

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E971D-MC
PRODUCT NAME: CXRMADO RH11,70/RM03,02 S P M
PRODUCT DATE: FEB 1979
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) 1978,1979 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RMA IS AN IOMOD THAT EXERCISES RM03/RM02 DISK DRIVES ON AN RH11/RH70 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RM03/RM02 WITH AN RH11/RH70 CONTROLLER

STORAGE:: RMA REQUIRES:

1. DECIMAL WORDS: 1825
2. OCTAL WORDS: 03441
3. OCTAL BYTES: 7102

3. PASS DEFINITION

ONE PASS OF THE RMA MODULE CONSISTS OF 1300 CYCLES OF THE BASIC TEST SEQUENCE (WRITE, 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 RMA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 0.9 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. 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

- SR1 BIT0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR
IS ASSUMED AND THE DRIVE IS DROPPED
- SR1 BIT0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED, THE FUNCTION IS ABORTED AND
THE TESTING CONTINUES
- SR1 BIT2 SET(1):
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT
- SR1 BIT2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND COUNT THEM
- SR1 BIT5 CLEAR (0) ;NORMAL FOR RM03 PACKS THAT ARE FORMATED
;FOR 16 BIT MODE (PDP-11)
- SR1 BIT5 SET (1) ;FOR RM03 18 BIT FORMATED PACKS
- SR1 BIT15 SET (1) ;32 REGISTERS ON RH70
- SR1 BIT15 CLEAR (0) ;22 REGISTERS ON RH70

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 20 RH11/RM03 REGISTERS IN THE FOLLOWING ORDER:
RMCS1 RMWC RMBA RMDA RMCS2 RMDS RMER1 RMAS
RMLA RMDB RMMR1 RMDT RMSN RMOF RMDC RMHR
RMMR2 RMER2 RMEC1 RMEC2 RMBAE RMCS3 XFER CNT

10. BAD SPOTS

- A. LOCATION 256 THROUGH 452 CONTAIN ROOM FOR 32. BAD SPOTS.
EACH BAD SPOT TAKES TWO WORDS.
THE FIRST WORD SPECIFIES THE CYLINDER ADDRESS.
THE LOW BYTE OF THE SECOND WORD SPECIFIES THE SECTOR ADDRESS.
THE HIGH BYTE OF THE SECOND WORD SPECIFIES THE TRACK ADDRESS.
- B. THE RMA MODULE DOES NOT ACCESS THESE SPOTS ON ANY OF THE DRIVE UNDER TEST.
- C. THE BAD SPOT FILE IS RETRIEVED FROM ALL DRIVES ASSIGNED IN THE BIT MAP "DVID1".
(CYLINDER 822, TRACK 4, SECTOR 0 - MANUFACTUR BAD SPOT FILE)
(CYLINDER 822, TRACK 4, SECTOR 12 - USER BAD SPOT FILE)
LOCATION 1760 MUST BE SET TO 10 TO SEARCH USER BAD SPOT FILE

```

000000* 000000* IOMODX <RMAD > 176700,254,5,0,0,1300,144,BUFIN,256,,1024.
000000* 000000* MODULE 150000, RMAD 176700,254,5,0,0,1300,144,BUFIN,256,,1024.
; TITLE RMAD DEC/X11 SYSTEM EXERCISER MODULE
DDXCOM VERSION 6 23-MAY-78
;*****LIST*****
000000* BEGIN:
000000* 046522 042101 040 MODNAM: ASCII /RMAD / ;MODULE NAME
000005* 000 XFLAG: BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006* 176700 ADDR: 176700+0 ;1ST DEVICE ADDR
000010* 000254 ;VECTOR: 254+0 ;1ST DEVICE VECTOR.
000012* 240 BR1: BYTE PRTYS+0 ;1ST BR LEVEL
000013* 000 BR2: BYTE PRTY0+0 ;2ND BR LEVEL
000014* 000001 DVID1: 0+1 ;DEVICE INDICATOR 1.
000016* 000000 SR1: OPEN ;SWITCH REGISTER 1
000018* 000000 SR2: OPEN ;SWITCH REGISTER 2
000022* 000000 SR3: OPEN ;SWITCH REGISTER 3
000024* 000000 SR4: OPEN ;SWITCH REGISTER 4
;*****LIST*****
000026* 150000 STAT: 150000 ;STATUS WORD.
000030* 001662 INIT: START ;MODULE START ADDR.
000032* 000252 SPOINT: MODDSP ;MODULE STACK POINTER.
000034* 000000 PASCNT: 0 ;PASS COUNTER
000036* 002472 ICONF: 1300. ;# OF ITERATIONS PER PASS=1300.
000040* 000000 ICOUNT: 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 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050* 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052* 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054* 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056* 000000 CONFIG: ;RESERVED FOR MONITOR USE
000058* 000000 RPS1: 0 ;RESERVED FOR MONITOR USE
000060* 000000 RPS2: 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: ;ADDR OF GOOD DATA, OR
000104* 000000 ACSR: OPEN ;CONTENTS OF CSR.
000106* 000000 BSADR: ;ADDR OF BAD DATA, OR
000108* 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
000110* 000000 ERRTP: ;TYPE OF ERROR
000112* 000000 ASB: OPEN ;EXPECTED DATA.
000114* 000000 AWAS: OPEN ;ACTUAL DATA.
000116* 002532 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
000118* 000000 WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
000120* 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000122* 000144 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000124* 000600 RBUFFA: BUFIN ;MODULE IDENTIFICATION NUMBER=144
000126* 000000 RBUFFP: OPEN ;READ BUFFER VIRTUAL ADDRESS
;READ BUFFER PHYSICAL ADDRESS

```

```

000130* 000000 RBUFFEA: OPEN ;READ BUFFER EA BITS
000132* 000400 RBUFFSZ: 256. ;SIZE OF THE READ BUFFER
000134* 000000 WBUFFP: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
000136* 000000 WRUFFEA: OPEN ;WRITE BUFFER EA BITS
000140* 002000 WBUFFRQ: 1024. ;WRITE BUFFER SIZE REQUESTED
000142* 000000 WBUFFSZ: OPEN ;WRITE BUFFER SIZE AVAILABLE
000144* 000000 CDWRCT: OPEN ;CDATA/DATCK ERROR COUNT
000146* 000000 CDWDC: OPEN ;CDATA/DATCK WORD COUNT
000150* 000000 FREE: OPEN ;RESERVED FOR FUTURE USE
;MODULE STACK STARTS HERE.
;REPT SPSIZ
;LIST 0
;WORD 0
;LIST
;ENDR
000252* MODSP:
;*****LIST*****

```


360
361
362 000552* 000000
363 000554* 000000
364 000556* 000000
365 000560* 000000
366 000562* 000000
367 000564* 000000
368 000566* 000000
369 000570* 000000
370 000572* 000000
371 000574* 000000
372 000576* 000000
373 000600* 000400

ZFRD: 0
DSKADR: 0
DVIC: 0
DRIVE: 0
BLKSAV: 0
TRUP: 0
WCNT1: 0
WCNT2: 0
UNIN0: 0
MIXDV: 0
MOD1: 0
BUFIN: .BLKW 256.

;MIXDV=-1, IF NOT A RM03 OR RM02

374 001600* 000000
375 001600* 000000
376 001602* 000000
377 001604* 000000
378 001606* 000000
379 001610* 000000
380 001612* 000000
381 001614* 000000
382 001616* 000000
383 001620* 000000
384 001622* 000000
385 001624* 000000
386 001626* 000000
387 001630* 000000
388 001632* 000000
389 001634* 000000
390 001636* 000000
391 001640* 000000
392 001642* 000000
393 001644* 000000
394 001646* 000000
395 001650* 000000
396 001652* 000000
397 001654* 000500
398 001656* 000470
399 001660* 177777

TABLE:
RMCS1: 0
RMWC: 0
RMBA: 0
RMDA: 0
RMC2: 0
RMDS: 0
RMER1: 0
RMAS: 0
RMLA: 0
RMDR: 0
RMMR1: 0
RMDT: 0
RMSN: 0
RMOP: 0
RMDC: 0
RMHR: 0
RMMR2: 0
RMER2: 0
RMEC1: 0
RMEC2: 0
RMBAE: 0
RMCS3: 0
XPERAD: PERADR
XPERCT: CNT
177777

```

XRMADO.P11 12-DEC-78 16:17
400 001662 012767 002000 176226 START: MOV #1024, WDFR ;1024. WORDS FROM MEM/ITERATION
401 001670 012767 000400 176218 MOV #256, WDFD ;256 WORDS TO MEM/ITERATION
402 001676 012767 000003 176214 MOV #3, INTR ;3 INTERRUPTS/ITERATION
403 001704 005067 176560 CLR CNT ;ZERO END OF PASS TESTER
404 001710 005067 176540 CLR DLT CNT ;CLEAR DATA LATE ERROR COUNTER
405 001714 012767 010000 176654 MOV #112, MOD1 ;SETUP FOR 16 BIT MODE (NORMAL)
406 001722 012767 000526 MOV #16, R0 ;GET TABLE OF VALUES FOR 16 BIT MODF
407 001726 032767 000040 176062 BIT #R15, SR1 ;16 BIT MODE?
408 001734 001404 BEQ ZS ;YES
409 001736 005067 CLR MOD1 ;NOT 16 BIT MODE. CLEAR FMT BIT
410 001742 012700 MOV #18, R0 ;GET TABLE OF VALUES FOR 18 BIT MODE
411 001746 012701 000514 2S: MOV #MBLKRV, R1 ;BEGIN OF TABLE ENTRIES
412 001752 012702 MOV #5, R2 ;NO. OF ENTRIES
413 001756 012021 3S: MOV (R0)+, (R1)+ ;STORE AN ENTRY
414 001760 005302 DEC R2 ;COUNT IT
415 001762 001375 BNE ZS ;DO MORE
416 001764 012767 000007 176504 1S: MOV #7, ONCEE ;SET ONE TIME ONLY FLAGS
417 001772 005102 CLR RB ;CLEAR FLAGS
418 001776 016767 176514 176552 MOV DVID1, DVICE ;GET DRIVE INDICATOR
419 002004 122737 000016 000041 CWPB #16, #41 ;IF RM IS LOAD MEDIUM THEN
420 002012 001021 BNE BEGIN ;BEGIN
421 002014 113700 MOVB #40, R0 ;GET LOAD-DRIVE NUMBER
422 002020 012701 MOV #1, R1 ;INITIALIZE DRIVE POINTER
423 002024 105700 TSTB R0 ;WHILE NOT POINTING AT LOAD-DRIVE DO
424 002030 001403 BNE BEGIN ;BEGIN
425 002032 105300 ASL R1 ;POINT TO NEXT DRIVE
426 002034 000773 DECR R0 ;COUNT SHIFTS
427 002036 000773 BR 10S ;END
428 002036 130167 176514 12S: BIT #1, DVICE ;IF LOAD-DRIVE SELECTED THEN
429 002042 001403 BNE BEGIN ;BEGIN
430 002044 140167 176506 BICR #1, DVICE ;DROP THE DEVICE
431 002050 104403 000000 006746 MSGNS, BEGIN, LDRIVE ;ASCII MESSAGE CALL WITH COMMON HEADER
432 END
433 002056 TST DVICE ;END
434 002062 005767 176474 13S: BNE DAOST ;NO DRIVES ARE SELECTED THEN
435 002064 000102 JMP FINI ;BEGIN
436 002064 000167 000752 ;DROP MODULE
437 ;END
438 002070 016767 176462 DAOST: MOV DVICE, DRIVE ;ALSO SAVE IT IN DRIVE
439 002076 012700 MOV SPOIN, R0 ;RESTORE STACK POINTER
440 002109 012767 175730 MOV #-1, BLK1 ;INITIALIZE BLOCK COUNTER
441 002110 012767 177777 176454 MOV #-1, UNITNO ;INITIALIZE DEVICE COUNTER
442 002116 004767 003602 JSR PC, SETUP ;SET UP CONTROLLER REGISTER
443 002122 004767 003504 JSR PC, RESET ;SET CONTROLLER AND DRIVES
444 002126 004767 003070 JSR PC, DRBIT ;FOUND THE FIRST DRIVE AVAILABLE
445 002132 132767 000010 004740 BITB #BIT3, FLAG ;NO DRIVE?
446 002140 001402 BEQ +6 ;DRIVE AVAILABLE
447 002142 000167 000674 JMP FINI ;DROP THE MODULE
448 002150 004567 003330 JSR R, READY ;DEVICE READY?
449 002154 004767 003120 BR CT ;YES
450 002154 004767 003120 JSR PC, NOTRDY ;NO, TRY NOT READY ROUTINE
451 002160 CT: ;
452 002160 005767 176372 TST DVICE ;DROP THE MODULE?
453 002160 000102 BNE BDFL ;NO
454 002164 001002 000650 JMP FINI ;BRANCH IF DRIVE AVAILABLE
455 ;YES DROP MODULE

```

```

XRMADO.P11 12-DEC-78 16:17
456 002172 012701 000254 1S: MOV #BADSP, R1 ;TABLE ADDRESS
457 002176 012702 000100 MOV #64, R2 ;32 WORDS
458 002202 012721 177777 2S: MOV #-1, (R1)+ ;RESET TO -1
459 002206 005302 DEC R2 ;DECREMENT ONE WORD
460 002210 001374 BNE ZS ;BRANCH IF NOT DONE
461 002214 012767 001466 176242 BADFL: MOVB #22, CVL ;RETRIEVE THE BAD SPOT FILE
462 002216 012767 000004 176327 MOVB #4, DSKADR+1 ;FROM THE PACK CVL-822, TRK-4
463 002222 012767 000000 176320 MOVB #0, DSKADR ;SEC-0 (SEC-12 FOR USER BAD SPOT FILE)
464 002234 005767 176336 TST MOD1 ;IN 16 BIT MODE?
465 002240 005302 BNE ZS ;BRANCH IF YES
466 002242 005302 LDC DSRADR ;OTHERWISE START FROM SECTOR 1
467 002246 016767 176320 177334 MOV UNITNO, QRMCS2 ;LOAD DRIVE NUMBER
468 002254 012777 000000 177352 MOV #0, QRMDC ;SET TO CVL 0
469 002262 012777 000000 177316 MOV #0, QRMDA ;SET TO TRK 0, SEC 0 AT START TIME
470 002270 BADFL2: ;
471 002276 104415 000000 000124 GETPAS, BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
472 002276 004767 002636 JSR PC, RM03CK ;CHECK IF RM03 OR RM02
473 002302 005767 176266 TST #IXD ;NOT THEM?
474 002306 100455 BMI 4S ;BRANCH IF NOT
475 002310 016767 175616 176252 MOV RBUFVSZ, WCNT2 ;WORD CTR
476 002316 005467 176246 NEG WCNT2 ;2'S COMPLEMENT
477 002322 004567 001104 JSR R5, READ ;READ ONE SECTOR AT A TIME
478 002326 000167 000150 JMP RETRX ;BRANCH IF ERROR
479 002332 012701 000600 MOV #BUPIN, R1 ;BUFFER ADDRESS
480 002336 012702 000254 MOV #BADSP, R2 ;TABLE ADDRESS
481 002342 012703 000200 MOV #B2, R3 ;MAX 16 BAD SECTORS, EACH SECTOR DEFINED BY TWO
482 002346 060203 ADD R2, R3 ;LAST ADDRESS
483 002350 012704 000400 MOV #256, R4 ;END OF BAD SPOT FILE
484 002354 060104 ADD R1, R4 ;LAST ADDRESS OF BAD SPOT FILE
485 002356 005701 000010 ADD #10, R1 ;BAD SECTOR STARTS AT 5TH WORD
486 002362 022711 177777 CWPB #10, (R1) ;TABLE IS EMPTY?
487 002366 001425 BEQ 4S ;BRANCH IF IT IS
488 002370 022712 CMP #1, (R2) ;ENTRY FOR BAD SPOT TABLE?
489 002374 001006 BNE ZS ;BRANCH IF NOT
490 002376 011112 MOV (R1), (R2) ;LOAD CYLINDER ADDRESS
491 002400 016767 000002 000002 2(R1), 2(R2) ;LOAD THE TRACK AND SECTOR ADDRESSES
492 002406 062701 000004 ADD #4, R1 ;ADJUST POINTER
493 002412 062702 ADD #4, R2 ;ADJUST POINTER
494 002416 020104 CMP R1, R4 ;END OF TABLE?
495 002420 030103 BHS 4S, R3 ;BRANCH IF SO
496 002422 030203 CMP R2, R3 ;OVER 32 BAD SECTORS RECORDED?
497 002424 103001 BHS 3S ;BRANCH IF SO
498 002426 000755 BR 1S ;LOOP BACK
499 002430 104403 007044 000000 3S: MSGNS, MES14, BEGIN ;MESSAGE: OVER 32 BAD SECTORS
500 002436 000167 000400 JMP FINI ;DROP THE MODULE
501 002436 004767 00554 4S: JSR PC, FDNIT ;FOUND NEXT DRIVE
502 002446 132767 000010 004424 BITB #BIT3, FLAG ;DRIVE AVAILABLE?
503 002454 001006 BNE 5S ;BRANCH IF NOT
504 002456 004567 003020 JSR R5, READY ;LET DRIVE READY
505 002460 004767 002610 BR BDFL ;BRANCH IF DRIVE READY
506 002464 004767 002610 JSR PC, NOTRDY ;TRY DRIVE NOT READY
507 002470 000650 BR BADFL ;BRANCH IF SUCCESSFUL
508 002472 012767 177777 176072 5S: MOV #-1, UNITNO ;RESET UNIT NUMBER
509 ;ALL ON LINE DRIVE HAS BEEN RETRIEVED THE BAD SPOT FILE
510 ;* SUPPORT - DT03
511 BR RSTRT1

```

```

RMAD0.P11 12-DEC-78 16:17
512 002502* 105267 176046
513 002506* 105267 176042
514 002511* 122767 000035 176034
515 002520* 101763
516 002522* 104403 007050* 000000*
517 002530* 000167 000306

```

```

RETRX: INCR DSKADR ;ADJUST 2 SECTOR
        INCR DSKADR ;ADJUST TWO SECTOR
        CMPR #30, DSKADR ;SECTORS 0,2,4,.....30, ALL TRIED
        BHI BADFL ;BRANCH IF NOT
        MSGNS, MCS15 BEGIN ;MESSAGE : RETRIEVE FAILS
        JMP FIN ;DROP THE MODULES

```

```

RMAD0.P11 12-DEC-78 16:17
518 002534* 005767 175732
519 002540* 001005
520 002542* 005767 175722
521 002546* 001002
522 002550* 000167 177106
523 002554*
524 002554* 104415 000000* 000124*
525 002562* 016767 175344 176000
526 002570* 005467 175774
527 002574* 016767 175660
528 002602* 104414 000000*
529 002606* 016767 175330 175752
530 002614* 005467 175745
531 002620* 004767 002376
532 002624* 016777 175742 176756
533 002632* 005767 175720
534 002636* 001002
535 002640* 000167 000176
536 002644* 132767 000010 004226
537 002653* 001404
538 002654* 012767 177777 175710
539 002662* 000744
540 002669* 004767 002250
541 002670* 004567 002606
542 002674* 000402
543 002676* 004767 002376
544 002705* 005067 004156
545 002711* 005067 004152
546 002716* 005067 004150
547 002722* 005067 004146
548 002726* 005067 004144
549 002732* 004567 000466

```

```

RFRSTRT: TST CNT1 ;+ / SUPPORT
          BNE RFRSTRT1 ;+ / FOR
          TST CNT ;+ / DT03
          BNE RFRSTRT1 ;+ / BUS
          JMP START ;+ / SWITCH
          ;+ / OPTION
RSTRT1: GETPAS, REGIN, RBUPVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUPVA
        MOV RBUFSSZ, WCNT2 ;SAVE READ BUFFER SIZE
        NEG WCNT2 ;GET THE 2'S COMPLEMENT
        MOV BLK1, RLKSAV ;INIT BLOCK COUNTER
STRT1: GWBUFS, BEGIN ;GET WRITE BUFFER INFORMATION
        MOV WBUFSZ, WCNT1 ;SAVE WRITE BUFFER SIZE
        NEG WCNT1 ;GET THE 2'S COMPLEMENT
        JSR PC, PDUNIT ;FIND UNIT #
        MOV UNITNO, @RMCS2 ;GET UNIT # SO BOTH PORTS
        ;WILL BE LOOKING FOR SAME DRIVE
        TST DVICE ;ANY DRIVES LEFT?
        BNE IS ;YES
        JMP FINI ;NO DROP MODULE
        BITB #BIT3, FLAG ;MORE DRIVES ON SYS?
        BEQ ABW ;YES
        MOV #-1, UNITNO ;RESET DRIVE NUMBER
        BR START ;YES CORRECT SETUP FOR DUEL PORT RETURN
        JSR PC, RM03CK ;UPDATE DSK ADDR
        JSR R5, READY ? ;IS DRIVE READY?
        BR IS ;YES, CONTINUE
        JSR PC, NOTRDY ;NOT READY, GO WAIT UNTIL IT IS
        CLR TRV1 ;ZERO RETRY COUNTERS
        CLR TRV3
        CLR TRV5
        CLR TRV7
        CLR TRV9
        CLR TRV11
        JSR R5, UPDAT ;TO SEE IF RM03 OR RM02
        ;FOR THE SELECTED DRIVE

```

```

554 002736 004567 000272 GOC: JSR R5,WRITE ; WRITE SOME DATA
555 002744 000167 000104 JMB RETRY ; IF ERRORS, TRY IT AGAIN
556 002746 132767 000004 004124 BITB #BIT2,FLAG ; DISK OVERFLOW ?
557 002754 001410 BEQ GOA ; NO, CONTINUE
558 002756 142767 000004 004114 DAOCL: BICR #BIT2,FLAG ; YES, CLEAR OVERFLOW FLAG
559 002764 012767 177777 175466 MOV #-1,BLK1 ; RESET BLOCK #
560 002772 000167 177576 JMB STR1 ; CONTINUE
561 002776 004567 000326 GDA: JSR R5,WRITCK ; WRITE-CHECK THE DATA
562 003002 000167 000114 JMB RETRY2 ; IF ERRORS, TRY AGAIN
563 003006 004567 000420 GOB: JSR R5,READ ; READ THE DATA WRITTEN
564 003012 000167 000136 JMB RETRY3 ; IF ERRORS, TRY AGAIN
565 003016 104412 000000 000126 CDATA$,BEGIN,RBMPA ; REQUEST FOR MONITOR TO CHECK DATA
566 003024 003026 *-+2 ; IF ERROR, CONTINUE
567 003026
568 003026 005267 175440 CVCLE: INC CNT1 ; *BUMP INDICATOR FOR DT03
569 003032 104413 000000 ENDS$,BEGIN ; SIGNAL END OF ITERATION.
570
571 003036 000167 177556 JMB NEXT ; MONITOR SHALL TEST END OF PASS
572 003042
573 003042 104410 000000 FINI: ENDS$,BEGIN ; DROP THE MODULE
574
575 003046 000167 177546 ;
576 003052 005767 175422 NXT1: JMB NEXT ; GET NEXT DRIVE
577 003056 001404 RESTRY1: TST BADSEC ; BAD SECTOR DETECTED ?
578 003060 005067 175414 BEQ +10 ; DETORE
579 003064 000167 000116 CLR BADSEC ; CLEAR THE FLAG
580 003068 105267 003770 JMB NEXTA ; EXIT
581 003074 122767 000003 003762 INCR TRV1 ; COUNT THE RETRYS
582 003102 001402 CMPB #3,TRV1 ; LIMIT EXCEEDED
583 003104 000167 177626 BEQ IS ; YES
584 003110 000000 006766 JS: MSGNS$,BEGIN,EXCED1 ; ASCII MESSAGE CALL WITH COMMON HEADER
585 003116 000167 000064 JMB NEXTA ; GO TO NEXT DRIVE
586 003122 105267 003737 RESTRY2: INCB TRV2 ; COUNT THE RETRYS
587 003126 122767 000003 003731 CMPR #3,TRV2 ; LIMIT EXCEEDED
588 003134 001402 BEQ IS ; YES
589 003136 000167 177634 JMB GOA ; NO RETRY
590 003142
591 003142 104403 000000 006774 JS: MSGNS$,BEGIN,EXCED2 ; ASCII MESSAGE CALL WITH COMMON HEADER
592 003150 000167 000032 JMB NEXTA ; GO TO NEXT DRIVE
593 003154 105267 003706 RESTRY3: INCR TRV3 ; COUNT THE RETRYS
594 003160 122767 000003 003700 CMPB #3,TRV3 ; LIMIT EXCEEDED
595 003166 001402 BEQ IS ; YES
596 003170 000167 177612 JMB GOB ; NO RETRY
597 003174
598 003174 104403 000000 007002 JS: MSGNS$,BEGIN,EXCED3 ; ASCII MESSAGE CALL WITH COMMON HEADER
599 003202 000167 000000 JMB NEXTA ; GO TO NEXT DRIVE
600

```

```

601 003206 032767 000001 174602 NEXTA: BIT #BIT0,SRI ; DROP THE DRIVE
602 003214 001405 BEQ IS ; NO, SKIP TO NEXT DRIVE
603 003216 004767 000634 JSR PC,DROP ; YES, DROP OPENDING DRIVE
604 003222 104403 000000 007024 MSGNS$,BEGIN,DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
605 003230 000167 177364 JS: JMB NEXT ; GO ON TO NEXT DRIVE
606
607
608
609
610
611 ; .MACRO LINEUP 2ABITS ; LINE UP EA BITS FOR RHCSI
612 LINEUP EABITS,RO
613 MOV EABITS,RO ; GET EXTENDED MEMORY BITS
614 ASL RO ; SHIFT 4 PLACES TO THE LEFT
615 ASL RO ;
616 ASL RO ; TO LINE UP WITH RHCSI
617 ASL RO ;
618 MOV RO,XMEM ; SAVE THE SHIFTED BITS
619 .ENDM LINEUP

```

```
----- R4 DISK DRIVERS -----  
619  
620  
621  
622 003234* 012767 000161 175232 WRITE: MOV #161, FUNC ; LOAD WRITE FUNCTION  
623 003232* 016777 175320 175332 MOV WCNTR, @RMWC ; LOAD WORD COUNT  
624 003250* 016777 174660 175332 MOV #BUFA, @RMBA ; LOAD BUFFER ADDRESS  
625 003256* 016777 175272 176322 MOV DSKADR, @RMDA ; LOAD DISK ADDRESS  
626 003264* 004767 001650 JSR PC, @RM3CK ; CHECK FOR RM03 DRIVE  
627 003270* 016777 175166 176336 MOV CYL, @RMDC ; LOAD CYLINDER ADDRESS  
628 003276* 016767 175162 175158 MOV CYLSAV, CYL ; RESTORE CALCULATED CYLINDER ADDRESS  
629  
630 003304* 016700 174626 ; LINEUP  
631 003310* 006300 MOV #BUFEA, RO ; GET EXTENDED MEMORY BITS  
632 003312* 006300 ASL RO ; SHIFT 4 PLACES TO THE LEFT  
633 003314* 006300 ASL RO ; TO LINE UP WITH RHCSI  
634 003316* 006300 ASL RO ;  
635 003320* 010067 175162 MOV RO, XMEM ; SAVE THE SHIFTED BITS  
636 003324* 000167 000300 JMP GOGO ; CONTINUE  
637 003330* 012767 000151 175136 WRITCK: MOV #151, FUNC ; LOAD WRITE-CHECK FUNCTION  
638 003336* 016777 175224 176236 MOV WCNTR, @RMWC ; LOAD WORD COUNT  
639 003344* 016777 174564 176232 MOV #BUFA, @RMBA ; LOAD BUFFER ADDRESS  
640 003352* 016777 175176 176226 MOV DSKADR, @RMDA ; LOAD DISK ADDRESS  
641 003360* 004767 001554 JSR PC, @RM3CK ; CHECK FOR RM03 DRIVE  
642 003364* 016777 175072 176242 MOV CYL, @RMDC ; LOAD CYLINDER ADDRESS  
643 003372* 016767 175066 175062 MOV CYLSAV, CYL ; RESTORE CALCULATED CYLINDER ADDRESS  
644  
645 003400* 016700 174532 ; LINEUP  
646 003404* 006300 MOV #BUFEA, RO ; GET EXTENDED MEMORY BITS  
647 003406* 006300 ASL RO ; SHIFT 4 PLACES TO THE LEFT  
648 003410* 006300 ASL RO ; TO LINE UP WITH RHCSI  
649 003412* 006300 ASL RO ;  
650 003414* 010067 175066 MOV RO, XMEM ; SAVE THE SHIFTED BITS  
651 003420* 000167 000204 JMP GOGO ; CONTINUE  
652 003424* 004767 001010 UPDAT: JSR PC, @BLOCK  
653 003430* 000205 RTS  
654  
655 003432* 012767 000171 175034 READ: MOV #171, FUNC ; LOAD READ FUNCTION  
656 003440* 016777 175124 176134 MOV WCNTR, @RMWC ; LOAD WORD COUNT  
657 003446* 016777 174454 176130 MOV #BUFA, @RMBA ; LOAD BUFFER ADDRESS  
658 003454* 016777 175074 176124 MOV DSKADR, @RMDA ; LOAD DISK ADDRESS  
659 003462* 004767 001452 JSR PC, @RM3CK ; CHECK FOR RM03 DRIVE  
660 003466* 016777 174770 176140 MOV CYL, @RMDC ; LOAD CYLINDER ADDRESS  
661 003474* 016767 174764 174760 MOV CYLSAV, CYL ; RESTORE CALCULATED CYLINDER ADDRESS  
662  
663 003502* 016700 174422 ; LINEUP  
664 003506* 006300 MOV #BUFEA, RO ; GET EXTENDED MEMORY BITS  
665 003510* 006300 ASL RO ; SHIFT 4 PLACES TO THE LEFT  
666 003512* 006300 ASL RO ; TO LINE UP WITH RHCSI  
667 003514* 006300 ASL RO ;  
668 003516* 010067 174764 MOV RO, XMEM ; SAVE THE SHIFTED BITS  
669 003522* 000442 BR GOGO ; CONTINUE
```

```
670 003524* 016777 175042 176056 CLEAR: MOV UNITNO, @RWCS2 ; LOAD UNIT ADDRESS  
671 003532* 012777 000011 176040 MOV #11, @RMCS1 ; ISSUE A DRIVE CLEAR  
672 003540* 000240 NOP ; WAIT  
673 003542* 000240 NOP ; FOR DRIVE CLEAR TO FINISH  
674 003544* 004407 000000* BREAK$, BEGIN ; TEMPORARY RETURN TO MONITOR  
675 003546* 004407 000000* BREAK$, BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION  
676 003554* 012777 000021 176016 MOV #21, @RMCS1 ; ISSUE A PACK ACK  
677 003562* 032777 000200 176010 1S: BIT #17, @RMCS1 ; FUNCTION DONE ?  
678 003570* 001005 BNE CS ; YES, CONTINUE  
679 003572* 004407 000000* BREAK$, BEGIN ; TEMPORARY RETURN TO MONITOR  
680 003576* 004407 000000* BREAK$, BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION  
681 003602* 000767 BR 1S ; NO, WAIT TILL DONE  
682 003604* 012777 177777 176004 2S: MOV #-1, @RMAS ; CLEAR AS BIT  
683 003612* 012777 040000 175760 MOV #BIT4, @RMCS1 ; CLEAR ANY CONTROLLER ERRORS  
684 003620* 016777 174752 176004 MOV MOD1, @RMDF ; SET BIT FOR 11 FORMAT  
685 003626* 000205 RTS ; RETURN  
686  
687 003630* 016777 174736 175752 GOGO: MOV UNITNO, @RMCS2 ; LOAD UNIT SELECT  
688 003636* 012777 003764 174144 MOV #NTRPT, @VECTOR ; SET INTERRUPT ENTRY POINTER  
689 003644* 032767 001000 174204 BIT #ADDR22, RES1 ; #11/70?  
690 003652* 001434 BEQ 1S ; NO  
691 003654* 017767 175724 174622 MOV @RMBA, PA18 ; GET 18 BIT ADDR  
692 003662* 006267 174620 ASR XMEM ; SHIFT EA BITS TO POSITION 4,5  
693 003666* 006267 174614 ASR XMEM ;  
694 003672* 006267 174610 ASR XMEM ;  
695 003676* 006267 174604 ASR XMEM ;  
696 003702* 004416 000000* 000504* MAP22$, BEGIN, PA18 ; GET 22-BIT ADDR FROM 18-BIT ADDR  
697 003710* 016777 174574 175666 MOV PA22, @RMBA ; LOAD BA REG  
698 003716* 016777 174570 175724 MOV EA22, @RMBAE ; LOAD BAE REG  
699 003724* 012767 000034 174560 BIC #34, EA22 ; CLEAR UNWANTED BITS  
700 003732* 000367 174554 SWAB ; LOAD INTO BITS 8,9  
701 003736* 016767 174550 174542 MOV EA22, XMEM ; LOAD XMEM TO SET INTO FUNCTION CODE  
702 003744* 056767 174536 174522 1S: BIS XMEM, FUNC ; LOAD EXTENDED MEMORY BITS  
703 003752* 016777 174536 175620 MOV FUNC, @RMCS1 ; EXECUTE THE FUNCTION  
704 003760* 014400 000000* EXITS, BEGIN ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.  
705  
706 003764* NTRPT: ;  
707 ;-----  
708 003764* 000004 000000* 003772* PIRQ$, BEGIN, 1S ; QUEUE UP TO CONTINUE AT 1S AND RTI  
709 ;-----  
710  
711 003772* 004567 000116 1S: JSR R5, ERRORS ; GO CHECK FOR ERRORS  
712 003776* 000205 RTS R5 ; ERRORS DETECTED, RETURN  
713 004000* 005725 TST (R5)+ ; NO ERRORS, SKIP RETRY  
714 004002* 005725 TST (R5)+ ;  
715 004004* 000205 RTS R5 ; RETURN OK
```

```

RMAD0.P11 12-DEC-78 16:17
716 004006 016700 174446 ROOM: MOV BLK1,R0 ; SAVE THE CURRENT BLOCK NUMBER
717 004012 016703 174124 MOV WBUF57,R3 ; GET THE TRANSFER SIZE
718 004016 132767 000002 003054 BITR #BIT1,FLAG ; PLENTY OF ROOM LEFT ?
719 004024 001406 BEQ 4S ; YES, CONTINUE
720 004026 142767 000001 003044 BITR #BIT0,FLAG ; CLEAR 32K INDICATOR
721 004034 016701 174464 MOV WHICV,R1 ; LOAD MAX. NUMBER OF BLOCKS
722 004040 001402 BEQ 5S ; EXHAUST OF ALL BLOCKS
723 004042 005725 4S: TST (R5)+ ; YES, MUST BE A REAL ERROR
724 004044 000205 RTS R5 ; RETURN ERROR
725 004046 152767 000004 003024 5S: BITB #BIT2,FLAG ; SET OVERFLOW FLAG
726 004054 000205 RTS R5 ; RETURN OK
727 ;
728 ;
729 004056 012701 000001 DROP: MOV #1,R1 ; INITIALIZE DROP PICKER
730 004062 016700 174504 MOV UNITNO,R0 ; GET THE DRIVE NUMBER
731 004066 001403 BEQ 2S ; IF DRIVE 0 GO DROP IT
732 004070 006301 1S: ASL R1 ; POINT TO NEXT DRIVE
733 004072 005300 DEC R0 ; IS THIS THE ONE ?
734 004074 001375 BNE 1S ; NO, LOOK AGAIN
735 004076 040167 174454 2S: BIC R1,DEVICE ; DROP THE DRIVE
736 ;
737 ;
738 ;
739 004102 104420 000000 000572 OTDAS,BEGIN,UNITNO,ADR1 ; CONVERT UNITNO TO ASCII AND
740 004110 007054 ***** ; STORE AT ADR1
741 ;
742 004112 000207 RTS PC ; RETURN
743 ;

```

```

RMAD0.P11 12-DEC-78 16:17
744 004114 005067 174360 ERRORS: CLR BADSEC ; CLEAR THE BAD SECTOR FLAG
745 004120 005777 175454 TST @RMCS1 ; ATTENTION OR ERROR ?
746 004124 100402 BMI 22S ; YES
747 004126 000167 000302 JMP RESYNC ; NO GO ON TO NEXT FUNTION
748 004132 032777 001000 175454 22S: BIT #BIT9,@RMER1 ; ADDRESS OVERFLOW ?
749 004140 001403 BEQ 1S ; YES, IS IT A REAL ERROR ?
750 004142 004567 177640 JSR R5,ROOM ; NO, CONTINUE
751 004146 000532 RESYNC ; DID LBT SET?
752 004150 032777 002000 175434 1S: BIT #BIT10,@RMS ; YES
753 004156 001126 BNE RESYNC ; LOAD ERROR INFORMATION
754 004160 001430 JSR PC,RSUB1 ; IS THIS A DATA LATE ERROR?
755 004164 005777 175420 TST @RMCS2 ; NO
756 004170 100012 BPL 11S ; ADD 1 TO DATA LATE COUNTER
757 004172 005267 174256 INC DLTCNT ; TYPE ERROR AND COUNT IT?
758 004176 032767 000004 173612 BIT #BIT2,SR1 ; ASCII MESSAGE CALL WITH COMMON HEADER
759 004204 001102 MSGNS,BEGIN,DLTERR ; CONT
760 004206 104403 000000 007010 BR 5S ; MASSBUS CONTROL PARITY ERROR ?
761 004214 000450 BR 5S ; MASSBUS DATA PARITY ERROR ?
762 004216 032777 020000 175354 11S: BIT #BIT13,@RMCS1 ; YES
763 004224 001035 BNE 3S ; YES
764 004226 032777 000400 175354 BIT #BIT8,@RMCS2 ; TRANSFER ERROR ?
765 004234 001035 BNE 4S ; YES
766 004236 032777 040000 175334 BIT #BIT14,@RMCS1 ; ANY DRIVE ERRORS ?
767 004244 001015 BNE 2S ; YES
768 004246 032777 040000 175336 BIT #BIT14,@RMS ; ANY ATTENTIONS ACTIVE ?
769 004254 001030 BNE 5S ; YES, CONTINUE
770 004256 005777 175334 TST @RMAS ; ANY ATTENTIONS ACTIVE ?
771 004262 001025 BNE 5S ; YES, CONTINUE
772 004264 005067 173616 CLR ERRTP ; UNKNOWN ERROR
773 ;
774 004270 104405 000000 001600 ***** ; SPECIAL CONDITION SET BUT NO REASON FOUND
775 ***** ;
776 004276 000445 BR 8S ; RETURN
777 004300 000445 BR 8S ; RETURN
778 004306 032777 100000 175334 2S: BIT #BIT15,@RMER2 ; A BAD SPOT ? *****
779 004310 001044 BNE 8S ; BRANCH IF SO *****
780 004312 104403 000000 006752 MSGNS,BEGIN,TRERR ; ASCII MESSAGE CALL WITH COMMON HEADER
781 004316 000407 BR 5S ; GO DUMP REGISTERS
782 004320 104403 000000 006756 3S: MSGNS,BEGIN,MCPERR ; ASCII MESSAGE CALL WITH COMMON HEADER
783 004326 000403 BR 5S ; GO DUMP REGISTERS
784 004330 104403 000000 006762 4S: MSGNS,BEGIN,MDPERR ; ASCII MESSAGE CALL WITH COMMON HEADER
785 004336 005777 175254 TST @RMAS ; ANY ATTENTIONS ACTIVE ?
786 004342 001402 BNE 6S ; NO, CONTINUE
787 004344 004767 JSR PC,WHO ; YES, FIND OUT WHICH DRIVE IT IS
788 004350 016700 175246 MOV RMDB,R0 ; SAVE ADDRESS OF DATA BUFFER
789 004354 032777 000200 175226 6S: BIT #BIT7,@RMCS2 ; CAN DATA BUFFER BE READ ?
790 004362 001003 BNE 7S ; YES, CONTINUE
791 004364 012767 000552 175230 MOV #ZERO,RMDR ; NO, LOAD ADDRESS OF ZERO
792 004372 012767 000001 173506 7S: MOV #1,ERRTP ; DATA ERROR
793 ***** ;
794 ***** ;
795 ***** ;
796 004400 104406 000000 001600 SOPS,BEGIN,TABLE ; DUMP R11 AND R10 REGISTERS
797 ***** ;
798 004406 010067 175210 MOV R0,RMDB ; RESTORE DATA BUFFER ADDRESS
799 004412 004567 177106 JSR R5,CLEAR ; GO CLEAR OUT ERRORS

```

```

000 004416* 000205
001 004420* 004567 177100
002 004424* 012767 177777 174046
003 004432* 000205
9S: RTS R5 ; ERRORS DETECTED, RETURN
    JSR R5 ; CLEAR ; CLEAR THE BAD SPOT ERROR
    MOV #1,RADSEC ; SET THE BAD SPOT ERROR FLAG
    RTS R5 ; EXIT
  
```

```

004 004434* 005725
005 004436* 000205
006
007
008
009
010
011
012 004440* 005267 174014
013 004444* 132767 000002 002425
014 004452* 001012
015 004454* 026767 174042 173776
016 004462* 101017
017 004464* 005067 173770
018 004470* 152767 000002 002402
019 004476* 000411
020
021 004500* 026767 174020 173752
022 004506* 101005
023 004510* 005067 173744
024 004514* 142767 000002 002355
025 004522* 016700 173732
026 004526* 005767 174042
027 004532* 001432
028 004534* 016777 174032 175046
029 004542* 005067 173714
030 004546* 005067 173712
031 004552* 005067 173716
032 004556* 022777 000620 175050
033 004564* 101414
034 004566* 017767 175042 173666
035
036 004574* 062767 000001 173660
037
038 004602* 017767 175000 173744
039 004610* 017767 174772 173736
040 004616* 000207
3S: RESYNC: TST (R5)* ; SKIP RETRY
    RTS R5 ; RETURN OK

;THIS ROUTINE DETERMINES IF THERE IS ENOUGH ROOM ON THE DISK TO
;DO ANOTHER TRANSFER. IF NOT, PROGRAM GOES TO RESYNC TO
;ALLOW BOTH PROCESSORS TO RESYNC AND TO RESTART.
BLOCK: INC BLK1 ; STEP TO NEXT BLOCK
       BITR #BIT1,FLAG ; BLOCK # IN CYLINDER 410. OR HIGHER ?
       BNE IS ; YES, GO ADJUST BLOCK #
       CMP #LOWCY,BLK1 ; BLOCK # IN CYLINDER 410. OR LOWER ?
       BHI ZS ; YES, RETURN
       CLR BLK1 ; NO, RESET BLOCK #
       BISR #BIT1,FLAG ; SET HIGH RANGE FLAG
       BR ZS ; RETURN

021: CMP #HICY,BLK1 ; BLOCK # IN WITHIN RANGE ?
     BHI ZS ; YES, RETURN
     CLR BLK1 ; NO, RESET BLOCK #
     BICR #BIT1,FLAG ; SET FLAG TO LOWER RANGE FOR NEW PACK CYCLE
2S: MOV BLK1,R0 ; TRANSFER PARAMETER FOR CONVRT
     TST #HXDV ; IS AN RM03/RW02 ?
     BEQ CONVRT ; BRANCH IF SO
     MOV UNITNO,@RMCS2 ; LOAD THE DRIVE ADDRESS
     CLR CYL ; RESET CYLINDER ADDRESS
     CLR CYLSAV ; RESET THE SAVED CYLINDER ADDRESS
     CLR DSKADR ; RESET THE TRACK AND SECTOR ADDRESS
     CMP #400,@RMDC ; OVER CYL 400 ?
     BLS ; BRANCH IF SO
     MOV @RMDC,CYL ;
     MOV @RMDC,CYLSAV ;
     ADD #1,CYL ; THE NUMBER CAN MODIFY TO INCREMENT
     ; THE CYLINDER ADDRESS
     MOV @RMDA,DSKADR ;
     MOV @RMDA,DSKADR ;
3S: RTS PC ;
  
```

```

XRMAD0.P11 12-DEC-78 16:17
R41 004620 005067 173636 CONVRT: CLR CYL ; CLEAR DISK ADDRESSES
R42 004624 005067 173636 CLR SEC
R43 004630 105067 002245 CLR TRK
R44 004634 016701 173656 MOV MBLKTR,R1 ; LOAD REG. 1 WITH BLK PER TRACK
R45 ; REG. 0 HAS BLK1 (BLOCK #) IN IT
R46 004640 016702 173650 MOV MRLKRV,R2 ; LOAD REG. 2 WITH BLK PER REVOLUTION
R47 004644 132767 000002 002226 BITB #BIT1,FLAG ; BLOCK # ON CYLINDER 410. OR LOWER?
R48 004652 001403 BEQ IS ; YES, CONTINUE
R49 004654 012767 000632 173600 MOV #10,,CYL ; NO, LOAD HIGH BASE VALUE FOR CYLINDER ADR.
R50 004662 020001 CMP #0,R1 ; CORRECT CYLINDER FOUND?
R51 004664 103413 BLO #5 ; YES, CONTINUE
R52 004666 005267 173570 INC CYL ; NO, STEP TO NEXT CYLINDER
R53 004672 160100 SUB R1,R0 ; SUBTRACT 1 CYLINDER FROM BLOCK #
R54 004674 000772 BR JS ; CONTINUE UNTIL CORRECT CYLINDER IS FOUND
R55 004676 022767 001440 173556 CMP #800,,CYL ; NOT OVER CYL 800 FOR PROTECT BAD SE.FILE
R56 004704 101003 BHI #5 ; YES
R57 004706 012767 000632 173546 MOV #410,,CYL ; RESET TO 410
R58 004714 020002 CMP #0,R2 ; CORRECT TRACK FOUND?
R59 004716 002404 BLT JS ; YES, CONTINUE
R60 004720 105267 002155 INCR TRK ; NO, STEP TO NEXT TRACK
R61 004724 160200 SUB R2,R0 ; SUBTRACT 1 TRACK FROM BLOCK #
R62 004726 000772 BR #5 ; CONTINUE UNTIL CORRECT TRACK IS FOUND
R63 004730 005700 TST R0 ; CORRECT SECTOR FOUND?
R64 004732 001405 BEQ #5 ; YES, CONTINUE
R65 004734 002767 000004 173524 ADD #1,SEC ; NO, STEP TO NEXT 1024 WORDS
R66 004742 005300 DEC R0 ; DECREASE BLOCK # BY 1
R67 004744 000771 BR #5 ; CONTINUE UNTIL CORRECT SECTOR IS FOUND
R68 004746 022767 000005 002125 4S: CMPIR #5,TRK ; LAST TRACK?
R69 004750 003007 BGT #5 ; NO, CONTINUE
R70 004756 026767 173536 173502 CMP MODE,SEC ; YES, LAST SECTOR?
R71 004764 003003 BCT #5 ; NO, CONTINUE
R72 004766 162767 000004 173472 SUB #4,SEC ; YES, ADJUST SECTOR SO NO OVERFLOW
R73 004774 016767 173466 173552 MOV SEC,DSKADR ; LOAD SECTOR INTO DISK ADDRESS
R74 005002 116767 02073 173545 MOVR TRK,DSKADR+1 ; LOAD TRACK INTO DISK ADDRESS
R75 ; CHECK THE THE SELECT ADDRESS IS IN THE RAD SPOT?
R76 005010 005000 CLR R0 ; INDEX VALUE TO THE BAD SPOT TABLE
R77 005012 026760 173444 000254 BADL0P: CMP CYL,BADSPT(R0) ; CYLINDER ADDRESS IN THE SPOT TABLE?
R78 005014 001405 BEQ #1 ; YES
R79 005016 001405 ADD #1,R0 ; CHECK THE NEXT SPOT
R80 005026 000434 BR #5 ; COMMON BRANCH POINT
R81 005030 002700 000002 000255 1S: ADD #2,R0 ; CHECK THE TRAK AND SEC
R82 005034 126760 002041 000255 CMPB TRK,BADSPT+1(R0) ; ON THE BAD SPOT TRACK?
R83 005042 001405 BEQ #5 ; YES
R84 005044 003007 ADD #2,R0 ; CHECK THE NEXT SPOT
R85 005050 000423 BR #5 ; COMMON BRANCH POINT
R86 005052 126760 173410 000254 2S: CMPIR SEC,BADSPT(R0) ; CHECK THE SECTOR ADDRESS IS IN THIS BLOCK RANGE?
R87 005060 101015 BHI #4 ; NO, NOT IN THIS BLOCK
R88 005062 002767 000003 173376 ADD #1,SEC ; WITHIN THE BLOCK?
R89 005070 126760 173372 000254 CMPIR SEC,BADSPT(R0) ; NOT IN THE BLOCK IF LESS
R90 005076 103403 BLD #3 ; YES
R91 005100 012700 MOV #-1,R0 ; ERROR FLAG
R92 005104 000405 BR #5 ; COMMON BRANCH POINT
R93 005106 002767 000003 173352 3S: SUB #2,SEC ; RESTORE THE SECTOR VALUE
R94 005114 002700 000002 4S: ADD #2,R0 ; CHECK NEXT BAD SPOT
R95 005120 005700 5S: TST R0 ; MINUS SIGN SET, IF ON A BAD SPOT
R96 005122 100002 BPL #+6 ; SKIP CURRENT SELECTED BLOCK IF BAD SPOT

```

```

XRMAD0.P11 12-DEC-78 16:17
R97 005124 000167 177310 JMP BLOCK ; SELECT OTHER BLOCK
R98 005130 022700 000200 CMP #12R,,R0 ; END OF TABLE
R99 005134 101326 BHI BADL0P ; NO, THEN BRANCH BACK
R00 005136 000207 RTS PC ; RETURN
R01 ;
R02 ;
R03 005140 016767 173316 173316 RM03CK: MOV CYL,CYLSAV ; SAVE THE CALCULATED ADDRESS
R04 005146 005067 173422 CLR MIXDV ; RESET MIX DRIVE FLAG
R05 005152 022777 024025 174446 CMP #24025,@R4DT ; DUAL PORT RM02?
R06 005160 001417 BEQ IS ; BRANCH IF SO
R07 005162 022777 020025 174436 CMP #20025,@RMDT ; SINGLE PORT RM02?
R08 005170 001413 BEQ IS ; BRANCH IF SO
R09 005172 022777 024024 174426 CMP #24024,@R4DT ; DUAL PORT RM03?
R10 005200 001407 BEQ IS ; YES
R11 005202 022777 020024 174416 CMP #20024,@RMDT ; NO, SINGLE PORT RM03?
R12 005210 001403 BEQ IS ; YES
R13 ; JSR PC,DROP ; DROP THE DRIVE ; IF NOT A RM03
R14 ; MSGNS,DRP1,BEGIN ; MESSAGE - NOT A RM03
R15 ; TST (SP)+ ; CLEAR STACK FOR WHOEVER CALLING RM03CK
R16 ; MOV (SP)+,R5 ; CLEAR STACK FOR WHOEVER CALLING "WRITE",
R17 ; ; "WRITCK" AND "READ" ROUTINES
R18 ; JMP NEXT ; TO NEXT DRIVE
R19 005212 012767 177777 173354 1S: MOV #-1,MIXDV ; SET NON-RM03, RM02 DRIVE FLAG
R20 005220 000207 RTS PC ;
R21 ;
R22 ;

```

```

023 005222* 005267 173344 FDUNIT: INC UNITNO ; COUNT A DRIVE
024 005226* 142767 000010 BIT3,FLAG ; CLEAR END OF DRIVES FLAG
025 005234* 022767 000010 173330 CMP #R,UNITNO ; ALL DRIVES CHECKED ?
026 005242* 003404 BLE IS ; YES, GO FLAG END OF DRIVES
027 005244* 006267 173310 ASR DRIVE ; NO, IS NEXT DRIVE CHOSEN ?
028 005250* 103364 BCC FDUNIT ; NO, GO TRY ANOTHER DRIVE
029 005252* 000411 BR 25 ; RETURN
030 005254* 152767 000010 001616 1S: BISR #BIT3,FLAG ; SET END OF DRIVES FLAG
031 005262* 012767 177777 173302 MOV #1,UNITNO ; RESET DRIVE COUNTER
032 005270* 016767 173262 173262 MOV DEVICE,DRIVE ; RESTORE CHOSEN DRIVES
033 005276* 000207 RTS PC ; RETURN
034
035
036
037
038
039 ;SEIZE THE DRIVE BY READING CSI REG
040 005300* 012767 077777 173150 NOTRDY: MOV #77777,CLX ; SET THE TIMER
041 005306* 005777 174266 4S: TST @RMCS1 ; GRAB THE DRIVE
042 005312* 032777 004000 174260 BIT #BIT11,@RMCS1 ; DO I HAVE THE DRIVE DVA?
043 005320* 001013 BNE 25 ; YES
044 005326* 104407 000000* BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR...
045 005332* 005367 173120 DPC CLK ; THEN CONTINUE AT NEXT INSTRUCTION.
046 005336* 001363 BNE 45 ; COUNT # OF TRIES
047 005340* 104403 000000* 007014* MSGNS,BEGIN,NOT ;ASCII MESSAGE CALL WITH COMMON HEADER
048 005346* 000405 BR 25 ; COULD NOT GET DRIVE
049 005350* 004567 176150 2S: JSR R5,CLEAR ; RESET THE CONTROLLER AND DRIVE
050 005354* 004567 000122 JSR R5,READY ; IS DRIVE READY ?
051 005360* 000422 BR 15 ; YES, CONTINUE
052 005362* 004767 000226 5S: JSR PC,ERSUB1 ; LOAD ERROR INFORMATION
053 005366* 012767 000000 JSR #6,ERRTYP ; DRIVE NOT AVAILABLE
054 *****
055 005374* 104405 000000* 001600* HDRS,BEGIN,TABLE ; COULD NOT GET DRIVE --- TIME-OUT
056 *****
057 005402* 012777 000013 174170 MOV #13,@RMCS1 ; RELEASE DRIVE
058 005410* 004767 176442 JSR PC,DRP ; NO, DROP THE DRIVE
059 005414* 104403 000000* 007024* MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
060 005422* 000167 175414 JMP FINI ; COULD NOT GET DRIVE DROP MODULE
061 005426* 000207 1S: RTS PC ; RETURN
062

```

```

063 005430* 017701 174162 WHO: MOV @RMAS,R1 ; GET THE ATTENTION SUMMARY
064 005434* 017704 174150 MOV @RMCS2,R4 ; SAVE THE STATUS REGISTER
065 005440* 012702 000001 MOV #BIT0,R2 ; SET POINTER TO DRIVE 0
066 005444* 005003 CLR R3 ; ZERO THE DRIVE COUNTER
067 005446* 030201 1S: BIT R2,R1 ; FIND IT ?
068 005450* 001006 BNE 25 ; YES, CONTINUE
069 005454* 005203 MOV #1,R2 ; YES, INCREMENT THE DRIVE COUNTER
070 005458* 006502 ASL R2 ; SET POINTER TO NEXT DRIVE
071 005456* 032702 000400 BIT #BIT8,R2 ; ALL DONE ?
072 005462* 001771 BEQ IS ; NO, GO AGAIN
073 005464* 000207 000007 2S: RTS PC ; SOMEBODY LIED -- NO ATTENTIONS SET
074 005466* 042704 BIT #7,R4 ; CLEAR OUT OLD UNIT NUMBER
075 005470* 050304 BIS R4,R4 ; LOAD THE NEW UNIT NUMBER
076 005474* 010477 174110 MOV R4,@RMCS2 ; RESTORE THE STATUS REGISTER
077 005500* 000207 RTS PC ; RETURN
078
079
080 005502* 016777 173064 174100 READY: MOV UNITNO,@RMCS2 ; LOAD UNIT ADDRESS
081 005510* 017700 174076 MOV @RMDS,R0 ; SAVE STATUS IN R0
082 005514* 105700 TSTB R0 ; DRIVE READY ?
083 005516* 100022 BPL IS ; NO
084 005520* 032700 BIT #BIT6,R0 ; VOLUME VALID ?
085 005524* 001417 BEQ IS ; NO
086 005526* 032700 000400 BIT #BIT8,R0 ; DRIVE PRESENT ?
087 005532* 001414 BEQ IS ; NO
088 005534* 032700 004000 BIT #BIT11,R0 ; WRITE LOCKED ?
089 005540* 001011 BNE IS ; YES
090 005542* 032700 010000 BIT #BIT12,R0 ; MEDIUM ON LINE ?
091 005546* 001406 BEQ IS ; NO
092 005550* 032700 040000 BIT #BIT14,R0 ; ANY ERRORS ?
093 005554* 001003 BNE IS ; YES
094 005556* 005700 TST R0 ; ATTENTION SET ?
095 005560* 100401 BMI IS ; YES
096 005562* 000205 RTS R5 ; RETURN READY
097 005564* 005725 1S: TST (R5)+ ; SKIP INSTRUCTION FOLLOWING CALL
098 005566* 000205 RTS R5 ; RETURN AS NOT READY
099
1000
1001
1002 005570* 014167 172312 ERSUB2: MOV -(R1),ASB ; LOAD THE DATA
1003 005574* 010167 172302 R1,SBADR ; LOAD ADDRESS OF DATA WRITTEN
1004 005600* 014267 172304 MOV -(R2),AWAS ; LOAD THE DATA
1005 005604* 010267 172274 R2,WASADR ; LOAD ADDRESS OF DATA READ
1006 005610* 005721 TST (R1)+ ; RESET REG. 1
1007 005612* 005722 TST (R2)+ ; RESET REG. 2
1008
1009 005614* 016767 173760 172256 ERSUB1: MOV RMCS1,CSRA ; LOAD ADR OF CURRENT CSR
1010 005622* 017767 173752 172252 MOV @RMCS1,ACSR ; LOAD CONTENTS OF CURRENT CSR
1011 005630* 000207 RTS PC ; RETURN

```

```

1012
1013
1014
1015
1016
1017 005632* 012777 000040 173750 REZET: MOV #BITS,@RMCS2 ;ISSUE AN RH11 INIT
1018 005640* 012777 177777 173750 MOV #-1,@RMS ;CLEAR ALL ATA BITS
1019 005646* 012767 077777 172602 MOV #77777,CLK ;SET THE TIMER
1020 005654* 032777 000200 173716 1S: BIT #BIT7,@RMCS1 ;CONTROLLER READY?
1021 005664* 010017 BNE IS ;YES, CONTINUE
1022 005674* 104407 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
1023 005670* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1024 005674* 005367 172556 DEC CLK ;WAIT SOME MORE?
1025 005700* 001365 BNE IS ;YES
1026 005702* 005067 CLR DVICE ;NO, SET TO DROP THE MODULE
1027 005706* 012767 000004 172172 MOV #4,ERRTYP ;CONTROLLER NOT READY
*****
1029 005714* 104405 000000* 001600* HDRS,BEGIN,TABLE ;CONTROLLER NOT READY
*****
1030 005722* 000207 2S: RTS PC ;RETURN
1031
1032

```

```

1033 005724* 016700 172056 SETUP: MOV ADDR,RO ;GET DEVICE ADDRESS
1034 005730* 010067 173644 MOV RO,RMCS1 ;GENERATE REGISTER ADDRESSES
1035 005734* 005720 TST (RO)+
1036 005736* 010067 173640 MOV RO,RMWC
1037 005742* 005720 TST (RO)+
1038 005744* 010067 173634 MOV RO,RMRA
1039 005750* 005720 TST (RO)+
1040 005752* 010067 173630 MOV RO,RMDA
1041 005756* 005720 TST (RO)+
1042 005760* 010067 173624 MOV RO,RMCS2
1043 005764* 005720 TST (RO)+
1044 005766* 010067 173620 MOV RO,RMDS
1045 005772* 005720 TST (RO)+
1046 005774* 010067 173614 MOV RO,RMER1
1047 006000* 005720 TST (RO)+
1048 006002* 010067 173610 MOV RO,RMAS
1049 006006* 005720 TST (RO)+
1050 006010* 010067 173604 MOV RO,RMLA
1051 006014* 005720 TST (RO)+
1052 006016* 010067 173600 MOV RO,RMDB
1053 006022* 005720 TST (RO)+
1054 006024* 010067 173574 MOV RO,RMMR1
1055 006030* 005720 TST (RO)+
1056 006032* 010067 173570 MOV RO,RMDT
1057 006036* 005720 TST (RO)+
1058 006040* 010067 173564 MOV RO,RMSN
1059 006044* 005720 TST (RO)+
1060 006046* 010067 173560 MOV RO,RMNF
1061 006052* 005720 TST (RO)+
1062 006054* 010067 173554 MOV RO,RMNC
1063 006060* 005720 TST (RO)+
1064 006062* 010067 173550 MOV RO,RMHR
1065 006066* 005720 TST (RO)+
1066 006070* 010067 173544 MOV RO,RMMR2
1067 006074* 005720 TST (RO)+
1068 006076* 010067 173540 MOV RO,RMER2
1069 006102* 005720 TST (RO)+
1070 006104* 010067 173534 MOV RO,RMEC1
1071 006110* 005720 TST (RO)+
1072 006112* 010067 173530 MOV RO,RMEC2
1073 006114* 032767 001000 171732 BIT #ADDR22,RFS1 ;11/70 MONITOR?
1074 006124* 001417 BEQ IS ;NO
1075 006126* 016700 171654 MOV ADDR,RO ;LOCATE THE RMBAE
1076 006132* 062700 000050 ADD #50,RO ;ASSUME 22 REGISTERS
1077 006136* 032767 100000 171652 BIT #BIT15,SR1 ;32 REGISTER?
1078 006144* 001402 BEQ IS ;BRANCH IF NOT 32 REGISTERS
1079 006146* 062700 000024 ADD #24,RO ;ADJUST THE RMBAE ADDRESS
1080 006152* 010067 173472 3S: MOV RO,RMBAE
1081 006156* 005720 TST (RO)+
1082 006162* 010067 173466 MOV RO,RMCS3
1083 006164* 016700 171620 1S: MOV VECTOR,RO ;GET VECTOR ADDRESS
1084 006170* 012720 002574* #STR1,(RO)+ ;SET POINTER JUST IN CASE
1085 006174* 116710 171612 MOVV #1,(RO) ;SET PRIORITY
1086 006200* 000207 2S: RTS PC ;RETURN

```

XRMANO.P11	12-DEC-78	16:17				
1087	006202	020040	051124	047101	MES1:	.ASCIZ * TRANSFER ERROR*
1088	006210	043123	051105	020040		
1089	006216	051105	047522	022522		
1090	006224	000				
1091	006225	040	046440	051501	MES2:	.ASCIZ * MASSBUS PARITY ERROR*
1092	006232	041123	051522	020040		
1093	006240	040520	044522	054524		
1094	006246	020040	051105	047522		
1095	006254	022522	000			
1096	006257	040	046440	051501	MFS3:	.ASCIZ * MASSBUS DATA PARITY ERROR*
1097	006264	041123	051525	020040		
1098	006272	040504	040524	020040		
1099	006300	040520	044522	054524		
1100	006306	020040	051105	047522		
1101	006314	022522	000			
1102	006317	040	042040	044522	MFS4:	.ASCIZ * DRIVE *
1103	006324	042526	020040	000		
1104	006331	040	042040	047522	MFS5:	.ASCIZ * DROPPED*
1105	006336	050120	042105	000045		
1106	006344	051040	052105	054522	MFS6:	.ASCIZ * RETRY EXCEEDED*
1107	006352	042440	041530	042505		
1108	006360	042504	022504	000		
1109	006365	040	020040	051127	MES7:	.ASCIZ * WRITE*
1110	006372	052111	000105			
1111	006376	020040	053440	044522	MES8:	.ASCIZ * WRITE-CHECK*
1112	006384	042524	041505	042510		
1113	006412	045503	000			
1114	006415	040	020040	042522	MES9:	.ASCIZ * READ*
1115	006422	042101	000			
1116	006425	040	040504	040524	MFS10:	.ASCIZ * DATA LATE ERROR*
1117	006430	046040	020105	020105		
1118	006440	051105	047522	022522		
1119	006446	000				
1120	006447	040	051104	053111	MFS11:	.ASCIZ * DRIVE NOT READY*
1121	006454	020105	047516	020124		
1122	006457	042522	042101	022531		
1123	006470	000				
1124	006471	040	047503	046125	MFS12:	.ASCIZ * COULD NOT GET DRIVE*
1125	006476	020104	047516	020124		
1126	006504	042507	020124	051104		
1127	006511	053117	022505	000		
1128	006517	137	057537	020137	MFS13:	.ASCIZ * ____ NOT A RM03/RM02*
1129	006524	047516	020124	020101		
1130	006532	046522	031460	051057		
1131	006540	030400	022460	000	MESLDP:	.ASCIZ *DROPPED RMDP LOAD DRIVE*
1132	006545	045	051104	050117		
1133	006552	042520	020104	046522		
1134	006560	050104	046040	040517		
1135	006566	020104	051104	053111		
1136	006577	022505	042122	042105		
1137	006577	022505	053117	051105	XMES14:	.ASCIZ *OVER 32 BAD SECTORS RECORDED,PACK NOT ACCEPTABLE*
1138	006604	031440	020062	040502		
1139	006612	020104	042523	052103		
1140	006620	051117	020123	042522		
1141	006626	046440	042122	042105		
1142	006634	050054	041501	020113		

XRMANO.P11	12-DEC-78	16:17				
1143	006642	047516	020124	041501		
1144	006643	047516	052120	041101		
1145	006656	042514	000045			
1146	006662	051045	052105	044522	XMES15:	.ASCIZ *RETRIEVING THE BAD SPOT FILE FAILS-- DROP MODULE*
1147	006670	053105	047111	020107		
1148	006676	044124	020105	040502		
1149	006704	020123	050123	052114		
1150	006712	043040	046111	020105		
1151	006720	040506	046111	026523		
1152	006726	020055	051104	050117		
1153	006734	046440	042117	046125		
1154	006742	022505	000			
1155	006746	006746				
1156	006746	006545			LDRIVE:	.EVEN
1157	006750	177777			MESLDP:	-1
1158	006752	006202			TRERR:	MES1
1159	006752	177777				177777
1160	006755	006225			MCPERR:	MES2
1161	006760	177777				177777
1162	006762	006257			MDFERR:	MES3
1163	006764	177777				177777
1164	006766	106365	000		EXCED1:	MES7
1165	006770	006344				MES6
1166	006772	177777				177777
1167	006774	006375			EXCED2:	MES8
1168	006776	006344				MES6
1169	007000	177777				177777
1170	007002	006415			EXCED3:	MES9
1171	007004	006344				MES6
1172	007006	177777				177777
1173	007010	006425			DLTERR:	MES10
1174	007012	177777				177777
1175						
1176	007014	006447			NOT:	MES11
1177	007016	177777				177777
1178	007020	006471			TOUT:	MES12
1179	007022	177777				177777
1180						
1181	007024	006317			DRP:	MES4
1182	007026	007061				NUMB
1183	007030	006331				MES5
1184	007032	177777				177777
1185	007034	006317			DRP1:	MES4
1186	007036	007061				NUMB
1187	007040	006517				MES13
1188	007042	177777				177777
1189	007044	006577			MES14:	XMES14
1190	007046	177777				177777
1191	007050	006662			MES15:	XMES15
1192	007052	177777				177777
1193						
1194	007054	000005			ADR1:	.BLKB 5
1195	007061	000			NUMB:	.BYTE 0
1196	007062	000				.BYTE 0
1197						.EVEN
1198	007064	000			TRY1:	.BYTE 0

DAGST	002070R	435	438#																	
DATCKC=	104411	248#																		
DATERC=	104411	248#																		
DLTCTR	000454R	316#	404*	757*																
DLTEPR	007010R	760	1173#																	
DRIVE	000560R	365#	438*	927*	932*															
DRDP	004056R	604	729#																	
DRP	007024R	605	959	1181#																
DRP1	007034R	1185#																		
DSKADR	000554R	363#	462*	463*	466*	512*	513*	514	625	640	658	831*	838*	839*						
		873#	874*																	
DVICE	000556R	364#	418*	428	430*	434	438	452	535	735*	932	1026*								
DVLD1	000914R	354#	418																	
EA22	000512R	335#	698	699*	700*	701														
ENDITS=	104413	248#																		
ENDS =	104410	248#																		
ERRORS	004114R	711	744#																	
ERRTP	000106R	712#	772*	794*	953*	1027*														
ERSUB1	005614R	754	952	1009#																
ERSUB2	005570R	1002#																		
EXCED1	006766R	585	1164#																	
EXCED2	006774R	599	1167#																	
EXCED3	007002R	599	1170#																	
EXITS =	104400	249#	704																	
FDUNIT	005222P	444	501	532	923#	928														
FERRADR	000502R	327#	397																	
FILE	000224R	248#																		
FINI	003042R	436	447	455	500	517	537	572#	960											
FIX	007102P	1212#																		
FLAG	007100P	417*	445	502	538	556	558*	718	720*	725*	813	818*	824*	847						
		924	930*	1210#																
FRFF	000150R	240#																		
FUNC	000474R	234#																		
GETPAS=	104415	248#																		
GNA	002776R	557	561#																	
GOB	003006R	563	567																	
GOC	007070R	554#	563																	
GOCN	003630R	636	651	669	697#															
GWRUPFS=	104414	248#	529																	
HRDCNT	000044R	202#																		
HRDEFS=	104405	248#	774	955	1029															
HRDPAS	000050R	204#																		
ICONT	000036R	199#																		
ICOUNT	000040R	200#																		
IDNUM	000122R	229#																		
IMODX =	000000	241	530																	
INIT	000030R	196#																		
INTR	000120R	228#																		
LDRIVE	006746R	431	1156#																	
MAP22S=	104416	248#	696																	
MBLRW	000514R	336#	411	846																
MBLKR	000516R	337#	844																	
MCPERR	006756R	783	1160#																	
MDPPRR	006762R	786	1162#																	
MESLDP	006545R	1132#	1156																	
MESI	006202R	1087#	1158																	

MES10	006425R	1116#	1173																		
MES11	006447R	1120#	1176																		
MES12	006571R	1128#	1178																		
MES13	006571R	1128#	1177																		
MES14	007044R	499	1189#																		
MES15	007050R	516	1191#																		
MES2	006225R	1091#	1160																		
MES3	006225R	1096#	1162																		
MES4	006317R	1102#	1181	1185																	
MES5	006331R	1104#	1183																		
MES6	006344R	1106#	1165	1168	1171																
MES7	006365R	1109#	1164																		
MES8	006376R	1111#	1167																		
MES9	006415R	1114#	1170																		
MHICY	000524R	340#	721	821																	
MIXDV	000574R	371#	473	826	904*	919*															
MLOMVCV	000522R	338#	815																		
MDF	000520R	338#	870																		
MODNAM	000000R	183#																			
MODSP	000252R	197	246#																		
MOD1	000576R	372#	405*	409*	464	684															
MSGNS =	104403	248#	431	499	516	585	592	599	605	760	780	783	786	947							
		659																			
MSGSS =	104402	248#																			
MSGC =	104401	248#																			
NEXT	002620R	532#	571	575	606																
NEXTA	003206R	579	588	593	600	602#															
NOT	007014R	947	1176#																		
NOTRCY	005300R	450	506	545	939#																
INTRUPT	003764R	688	706#																		
NULL	000000	248#																			
NUM	007061R	1182	1196	1195#																	
NXT1	003046R	573#																			
DNCEE	000476R	325#	416*																		
OPEN =	000000	184	191	192	193	210	211	212	213	214	215	216	217								
		219	221	224	226	227	228	231	232	234	235	237	238								
		239	240	248#																	
OTGAS =	104420	248#	739																		
PASCNT	000034R	198#																			
PALR	000504R	332#																			
PA22	000510R	334#	691*	606																	
PIROS =	000004	248#	697																		
POPSP =	005726	248#	708																		
POPSP2=	022626	248#																			
PRTV =	000000	248#																			
PRTV0 =	000000	188	248#																		
PRTV1 =	000040	248#																			
PRTV2 =	000100	248#																			
PRTV3 =	000140	248#																			
PRTV4 =	000200	248#																			
PRTV5 =	000240	187	248#																		

RMAD DEC/111 SYSTEM EXERCISER MODULE
XRMADO.P11 12-DEC-78 16:17

MACY11 30A(1052) 12-DEC-78 16:22 PAGE 40
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0037

ZERN = 000552R 362# 793
 = 007104R 373# 446 465 566 577 896 1155# 1194# 1197# 1213#

. ABS. 000000 000
 007104 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XRMADO,XRMADO/SOL/CRF:SYM=DDXCOM,XRMADO
RUN-TIME: 23.5 SECONDS
RUN-TIME RATIO: 28/6=4.5
CORE USED: 7K (13 PAGES)