

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E673G-MC
PRODUCT NAME: CXRFAGO RF11 MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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

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

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

COPYRIGHT (C) 1973,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RFA IS AN IOMODX THAT EXERCISES RS DISK DRIVES ON AN RF11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE DATA COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RS DISK DRIVES WITH AN RF11 CONTROLLER

STORAGE:: RFA REQUIRES:
1. DECIMAL WORDS: 978
2. OCTAL WORDS: 1722
3. OCTAL BYTES: 3644

3. PASS DEFINITION

ONE PASS OF THE RFA MODULE CONSISTS OF 740 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 RFA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177460, VECTOR: 204, RR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

7. MODULE OPERATION

TEST SEQUENCE:

- A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
- B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
- C. GET A STARTING DISK ADDRESS AND FRESH BLOCK OF DATA
- 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 CONTINUE
- 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:

- A. BIT 0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON A HARD ERROR, THE FUNCTION
IS ABORTED AND THE TEST CONTINUES
- BIT 0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED ON A HARD ERROR, THE
DRIVE WILL BE DROPPED
- B. BIT 1 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON A SOFT ERROR, THE
DRIVE WILL BE DROPPED
- BIT 1 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED ON A SOFT ERROR, THE
FUNCTION IS ABORTED AND THE TEST CONTINUES
- C. BIT 2 SET(1)
WILL NOT TYPE OUT DATA LATE ERRORS AND WILL KEEP
ERROR COUNT IN LOCATION DLTCNT
- BIT 2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND KEEP COUNT OF
DATA LATE ERRORS IN DLTCNT

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 8 RF11 REGISTERS IN THE FOLLOWING ORDER:

RFCS RFWC RFMA RFDA RFAE RFDB RFMR RFDS

```

000000- IOMODX <RFAG > 177460,204,5,0,0,740,4,4,0,256,1024.
000000- MODULE 150000,RFAG 177460,204,5,0,0,740,4,4,0,256,1024.
; -TITLE RFAG DEC/X11 SYSTEM EXERCISER MODULE
; DDXCO4 VERSION 6 23-WAV-78
;*****LIST BIN*****
000009- 043122 043501 040 REGIN:
000009- 000000 MODNAM: ASCII /RFAG / ;MODULE NAME
000006- 177460 XPLAG: BYTE OPEN USED TO KEEP TRACK OF WBUFU USAGE
000019- 000204 ADDR: 177460+0 ;1ST DEVICE ADDR.
000013- 000000 VECTOR: 204+0 ;1ST DEVICE VECTOR.
000014- 000001 BR1: BYTE PPTV5+0 ;1ST BR LEVEL.
000013- 000000 BR2: BYTE PPTV0+0 ;2ND BR LEVEL.
000014- 000001 DVID1: 0+1 ;DEVICE INDICATOR 1.
000010- 000000 SR1: OPEN ;SWITCH REGISTER 1
000010- 000000 SR2: OPEN ;SWITCH REGISTER 2
000022- 000000 SR3: OPEN ;SWITCH REGISTER 3
000024- 000000 SR4: OPEN ;SWITCH REGISTER 4
;*****LIST BIN*****
000026- 150000 STAT: 150000 ;STATUS WORD
000030- 000252 INIT: START ;MODULE START ADDR.
000032- 000252 SPOINT: MODSP ;MODULE STACK POINTER.
000034- 000000 PASCNT: 0 ;PASS COUNTER.
000034- 001344 ICOUNT: 740. ;# OF ITERATIONS PER PASS=740.
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 SPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050- 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052- 000000 SVSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054- 000000 RANUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056- 000000 CDFIG: 0 ;RESERVED FOR MONITOR USE
000060- 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062- 000000 SVR0: OPEN ;LOC TO SAVE R0.
000064- 000000 SVR1: OPEN ;LOC TO SAVE R1.
000066- 000000 SVR2: OPEN ;LOC TO SAVE R2.
000070- 000000 SVR3: OPEN ;LOC TO SAVE R3.
000072- 000000 SVR4: OPEN ;LOC TO SAVE R4.
000074- 000000 SVR5: OPEN ;LOC TO SAVE R5.
000076- 000000 SVR6: OPEN ;LOC TO SAVE R6.
000100- 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102- 000000 SBADR: OPEN ;ADDR OF GOOD DATA, OR
000104- 000000 ACSR: OPEN ;CONTENTS OF CSR.
000104- 000000 WASADR: OPEN ;ADDR OF BAD DATA, OR
000106- 000000 ASTAT: OPEN ;STATUS REG CONTENTS.
000108- 000000 ERRTP: OPEN ;TYPE OF ERROR
000110- 000000 ASB: OPEN ;EXPECTED DATA.
000112- 000432 AWAS: OPEN ;ACTUAL DATA.
000114- 000000 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
000116- 000000 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION
000118- 000000 WFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000120- 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122- 000004 IDNUM: 4 ;MODULE IDENTIFICATION NUMBER=4

```

```

000124- 002422 RBUFVA: BUFIN ;READ BUFFER VIRTUAL ADDRESS
000126- 000000 RBUFPA: OPEN ;READ BUFFER PHYSICAL ADDRESS
000130- 000000 RBUFEA: OPEN ;READ BUFFER EA BITS
000132- 000400 RBUFSZ: 256. ;SIZE OF THE READ BUFFER
000134- 000000 WBUFPA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
000136- 000000 WBUFEA: OPEN ;WRITE BUFFER EA BITS
000140- 002000 WBUFRQ: 1024. ;WRITE BUFFER SIZE REQUESTED
000142- 000000 WBUFSZ: OPEN ;WRITE BUFFER SIZE AVAILABLE
000144- 000000 CDRECT: OPEN ;CDATA/DATCK ERROR COUNT
000146- 000000 CDWDCT: OPEN ;CDATA/DATCK WORD COUNT
000150- 000000 FREE: OPEN ;RESERVED FOR FUTURE USE
;*****LIST BIN*****
;REPT SPSIZ ;MODULE STACK STARTS HERE.
;WORD 0
;LIST
;ENDR
000252- MODSP:
;*****LIST BIN*****

```

```

234 000252* 012767 000400 177634 START: MOV #256,WDFO ;256 WORDS TO MEM PER ITERATION
235 000260* 012767 000400 177630 MOV #3024,WDFR ;3024 WORDS FROM MEM PER ITERATION
236 000268* 012767 000400 177624 MOV #3,INT4 ;3 INTERRUPTS PER ITERATION
237 000274* 105067 003337 CLR B FLAG ;CLEAR FLAGS
238 000300* 005067 002062 CLR DLTCNT ;CLEAR DATE LATE ERROR COUNTER
239 000304* 012767 002062 MOV DVICE,DRIVE ;GET DRIVE INDICATOR
240 000311* 012767 002062 MOV DVICE,DRIVE ;ALSO SAVE IT IN DRIVE
241 000320* 012767 177777 MOV #-1,BLK1 ;INITIALIZE BLOCK COUNTER
242 000326* 012767 177777 MOV #-1,DRVVE ;INITIALIZE DEVICE COUNTER
243 000334* 012767 177774 MOV #17774,DRVSFT ;INITIALIZE THE SHIFTED DRIVE #
244 000342* 004767 001174 JSR PC,PEZET ;GENERATE REGISTER ADDRESSES
245 000346* 004767 001174 JSR PC,PEZET ;INITIALIZE RR REGS. AND ALL DRIVES
246 000352* 123727 000041 CMPB #41,#12 ;IF RS IS LOAD MEDIUM THEN
247 000360* 001174 RNE J5 ;BEGIN
248 000366* 112701 000040 MOVB #40,RO ;GET LOAD-DRIVE NUMBER
249 000368* 012701 000001 MOV #1,R1 ;INITIALIZE DRIVE MASK
250 000372* 105700 1S: TSTB RO ;WHILE RO>0 DO
251 000374* 001403 RFB ZS ;BEGIN
252 000376* 000300 ASL R1 ;SHIFT DRIVE MASK
253 000400* 105300 DECR RO ;DOWNCOUNT RO
254 000402* 000773 RR ;END
255 000410* 001174 BITB R1,DRIVE ;IF LOAD-DRIVE SELECTED THEN
256 000412* 001174 BEQ #40,DRVVE ;BEGIN
257 000412* 013767 MOV #40,DRVVE ;MOVE LOAD-DRIVE NUMBER TO DRVVE
258 000420* 004767 JSR PC,DROP ;DROP LOAD-DRIVE
259 000424* 004767 JSR PC,DROP ;END
260 000424* 004767 JSR PC,DROP ;END
261 000424* 005767 001750 3S: TST DVICE ;DROP THE MODULE ?
262 000430* 005152 BEQ FINI ;YES
263 000432* 004415 000000* 000124* RESTRST: GETPAS,BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
264 000440* 012767 177466 MOV RROPSZ,WCNT2 ;SAVE READ BUFFER SIZE
265 000446* 005467 001746 NEG WCNT2 ;GET THE 2'S COMPLEMENT
266 000452* 004767 000634 STRT: JSR PC,BLOCK ;GET NEXT BLOCK NUMBER
267 000456* 104415 000000* 0001726* MOV #RROPSZ,WCNT1 ;GET WRITE BUFFER INFORMATION
268 000460* 005467 001722 NEG WCNT1 ;SAVE THE WRITE BUFFER SIZE
269 000474* 016700 001710 MOV BLK1,RO ;GET THE 2'S COMPLEMENT
270 000500* 004767 001412 JSR PC,CONVRT ;LOAD BLOCK # FOR CONVRT
271 000504* 004767 000632 NEXT: JSR PC,DRVADR ;GET A DRIVE ADDRESS
272 000510* 005767 001664 TST DVICE ;ANY DRIVES LEFT ?
273 000514* 001460 BEQ FINI ;NO, GO DROP THE MODULE
274 000524* 001352 BITB #R13,FLAG ;ALL DRIVES DONE ?
275 000526* 004767 RNE STRT ;YES, GO GET ANOTHER BLOCK
276 000534* 005767 001634 R1C #34,DSK2 ;CLEAR DRIVE ADDRESS
277 000540* 005067 003072 RTS DRVSFT,DSK2 ;LOAD DRIVE ADDRESS
278 000546* 105067 003070 CLP TRV1 ;ZERO RETRY COUNTERS
279 000554* 142767 000003 003057 CLP TRV3 ;CLEAR DROP FLAGS
280 000560* 004567 001416 R1CR #3,FLAG ;IS CONTROLLER READY ?
281 000564* 000401 JSR PC,READY ;YES, CONTINUE
282 000568* 000433 BR ;NO, GO DROP THE MODULE

```

```

290 000570* 004567 000250 GO: JSR R5,WRITE ;WRITE SOME DATA
291 000574* 000432 BR RETRY1 ;IF ERRORS, TRY IT AGAIN
292 000576* 132767 000200 003033 BITB #R14,FLAG ;DID THE DISK OVERFLOW ?
293 000604* 001407 BEQ GOA ;NO, CONTINUE
294 000606* 142767 000020 003023 BICB #R14,FLAG ;YES, CLEAR THE OVERFLOW FLAG
295 000614* 132767 177777 001566 MOV #-1,BLK1 ;RESET THE BLOCK NUMBER
296 000622* 000713 RR STRT ;START OVER AT BEGINNING OF DISK
297 000624* 004567 000245 GOA: JSR R5,BITCK ;WRITE-CHECK THE DATA
298 000630* 004567 BR RETRY2 ;IF ERRORS, TRY AGAIN
299 000632* 004567 JSR R5,READ ;READ THE DATA WRITTEN
300 000636* 000435 BR RETRY3 ;IF ERRORS, TRY AGAIN
301 000640* 104412 000000* 000126* CDATAS,BEGIN,RBUFFA ;REQUEST FOR MONITOR TO CHECK DATA
302 000646* 000650 .+2 ;IF ERROR, CONTINUE

```

```

304
305
306 000650* 104413 000000* PASS: ENDDTS,BEGIN ; SIGNAL END OF ITERATION.
307 000650* 104413 000000* BR NEXT ; MONITOR SHALL TEST END OF PASS
308 000654* 000713 ; CONTINUE
309
310 000656* 104410 000000* FINI: ENDS,BEGIN ; DROP THE MODULE
311 000656* 104410 000000* ;
312
313
314
315
316
317
318 000662* 105267 002752 RETRV1: INCR TRY1 ; COUNT THE RETRYS
319 000666* 122767 000003 002744 CMPR #3,TRV1 ; LIMIT EXCEEDED ?
320 000674* 001335 BNE GOB ; NO, GO TRY IT AGAIN
321 000674* 104403 000000* 003572* MSGNS,BEGIN,EXCED1 ; ASCII MESSAGE CALL WITH COMMON HEADER
322 000704* 000424 BR NEXT ; GO ON TO NEXT DRIVE
323
324
325
326
327 000706* 105267 002727 RETRV2: INCR TRY2 ; COUNT RETRYS
328 000712* 122767 000003 002721 CMPR #3,TRV2 ; LIMIT EXCEEDED ?
329 000720* 001341 BNE GOA ; NO, TRY AGAIN
330 000736* 000412 000000* 003600* MSGNS,BEGIN,EXCED2 ; ASCII MESSAGE CALL WITH COMMON HEADER
331 000736* 000412 BR NEXT ; GO ON TO NEXT DRIVE
332
333
334
335 000732* 105267 002704 RETRV3: INCR TRY3 ; COUNT RETRYS
336 000738* 122767 000003 002676 CMPR #3,TRV3 ; LIMIT EXCEEDED ?
337 000744* 001332 BNE GOB ; NO, GO TRY AGAIN
338 000754* 000400 000000* 003606* MSGNS,BEGIN,EXCED3 ; ASCII MESSAGE CALL WITH COMMON HEADER
339 000754* 000400 BR NEXT ; GO ON TO NEXT DRIVE
340
341
342
343 000756* 032767 000001 177032 NEXTA: BIT #R10,SRI ; IS HARD DROP OPTION CHOSEN ?
344 000764* 001000 000001 002643 BNE ZS ; NO, CHECK THE OTHER OPTION
345 000770* 001007 000000 002643 BITR #R10,FLAG ; HARD ERROR DROP SET ?
346 000776* 000167 177502 1S: JMP NEXT ; YES, GO DROP THE DRIVE
347 001000* 032767 000002 177006 BIT #R11,SRI ; IS SOFT DROP OPTION CHOSEN ?
348 001010* 001001 000000 177006 BNE ZS ; YES, GO DROP THE DRIVE
349 001010* 001001 000000 177006 BR PC,DROP ; NO, GO ON TO NEXT DRIVE
350 001014* 004767 000234 3S: JSP PC,DROP ; DROP PENDING DRIVE
351 001020* 012767 000001 177060 MOV #1,ERRTP ; DATA ERROR
352 *****
353 *****
354 *****
355 *****
356 *****
357 *****
358 *****
359 *****
360 *****
361 *****
362 *****
363 *****
364 *****
365 *****
366 *****
367 *****
368 *****
369 *****
370 *****
371 *****
372 *****
373 *****
374 *****
375 *****
376 *****
377 *****
378 *****
379 *****
380 *****
381 *****
382 *****
383 *****
384 *****
385 *****
386 *****
387 *****
388 *****
389 *****
390 *****
391 *****
392 *****
393 *****
394 *****
395 *****
396 *****
397 *****
398 *****
399 *****
400 *****
401 *****
402 *****
403 *****
404 *****
405 *****
406 *****
407 *****
408 *****
409 *****
410 *****
411 *****
412 *****
413 *****

```

```

358
359
360
361
362 001044* 012767 000103 001316 WRITE: MOV #103,FUNC ; LOAD WRITE FUNCTION
363 001052* 016777 001340 002346 MOV WCNT1,RRFMC ; LOAD WORD COUNT
364 001060* 016777 177030 002342 MOV WRUPEA,RRFMA ; LOAD RUFFER ADDRESS
365 001068* 016777 177044 001276 MOV WRUPEA,XMEM ; LOAD EXTENDED MEMORY BITS
366 001074* 000436 BR ; CONTINUE
367 001076* 012767 000107 001264 WRITCK: MOV #107,FUNC ; LOAD WRITE-CHECK FUNCTION
368 001104* 016777 001306 002314 MOV WCNT1,RRFMC ; LOAD WORD COUNT
369 001112* 016777 177016 002310 MOV WRUPEA,RRFMA ; LOAD RUFFER ADDRESS
370 001120* 016767 177012 001244 MOV WRUPEA,XMEM ; LOAD EXTENDED MEMORY BITS
371 001126* 000421 BR ; CONTINUE
372 001130* 012767 000105 001232 READ: MOV #105,FUNC ; LOAD READ FUNCTION
373 001136* 016777 001306 002282 MOV WCNT1,RRFMC ; LOAD WORDCOUNT
374 001144* 016777 176756 002256 MOV WRUPEA,RRFMA ; LOAD RUFFER ADDRESS
375 001152* 016767 176752 001212 MOV WRUPEA,XMEM ; LOAD EXTENDED MEMORY BITS
376 001160* 000408 BR ; CONTINUE
377 001162* 012777 000400 002234 CLEAR: MOV #R10,RRFCS ; ISSUE A CONTROL RESET
378 001170* 000205 RTS ; RETURN
379
380 001172* 012777 001234* 176610 GOGO: MOV #NTRUPT,AVECTOR ; SET INTERRUPT ENTRY POINTER
381 001206* 016777 001170 002234 MOV DSK1,RRFA ; LOAD THE DISK ADDRESS
382 001206* 016777 001164 002220 MOV DSK1,RRFAE ; LOAD DRIVE AND DISK ADDRESS
383 001214* 056767 001152 001146 BTS XMEM,FUNC ; LOAD EXTENDED MEMORY BITS
384 001222* 016777 001142 002174 MOV FUNC,RRFCS ; EXECUTE THE FUNCTION
385 001232* 104400 000000* EXIT, BEGIN ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
386
387
388
389
390 001234* 000004 000000* 001242* NTRUPT: ;
391 ;IFQS,BEGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
392 ;
393
394
395 001242* 004567 000372 1S: JSR R5,RRFRORS ; GO CHECK FOR ERRORS
396 001242* 004567 000372 RTS ; ERRORS DETECTED, RETURN
397 001252* 000205 TST (R5)+ ; NO ERRORS, SKIP RETRY
398 001252* 000205 RTS ; RETURN OK
399 ;
400
401
402 001254* 012701 000001 001120 DROP: MOV #1,R1 ; INITIALIZE DROP PICKER
403 001264* 001403 MOV #1,DRIVE,R0 ; GET THE DRIVE NUMBER
404 001266* 000301 BEQ ZS ; IF DRIVE 0 GO DROP IT
405 001270* 005300 ASL R1 ; NO, AIM AT THE NEXT DRIVE
406 001274* 040167 001100 2S: DEC R0 ; IS THIS THE ONE ?
407 001274* 040167 001100 BNE 1S ; NO, LOOK AGAIN
408 001274* 040167 001100 BIT R1,DRIVE ; DROP THE DRIVE
409 *****
410 *****
411 *****
412 *****
413 *****
414 *****
415 *****
416 *****
417 *****
418 *****
419 *****
420 *****
421 *****
422 *****
423 *****
424 *****
425 *****
426 *****
427 *****
428 *****
429 *****
430 *****
431 *****
432 *****
433 *****
434 *****
435 *****
436 *****
437 *****
438 *****
439 *****
440 *****
441 *****
442 *****
443 *****
444 *****
445 *****
446 *****
447 *****
448 *****
449 *****
450 *****
451 *****
452 *****
453 *****
454 *****
455 *****
456 *****
457 *****
458 *****
459 *****
460 *****
461 *****
462 *****
463 *****
464 *****
465 *****
466 *****
467 *****
468 *****
469 *****
470 *****
471 *****
472 *****
473 *****
474 *****
475 *****
476 *****
477 *****
478 *****
479 *****
480 *****
481 *****
482 *****
483 *****
484 *****
485 *****
486 *****
487 *****
488 *****
489 *****
490 *****
491 *****
492 *****
493 *****
494 *****
495 *****
496 *****
497 *****
498 *****
499 *****
500 *****

```

414

```

415 001312 005267 001072 BLOCK: INC BLK1 ; STEP TO NEXT BLOCK
416 001316 022767 000400 001064 CMP #256.,BLK1 ; BLOCK LIMIT REACHED ?
417 001320 005267 001056 RPL AS ; NO, CONTINUE
418 001324 005267 001056 CLR BLK1 ; YES, RESET BLOCK #
419 001332 016767 001052 001052 1S: MOV BLK1,BLK2 ; READ WHERE WRITE
420 001340 000207 001052 001052 RTS PC ; RETURN
-----
421 001342 005267 001036 DRVADR: INC DRVVE ; COUNT A DRIVE
422 001346 062767 000004 001032 ADD #RIT2,DRVSFT ; DRIVE # LINED UP WITH RFDA
423 001350 142767 000010 002255 RICR #RIT3,FLAG ; CLEAR END OF DRIVES FLAG
424 001354 142767 000010 001814 CMP #R.,DRVVE ; ALL DRIVES CHECKED ?
425 001358 061404 001004 BEQ IS ; YES, GO FLAG END OF DRIVES
426 001372 006267 001004 ASP DRIVE ; NO, IS NEXT DRIVE CHOSEN ?
427 001376 103367 000010 002227 DRVADR ; NO, GO TRY ANOTHER DRIVE
428 001380 103367 000010 000766 RTS PC ; YES, RETURN
429 001400 006267 000010 002227 1S: RISR #RIT3,FLAG ; SET END OF DRIVES FLAG
430 001404 152767 000010 000766 MOV #-1,DRVVE ; RESET DRIVE COUNTER
431 001410 012767 177777 000762 MOV #177774,DRVSFT ; RESET SHIFTED DRIVE #
432 001416 012767 177777 000762 MOV DVICE,DRIVE ; RESTORE CHOSEN DRIVES
433 001420 016767 000750 000750 RTS PC ; RETURN
-----
434 001432 006267 000750 ROOM: MOV BLK1,R0 ; SAVE THE CURRENT BLOCK NUMBER
435 001434 016700 000750 MOV #256.,R1 ; LOAD MAX. NUMBER OF BLOCKS
436 001440 012701 000400 CLR R2 ; ZERO REG. 2
437 001444 005202 000001 002163 RICR #RIT0,FLAG ; CLEAR 32K INDICATOR
438 001448 142767 000001 176462 MOV WRUPSZ,R3 ; GET THE TRANSFER SIZE
439 001452 016703 000001 176462 SUB R0,R1 ; GET # OF BLOCKS LEFT ON DISK
440 001456 180001 000040 CMP #32.,R1 ; MORE THAN 32K LEFT?
441 001460 022701 000040 RGT IS ; NO, CONTINUE
442 001464 003010 000001 002141 RISR #RIT0,FLAG ; YES, SET THE INDICATOR
443 001468 152767 000001 000040 SUB #32.,R1 ; SUBTRACT 32K WORTH OF BLOCKS
444 001472 152767 000001 000040 CMP #64.,R1 ; WAS THERE MORE THAN 64K LEFT?
445 001476 162701 000040 BLT 5S ; YES, MUST BE AN ERROR
446 001500 022701 000040 TST R1 ; ANY BLOCKS LEFT ON DISK?
447 001504 002425 000001 000040 RLE 6S ; NO, RETURN OK
448 001510 005701 000001 002006 ADD #1024.,R2 ; GET TOTAL # OF WORDS LEFT
449 001514 003425 000001 002006 DEC R1 ; ALL BLOCKS ADDED IN?
450 001518 005301 000001 002006 RGT 2S ; NO, KEEP ADDING
451 001522 003374 000001 002006 TST R3 ; REQUEST LARGER THAN 32K?
452 001526 005703 000001 002006 RPL 3S ; NO, GO CHECK THAT CONDITION
453 001530 042703 100000 002075 RICR #RIT15,R3 ; YES, GET RID OF 32K
454 001534 132767 000001 002075 BITR #RIT0,FLAG ; MORE THAN 32K LEFT?
455 001538 001411 000001 002075 BEQ 4S ; NO, RETURN OK
456 001542 004404 000001 002063 BR 4S ; YES, GO COMPARE
457 001546 132767 000001 002063 3S: BITR #RIT0,FLAG ; MORE THAN 32K?
458 001550 001402 000001 002063 BNE 5S ; YES, PLENTY OF ROOM LEFT, ERROR
459 001554 005703 000001 002063 CMP R2,R3 ; ENOUGH ROOM FOR THE TRANSFER?
460 001558 005703 000001 002063 BLT 6S ; NO, RETURN OK
461 001562 005703 000001 002063 TST (R5)+ ; MUST BE A REAL ERROR
462 001566 005703 000020 002043 RTS R5 ; RETURN INDICATING THE ERROR
463 001570 152767 000020 002043 6S: RISR #RIT4,FLAG ; SET OVERFLOW FLAG
464 001574 000205 000020 002043 RTS PC ; RETURN OK
-----

```



```

RFAG0.P11 12-OCT-78 12:07
469
470
471 001576* 014167 176304 ERSUB2: MOV -(R1),ASB ; LOAD THE DATA
472 001600* 010167 176274 MOV R1,ASADR ; LOAD ADDRESS OF DATA WRITTEN
473 001600* 014267 176276 MOV -(R2),AWAS ; LOAD THE DATA
474 001612* 010267 176276 MOV R2,WAADR ; LOAD ADDRESS OF DATA READ
475 001620* 005722 TST (R1)+ ; RESET REG. 1
476 001620* 005722 TST (R2)+ ; RESET REG. 2
477 001622* 016767 001576 176250 ERSUB1: MOV RFCS,CSRA ; LOAD ADR. OF CURRENT CSR
478 001636* 011767 001570 176244 MOV BRFC5,ACSR ; LOAD CONTENTS OF CURRENT CSR
479 001636* 000207 RTS ; RETURN
480
481
482
483
484
485
486 001640* 005777 001560 ERRORS: TST BRFC5 ; ANY ERRORS ?
487 001644* 100016 RPL ZS ; NO, RETURN
488 001646* 032777 004000 001550 BIT #BIT11,ARFCS ; NON-EXISTENT DISK?
489 001654* 001403 BEQ IS ; NO CONTINUE
490 001666* 000467 JSR R5,ROOM ; YES, MAKE SURE ENOUGH ROOM FOR TRANSFER
491 001666* 000467 JSR PC,ERSUB1 ; OVERFLOW, DON'T REPORT AS ERROR
492 001666* 000467 JSR PC,ERSUB1 ; LOAD ERROR INFORMATION
493 001670* 032777 040000 001526 BIT #BIT14,ARFCS ; HARD ERROR ?
494 001676* 001067 BR ZS ; YES, GO REPORT
495 001700* 000467 BR JZ ; MUST BE SOFT ERROR
496 001700* 000467 JSR R5,CLEAR ; GO CLEAR OUT ANY ERRORS
497 001706* 005725 TST (R5)+ ; SKIP RETRY
498 001712* 005087 CLR ERRTYP ; RETURN OK
499 001712* 005087 CLR ERRTYP ; UNKNOWN ERROR
500
501 001716* 104405 000000* 003424* HRDRS,REGIN,TABLE ; *****
502 001724* 004567 JSR R5,CLEAR ; GO CLEAR OUT ERRORS
503 001730* 152767 000001 001701 BISB #BIT0,FLAG ; SET HARD ERROR DROP FLAG
504 001736* 000426 BR ZS ; RETURN
505 001740* 032777 000200 001466 4S: BIT #BIT7,ARFAE ; DATA REQUEST LATE?
506 001746* 001411 BEQ NO ; NO
507 001750* 005267 INC DLTCNT ; INCREMENT ERROR COUNTER
508 001754* 032767 000004 176034 BIT #BIT2,SRI ; TYPE OUT ERROR?
509 001762* 001014 BNE MSGNS,BEGIN,DLTERR ; NO
510 001764* 104443 000000* 003624* MOV #2,ERRTYP ; ASCII MESSAGE CALL WITH COMMON HEADER
511 001772* 012767 000002 176106 6S: ; *****
512
513 002000* 104406 000000* 003424* SOFTS,REGIN,TABLE ; *****
514 002006* 152767 000002 001623 5S: BISB #BIT1,FLAG ; SET SOFT ERROR DROP FLAG
515 002014* 000205 RTS R5 ; RETURN, ERRORS
516
517

```

```

RFAG0.P11 12-OCT-78 12:07
519
520
521 002016* 016700 175754 SETUP: MOV ADDR,R0 ; GET DEVICE ADDRESS
522 002022* 000067 MOV R0,RFCS ; GENERATE CONTROLLER REGS. ADDRESSES
523 002026* 005720 TST (R0)+
524 002030* 010067 MOV R0,RFWC 001372
525 002034* 005720 TST (R0)+
526 002036* 010067 MOV R0,RFWA 001366
527 002040* 005720 TST (R0)+
528 002044* 010067 MOV R0,RFDA 001362
529 002050* 005720 TST (R0)+
530 002054* 010067 MOV R0,RFEA 001356
531 002058* 005720 TST (R0)+
532 002060* 010067 MOV R0,RFDB 001352
533 002064* 005720 TST (R0)+
534 002066* 010067 MOV R0,RFMR 001346
535 002070* 005720 TST (R0)+
536 002074* 010067 MOV R0,RFDS 001342
537 002100* 016700 MOV VECTOR,R0 ; GET THE VECTOR ADDRESS
538 002104* 175704 MOV #STPT,(R0)+ ; SET POINTER JUST IN CASE
539 002110* 116710 MOV#B R1,(R0) ; SET PRIORITY
540
541 002114* 000207 1S: RTS PC ; RETURN
542
543
544
545
546
547 002116* 005067 000252 CONVRT: CLR DSK1 ; RESET DISK ADDRESSES
548 002122* 005067 CLR DSK2 ;
549 002126* 006200 ASR R0 ; DIVIDE BLOCK # BY 2 TO GET TRACK #
550 002130* 103093 BCC R0 ; 1ST TIME ON THIS TRACK ?
551 002132* 000000 002000 000234 1S: MOV #1024,DSK1 ; 2ND TIME, STEP TO NEXT 1024 WORDS
552 002140* 010067 000011 2S: MOV R0,R1 ; SHIFT 9 PLACES TO THE LEFT
553 002144* 006300 ASL R0 ; TO LINE UP TRACK ADDRESS
554 002146* 005301 DEC R1 ; DONE ?
555 002150* 003375 BGT ZS ; NO, KEEP SHIFTING
556
557 002152* 006300 ASL R0 ; YES, GET MOST SIGNIFICANT BIT FOR RFAE
558 002154* 103093 BCC R0 ; BIT SET ?
559 002156* 012767 MOV #BIT1,DSK2 ; YES, SET IT FOR RFAE
560 002164* 006300 ASL R0 ; GET NEXT SIGNIFICANT BIT FOR RFAE
561 002166* 103092 BCC R0 ; BIT SET ?
562 002170* 005267 INC DSK2 ; YES, SET IT FOR RFAE
563 002174* 050067 RFS R0,DSK1 ; GET LEAST SIG. BITS OF TRACK ADR
564 002200* 000207 RTS ; RETURN
565

```

```

566
567
568 002202 012767 077777 001212 REZY: MOV #77777,CLK ; SET THE TIMER
569 002210 105777 001210 1S: TSTR BRPCS ; CONTROLLER READY ?
570 002214 100001 ; NO CONTINUE
571 002216 000204 ; YES, RETURN READY
572
573 002220 004567 176736 2S: JSP R5,CLEAR ; GO CLEAR OUT ANY ERRORS
574 002234 104407 000000 ; TEMPORARY RETURN TO MONITOR
575 002236 104407 000000 ; THEN CONTINUE AT NEXT INSTRUCTION.
576 002234 005367 001162 DEC CLK ; WAIT SOME MORE ?
577 002240 001383 ; YES
578 002242 012767 175636 MOV #3,FRPTYP ; CONTROLLER NOT READY CODE
579 ***** ; *****
580 002250 104405 000000 003424 HDRRS,REGIN,TABLE ; CONTROLLER NOT READY
581 ***** ; *****
582 002256 005067 000116 CLR DVICE ; SET TO DROP THE MODULE
583 002262 005725 TST (P5)+ ; SKIP INSTRUCTION FOLLOWING CALL
584 002264 000205 ; RETURN
585 ;
586
587
588
589 002266 004567 176670 REZET: JSP R5,CLEAR ; EXECUTE CONTROLLER RESET
590 002272 004567 177704 JSR R5,READV ; IS CONTROLLER READY ?
591 002276 000401 RR 1S ; YES, CONTINUE
592 002300 000207 RTS PC ; NO, RETURN ERROR
593
594 002302 004767 177034 1S: JSR PC,DRVADR ; GET A DRIVE ADDRESS
595 002306 132767 000310 BITR #R13,FLAG ; ALL DRIVES DONE ?
596 002314 001023 BNE ZS ; YES, RETURN
597 002316 000064 001110 MOV DRVSFT,DRFAE ; NO, LOAD DISK ADDRESS REG.
598 002320 032777 001072 BIT #R11,BRFCS ; DRIVE EXIST ?
599 002332 001763 BEO C ; YES, CONTINUE
600 002334 004767 176714 JSR PC,DROP ; NO, DROP THE DRIVE
601 002340 012767 000066 MOV #6,FRPTYP ; NON EXISTENT DEVICE
602 ***** ; *****
603 002346 104405 000000 000000 HDRRS,REGIN,NULL ; NON EXISTENT DEVICE
604 ***** ; *****
605 002354 104403 000300 003614 MSGNS,REGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
606 002362 000447 ; WARE CALL ALL GET CHECKED
607 002364 000207 RTS PC ; RETURN
608 ;
  
```

```

609
610
611
612
613 002366 000000 DLTCNT: C
614 002370 000000 FUNC: C
615 002372 000000 XMEM: C
616 002374 000000 DSK1: C
617 002376 000000 DSK2: C
618 002400 000000 DVICE: C
619 002402 000000 DRIVE: C
620 002406 000000 DRWE: C
621 002408 000000 DRVSFT: C
622 002410 000000 BLK1: C
623 002412 000000 BLK2: C
624 002414 000000 TRUB: C
625 002416 000000 WCNT1: C
626 002420 000000 WCNT2: C
627 002422 000400 BUFIN: ;BLKW 256.
628 002424 000000 CLK: C
629 002426 000000 TABLE:
630 002428 000000 RPCS: C
631 002430 000000 RFWC: C
632 002432 000000 RFWA: C
633 002434 000000 RFDI: C
634 002436 000000 RFAE: C
635 002438 000000 RFDI: C
636 002440 000000 RFWA: C
637 002442 000000 RFDI: C
638 002444 177777 RFDI: 177777
639
640
641
642 003446 020040 051104 053111 MES3: .ASCIZ " DRIVE "
643 003454 020105 000040 MES4: .ASCIZ " DROPPED*"
644 003460 020040 051124 050117 MES5: .ASCIZ " RETRY EXCEEDED*"
645 003466 042920 042522 051124 MES6: .ASCIZ " WRITE"
646 003473 040000 042522 042522 MES7: .ASCIZ " WRITE-CHECK"
647 003500 020131 054105 042563 MES8: .ASCIZ " READ"
648 003506 042105 042105 000045 MES9: .ASCIZ " DATA REQUEST LATER"
649 003514 053440 044522 042524
650 003522 000000
651 003523 040000 051127 052111
652 003530 026505 044103 041505
653 003536 000113
654 003540 051040 040505 000104
655 003546 042046 052101 020101
656 003554 042522 052521 051505
657 003562 042522 042514 042524
658 003570 000045
659
660 003572 003514 EXCED1: .EVEN
661 003574 003473 MES6
662 003576 177777 MES7
663 003600 003523 EXCED2: MES7
664 003602 003473 MES5
  
```

```

RFAG0.P11 12-OCT-78 12:07
665 003604* 177777
666 003606* 003540*
667 003610* 003473*
668 003612* 177777
669 003614* 003446*
670 003616* 003635*
671 003620* 003460*
672 003622* 177777
673 003624* 003546*
674 003626* 177777
675 003630* 000005
676 003635* 000
677 003636* 000
678 003639* 000
679
680 003640* 000
681 003641* 000
682 003642* 000
683
684
685 000001
    
```

```

EXCED3: 177777
        MESA
        MESA
DRP:    177777
        MESA
        NUMR
        MESA
DLTERR: MESA
        177777
ADRI:   .BLKB      5
NUMR:   .BVTE      0
        .BVTE      0
FLAG:   .BVTE      0
        .BVTE      0
        .EVEN
TRV1:   .BVTE      0
TRV2:   .BVTE      0
TRV3:   .BVTE      0
        .EVEN
        .END
    
```

```

RFAG0.P11 12-OCT-78 12:07
ACSR 000102R 295# 478*
ADDR 000006R 171# 521
ADDR22= 001000 234#
ADRI 003630R 410 675#
ASB 000166R 209# 470*
ASTA* 000104R 207#
ANAS 000110R 210#
BRGIN 000000R 168# 472*
BIT0 = 000001 410 270 302 307 312 321 329 337 353 355 385 389
BIT1 = 000002 410 501 511 514 574 575 580 603
BIT10 = 002000 234#
BIT11 = 004000 234#
BIT12 = 000004 234#
BIT13 = 000000 234#
BIT14 = 000000 234#
BIT15 = 000000 234#
BIT2 = 000004 234#
BIT3 = 000010 234#
BIT4 = 000020 234#
BIT5 = 000040 234#
BIT6 = 000100 234#
BIT7 = 000200 234#
BIT8 = 000400 234#
BIT9 = 001000 234#
BLK1 002410R 231# 273
BLK2 002412R 419* 296* 415* 416 418* 419 436 622#
BLOCK 001312R 269 415#
BREAKS= 104407 234# 575
BR1 000012R 173#
BR2 000013R 174#
BRDDS = 104421 234#
BOFLN 002422R 216 627#
DATAS = 104412 234# 302
CDERCT 000144R 234#
CDWDCT 000146R 225#
CLEAR 001162R 377# 496 503 573 589
CLK 003422R 568* 576*
CONFIC 000056R 103#
CONVRT 002116R 274 547#
CSRA 000100R 203# 477*
DATCS = 104411 234#
DATERS = 104404 234#
DLTCTR 002366R 238* 508* 613#
DLTERR 003624R 511 673#
DRIVE 002402R 240* 427* 433* 619#
DROP 001254R 258 350 400# 600
DRP 003614R 355 605 669#
DRVADR 001342R 276 422# 428 594
DRVST 002408R 243* 401 432* 497 621#
DRYVE 002404R 242* 401 422* 425 431* 620#
DSK1 002374R 381 547* 551* 563* 597 616#
DSK2 002376R 282* 293* 382 548* 559* 562* 617#
DWICE 002400R 240 240 261 277 406* 433 582* 618#
DWID1 000604R 239#
ENDITS= 104413 234# 307
    
```


SVR4	000072R	200#					
SVR5	000074R	201#					
SVR6	000076R	202#					
SYSCNT	000052R	101#					
TABLE	003424R	501#	514	580	629#		
IBUF	002414R	624#					
TRPDPD=	000022	234#					
TRV1	003640R	284#	318*	319	680#		
TRV2	003641R	326*	327	681#			
TRV3	003642R	285*	354*	335	682#		
VECTOR	000010R	172#	380*	537			
WASADR	000104R	226#	473*				
WBUPEA	000136R	221#	365	370			
WBUPEA	000134R	520#	364	369			
WBUFRQ	000140R	492#					
WBUFSZ	000142R	223#	271	440			
WCNT1	002416R	221*	272*	363	368	625#	
WCNT2	002420R	226*	249*	373	626#		
WDFR	000116R	213#	235*				
WDFO	000114R	212#	234*				
WRITCK	001076R	298	367#				
WRITC	001048R	291	362#				
XPLAG	000005R	170#					
XMEM	002372R	365*	370*	375*	383	615#	
.	= 003644R	303	627#	675#	683#		

. ABS. 000000 000
 003644 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XRFAGO XRFAGO/SQL/CRF:SYM=DDXCOM,XRFAGO
 RUN-TIME: 1 2 .3 SECONDS
 RUN-TIME RATIO: 15/4=3.8
 CORE USED: 7K (13 PAGES)