNE #S+16 ;YES, CONTINUE
873 004824 005767 17326 11S: CLR #ERRTYE ;UNKNOWN ERROR
874
875 005056 104405 000000* 001444* ;*****
876 005064 000444 000000* 001444* ;*****
877 005066 000444 000000* 001444* BR #S ; SPECIAL CONDITION SET BUT NO REASON FOUND
878
879 005068 002767 100000 17326 2S: BR #S ; RETURN
880 005072 001045 100000 17326 2S: BIT #BIT15,S+42 ;IS A BAD SPOT *****?
881 005076 001403 100000 17326 2S: BSENS,BEGIN,TRERR ;BRANCH IF GO ****
882 005080 001407 100000 17326 2S: BR #S ;ASCII MESSAGE CALL WITH COMMON HEADER
883
884 005104 104403 100000* 006714* 3S: ; GO DUMP REGISTERS
885 005108 000403 100000* 006714* 3S: MSGNS,BEGIN,MCERRR ;ASCII MESSAGE CALL WITH COMMON HEADER
886 005112 000403 100000* 006714* 3S: BR #S ; GO DUMP REGISTERS
887
888 005116 104403 000000* 00672 4S: MSGNS,BEGIN,MDPERR ;ASCII MESSAGE CALL WITH COMMON HEADER
889 005120 001004 173200 00672 4S: TST #S+16 ;ANY ATTENTIONS ACTIVE?
890 005124 001004 173172 174374 4S: BNE #S ;YES, CONTINUE
891 005128 005777 173172 174374 4S: MOV #S+16,RMAS ;KILL ATT BITS
892 005132 001670 174374 6S: MOV #RDR,RO ;SAVE ADDRESS OF DATA BUFFER
893 005136 005767 173150 174374 6S: BNE #S+10 ;CAN DATA BUFFER BE READ?
894 005140 001004 000430 17436 7S: MOV #ZERO,RMDR ;NO, LOAD ADDRESS OF ZERO
895 005144 012767 000430 172716 7S: MOV #1,ERRTPY ;DATA ERROR
896 005148 012767 000430 172716 7S: CLR #ERRTYE ;*****
897 005152 104405 000000* 001444* ;*****
898 005156 010067 174374 8S: ;*****
899 005160 004767 000000* 001444* ;*****
900 005164 001004 174374 8S: MOV RO,RMDR ;RESTORE DATA BUFFER ADDRESS
901 005168 004767 000000* 001444* JSR PC,NOTRDY ;GO CLEAR OUT ERRORS
902 005172 005728 174374 9S: RTS R5 ; ERRORS DETECTED, RETURN
903 005176 005728 174374 9S: TST #R5+ ;IF BAD SCPT DON'T REPORT

```

```

904 005212 012767 177777 173222 MOV #1,RADSECT ;SET BAD SECTOR DETECT FLAG
905 005220 010205 RTS #5 ;EXIT***
906
907 005222 012767 077777 173222 NOTRDY: MOV #77777,CLK ;SET THE TIMER
908 005230 016777 173160 174274 4S: MOV UNITNO,@RMCS2 ;MOVE DRIVE NUMBER TO RM11/70
909 005234 032777 000000 174274 MOV #11,@RMCS1 ;ISSUE DRIVE CLEAR COMMAND
910 005238 032777 000000 174274 BIT #11,@RMCS1 ;DO I HAVE THE DRIVE DVA?
911 005242 032777 000000 174274 BNC #0 ;NO
912 005252 031406 000000 174274 MOV #0-(SP),RO ;JUST WASTE A LITTLE TIME
913 005254 014046 000000 174274 MOV #0,RO ;STILL GOT IT?
914 005256 012600 000000 174274 BIT #11,@RMCS1 ;JUST WASTE A LITTLE TIME
915 005260 004000 174274 BNE #2S ;STILL GOT IT?
916
917 005270 104407 000000 174274 6S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
918 005274 104407 000000 174274 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
919 005278 104407 000000 174274 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
920 005282 104407 000000 174274 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
921 005286 032777 000000 174274 BIT #11,@RMCS2 ;NEED SET?
922 005290 032777 000000 174274 BNE #1 ;YES
923 005310 032777 000000 174274 DEC #1 ;COUNT # OF TRIES
924 005314 032777 000000 174274 BNE #4S ;COUNT DONE YET
925 005318 032777 000000 174274 MSGNS,BEGIN,NOT ;ASCII MESSAGE CALL WITH COMMON HEADER
926 005322 032777 000000 174274 BR #7S ;COULD NOT GET DRIVE
927 005326 032777 000000 174274 JSR #5,CLEAR ;SET THE CONTROLLER AND DRIVE
928 005330 032777 000000 174274 JSP #5,READY ;IS DRIVE READY?
929 005334 032777 000000 174274 BNE #5S ;YES, CONTINUE
930 005338 032777 000000 174274 JSR #5,PC,ERSUR1 ;LOAD ERROR INFORMATION
931 005342 032777 000000 174274 CLD #0 ;MOVE DRIVE REG INTO TABLE
932 005346 032777 000000 174274 MOV #RMCS1,R1
933 005350 032777 000000 174274 TST (R1),S(RO) ;READ AND STORE ALL
934 005354 032777 000000 174274 CMP #46,RO ;RM11/70 AND RM03
935 005358 032777 000000 174274 BNE #22S ;REGISTERS
936
937 005400 104407 000000 172502 MOV #4,ERRIYP ;DRIVE NOT READY
938 005404 104407 000000 172502 HPER$,BEGIN,TABLE ;DRIVE NOT READY
939 005408 104407 000000 172502 ***** ;*****
940 005412 104407 000000 172502 MOV #13,@RMCS1 ;RELEASE DRIVE
941 005416 104407 000000 172502 JSR #5,DROP ;NO, DROP THE DRIVE
942 005420 104407 000000 172502 MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
943 005424 104407 000000 172502 MOV #0,FLAG ;IN PROCESS OF RETRIEVING BAD SPOT
944 005428 104407 000000 172502 RNE #6 ;BRANCH IF NOT
945 005432 104407 000000 172502 JMP #5,RESX ;IF SO, BRANCH BACK
946 005436 104407 000000 172502 LOOP2
947 005440 104407 000000 172502 RTS PC ;RETURN
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 ;
1001 ;
1002 ;
1003 ;
1004 ;
1005 ;
1006 ;
1007 ;
1008 ;
1009 ;
1010 ;
1011 ;
1012 ;
1013 ;
1014 ;
1015 ;

```

```

960 005466 012777 172730 174042 MOV #11,@RMCS1 ;CLEAR DRIVE AND GRASP THE DRIVE
961 005470 012777 172730 174042 MOV #RMCS1,RO ;RO = BASE ADDRESS OF RM CONTROLLER
962 005474 012777 172730 174042 BIT #11,RO ;DVA BIT = 1?
963 005478 012777 172730 174042 BNC #0 ;NO
964 005482 012777 172730 174042 MOV @R0,R0 ;SAVE STATUS IN RO
965 005486 012777 172730 174042 TSTR #1,R0 ;DRIVE READY?
966 005490 012777 172730 174042 RPL #1,R0 ;NO
967 005494 012777 172730 174042 BIT #6,RO ;VOLUME VALID?
968 005498 012777 172730 174042 BNC #0 ;NO
969 005502 012777 172730 174042 MOV #1,RO ;DRIVE PRESENT?
970 005506 012777 172730 174042 BIT #1,RO ;WRITE LOCKED?
971 005510 012777 172730 174042 BNE #1 ;YES
972 005514 012777 172730 174042 BIT #12,RO ;MEDIUM ON LINE?
973 005518 012777 172730 174042 BNC #0 ;NO
974 005522 012777 172730 174042 BIT #14,RO ;ANY ERRORS?
975 005526 012777 172730 174042 BNE #1 ;YES
976 005530 012777 172730 174042 TST #0,RO ;ATTENTION SET?
977 005534 012777 172730 174042 BNE #1 ;YES
978 005538 012777 172730 174042 BIT #11,@RMCS1 ;DVA SET?
979 005542 012777 172730 174042 BNC #0 ;BR IF NOT
980 005546 012777 172730 174042 BIT #12,@RMCS2 ;DVA BIT SET?
981 005550 012777 172730 174042 BNE #0 ;YES
982 005554 012777 172730 174042 RTS #5 ;RETURN READY
983 005558 012777 172730 174042 TST #5,RO ;SKIP INSTRUCTION FOLLOWING CALL
984 005562 012777 172730 174042 RTS #5 ;RETURN AS NOT READY
985
986 ;
987 ;
988 ;
989 ;
990 ;
991 ;
992 ;
993 ;
994 ;
995 ;
996 ;
997 ;
998 ;
999 ;
1000 ;
1001 ;
1002 ;
1003 ;
1004 ;
1005 ;
1006 ;
1007 ;
1008 ;
1009 ;
1010 ;
1011 ;
1012 ;
1013 ;
1014 ;
1015 ;

```

1016	005736	012767	000004	172142					
1017						MOV	#4,ERRTY	CONTROLLER NOT READY	
1018	005744	014405	000000	000000		RDERS	BEGIN,NDI	CONTROLLER NOT READY	
1019						ENDS	BEGIN		
1020	005752	013410	000000			MOV	RRNAS,-(SP)	CLEAR ALL BITS	
1021	005756	013746	173554		2S:	MOV	(SP)+,RRNAS	IN THE SUMMARY REGISTER	
1022	005766	012207	173550			RTS	PC	RETURN	
1023									
1024									
1025									
1026									
1027	005776	016700	172312		SETUP:	MOV	ADDR,RO	GET DEVICE ADDRESS	
1028	005774	012780	173550			MOV	RO,RRCS	GENERATE REGISTER ADDRESSES	
1029	006000	012780	173550			ADD	#2,RO		
1030	006004	016674	173512			MOV	RO,RRMC		
1031	006010	062700	000002			ADD	#2,RO		
1032	006014	010067	173504			MOV	RO,RRBA		
1033	006020	012700	173470			ADD	#2,RO		
1034	006024	012700	173470			MOV	RO,RRDA		
1035	006030	010067	173470			ADD	#2,RO		
1036	006034	010067	173470			MOV	RO,RRCS2		
1037	006040	016067	173462			ADD	#2,RO		
1038	006044	016067	173462			MOV	RO,RRDS		
1039	006050	012700	000002			ADD	#2,RO		
1040	006054	012700	000002			MOV	RO,RRER1		
1041	006060	012700	173442			ADD	#2,RO		
1042	006064	012700	173442			MOV	RO,RRAS		
1043	006070	012700	000002			ADD	#2,RO		
1044	006074	012700	173442			MOV	RO,RRLA		
1045	006080	012700	173435			ADD	#2,RO		
1046	006104	012700	000002			MOV	RO,RRDB		
1047	006110	010067	173424			ADD	#2,RO		
1048	006114	010067	173424			MOV	RO,RRMR1		
1049	006120	012700	000002			ADD	#2,RO		
1050	006124	010067	173424			MOV	RO,RRMT		
1051	006130	010067	000002			ADD	#2,RO		
1052	006134	010067	173412			MOV	RO,RRSN		
1053	006140	016067	173405			ADD	#2,RO		
1054	006144	016067	173405			MOV	RO,RRMF		
1055	006150	012700	000002			ADD	#2,RO		
1056	006154	012700	173371			MOV	RO,RRDC		
1057	006160	012700	173371			ADD	#2,RO		
1058	006164	012700	173371			MOV	RO,RRMR		
1059	006170	012700	000002			ADD	#2,RO		
1060	006174	012700	173362			MOV	RO,RRMR2		
1061	006180	012700	173362			ADD	#2,RO		
1062	006184	012700	000002			MOV	RO,RRMR2		
1063	006190	012700	173344			ADD	#2,RO		
1064	006194	012700	173344			MOV	RO,RRMC1		
1065	006200	012700	000002			ADD	#2,RO		
1066	006204	012700	173332			MOV	RO,RRMC2		
1067	006210	012700	173332			ADD	#2,RO		
1068	006214	012700	000001			RTS	PC		
1069	006220	012700	000001	172142	SETUP2:	BIT	#1,FLAG	FIRST TIME THROUGH?	
1070	006224	012700	000001			MOV	RR,IP	NO	
1071	006228	012700	000004			ADD	ADDR,RO	BASE ADDRESS OF RR11/RR70	
							#46,RO	INDEX VALUE	

1072	006252	032767	00100	171576		BIT	#ADDR22,RES1	11/70 SUPPORT?	
1073	006256	041908	000002			BEG		NO	
1074	006260	032767	10000	17152		ADD	#2,RO		
1075	006264	001402				RTG	#BIT15,SR1	SPECIFY 32 REGISTER ON RR70 ?	
1076	006270	062700	000024			ADD	#4,RO	BRANCH IF NOT	
1077	006274	062700	173262		3S:	MOV	RO,RRBAE	OTHERWISE ADJUST THE RRBAE ADDRESS	
1078	006278	062700	000002			ADD	#2,RO		
1079	006284	012700	000002			MOV	RO,RRCS3		
1080	006288	012700	171426		1S:	MOV	VECTOR,RO	GET VECTOR ADDRESS	
1081	006294	012700	004072			MOV	VECTOR,(RO)+	SET POINTER JUST IN CASE	
1082	006300	012700	004072			MOV	RR1,(RO)	SET PRICITY	
1083	006304	012700	171466		2S:	RTS	PC	RETURN	
1084	006308	012700	171466						
1085	006314	012700	000000						
1086	006320	012700	04710	04710	MES1:	.ASCIZ	" TRANSFER ERROR"		
1087	006324	012700	02004	02004					
1088	006328	012700	04752	02252					
1089	006334	012700	04644	05150	MES2:	.ASCIZ	" MASSBUS PARITY ERROR"		
1090	006338	012700	05152	02004					
1091	006344	012700	04452	04452					
1092	006348	012700	04452	04452					
1093	006354	012700	04644	05150	MES3:	.ASCIZ	" MASSBUS DATA PARITY ERROR"		
1094	006358	012700	05152	02004					
1095	006364	012700	04452	04452					
1096	006368	012700	04452	04452					
1097	006374	012700	04452	04452	MES4:	.ASCIZ	" DRIVE "		
1098	006378	012700	04452	04452	MES5:	.ASCIZ	" DROPPED"		
1099	006384	012700	04452	04452	MES6:	.ASCIZ	" RETPY EXCEEDED"		
1100	006388	012700	04452	04452					
1101	006394	012700	04452	04452					
1102	006398	012700	04452	04452	MES10:	.ASCIZ	" DATA LATE ERROR"		
1103	006404	012700	04452	04452					
1104	006408	012700	04452	04452					
1105	006414	012700	04452	04452	MES11:	.ASCIZ	" DRIVE NOT READY"		
1106	006418	012700	04452	04452					
1107	006424	012700	04452	04452	MES12:	.ASCIZ	" COULD NOT GET DRIVE"		
1108	006428	012700	04452	04452					
1109	006434	012700	04452	04452	MES14:	.ASCIZ	" OTHER PORT NOT UPDATING DRIVE"		
1110	006438	012700	04452	04452					
1111	006444	012700	04452	04452					
1112	006448	012700	04452	04452					
1113	006454	012700	04452	04452					
1114	006458	012700	04452	04452	MES15:	.ASCIZ	"%GVE: 16 BAD BLOCKS DETECTED"		
1115	006464	012700	04452	04452					
1116	006468	012700	04452	04452					
1117	006474	012700	04452	04452					
1118	006478	012700	04452	04452					
1119	006484	012700	04452	04452					
1120	006488	012700	04452	04452					
1121	006494	012700	04452	04452					
1122	006498	012700	04452	04452					
1123	006504	012700	04452	04452					
1124	006508	012700	04452	04452					
1125	006514	012700	04452	04452					
1126	006518	012700	04452	04452					
1127	006524	012700	04452	04452					







RMDT	001546R	355#	795																
RMC1	001546R	355#	797	1050*															
RMC2	001546R	355#																	
RMR1	0015334R	355#																	
RMR2	0015566R	355#																	
RMR3	0015566R	355#																	
RMR4	0015566R	355#																	
RMR5	0015566R	355#																	
RMR6	0015566R	355#																	
RMR7	0015566R	355#																	
RMR8	0015566R	355#																	
RMR9	0015566R	355#																	
RMR10	0015566R	355#																	
RMR11	0015566R	355#																	
RMSN	0015566R	355#	1054*																
RMC	0015566R	355#	599*	615*	631*	647*	663*	1030*											
RST	0015566R	355#																	
RSTR1	0015566R	355#																	
SADR	000102R	355#	448	450	452*														
SETUP	005270R	355#	450	450	452*														
SOP	006270R	355#	326	326	327	328	329	330	331	332	333								
SOPERS	104407R	355#	1027*	888	888	890	893	934*	341	329	330	331	332	333					
SOPPAS	000046R	355#	1068*																
SPOINT	000032R	355#																	
SISIZ	000040R	355#	374	734	945														
SSR1	000020R	355#	385	474	730	850	1075												
SSR2	000022R	355#																	
SSR3	000022R	355#																	
SSR4	000024R	355#																	
SSART	001604R	355#	370*	451															
SSRG	000026R	355#																	
SSVR1	000024R	355#																	
SSVR2	000066R	355#																	
SSVR3	000070R	355#																	
SSVR4	000072R	355#																	
SSVR5	000076R	355#																	
SSVR6	000076R	355#																	
SYSCNT	000052P	355#																	
TABLE	001444R	355#	875	898	940														
TIMER	000426P	355#	496*	498*	521*	541*													
TOUT	006740R	355#	1146*																
TRACK	000404R	355#	747*	760*	764	774*	778												
TERR	006710P	355#	1133*																
TRPDFD	006022	355#																	
TRY	000436P	355#	395*	461*	725*	726	693	711	740*	783*	784	787	829	838					
UNITNO	000414P	355#	377*	387*	455*	648													
VECTOP	000010R	355#	909																
WASADR	000104R	355#	208#	1081															
WBFPA	000134R	355#	244#	995*															
WBFPPA	000134R	355#	584	604															
WBFRQ	000140R	355#	584	600															
WBFSZ	000142R	355#	258#																
WDR	000142R	355#	581	597															
WDTO	000114R	355#	248#	370*															

WRITE	003452P	555	643#	644															
WRITE	003166P	485	595#	596															
WRITE	003072P	481	597#	598															
WRITE	003356P	490#	627#	628															
WRTOP	003546P	500#	627#	659#	660														
WRTOP	003546P	500#	627#	659#	660														
XBLK	001600R	368#																	
XFERAD	001574P	366#																	
XFERCT	001574P	367#																	
XFLAG	000084R	207#																	
XNEM	000376P	207#																	
XMES15	000651R	1126#	593*	609*	625*	641*	657*	673*	697*	698*	699*	700*	706*	707					
ZERO	000436R	313#	1160*																
ZERO	000436R	313#	401	483	488	518	737	947	1132*	1162*	1165*								

ARS. 000000 000  
007002 001

ERRORS DETECTED: 0  
DEFAULT GLOBALS REPAIRED: 0

XRMRCO XRMRCO/SOL/CRF:SVN=DDXCOM,XRMRCO  
RUN-TIME: 23.5 SECONDS  
RUN-TIME RATIO: 23/5=4.7  
CORE USED: 7K (13 PAGES)