

RXAF DEC/X11 SYSTEM EXERCISER MODULE  
XRXAE0.P11 12-OCT-78 12:16

MACV11 30A(1052) 12-OCT-78 17:04 PAGE 2

SEQ 0001

.RFM 1

IDENTIFICATION  
-----

PRODUCT CODE: AC-E736E-MC  
PRODUCT NAME: CXXAEO DEC/X11 MX01 FLOPPY DISK MODULE  
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 OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975, 1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

-----  
RXA IS AN IOMOD THAT EXERCISES TWO RX01 FLOPPY DISKS ON THE UNIBUS. IT EXERCISES BOTH DRIVES BY WRITING AND READING ALL AVAILABLE DRIVES.  
ERRORS ARE CHECKED FOR BUFFER FILL, WRITE, READ, AND DATA COMPARE. TWO RETRIES ARE DONE FOR EACH WRITE OR READ STATUS ERROR. ALL ERRORS ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

-----  
HARDWARE: 1 OR 2 DISKETTES WITH AN RX01 CONTROLLER

STORAGE:: RXA REQUIRES:  
1. DECIMAL WORDS: 1091  
2. OCTAL WORDS: 02103  
3. OCTAL BYTES: 4206

3. PASS IDENTIFICATION

-----  
ONE PASS OF THE RXA MODULE CONSISTS OF THREE WRITE AND READ PASSES OF THE AVAILABLE DRIVES. THE TEST SEQUENCE WRITES THEN READS EVERY THIRD SECTOR OF EVERY TENTH TRACK STARTING AT TRACK 1 SECTOR 1.  
THREE CYCLES OF THE DISKETTE ARE MADE FOR EACH PASS:  
1. STARTS AT SECTOR 1/TRACK 1  
2. STARTS AT SECTOR 2/TRACK 1  
3. STARTS AT SECTOR 3/TRACK 1  
RESTART SECTOR CYCLE AT SECTOR 1/TRACK 2

4. EXECUTION TIME

-----  
ONE PASS OF RXA RUNNING ALONE ON THE PDP-11/05 TAKES APPROXIMATELY .75 MINUTES FOR 2 DRIVES.

5. CONFIGURATION REQUIREMENTS

-----  
DEFAULT PARAMETERS:  
DEVADR: 177170, VECTOR: 264, BRI: 5, DEVCNT: 2  
REQUIRED PARAMETERS:

NONE

6. -----  
DEVICE/OPTION SETUP

ASSURE ALL DRIVES ARE POWERED UP, DISKETTES INSTALLED,  
AND READY.

7. -----  
MODULE OPERATION

TEST SEQUENCES:

- A. SETUP DRIVE REGISTER ADDRESSES AND MODULE VARIABLES
- B. SELECT DRIVES FOR TEST - IF NONE AVAILABLE, DROP MODULE
- C. INITIALIZE DRIVES
- D. SELECT A DRIVE
- E. WRITE DRIVE - IF ERROR, REPORT AND  
RETRY UP TO RETRY LIMIT.
- F. READ DRIVE - IF ERROR, REPORT AND  
RETRY UP TO RETRY LIMIT.
- G. DO DATA COMPARE FOR THE READS - IF ERROR, REPORT
- H. IF NOT DONE ALL DRIVES - GO TO D
- I. IF END OF PASS, REPORT AND GO TO B
- J. ELSE UPDATE STARTING ADDRESS, GO TO D

8. -----  
OPERATION OPTIONS

- SRI BIT 0 CLEAR(0):  
IF RETRY LIMIT IS EXCEEDED, CONTINUE WITH NEXT TEST.
- SRI BIT 0 SET(1):  
IF RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, REPORT A  
HARD ERROR AND DROP THE MODULE.
- SRI BIT 1 CLEAR(0):  
USE ALTERNATING DATA PATTERN OF ONES AND ZEROS.
- SRI BIT 1 SET(1):  
USE DECREMENTING DATA PATTERN STARTING WITH  
A RANDOM NUMBER.

9. NON-STANDARD PRINTOUTS  
-----

- A. ALL PRINTOUTS HAVE THE STANDARD FORMAT DESCRIBED IN THE DEC/X11 DOCUMENT.
- E. ERROR MESSAGES DUMP THE CONTENTS OF THE RX01 REGISTER IN THE FOLLOWING ORDER.  
RXCS (COMMAND REGISTER)  
RXES (ERROR REGISTER)  
RXSB (STATUS REGISTER)  
RYTA (STACK ADDRESS)  
RXSA (SECTOR ADDRESS)
- C. DATA ERRORS REPORT UP TO 3 BAD WORDS, THEN DISCONTINUE CHECK
- D. RETRIES: EACH WRITE OR READ STATUS ERROR IS ACCOMPANIED BY A RETRY NUMBER:  
RETRY 0: IS THE ORIGINAL ERROR  
RETRY 1: IS THE FIRST RETRY OF THAT ERROR (SAME ADDRESS)  
RETRY 2: IS THE SECOND RETRY OF THAT ERROR (SAME ADDRESS)  
NOW DROP THE MODULE IF SRI=1 OR  
CONTINUE TO NEXT ADDRESS IF SRI=0

!



```
245  
246  
247 000224* 012767 000100 177662 START: MOV #64, WDT0 ;64 WORDS TO MEM/ITERATION  
248 000232* 012767 000100 177656 MOV #0, WDFR ;64 WORDS FROM MEM/ITERATION  
249 000240* 012767 000002 177652 MOV #2, IINTR ;2 INTERRUPTS/ITERATION  
250 000246* 012767 000001 002522 MOV #1, TASAV ;PRESET TRACK ADDRESS SAVE  
251 000252* 012767 000001 002516 MOV #1, SASAV ;PRESET SECTOR ADDRESS  
252 000258* 012767 002510 002504 MOV #1, SASAV, TA ;"  
253 000270* 012767 002504 002476 MOV #1, SASAV, SA ;"  
254 000276* 012767 000001 002476 MOV #1, PCNTR ;PRESET INTERNAL PASS COUNTER = 0  
255 000304* 012767 000001 002474 MOV #0, WVFLG ;ONE INTERNAL PASS WITH DATA = 0  
256 000312* 012767 177476 002430 MOV DVI01, DVIDIX ;GET DRIVE COUNT  
257 000320* 122737 000010 000041 CMPB #10, #41 ;CHECK IF FLPPY WAS LOAD MEDIUM  
258 000326* 001012 000001 002412 BNE RESTR1 ;BR IF NO  
259 000330* 042767 000001 002412 BIC #1, DVIDIX ;KILL TEST FOR DRIVE 0  
260 000336* 032767 000001 177450 BIT #1, DVID1 ;WAS IT TO BE TESTED?  
261 000344* 001403 000000 003556* BEQ RESTR1 ;BR IF NO  
262 000346* 104403 000000 003556* MSGNS, BEGIN, NUNTO ;ASCII MESSAGE CALL WITH COMMON HEADER  
263  
264 000354* 016706 177452 RESTR1: MOV SPOINT, SP ;SET STACK POINTER  
265 000360* 012700 002752* MOV #UNITS, RO ;GET START OF TABLE  
266 000364* 012701 000010 000010 MOV #10, R1 ;SET SIZE OF TABLE  
267 000370* 005020 000000 000000 CLR (R0)+ ;CLEAR TABLES  
268 000374* 001375 000000 000000 BNE CFC ;SET IF DONE  
269 000376* 004767 000646 JSR PC, VSRT ;GC SETUP ADDRESS AND VECTOR  
270  
271 000402* 012777 640000 002400 MOV #40000, BRXCS ;INIT UNIT 0  
272 000410* 004767 002130 JSR PC, ANDN ;GC AWAIT DONE  
273 000414* 005777 002370 TST BRXCS ;GC ERROR ON INIT  
274 000420* 180056 000000 003530* BPL TGO ;IF NOT: BR  
275 000430* 017767 000000 003530* MSGNS, BEGIN, DRP1 ;ASCII MESSAGE CALL WITH COMMON HEADER  
276 000436* 052767 040000 002354 MOV #0, BRXCS ;GET BRXCS  
277 000444* 017767 002342 002350 BIT #40000, BRXCS ;SET BRXCS TO SHOW INIT COMMAND  
278 000448* 017767 002342 002350 MOV #4, SRXCS ;GET SRXCS  
279 000450* 017767 002342 002350 MOV #4, SRXSA ;GET SRXSA  
280 000460* 005067 002342 002350 CLR SRXTA ;SET TRACK ADDRESS  
281 000464* 016767 002330 177410 MOV SRXCS, ACSR ;SET CONTENTS OF BRXCS  
282 000472* 016767 002312 177400 MOV SRXCS, CSRA ;SET ADDRESS OF BRXCS  
283 000500* 005067 002312 177400 MOV SRXCS, ASTAT ;SET STATUS  
284 000506* 012777 000017 002274 MOV #17, BRXCS ;READ STATUS B  
285 000514* 004767 002024 JSR PC, ANDN ;GC AWAIT DONE  
286 000520* 017767 002266 002276 MOV #4, SRXSB ;GET STATUS B  
287 000526* 012767 000034 003032* HRDRS, BEGIN, TABLE ;INITIALIZE ERROR  
288  
289 000534* 104405 000000 003032* ;*****  
290 ;***** ;*****  
291 ;***** ;*****  
292 000542* 000167 000306 JMP FINI ;GC DROP MODULE  
293  
294 000546* 104403 000000 003556* LDROP: MSGNS, BEGIN, NUNTO ;ASCII MESSAGE CALL WITH COMMON HEADER  
295 000546* 004767 000000 003556* BR #1, ;GC Deselect UNIT 0  
296 000550* 032767 000001 002164 TGO: BIT #1, DVIDIX ;SEE IF UNIT 0  
297 000556* 001423 000000 002214 BEQ UNT0 ;IF NOT: BR  
298 000566* 012777 000013 002214 UNT0: MOV #13, BRXCS ;READ STATUS OF UNIT 0  
299 000574* 004767 001744 JSR PC, ANDN ;GC AWAIT DONE
```

```
301 000600* 017700 002206 MOV BRXDB, RO ;GET RXES  
302 000604* 032700 000200 BIT #0, RO ;GET RXES  
303 000610* 001014 000000 003556* BNE UNT1A ;IF SO: BR  
304 000612* 104403 000000 177260 MSGNS, BEGIN, NUNTO ;ASCII MESSAGE CALL WITH COMMON HEADER  
305 000620* 012767 000006 177260 MOV #4, ERRTP ;NOT AVAILABLE  
306  
307 000626* 104405 000000 000000 HRDRS, BEGIN, NULL ;UNIT 0 NOT AVAILABLE  
308 ;***** ;*****  
309 ;***** ;*****  
310 000634* 012767 177777 002110 UNT1: MOV #1, UNITS ;Deselect UNIT 0  
311 000638* 032767 000002 002100 UNT1A: BIT #2, DVIDIX ;SEE IF UNIT 1  
312 000650* 001423 000000 002130 BEQ UNT1 ;IF NOT: BR  
313 000652* 012777 000033 002130 MOV #33, BRXCS ;READ STATUS OF UNIT 1  
314 000660* 004767 001660 002122 JSR PC, ANDN ;GC AWAIT DONE  
315 000666* 032700 002122 002122 MOV BRXDB, RO ;READ RXES  
316 000670* 032700 000200 BIT #200, RO ;IF IDV  
317 000674* 001024 000000 003552* BNE TST ;IF SO: BR  
318 000676* 104403 000006 177174 MSGNS, BEGIN, NUNTO ;ASCII MESSAGE CALL WITH COMMON HEADER  
319 000704* 012767 000006 177174 MOV #4, ERRTP ;NOT AVAILABLE  
320  
321 000712* 104405 000000 000000 HRDRS, BEGIN, NULL ;UNIT 1 NOT AVAILABLE  
322 ;***** ;*****  
323 ;***** ;*****  
324 000720* 012767 177777 002026 1S: MOV #1, UNITS ;Deselect UNIT 1  
325 000726* 005767 002020 TST UNITS ;SEE IF UNIT 0 READY  
326 000732* 001405 000000 003536* BEQ TST ;IF SO: BR  
327 000734* 104403 000000 003536* MSGNS, BEGIN, DRP2 ;ASCII MESSAGE CALL WITH COMMON HEADER  
328 000742* 000167 000106 JMP FINI ;GC DROP MODULE  
329  
330 000746* 004767 000106 TST: JSR PC, DSUP ;SET UP THE DATA  
331 000752* 004767 000172 JSR PC, SELD ;GC SELECT UNIT  
332 000756* 012767 000259 MOV #260, RTYN ;RESET RETRY COUNTER  
333 000764* 004767 000522 JSR PC, WRITE ;GC WRITE  
334 000770* 012767 000260 MOV #260, RTYN ;SET RETRY LIMIT  
335 000776* 004767 000742 JSR PC, READ ;READ THE DATA  
336 001000* 004767 001362 JSR PC, DCHR ;CHECK THE DATA  
337 001006* 004767 000136 JSR PC, SELD ;SELECT ANOTHER DRIVE  
338 001012* 012767 000260 MOV #260, RTYN ;SET RETRY LIMIT  
339 001024* 004767 000466 JSR PC, WRITE ;WRITE AGAIN  
340 001028* 012767 000260 MOV #260, RTYN ;SET RETRY LIMIT  
341 001032* 004767 000706 JSR PC, READ ;HOW READ THE STUFF  
342 001036* 004767 001326 JSR PC, DCHR ;CHECK THE DATA  
343 001042* 104413 000000* ENDLTS, BEGIN ;SIGNAL END OF ITERATION.  
344 ;***** ;*****  
345 001046* 004767 000236 JSR PC, TSS ;MONITOR SHALL TEST END OF PASS  
346 001052* 000735 000000 000000 BR TST ;UPDATE TRACK & SECTOR  
347 ;***** ;*****  
348 ;***** ;*****  
349 ;***** ;*****  
350  
351 001054* 104410 000000* FINI: ENDS, BEGIN ;DROP THE MODULE  
352 ;***** ;*****  
353 ;***** ;*****  
354 001060* 012700 000100 DSUP: MOV #100, RO ;SET SIZE OF BUFFER  
355 001064* 012701 003602* MOV #WBUF, R1 ;SET START OF BUFFER  
356 001070* 005767 001712 TST WVFLG, R1 ;+INTERNAL PASS DATA = ZERO  
357 001074* 001402 BEQ 1S ;+ALLREADY DONE:BR  
358 001076* 005067 CLR #S ;+YES LOAD BUFFER  
359 001100* 000417 BR #S ;+WITH ZEROS
```

```

357 001102* 032767 000002 176706 1S: BTT #2,SR1 ;+ USE RANDOM DATA?
358 001110* 041417 ;+ NO: BRANCH
359 001116* 016703 000000*
360 001124* 010324 176732
361 001126* 005300
362 001130* 001374
363 001132* 000207
364 001140* 012703 125125
365 001142* 005300
366 001144* 001375
367 001146* 000207
370
371
372 ;SELECT UNIT FOR TEST*****
373
374 001150* 005767 001602 SELD: TST DRVN ;SEE IF DRIVE 0
375 001152* 001417 ;IF SO: BR
376 001156* 005767 001572 TST UNIT1 ;SEE IF UNIT 1 AVAILABLE
377 001162* 001403 ;IF SO: BR
378 001164* 005087 001566 CLR DRVN ;ELSE SET TO 0
379 001170* 005767 ;SET UNIT UNDER TEST
380 001172* 012767 000020 001562 1S: MOV #20,UTT ;SET UNIT UNDER TEST
381 001200* 012767 000261 002246 MOV #260,DNUM ;SET DRIVE NUMBER FOR PRINTS
382 001206* 005087 001544 CLR DRVN ;CLEAR DRIVE NUMBER
383 001210* 005767 ;RETURN
384 001214* 005767 001532 2S: TST UNIT0 ;SEE IF UNIT 0 AVAILABLE
385 001220* 001403 ;IF SO: BR
386 001222* 005167 001530 COM DRVN ;SWITCH UNITS
387 001224* 005767 BR SELD ;SELECT NEXT
388 001230* 005167 001522 3S: COM DRVN ;SWITCH UNITS
389 001234* 005087 001522 CLR UTT ;SELECT DRIVE 0
390 001240* 012767 000260 002206 MOV #260,DNUM ;SET DRIVE NUMBER FOR PRINTS
391 001246* 000207 ;RETURN
392
393 ;VECTOR SET UP
394
395 001250* 016700 176534 VSET: MOV VECTOR,RO
396 001254* 012770 001716 MOV #INTERRUPT,(RO)+ ;SET UP INTERRUPT HANDLER ADDRESS
397 001260* 116710 176526 MOVB BR1,(RO) ;SET BR1 LEVEL
398 001264* 005767 001516 MOV ADDL,RXCS ;SET ADDRESS OF RXCS
399 001270* 016767 176510 001512 MOV ADDR,RXDB ;SET ADDRESS OF RXCS
400 001300* 062767 000002 001504 ADD #2,RXDB ;SET ADDRESS OF DATA BUFFER
401 001306* 000207 ;RETURN
402
403 ;TRACK AND SECTOR UPDATE*****
404
405 001310* 062767 000003 001456 TSS: ADD #3,SA ;BUMP SECTOR ADDRESS
406 001316* 022767 000033 001450 CMP #35,SA ;SEE IF DONE SECTORS
407 001324* 101401 176526 BLOS #2 ;IF SO: BR
408 001326* 000470 ;RETURN
409 001330* 005767 001444 001436 2S: MOV #106,SA ;SET SA
410 001336* 062767 000012 001426 ADD #12,TA ;INCREMENT TRACK ADDRESS

```

```

413 001344* 022767 000114 001420 CMP #114,TA ;SEE IF DONE TRACKS
414 001352* 103056 105 ;IF NOT: BR
415 001354* 016767 001416 001410 MOV TASAV,TA ;RESET TRACK ADDRESS
416 001362* 005767 001414 INC BPNTR ;BUMP INTERNAL STARTING SECTOR
417 001364* 005767 001414 TST FFLG ;IS IT SPECIAL DATA PATTERN?
418 001372* 001411 51S: BEQ #1 ;NO:CONTINUE
419 001374* 005087 001406 ;YES:HAVE CYCLED THRU BOLT DRIVES ONCE
420 001376* 012767 000001 001372 MOV #1,SASAV ;NOW RESET POINTERS TO WRITE
421 001380* 112177 001320 MOV #1,TASAV ;SAME PLACES WITH REGULAR PATTERN
422 001414* 000427
423 001416* 022767 000004 001356 51S: CMP #4,PCNTR ;SEE IF DONE THREE
424 001422* 005767 ;IF SO: BUMP
425 001424* 003487 001350 001344 BR PCNTR,SASAV ;ELSE BUMP STARTING SECTOR ADDRESS
426 001434* 000417 9S: MOV #1,PCNTR ;SAVE TA AND SA AND LEAVE
427 001436* 012767 000001 001336 6S: MOV #1,PCNTR ;RESET PASS COUNTER
428 001440* 012767 000001 001326 MOV #1,SASAV ;RESET STARTING SECTOR
429 001442* 005767 001320 INC #1,TA ;RESET STARTING TRACK ADDRESS
430 001456* 022767 000013 001312 CMP #13,TASAV ;SEE IF DONE TRACKS
431 001464* 001003 9S: BNE #1 ;IF NOT: BR
432 001466* 012767 000001 001302 MOV #1,TASAV ;ELSE RESET TRACK STARTING ADDRESS
433 001468* 012767 001376 001270 MOV #16,TA ;SAVE TRACK ADDRESS
434 001502* 016767 001272 001264 9S: MOV SASAV,SA ;SAVE SECTOR
435 001510* 000207 10S: RTS PC
436
437 ;WRITE SUBROUTINE*****
438
439 001512* 012767 000001 001240 WRITE: MOV #1,CMD ;SET TO FILL BUFFER
440 001520* 012767 000001 001240 MOV #16,FFR1 ;SET WRITE FLAG
441 001522* 012701 003602 MOV #200,RO ;SET ADDRESS OF WRITE BUFFER
442 001532* 012700 000200 MOV #1,PCNTR ;SET SIZE OF BUFFER
443 001536* 016777 001216 1S: MOV CMD,RRXCS ;LOAD COMMAND
444 001544* 004767 001076 JSR #1,ANTR ;GO AWAIT TRANSFER READY
445 001546* 112177 001236 MOVB (R1),RRXDB ;FILL BUFFER
446 001554* 005300 DEC RO ;SEE IF DONE FILL
447 001556* 001372 BNE #1 ;IF NOT: BR
448 001560* 004767 000760 JSR PC,ANDN ;GO AWAIT DONE
449 001564* 005767 001220 TST RRXCS ;SEE IF ERROR
450 001570* 100025 BPL #1 ;IF NOT: BR
451 001572* 016767 001212 176300 MOV RRXCS,CSRA ;LOAD ADDRESS OF RXCS
452 001600* 022767 001154 176274 MOV CMD,ACSR ;LOAD CURRENT COMMAND
453 001606* 057767 001176 176266 BLS RRXCS,ACSR ;LOAD RXCS
454 001614* 017767 001172 176262 MOV RRXDB,ASTAT ;LOAD RXES
455 001622* 104403 000000* 003520 MSGNS,BEGIN,BFER ;ASCII MESSAGE CALL WITH COMMON HEADER
456 001630* 012767 000035 176250 MOV #3,ERRTYE ;SET ERROR TYPE
457
458 001636* 104405 000000* 000000 *****
459 ***** ;BUFFER FILL ERROR
460 *****
461 001644* 016767 001112 001106 2S: MOV UTT,CMD ;SELECT DRIVE
462 001652* 052767 000105 001100 BLS #105,CMD ;SET TO WRITE SECTOR + INT ENABLE
463 001660* 016777 001074 001122 MOV CMD,RRXCS ;LOAD COMMAND
464 001666* 004767 000754 JSR PC,ANTR ;GO AWAIT TRANSFER READY
465 001672* 016777 001076 001112 MOV SA,RRXDB ;LOAD SECTOR ADDRESS
466 001700* 004767 000742 JSR PC,ANTR ;GO AWAIT TRANSFER READY
467 001704* 016777 001062 001100 MOV TA,RRXDB ;LOAD TRACK ADDRESS
468 001712* 104400 000000* EXIT$,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

```

469 001716" NTRUPT:
470
471 001716" 000004 000000" 001724" ;IRQ$,BEGIN,1$ ; QUEUE UP TO CONTINUE AT 1$ AND RTI
472
473
474 001724" 004767 000072 1$: JSR PC,SECK ;GO CHECK FOR ERROR
475 001730" 016700 000054 MOV VECTOR,RO ;GET VECTOR ADDRESS
476 001734" 005720 TST (RO) ;BUMP POINTER
477 001736" 016710 176050 MOV BRL,(RO) ;ASSURE RESET TO BR LEVEL 5
478 001742" 000207 RTS ;RETURN
479
480 ;READ SUBROUTINE*****
481
482 001744" 005067 001016 READ: CLR WTF,CMD ;CLEAR WRITE FLAG
483 001750" 016767 001006 MOV #107,CMD ;SELECT DRIVE
484 001756" 052767 000107 BIS #107,CMD ;SET READ COMMAND + INT ENB
485 001764" 016777 000770 001016 MOV CMD,BRXCS ;LOAD COMMAND
486 001772" 004767 000650 JSR PC,ANTR ;GO AWAIT TRANSFER READY
487 001776" 016777 000772 001006 JSR PC,BRXDB ;LOAD SECTOR ADDRESS
488 002004" 004767 000636 JSR PC,ANTR ;GO AWAIT TRANSFER READY
489 002010" 016777 000756 000774 MOV TR,BRXDB ;LOAD TRACK ADDRESS
490 002016" 104400 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
491
492 ;STATUS ERROR CHECK SUBROUTINE*****
493
494 002022" 005967 000736 SECK: CLR SERFL ;CLEAR STATUS ERROR FLAG
495 002026" 057767 000720 BIS BRXCS,CMD ;GET RXCS
496 002034" 005767 000720 TST CMD ;SEE IF ERROR
497 002040" 100401 BMI 1$ ;IF SO: BR
498 002044" 000907 RTS ;ELSE RETURN
499 002052" 016767 000702 000746 1$: MOV CMD,SRXCS ;LOAD RXCS
500 002060" 016767 000724 176012 MOV RXCS,CSRA ;LOAD RXCS
501 002066" 016767 000700 000732 MOV TA,SRXTA ;LOAD ADDRESS OF RXCS
502 002072" 016767 000674 000726 MOV SA,SRXSA ;LOAD SECTOR
503 002102" 017767 000704 000712 MOV SRXDB,SRXES ;LOAD RXES
504 002110" 016767 000706 175766 MOV SRXDB,SRXSTAT ;LOAD RXES
505 002118" 016767 000640 000634 MOV UTT,CMD ;SELECT DRIVE
506 002124" 016767 000617 000626 BIS #17,CMD ;LOAD READ STATUS B COMMAND
507 002132" 016777 000622 000650 JSR PC,BRXCS ;EXECUTE COMMAND
508 002140" 004767 000400 JSR PC,ANTR ;GO AWAIT DONE
509 002144" 017767 000642 000652 MOV BRXDB,SRXSB ;LOAD STATUS B
510 002156" 001013 TST BNE ;SEE IF WRITE ERROR
511 002160" 104403 BNE ;IF SO: BR
512 002166" 012767 000036 175712 MSGNS,BEGIN,RTER ;ASCII MESSAGE CALL WITH COMMON HEADER
513
514 002174" 104405 000000" 003032" ;*****
515
516 002202" 000167 000022 2$: HDRER$,BEGIN,TABLE ;READ ERROR
517
518 002206" 104403 000000" 003464" JMP CLEAR ;GO CLEAR ERROR
519
520 002214" 012767 000037 175664 MSGNS,BEGIN,RTER ;ASCII MESSAGE CALL WITH COMMON HEADER
521
522 002222" 104405 000000" 003032" HDRER$,BEGIN,TABLE ;WRITE ERROR
523
524 ;CLEAR ERRORS*****

```

```

525 002230" 005267 000530 CLEAR: INC SERFL ;SET STATUS ERROR FLAG
526 002234" 005767 000564 BNE SRXSB ;SEE IF SB ERROR
527 002240" 001415 BEQ 1$ ;IF NOT: BR
528 002242" 012777 040000 000540 MOV #40000,BRXCS ;INITIALIZE
529 002246" 005767 000070 000530 JSR PC,ANDN ;GO AWAIT DONE
530 002252" 005767 000530 TST BRXCS ;SEE IF ERROR CN INIT
531 002260" 100005 BNE 1$ ;IF NOT: BR
532 002262" 104403 000000" 003544" MSGNS,BEGIN,DRP3 ;ASCII MESSAGE CALL WITH COMMON HEADER
533 002266" 005267 000530 JMP TDRC ;GO DROP MODULE
534 002274" 005267 001164 2$: INC RTYN ;BUMP RETRY COUNTER
535 002300" 016703 001154 MOV R3,RTYN ;GET COUNTER
536 002304" 042703 177770 BIC R1,177770,R3 ;MASK ASCII
537 002308" 042703 000003 CMP #3,R3 ;SEE IF DONE RETRIES
538 002314" 001011 BNE 2$ ;IF NOT: BR
539 002316" 104403 000000" 003552" MSGNS,BEGIN,HRDE ;ASCII MESSAGE CALL WITH COMMON HEADER
540 002324" 032767 000001 175464 BIT #1,SR1 ;SEE IF SHOULD DROP MODULE
541 002332" 001412 BNE 3$ ;IF NOT: BR
542 002334" 000167 176514 JMP FST ;GO DROP MODULE
543 002340" 005726 TST (SP)+ ;RESET STACK
544 002342" 005767 000420 TST WTF ;SEE IF WRITE TIME
545 002350" 000167 177136 BEQ 4$ ;IF NOT: BR
546 002354" 000167 177364 JMP WRITE ;RETRY WRITE
547 002360" 001267 000260 001072 4$: JMP READ ;RETRY READ
548 002366" 000207 RTS ;RESET RETRY COUNTER
549
550 ;DATA CHECK SUBROUTINE*****
551
552 002370" 005767 000370 DCHK: TST SERFL ;SEE IF STATUS ERROR
553 002374" 001401 BEQ 1$ ;IF NOT: BR
554 002376" 000207 RTS ;EXIT IF STATUS ERROR
555 002400" 005067 000364 CLR BMCNT ;CLEAR BAD WORD COUNTER
556 002404" 014700 MOV #200,RO ;SET SIZE OF BUFFER
557 002410" 014701 #RBUF,R1 ;ADDRESS OF READ BUFFER
558 002414" 012777 000003 MOV #3,BRXCS ;LOAD READ BUFFER COMMAND
559 002422" 004767 000220 2$: JSR PC,ANTR ;GO AWAIT TRANSFER READY
560 002430" 005767 000360 MOV#B,BRXDB,(R1)+ ;EMPTY BUFFER
561 002434" 001372 DEC RO ;SEE IF EMPTIED ENTIRE BUFFER
562 002436" 004767 000102 3$: JSR PC,ANDN ;IF NOT: BR
563 002444" 012701 003602" CLR RO ;GO AWAIT DONE
564 002450" 012702 004004" MOV #RBUF,R1 ;SET ADDRESS OF WRITE DATA
565 002454" 021112 #RBUF,R2 ;SET ADDRESS OF READ DATA
566 002458" 001006 CMP (R1),(R2) ;TEST DATA
567 002462" 005200 BNE 6$ ;IF DATA BAD: BR
568 002464" 022700 INC RO ;BUMP ADDRESS OF DATA POINTER
569 002470" 001377 000100 4$: CMP #100,RO ;BUMP WORD COUNTER
570 002474" 001016 175402 6$: RTS ;SEE IF DONE ALL
571 002500" 010267 175400 MOV R1,SBADR ;CHECK ALL
572 002504" 011167 175376 MOV (R1),ASB ;IF DONE: EXIT
573 002510" 011167 175374 MOV (R1),ASADR ;LOAD GOOD ADDRESS
574 002514" 104403 000000" 003510" MOV (R2),ASB ;LOAD BAD ADDRESS
575 MSGNS,BEGIN,RTER ;ASCII MESSAGE CALL WITH COMMON HEADER
576
577
578
579
580

```



```

581 002522 104404 000000
582
583 002526 005267 000236
584 002532 022767 000003 000230
585 002540 001347
586 002542 000207
587
588
589
590 002544 005067 000234
591 002550 032777 000040 000232
592 002556 001032
593 002560 104407 000000
594 002564 104407 000000
595 002570 005267 000210
596 002574 001365
597 002578 005067 175302
598 002605 016767 000144 175270
599 002610 016767 000144 175264
600 002616 104403 000000 003566
601 002624 012767 000040 175254
602
603 002632 104405 000000 000000
604
605 002640 000167 176210
606 002644 000207
607
608
609
610 002646 005067 000132
611 002652 032777 000200 000130
612 002660 001032
613 002664 104407 000000
614 002666 104407 000000
615 002672 005267 000106
616 002676 001365
617 002700 005067 175200
618 002704 016767 000100 175166
619 002712 016767 000042 175162
620 002720 104403 000000 003574
621 002726 012767 000040 175152
622
623 002734 104405 000000 000000
624
625 002742 000167 176106
626 002746 000207
627
628
629
630
631 002750 000000
632 002752 000000
633 002754 000000
634 002756 000000
635 002760 000000
636 002762 000000

```

```

DATER$,BEGIN
;*****CSRA*****
;*****DATA ERROR!!!*****
;*****SEE IF DONE SET*****
INC BWCNT ;BUMP BAD WORD COUNTER
CMP #3,BWCNT ;SEE IF 3 BAD WORDS
BNE $S ;IF NOT: BR
RTS PC ;RETURN
;AWAIT DONE BIT SUBROUTINE*****
;PRESET TIME OUT COUNTER
BIT #40,@RXCS ;SEE IF DONE SET
BNE $S ;IF $0: BR
BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR.....
BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
INC TOCNT ;BUMP TIME OUT COUNTER
BNE $S ;IF NOT TIMED OUT: BR
CLR A$STAT ;CLEAR STATUS WORD
MOV RXCS,CSRA ;SET ADDRESS OF RXCS
MOV CMD,CSR ;SET COMMAND WORD
MSGNS$,BEGIN,D$T0 ;ASCII MESSAGE CALL WITH COMMON HEADER
MOV #40,ERRTYP
;*****
;*****DCNE BIT TIME OUT*****
;*****DROP MODULE*****
;*****EXIT*****
JMP FINI ;DROP MODULE
RTS PC ;EXIT
;AWAIT TRANSFER READY SUBROUTINE*****
;PRESET TIME OUT COUNTER
BIT #200,@RXCS ;SEE IF TRANSFER READY SET
BNE $S ;IF $0: BR
BREAK$,BEGIN ;TEMPORARY RETURN TO MONITOR.....
BREAK$,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
INC TOCNT ;BUMP TIME OUT COUNTER
BNE $S ;IF NOT TIMED OUT: BR
CLR A$STAT ;CLEAR STATUS WORD
MOV RXCS,CSRA ;SET ADDRESS OF RXCS
MOV CMD,CSR ;SET COMMAND WORD
MSGNS$,BEGIN,TR$T0 ;ASCII MESSAGE CALL WITH COMMON HEADER
MOV #40,ERRTYP
;*****
;*****TRANSFER READY TIME OUT*****
;*****GO DROP MODULE*****
;*****RETURN*****
JMP FINI ;GO DROP MODULE
RTS PC ;RETURN
;FLAGS AND COUNTERS*****
DVID1X: 0 ;HOLDS WHICH DRIVES TO TEST
UNIT0: 0 ;UNIT 0 FLAG
UNIT1: 0 ;UNIT 1 FLAG
DRVN: 0 ;DRIVE NUMBER
CMD: 0 ;COMMAND SAVE
UTT: 0 ;UNIT UNDER TEST

```

```

637 002764 000000
638 002766 000000
639 002770 000000
640 002772 000000
641 002774 000000
642 002776 000000
643 003000 000000
644 003002 000000
645 003004 000000
646 003006 000001
647
648
649
650
651
652
653
654
655

```

```

SERFL: 0 ;STATUS ERROR FLAG
WTF: 0 ;WRITE FLAG
BWCNT: 0 ;BAD WORD COUNTER
TA: 0 ;CURRENT TRACK ADDRESS
SA: 0 ;CURRENT SECTOR ADDRESS
TASAV: 0 ;STARTING TRACK ADDRESS SAVE
SASAV: 0 ;STARTING SECTOR ADDRESS SAVE
PCNTR: 0 ;INTERNAL PASS COUNTER
TOCNT: 0 ;TIME OUT COUNTER
WVFLG: 1 ;WRITE VERIFY FLAG
;
; NOT EQ ZERO CA] DISABLE INTERNAL PASS
; TRACK & SECTOR OFFSETTING.
;
; EQUAL ZERO CA] ALLOW INTERNAL PASS
; TRACK & SECTOR OFFSETTING.
;
; USE DATA PATTERN DEFINED
; BY $1.
; (RANDOM OR ONES AND ZEROS).

```

```
656 ;CONSTANTS*****
657
658
659 003010* 177170 SRXCS: 177170 ;RX01 COMMAND REGISTER
660 003012* 177172 SRXDB: 177172 ;RX01 DATA BUFFER
661 003014* 000264 VEC: 264 ;VECTOR
662 003016* 000001 TPCNTR: 1 ;TOTAL PASS COUNTER
663
664 ;VARIABLES*****
665
666 003020* 000000 SRXCS: 0
667 003022* 000000 SRXES: 0
668 003024* 000000 SRXSB: 0
669 003026* 000000 SRXTA: 0
670 003030* 000000 SRXSA: 0
671
672 ;MESSAGE TABLE*****
673
674
675 003032* TABLE:
676 003032* 003020* ARXCS: SRXCS
677 003034* 003022* ARXES: SRXES
678 003036* 003024* ARXSB: SRXSB
679 003038* 003026* ARXTA: SRXTA
680 003042* 003030* ARXSA: SRXSA
681 003044* 177777 -1
682
683 003046* 053440 044522 042524 MSG1: .ASCIZ " WRITE ERROR: RETRY:"
684 003054* 042440 051122 051117
685 003062* 020072 042522 051124
686 003070* 035131 000
687 003072* 025040 047522 042101 MSG2: .ASCIZ " READ ERROR: RETRY:"
688 003100* 042440 051122 051117
689 003106* 020072 042522 051124
690 003114* 035131 000
691 003116* 025040 051100 MSG3: .ASCIZ " DRIVE "
692 003124* 020105 000
693 003127* 040 051104 050117 MSG4: .ASCIZ " DROP MODULE%"
694 003132* 046440 042117 046125
695 003133* 025040 000
696 003145* 045 044440 044516 MSG5: .ASCIZ "% INITIALIZE ERROR:"
697 003152* 044524 046101 055111
698 003160* 020105 051100 047522
699 003166* 035131 000
700 003171* 045 047040 020117 MSG6: .ASCIZ "% NO UNITS TO TEST:"
701 003176* 047125 052111 020123
702 003204* 047524 052044 051505
703 003205* 035131 000
704 003215* 045 051040 044505 MSG7: .ASCIZ "% REINITIALIZE ERROR:"
705 003222* 044516 044524 046101
706 003230* 030440 047125 051105
707 003233* 047125 020105 051105
708 003243* 040 040504 040524 MSG8: .ASCIZ " DATA ERROR:"
709 003250* 042440 051122 051117
710 003256* 000072 000
711 003260* 043046 046111 020114 MSG9: .ASCIZ " FILL BUFFER ERROR:"
```

```
712 003266* 052502 043106 051105
713 003274* 042440 051122 051117
714 003304* 000072 000
715 003308* 040440 051101 020104 MSG10: .ASCIZ " HARD ERROR:"
716 003312* 051105 047522 035122
717 003320* 000
718 003324* 030440 047125 052111 MSG11: .ASCIZ " UNIT 0 NOT AVAILABLE"
719 003328* 040440 047040 052117
720 003334* 040440 040526 046111
721 003342* 041101 042514 000
722 003352* 030440 047125 052111 MSG12: .ASCIZ " UNIT 1 NOT AVAILABLE"
723 003358* 030440 047040 052117
724 003362* 040440 040526 046111
725 003370* 041101 042514 000
726 003375* 040 047504 042516 MSG13: .ASCIZ " DONE BIT TIME OUT:"
727 003400* 041040 052111 052040
728 003410* 046511 020105 052517
729 003416* 035124 000
730 003422* 040 051124 047101 MSG14: .ASCIZ " TRANSFER READY TIME OUT:"
731 003426* 043123 051105 051040
732 003434* 040505 054504 052040
733 003442* 046511 020105 052517
734 003450* 035124 000
735 003454* 033454*
736 003454* 030040 000 DNUM: .EVEN
737 003460* 000 RTYN: .ASCIZ " 0"
738 003460* 030040 000
739
740 003464* 003464* WTER: .EVEN
741 003464* 003117* MSG3
742 003466* 003454* DNUM
743 003470* 003046* MSG1
744 003472* 003460* RTYN
745 003474* 177777* -1
746 003476* 003117* RTER: .EVEN
747 003500* 003454* MSG3
748 003502* 003072* DNUM
749 003504* 003460* MSG2
750 003506* 177777* RTYN
751 003510* 003117* -1
752 003512* 003454* DTER: .EVEN
753 003514* 003243* MSG3
754 003516* 177777* DNUM
755 003520* 003117* MSG8
756 003522* 003454* BFER: .EVEN
757 003524* 003260* DNUM
758 003526* 177777* MSG9
759 003530* 003117* -1
760 003532* 003117* DRP1: .EVEN
761 003534* 177777* MSG5
762 003536* 003117* -1
763 003540* 003117* DRP2: .EVEN
764 003542* 003117* MSG6
765 003544* 003215* -1
766 003546* 003127* DRP3: .EVEN
767 003550* 177777* MSG4
768 -1
```

768 003552\* 003304\*  
 769 003554\* 177777\*  
 770 003556\* 003321\*  
 771 003560\* 177777\*  
 772 003562\* 003347\*  
 773 003564\* 177777\*  
 774 003566\* 003375\*  
 775 003570\* 003177\*  
 776 003572\* 177777\*  
 777 003574\* 003421\*  
 778 003576\* 003177\*  
 779 003600\* 177777\*  
 780  
 781  
 782  
 783 003602\* 000000\*  
 784 004004\* 004004\*  
 785 004004\* 000000\*  
 786 004206\* 004206\*  
 787  
 788 000001

HRDE: MSG10  
 -1  
 NUNT0: MSG11  
 -1  
 NUNT1: MSG12  
 -1  
 DNT0: MSG13  
 MSG4  
 TRTO: MSG14  
 MSG4  
 -1  
 .EVEN  
 WBUF: 0 ;WRITE BUFFER  
 RBUF: 0 ;READ BUFFER  
 -=-+200  
 -=-+200  
 .END

ACSR 000102R 277#  
 ADDR 000006R 193#  
 ADDR22= 001000 245#  
 ARXS 003032R 279#  
 ARXS 003034R 279#  
 ARXSA 003042R 680#  
 ARXSB 003036R 678#  
 ARXTA 003040R 571#  
 ASB 000106R 229#  
 ASTAT 000104R 284\*  
 AWAS 000110R 578\*  
 ANDN 002548R 232#  
 ANTR 002646R 444#  
 BECN 000000R 190#  
 255#  
 595#  
 755#  
 BFER 003520R 245#  
 BIT0 000001 245#  
 BIT1 000002 245#  
 BIT10 002002 245#  
 BIT11 004000 245#  
 BIT12 010000 245#  
 BIT13 020000 245#  
 BIT14 040000 245#  
 BIT15 100000 245#  
 BIT2 000004 245#  
 BIT3 000010 245#  
 BIT4 000020 245#  
 BIT5 000040 245#  
 BIT6 000100 245#  
 BIT7 000200 245#  
 BIT8 000400 245#  
 BIT9 001000 245#  
 BREAK = 104407 245#  
 BR1 000012R 193#  
 BR2 000013R 193#  
 BTODS = 104421 193#  
 BWCNT = 002770R 555#  
 CDATA = 104415 245#  
 CFC 00370R 269#  
 CLEAR 00230R 255#  
 CMD 002760R 433\*  
 505\*  
 517\*  
 521\*  
 525\*  
 529\*  
 533\*  
 537\*  
 541\*  
 545\*  
 549\*  
 553\*  
 557\*  
 561\*  
 565\*  
 569\*  
 573\*  
 577\*  
 581\*  
 585\*  
 589\*  
 593\*  
 597\*  
 601\*  
 605\*  
 609\*  
 613\*  
 617\*  
 621\*  
 625\*  
 629\*  
 633\*  
 637\*  
 641\*  
 645\*  
 649\*  
 653\*  
 657\*  
 661\*  
 665\*  
 669\*  
 673\*  
 677\*  
 681\*  
 685\*  
 689\*  
 693\*  
 697\*  
 701\*  
 705\*  
 709\*  
 713\*  
 717\*  
 721\*  
 725\*  
 729\*  
 733\*  
 737\*  
 741\*  
 745\*  
 749\*  
 753\*  
 757\*  
 761\*  
 765\*  
 769\*  
 773\*  
 777\*  
 781\*  
 785\*  
 789\*  
 793\*  
 797\*  
 801\*  
 805\*  
 809\*  
 813\*  
 817\*  
 821\*  
 825\*  
 829\*  
 833\*  
 837\*  
 841\*  
 845\*  
 849\*  
 853\*  
 857\*  
 861\*  
 865\*  
 869\*  
 873\*  
 877\*  
 881\*  
 885\*  
 889\*  
 893\*  
 897\*  
 901\*  
 905\*  
 909\*  
 913\*  
 917\*  
 921\*  
 925\*  
 929\*  
 933\*  
 937\*  
 941\*  
 945\*  
 949\*  
 953\*  
 957\*  
 961\*  
 965\*  
 969\*  
 973\*  
 977\*  
 981\*  
 985\*  
 989\*  
 993\*  
 997\*  
 1001\*  
 1005\*  
 1009\*  
 1013\*  
 1017\*  
 1021\*  
 1025\*  
 1029\*  
 1033\*  
 1037\*  
 1041\*  
 1045\*  
 1049\*  
 1053\*  
 1057\*  
 1061\*  
 1065\*  
 1069\*  
 1073\*  
 1077\*  
 1081\*  
 1085\*  
 1089\*  
 1093\*  
 1097\*  
 1101\*  
 1105\*  
 1109\*  
 1113\*  
 1117\*  
 1121\*  
 1125\*  
 1129\*  
 1133\*  
 1137\*  
 1141\*  
 1145\*  
 1149\*  
 1153\*  
 1157\*  
 1161\*  
 1165\*  
 1169\*  
 1173\*  
 1177\*  
 1181\*  
 1185\*  
 1189\*  
 1193\*  
 1197\*  
 1201\*  
 1205\*  
 1209\*  
 1213\*  
 1217\*  
 1221\*  
 1225\*  
 1229\*  
 1233\*  
 1237\*  
 1241\*  
 1245\*  
 1249\*  
 1253\*  
 1257\*  
 1261\*  
 1265\*  
 1269\*  
 1273\*  
 1277\*  
 1281\*  
 1285\*  
 1289\*  
 1293\*  
 1297\*  
 1301\*  
 1305\*  
 1309\*  
 1313\*  
 1317\*  
 1321\*  
 1325\*  
 1329\*  
 1333\*  
 1337\*  
 1341\*  
 1345\*  
 1349\*  
 1353\*  
 1357\*  
 1361\*  
 1365\*  
 1369\*  
 1373\*  
 1377\*  
 1381\*  
 1385\*  
 1389\*  
 1393\*  
 1397\*  
 1401\*  
 1405\*  
 1409\*  
 1413\*  
 1417\*  
 1421\*  
 1425\*  
 1429\*  
 1433\*  
 1437\*  
 1441\*  
 1445\*  
 1449\*  
 1453\*  
 1457\*  
 1461\*  
 1465\*  
 1469\*  
 1473\*  
 1477\*  
 1481\*  
 1485\*  
 1489\*  
 1493\*  
 1497\*  
 1501\*  
 1505\*  
 1509\*  
 1513\*  
 1517\*  
 1521\*  
 1525\*  
 1529\*  
 1533\*  
 1537\*  
 1541\*  
 1545\*  
 1549\*  
 1553\*  
 1557\*  
 1561\*  
 1565\*  
 1569\*  
 1573\*  
 1577\*  
 1581\*  
 1585\*  
 1589\*  
 1593\*  
 1597\*  
 1601\*  
 1605\*  
 1609\*  
 1613\*  
 1617\*  
 1621\*  
 1625\*  
 1629\*  
 1633\*  
 1637\*  
 1641\*  
 1645\*  
 1649\*  
 1653\*  
 1657\*  
 1661\*  
 1665\*  
 1669\*  
 1673\*  
 1677\*  
 1681\*  
 1685\*  
 1689\*  
 1693\*  
 1697\*  
 1701\*  
 1705\*  
 1709\*  
 1713\*  
 1717\*  
 1721\*  
 1725\*  
 1729\*  
 1733\*  
 1737\*  
 1741\*  
 1745\*  
 1749\*  
 1753\*  
 1757\*  
 1761\*  
 1765\*  
 1769\*  
 1773\*  
 1777\*  
 1781\*  
 1785\*  
 1789\*  
 1793\*  
 1797\*  
 1801\*  
 1805\*  
 1809\*  
 1813\*  
 1817\*  
 1821\*  
 1825\*  
 1829\*  
 1833\*  
 1837\*  
 1841\*  
 1845\*  
 1849\*  
 1853\*  
 1857\*  
 1861\*  
 1865\*  
 1869\*  
 1873\*  
 1877\*  
 1881\*  
 1885\*  
 1889\*  
 1893\*  
 1897\*  
 1901\*  
 1905\*  
 1909\*  
 1913\*  
 1917\*  
 1921\*  
 1925\*  
 1929\*  
 1933\*  
 1937\*  
 1941\*  
 1945\*  
 1949\*  
 1953\*  
 1957\*  
 1961\*  
 1965\*  
 1969\*  
 1973\*  
 1977\*  
 1981\*  
 1985\*  
 1989\*  
 1993\*  
 1997\*  
 2001\*  
 2005\*  
 2009\*  
 2013\*  
 2017\*  
 2021\*  
 2025\*  
 2029\*  
 2033\*  
 2037\*  
 2041\*  
 2045\*  
 2049\*  
 2053\*  
 2057\*  
 2061\*  
 2065\*  
 2069\*  
 2073\*  
 2077\*  
 2081\*  
 2085\*  
 2089\*  
 2093\*  
 2097\*  
 2101\*  
 2105\*  
 2109\*  
 2113\*  
 2117\*  
 2121\*  
 2125\*  
 2129\*  
 2133\*  
 2137\*  
 2141\*  
 2145\*  
 2149\*  
 2153\*  
 2157\*  
 2161\*  
 2165\*  
 2169\*  
 2173\*  
 2177\*  
 2181\*  
 2185\*  
 2189\*  
 2193\*  
 2197\*  
 2201\*  
 2205\*  
 2209\*  
 2213\*  
 2217\*  
 2221\*  
 2225\*  
 2229\*  
 2233\*  
 2237\*  
 2241\*  
 2245\*  
 2249\*  
 2253\*  
 2257\*  
 2261\*  
 2265\*  
 2269\*  
 2273\*  
 2277\*  
 2281\*  
 2285\*  
 2289\*  
 2293\*  
 2297\*  
 2301\*  
 2305\*  
 2309\*  
 2313\*  
 2317\*  
 2321\*  
 2325\*  
 2329\*  
 2333\*  
 2337\*  
 2341\*  
 2345\*  
 2349\*  
 2353\*  
 2357\*  
 2361\*  
 2365\*  
 2369\*  
 2373\*  
 2377\*  
 2381\*  
 2385\*  
 2389\*  
 2393\*  
 2397\*  
 2401\*  
 2405\*  
 2409\*  
 2413\*  
 2417\*  
 2421\*  
 2425\*  
 2429\*  
 2433\*  
 2437\*  
 2441\*  
 2445\*  
 2449\*  
 2453\*  
 2457\*  
 2461\*  
 2465\*  
 2469\*  
 2473\*  
 2477\*  
 2481\*  
 2485\*  
 2489\*  
 2493\*  
 2497\*  
 2501\*  
 2505\*  
 2509\*  
 2513\*  
 2517\*  
 2521\*  
 2525\*  
 2529\*  
 2533\*  
 2537\*  
 2541\*  
 2545\*  
 2549\*  
 2553\*  
 2557\*  
 2561\*  
 2565\*  
 2569\*  
 2573\*  
 2577\*  
 2581\*  
 2585\*  
 2589\*  
 2593\*  
 2597\*  
 2601\*  
 2605\*  
 2609\*  
 2613\*  
 2617\*  
 2621\*  
 2625\*  
 2629\*  
 2633\*  
 2637\*  
 2641\*  
 2645\*  
 2649\*  
 2653\*  
 2657\*  
 2661\*  
 2665\*  
 2669\*  
 2673\*  
 2677\*  
 2681\*  
 2685\*  
 2689\*  
 2693\*  
 2697\*  
 2701\*  
 2705\*  
 2709\*  
 2713\*  
 2717\*  
 2721\*  
 2725\*  
 2729\*  
 2733\*  
 2737\*  
 2741\*  
 2745\*  
 2749\*  
 2753\*  
 2757\*  
 2761\*  
 2765\*  
 2769\*  
 2773\*  
 2777\*  
 2781\*  
 2785\*  
 2789\*  
 2793\*  
 2797\*  
 2801\*  
 2805\*  
 2809\*  
 2813\*  
 2817\*  
 2821\*  
 2825\*  
 2829\*  
 2833\*  
 2837\*  
 2841\*  
 2845\*  
 2849\*  
 2853\*  
 2857\*  
 2861\*  
 2865\*  
 2869\*  
 2873\*  
 2877\*  
 2881\*  
 2885\*  
 2889\*  
 2893\*  
 2897\*  
 2901\*  
 2905\*  
 2909\*  
 2913\*  
 2917\*  
 2921\*  
 2925\*  
 2929\*  
 2933\*  
 2937\*  
 2941\*  
 2945\*  
 2949\*  
 2953\*  
 2957\*  
 2961\*  
 2965\*  
 2969\*  
 2973\*  
 2977\*  
 2981\*  
 2985\*  
 2989\*  
 2993\*  
 2997\*  
 3001\*  
 3005\*  
 3009\*  
 3013\*  
 3017\*  
 3021\*  
 3025\*  
 3029\*  
 3033\*  
 3037\*  
 3041\*  
 3045\*  
 3049\*  
 3053\*  
 3057\*  
 3061\*  
 3065\*  
 3069\*  
 3073\*  
 3077\*  
 3081\*  
 3085\*  
 3089\*  
 3093\*  
 3097\*  
 3101\*  
 3105\*  
 3109\*  
 3113\*  
 3117\*  
 3121\*  
 3125\*  
 3129\*  
 3133\*  
 3137\*  
 3141\*  
 3145\*  
 3149\*  
 3153\*  
 3157\*  
 3161\*  
 3165\*  
 3169\*  
 3173\*  
 3177\*  
 3181\*  
 3185\*  
 3189\*  
 3193\*  
 3197\*  
 3201\*  
 3205\*  
 3209\*  
 3213\*  
 3217\*  
 3221\*  
 3225\*  
 3229\*  
 3233\*  
 3237\*  
 3241\*  
 3245\*  
 3249\*  
 3253\*  
 3257\*  
 3261\*  
 3265\*  
 3269\*  
 3273\*  
 3277\*  
 3281\*  
 3285\*  
 3289\*  
 3293\*  
 3297\*  
 3301\*  
 3305\*  
 3309\*  
 3313\*  
 3317\*  
 3321\*  
 3325\*  
 3329\*  
 3333\*  
 3337\*  
 3341\*  
 3345\*  
 3349\*  
 3353\*  
 3357\*  
 3361\*  
 3365\*  
 3369\*  
 3373\*  
 3377\*  
 3381\*  
 3385\*  
 3389\*  
 3393\*  
 3397\*  
 3401\*  
 3405\*  
 3409\*  
 3413\*  
 3417\*  
 3421\*  
 3425\*  
 3429\*  
 3433\*  
 3437\*  
 3441\*  
 3445\*  
 3449\*  
 3453\*  
 3457\*  
 3461\*  
 3465\*  
 3469\*  
 3473\*  
 3477\*  
 3481\*  
 3485\*  
 3489\*  
 3493\*  
 3497\*  
 3501\*  
 3505\*  
 3509\*  
 3513\*  
 3517\*  
 3521\*  
 3525\*  
 3529\*  
 3533\*  
 3537\*  
 3541\*  
 3545\*  
 3549\*  
 3553\*  
 3557\*  
 3561\*  
 3565\*  
 3569\*  
 3573\*  
 3577\*  
 3581\*  
 3585\*  
 3589\*  
 3593\*  
 3597\*  
 3601\*  
 3605\*  
 3609\*  
 3613\*  
 3617\*  
 3621\*  
 3625\*  
 3629\*  
 3633\*  
 3637\*  
 3641\*  
 3645\*  
 3649\*  
 3653\*  
 3657\*  
 3661\*  
 3665\*  
 3669\*  
 3673\*  
 3677\*  
 3681\*  
 3685\*  
 3689\*  
 3693\*  
 3697\*  
 3701\*  
 3705\*  
 3709\*  
 3713\*  
 3717\*  
 3721\*  
 3725\*  
 3729\*  
 3733\*  
 3737\*  
 3741\*  
 3745\*  
 3749\*  
 3753\*  
 3757\*  
 3761\*  
 3765\*  
 3769\*  
 3773\*  
 3777\*  
 3781\*  
 3785\*  
 3789\*  
 3793\*  
 3797\*  
 3801\*  
 3805\*  
 3809\*  
 3813\*  
 3817\*  
 3821\*  
 3825\*  
 3829\*  
 3833\*  
 3837\*  
 3841\*  
 3845\*  
 3849\*  
 3853\*  
 3857\*  
 3861\*  
 3865\*  
 3869\*  
 3873\*  
 3877\*  
 3881\*  
 3885\*  
 3889\*  
 3893\*  
 3897\*  
 3901\*  
 3905\*  
 3909\*  
 3913\*  
 3917\*  
 3921\*  
 3925\*  
 3929\*  
 3933\*  
 3937\*  
 3941\*  
 3945\*  
 3949\*  
 3953\*  
 3957\*  
 3961\*  
 3965\*  
 3969\*  
 3973\*  
 3977\*  
 3981\*  
 3985\*  
 3989\*  
 3993\*  
 3997\*  
 4001\*  
 4005\*  
 4009\*  
 4013\*  
 4017\*  
 4021\*  
 4025\*  
 4029\*  
 4033\*  
 4037\*  
 4041\*  
 4045\*  
 4049\*  
 4053\*  
 4057\*  
 4061\*  
 4065\*  
 4069\*  
 4073\*  
 4077\*  
 4081\*  
 4085\*  
 4089\*  
 4093\*  
 4097\*  
 4101\*  
 4105\*  
 4109\*  
 4113\*  
 4117\*  
 4121\*  
 4125\*  
 4129\*  
 4133\*  
 4137\*  
 4141\*  
 4145\*  
 4149\*  
 4153\*  
 4157\*  
 4161\*  
 4165\*  
 4169\*  
 4173\*  
 4177\*  
 4181\*  
 4185\*  
 4189\*  
 4193\*  
 4197\*  
 4201\*  
 4205\*  
 4209\*  
 4213\*  
 4217\*  
 4221\*  
 4225\*  
 4229\*  
 4233\*  
 4237\*  
 4241\*  
 4245\*  
 4249\*  
 4253\*  
 4257\*  
 4261\*  
 4265\*  
 4269\*  
 4273\*  
 4277\*  
 4281\*  
 4285\*  
 4289\*  
 4293\*  
 4297\*  
 4301\*  
 4305\*  
 4309\*  
 4313\*  
 4317\*  
 4321\*  
 4325\*  
 4329\*  
 4333\*  
 4337\*  
 4341\*  
 4345\*  
 4349\*  
 4353\*  
 4357\*  
 4361\*  
 4365\*  
 436



TDRP	000430R	277#	533			
TGO	000556R	275#	297#			
TDCNT	003004R	590*	595*	610*	615*	645#
TPCHTR	003016R	662#	648#			
TRPDRD=	000022	448#				
TRTO	003574R	620	777#			
TSS	001310R	349	407#			
TST	000746R	316	324	327#	344	
UNIT0	002752R	365	309*	394	384	632#
UNIT1	002754R	322*	376	633#		
UNIT0	000566R	296#				
UNIT1	000634R	296	298	309#		
UNITA	000642R	303	310#			
UTI	002764R	380*	389#	460	483	505 636#
VEC	003014R	664#				
VECTOR	000010R	194#	396	475		
VSET	001750R	470	396#			
WASADR	000104R	428#	576#			
WBUF	003602R	352	441	566	783#	
WDR	000116R	532#	248*			
WDT0	000114R	532#	247*			
WRITE	001512R	333	337	439#	546	
WTR	003466R	519	783#			
WTF	002766R	340*	483#	510	544	638#
WVFLG	003006R	255*	353	417	419*	646#
XPLAC	000005R	192#				
.	= 004206R	735#	737#	740#	784#	786#

. ABS. 000000 000  
 004206 001

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0  
 XRAXEO, XRAXEO/SOL/CRF:SYM=DDXCOM, XRAXEO  
 RUN-TIME: 1.2.3 SECONDS  
 RUN-TIME RATIO: 22/4=5.0  
 CORE USED: 7K (13 PAGES)