

.HEM -

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 RFI1 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 RFI1 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 124 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, RRI: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

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.  DO A WRITE ADDRESS
E.  DO A WRITE--CHECK IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
F.  DO A READ--CHECK IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
G.  DO A DATA-CHECK--IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
H.  IF END OF PASSES, REPORT AND GO TO C
I.  IF END OF DRIVES, GO TO C ELSE GO TO D
J.  OPERATION OPTIONS
-----
SR1:
A.  BIT 0 SET(1): 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): 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
```

RFAG DFC/X11 SYSTEM EXERCISER MODULE MACV11 3CA(1052) 12-OCT-78 16:58 PAGE 5
XRPAGC.P11 12-OCT-78 12:07

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 RFL1 REGISTERS IN THE FOLLOWING ORDER:
RPCS RFWC RFMA RFDA RFAE RFDB RFMP RFDS

SEQ 0004

```

000000- IOMODX <RFRAG > 177460,204,5,0,0,740,4,4,256,1024.
000000- MODULE 150000, RFRAG 177460,204,5,0,0,740,4,4,256,1024.
; TITLE RFRAG DEC/Y11 SYSTEM EXERCISER MODULE
DOXCOM VERSION 6 23-MAV-78
LIST BIN
*****
000000- 043122 043501 040
000000- 000000
000006- 177460
000010- 000204
000012- 740
000013- 000
000015- 000001
000016- 000000
000020- 000000
000022- 000000
000026- 150000
000030- 000252
000032- 000000
000034- 000000
000036- 001344
000040- 000000
000042- 000000
000044- 000000
000046- 000000
000050- 000000
000052- 000000
000054- 000000
000056- 000000
000058- 000000
000060- 000000
000062- 000000
000064- 000000
000066- 000000
000070- 000000
000072- 000000
000074- 000000
000076- 000000
000100- 000000
000102- 000000
000104- 000000
000106- 000000
000108- 000000
000110- 000000
000112- 000432
000114- 000000
000116- 000000
000120- 000000
000122- 000004

RECIN:
MODNAM: ASCII /RFRAG / ;MODULE NAME
XFLAG: BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR: 177460+0 ;1ST DEVICE ADDR.
VECTOR: 204+0 ;1ST DEVICE VECTOR.
RR1: BYTE PPTY5+0 ;1ST RR LEVEL.
RR2: BYTE PPTY0+0 ;2ND RR LEVEL.
DVID1: 0+1 ;DEVICE INDICATOR 1.
SR1: OPEN ;SWITCH REGISTER 1
SR2: OPEN ;SWITCH REGISTER 2
SR3: OPEN ;SWITCH REGISTER 3
SR4: OPEN ;SWITCH REGISTER 4
*****
STAT: 150000 ;STATUS WORD.
INIT: START ;MODULE START ADDR.
SPONT: MODSP ;MODULE STACK POINTER.
PASCN: 0 ;PASS COUNTER.
ICOUNT: 0 ;# OF ITERATIONS PER PASS=740.
SDFCNT: 0 ;LOC TO COUNT ITERATIONS
HRDCW: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
SOPPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
HRDPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS PER PASS
SYSCNT: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
RANUM: 0 ;# OF SYS ERRORS ACCUMULATED
CONFIC: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
SVR1: OPEN ;LOC TO SAVE R0.
SVR2: OPEN ;LOC TO SAVE R1.
SVR3: OPEN ;LOC TO SAVE R2.
SVR4: OPEN ;LOC TO SAVE R3.
SVR5: OPEN ;LOC TO SAVE R4.
SVR6: OPEN ;LOC TO SAVE R5.
CSRA: OPEN ;LOC TO SAVE R6.
SADR: OPEN ;ADDR OF CURRENT CSR.
ACSR: OPEN ;ADDR OF GOOD DATA, OR
WASADR: OPEN ;CONTENTS OF CSR.
ASAT: OPEN ;ADDR OF BAD DATA, OR
ERRTYP: OPEN ;STATUS REG CONTENTS.
ASB: OPEN ;TYPE OF ERROR
AWAS: OPEN ;EXPECTED DATA.
RSRT: RESTRT ;ACTUAL DATA.
WDTO: OPEN ;RESTART ADDRESS AFTER END OF PASS
WDFR: OPEN ;WORDS TO MEMORY PER ITERATION
INTP: OPEN ;WORDS FROM MEMORY PER ITERATION
IDNUM: 4 ;# OF INTERRUPTS PER ITERATION
;MODULE IDENTIFICATION NUMBER=4

```

```

000124- 002422-
000126- 000000
000130- 000000
000132- 000400
000134- 000000
000136- 000000
000140- 002000
000142- 000000
000144- 000000
000146- 000000
000150- 000000
000152- 000004

RBUFVA: BUFIN ;READ BUFFER VIRTUAL ADDRESS
RBUFA: OPEN ;READ BUFFER PHYSICAL ADDRESS
RBUFSZ: 256 ;READ BUFFER EA BITS
RBUFA: OPEN ;SIZE OF THE READ BUFFER
WBUFA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
WBUFA: OPEN ;WRITE BUFFER EA BITS
WBUFSZ: OPEN ;WRITE BUFFER SIZE REQUESTED
CDERCT: OPEN ;WRITE BUFFER SIZE AVAILABLE
CMDCT: OPEN ;CDATA/DATCK ERROR COUNT
FREE: OPEN ;CDATA/DATCK WORD COUNT
FREE: OPEN ;RESERVED FOR FUTURE USE
FREE: REPT SPSIZ ;MODULE STACK STARTS HERE.
FREE: .NLIST
FREE: .WORD 0
FREE: .LIST
FREE: .ENDR

MODSP:
;*****

```

```

234 000252 012767 000400 177634 START: MOV #256,WDT0 ;256 WORDS TO MEM PER ITERATION
235 000260 012767 002000 177630 MOV #1024,WDFR ;1024 WORDS FROM MEM PER ITERATION
236 000265 012767 000003 177624 MOV #1,INTR ;3 INTERRUPTS PER ITERATION
237 000274 105067 003337 CLRFB FLAG ;CLEAR FLAGS
238 000300 005067 002062 DLT CNT ;CLEAR DATE LATE ERROR COUNTER
239 000304 012767 177504 MOV DVID1, DVICE ;GET DRIVE INDICATOR
240 000312 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 PC, SETUP ;INITIALIZE THE SHIFTED DRIVE #
244 000342 004767 001450 JSR #17774, DRVSFT ;GENERATE REGISTER ADDRESSES
245 000348 004767 001450 JSR PC, RE7ET ;INITIALIZE RA REGS. AND ALL DRIVES
246 000356 123727 000041 RMPB #41, #12 ;IF RS IS LOAD MEDIUM THEN
247 000360 001271 RNF ;BEGIN
248 000362 113760 000040 MVB #40, RO ;GET LOAD-DRIVE NUMBER
249 000368 012701 000001 MOV #1, RI ;INITIALIZE DRIVE MASK
250 000374 105700 1S: TSTB RC ;WHILE RO>0 DO
251 000376 001403 BEQ 2S ;REG I
252 000376 006301 ASL 2S ;SHIFT DRIVE MASK
253 000400 105300 DECR RO ;DOWNCOUNT RO
254 000404 000773 BR 1S ;END
255 000404 000773 BITB RI, DRIVE ;IF LOAD-DRIVE SELECTED THEN
256 000410 001405 BEQ 3S ;BEGIN
257 000412 013767 000040 MOV #40, DRVVE ;MOVE LOAD-DRIVE NUMBER TO DRVVE
258 000420 004767 000630 JSR PC, DROP ;DROP LOAD-DRIVE
259 000424 3S: ENDW ;END
260 000424 005767 001750 TST DVICE ;DROP THE MODULE ?
261 000430 001512 BEQ FINI ;YES
262 000432 000000 000124 RESTR: GETPAS, BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
263 000432 104415 000000 MOV RBUF, SZ, WCNT2 ;SAVE READ BUFFER SIZE
264 000440 012767 177466 NEG MCNT2 ;GET THE 2'S COMPLEMENT
265 000446 005467 001746 ;
266 000452 004767 000634 STRT: JSR PC, RLOCK ;GET NEXT BLOCK NUMBER
267 000452 102424 000000 MOV RBUF, SZ, WCNT1 ;GET WRITE BUFFER INFORMATION
268 000452 005467 001722 NEG MCNT1 ;SAVE THE WRITE BUFFER SIZE
269 000470 005467 001722 MOV BLK1, RO ;GET THE 2'S COMPLEMENT
270 000474 016700 001710 JSR PC, CONVRT ;LOAD BLOCK # FOR CONVRT
271 000500 004767 001412 ;GENERATE DISK ADR. FROM BLOCK #
272 000504 004767 000632 NEXT: JSR PC, DRVADR ;GET A DRIVE ADDRESS
273 000510 005767 001664 TST DVICE ;ANY DRIVES LEFT ?
274 000514 001460 BEQ FINI ;NO, GO DROP THE MODULE
275 000514 132767 BITB #13, FLAG ;ALL DRIVES DONE ?
276 000524 001352 RNE STRT ;YES, GO GET ANOTHER BLOCK
277 000526 042767 000234 BIC #34, DSK2 ;CLEAR DRIVE ADDRESS
278 000534 050767 001646 RTS DRVSFT, DSK2 ;LOAD DRIVE ADDRESS
279 000542 005067 003072 CLP TRV1 ;ZERO RETRY COUNTERS
280 000546 105067 003070 CLPP TRV3 ;
281 000552 142767 000033 RICR #2, FLAG ;CLEAR DROP FLAGS
282 000564 000401 BR ;IS CONTROLLER READY ?
283 000568 000433 BP GO ;YES, CONTINUE
284 ; ;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 000020 003033 BITB #14, FLAG ;DID THE DISK OVERFLOW ?
293 000604 001407 BEQ GOA ;NO, CONTINUE
294 000606 142767 000020 003023 BICB #14, FLAG ;YES, CLEAR THE OVERFLOW FLAG
295 000614 012767 177777 001566 MOV #1, BLK1 ;RESET THE BLOCK NUMBER
296 000622 000713 RR STRT ;START OVER AT BEGINING OF DISK
297 000624 004567 000245 GOA: JSR R5, WRITCK ;WRITE CHECK THE DATA
298 000630 004268 BR RETRY2 ;IF ERRORS, TRY AGAIN
299 000632 004567 000272 COR: JSR R5, READ ;READ THE DATA WRITTEN
300 000636 000435 BR RETRY3 ;IF ERRORS, TRY AGAIN
301 000640 104412 000000 000126 CDATAS, BEGIN, RBUFPA ;REQUEST FOR MONITOR TO CHECK DATA
302 000646 000650 .+2 ;IF ERROR, CONTINUE

```

```
304
305
306 000650* 104413 000000* PASS: ENDTLS,BEGIN ; SIGNAL END OF ITERATION.
307 000650* 104413 000000* BR NEXT ; MONITOR SHALL TEST END OF PASS
308
309 000654* 000713 FINI: ENDS,BEGIN ; DROP THE MODULE
310
311 000656* 104410 000000* ;
312
313
314
315
316 000662* 105267 002752 RETRY1: INCR TRV1 ; COUNT THE RETRYS
317 000666* 122767 000003 002744 CMPR #3,TRV1 ; LIMIT EXCEEDED ?
318 000674* 001335 BNE CO ; NO, GO TRY IT AGAIN
319 000676* 104403 000000* 003572* MSGNS,BEGIN,EXCED1 ; ASCII MESSAGE CALL WITH COMMON HEADER
320 000704* 000424 BR NEXT ; GO ON TO NEXT DRIVE
321
322
323
324
325
326 000706* 105267 002727 RETRY2: INCR TRV2 ; COUNT RETRYS
327 000712* 122767 000003 002721 CMPR #3,TRV2 ; LIMIT EXCEEDED ?
328 000720* 001341 BNE CO ; NO, TRY AGAIN
329 000722* 104403 000000* 003600* MSGNS,BEGIN,EXCED2 ; ASCII MESSAGE CALL WITH COMMON HEADER
330 000730* 000412 BR NEXT ; GO ON TO NEXT DRIVE
331
332
333
334
335
336 000732* 105267 002704 RETRY3: INCR TRV3 ; COUNT RETRYS
337 000744* 001341 CMPR #3,TRV3 ; LIMIT EXCEEDED ?
338 000746* 104403 000000* 003600* BNE CO ; NO, GO TRY AGAIN
339 000754* 000400 MSGNS,BEGIN,EXCED3 ; ASCII MESSAGE CALL WITH COMMON HEADER
340 BR NEXT ; GO ON TO NEXT DRIVE
341
342
343
344
345
346 000756* 032767 000001 177032 NEXTA: BIT #R10,SRI ; IS HARD DROP OPTION CHOSEN ?
347 000764* 002904 BNE ZS ; NO, CHECK THE OTHER OPTION
348 000774* 001007 BITR #R10,FLAG ; HARD ERROR DROP SET ?
349 000776* 000167 177502 15: JMP NEXT ; YES, GO DROP THE DRIVE
350 001002* 032767 000002 177006 25: BIT #R11,SRI ; GO ON TO NEXT DRIVE
351 001012* 000771 BNE ZS ; IS SOFT DROP OPTION CHOSEN ?
352 001014* 004767 000234 35: JSP PC,DROP ; YES, GO DROP THE DRIVE
353 001020* 012767 000001 177060 BR ; NO, GO ON TO NEXT DRIVE
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
```

```
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
```

```

; ----- RF11 DISK DRIVERS -----
WRITE: MOV #103,FUNC ; LOAD WRITE FUNCTION
MOV #CNT1,@R0 ; LOAD WORD COUNT
MOV #RUFPA,@R1 ; LOAD BUFFER ADDRESS
MOV #XMEM ; LOAD EXTENDED MEMORY BITS
BR ; CONTINUE
WRITECK: MOV #107,FUNC ; LOAD WRITE-CHECK FUNCTION
MOV #CNT1,@R0 ; LOAD WORD COUNT
MOV #RUFPA,@R1 ; LOAD BUFFER ADDRESS
MOV #XMEM ; LOAD EXTENDED MEMORY BITS
BR ; CONTINUE
READ: MOV #105,FUNC ; LOAD READ FUNCTION
MOV #CNT1,@R0 ; LOAD WORDCOUNT
MOV #RUFPA,@R1 ; LOAD BUFFER ADDRESS
MOV #XMEM ; LOAD EXTENDED MEMORY BITS
BR ; CONTINUE
CLEAR: MOV #R17B,@R0 ; ISSUE A CONTROL RESET
RTS R5 ; RETURN
GOGO: MOV #INTPUT,@VECTOR ; SET INTERRUPT ENTRY POINTER
MOV #DSK1,@R0 ; LOAD THE DISK ADDRESS
MOV #DSK2,@R1 ; LOAD DRIVE AND DISK ADDRESS
BIS #XMEM,FUNC ; LOAD EXTENDED MEMORY BITS
MOV #FUNC,@R0 ; EXECUTE THE FUNCTION
EXIT: BR ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
NTRUPT: ;
;-----
;IRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI
;
15: JSP #R5,@R0 ; GO CHECK FOR ERRORS
RTS #R5 ; ERRORS DETECTED, RETURN
TST #R5 ; NO ERRORS, SKIP RETRY
RTS #R5 ; RETURN OK
;
DROP: MOV #R1 ; INITIALIZE DROP PICKER
MOV #DRIVE,R0 ; GET THE DRIVE NUMBER
BEG DS ; IF DRIVE 0 GO DROP IT
ASL R1 ; NO, AIM AT THE NEXT DRIVE
DEC R0 ; IS THIS THE ONE ?
BNE IS ; NO, LOOK AGAIN
R1C R1,DRVICE ; DROP THE DRIVE
;*****
;CONVERT DRIVE TO ASCII AND
;STORE AT ADRI
OTCAS,BEGIN,DRIVE,ADRI
;*****
PTS PC ; RETURN

```

414

```

415 001312 005267 001072 BLOCK: INC BLK1 ; STEP TO NEXT BLOCK
416 001316 022767 000400 001064 CMP #256,,BLK1 ; BLOCK LIMIT REACHED ?
417 001324 100002 000000 RPL IS ; NO, CONTINUE
418 001326 005087 001056 CLE BLK1 ; YES, RESET BLOCK #
419 001332 016767 001052 1S: MOV BLK1,BLK2 ; READ WHERE WRITE
420 001340 000207 000000 RTS PC ; RETURN
-----
421 001342 005267 001036 DRVADR: INC DRVVE ; COUNT A DRIVE
422 001346 062767 000004 001032 ADD #BIT2,DRVSFT ; DRIVE # LINED UP WITH RFDA
423 001354 142767 000010 002255 RICR #BIT3,FLAG ; CLEAR END OF DRIVES FLAG
424 001362 122767 000010 001014 IS,,DRVVE ; ALL DRIVES CHECKED ?
425 001370 001404 000000 REQ IS ; YES, GO FLAG END OF DRIVES
426 001372 006267 001004 ASP DRIVE ; NO, IS NEXT DRIVE CHOSEN ?
427 001376 103384 000000 RCC DRVADR ; NO, GO TRY ANOTHER DRIVE
428 001400 000207 000000 RTS PC ; YES, RETURN
429 001402 152767 000010 002227 1S: RISR #BIT3,FLAG ; SET END OF DRIVES FLAG
430 001410 012767 177777 000766 MOV #-1,DRVVE ; RESET DRIVE COUNTER
431 001416 012767 177774 000762 MOV #17774,DRVSFT ; RESET SHIFTED DRIVE #
432 001424 016767 000750 000750 NOV DVCE,DRIVE ; RESTORE CHOSEN DRIVES
433 001432 000207 000000 RTS PC ; RETURN
-----
434 001434 016700 000750 ROOM: MOV BLK1,R0 ; SAVE THE CURRENT BLOCK NUMBER
435 001440 012711 000400 MOV #256,,R1 ; LOAD MAX. NUMBER OF BLOCKS
436 001444 005002 000000 CLR R2 ; ZERO REG. 2
437 001446 142767 000001 002163 RICR #BIT0,FLAG ; CLEAR 32K INDICATOR
438 001454 016767 176462 000000 MOV #R0,R3 ; GET THE TRANSFER SIZE
439 001462 100001 000000 SUB #R2,,R1 ; GET # OF BLOCKS LEFT ON DISK
440 001466 003010 000000 CMP #32,,R1 ; MORE THAN 32K LEFT?
441 001470 152767 000001 002141 RGT IS ; NO, CONTINUE
442 001476 162701 000040 000000 RISR #BIT0,FLAG ; YES, SET THE INDICATOR
443 001484 003010 000000 SUB #32,,R1 ; SUBTRACT 32K WORTH OF BLOCKS
444 001486 002425 000040 000000 CMP #64,,R1 ; WAS THERE MORE THAN 64K LEFT?
445 001490 003010 000000 BLT SS ; YES, MUST BE AN ERROR
446 001494 002425 000000 TST R1 ; ANY BLOCKS LEFT ON DISK?
447 001498 003010 000000 BLE SS ; NO, RETURN OK
448 001502 002425 000000 ADD #1024,,R2 ; GET TOTAL # OF WORDS LEFT
449 001506 002425 000000 DEC R1 ; ALL BLOCKS ADDED IN?
450 001510 003010 000000 RGT SS ; NO, KEEP ADDING
451 001514 003010 000000 TST R3 ; REQUEST LARGER THAN 32K?
452 001518 100007 000000 RPL IS ; NO, GO CHECK THAT CONDITION
453 001522 003010 000000 RPL IS ; YES, GET RID OF 32K
454 001526 100007 000000 RPL IS ; YES, GET RID OF 32K
455 001530 042703 000001 002075 BIC #BIT15,R3 ; MORE THAN 32K LEFT?
456 001534 132767 000000 BITR #BIT0,FLAG ; NO, RETURN OK
457 001538 003010 000000 RR 4S ; YES, GO COMPARE
458 001542 003010 000000 RR 4S ; YES, GO COMPARE
459 001546 132767 000001 002063 3S: BITR #BIT0,FLAG ; MORE THAN 32K?
460 001550 003010 000000 HNE SS ; YES, PLENTY OF ROOM LEFT, ERROR
461 001554 003010 000000 CMP R2,R3 ; ENOUGH ROOM FOR THE TRANSFER?
462 001558 002425 000000 BLT SS ; YES, RETURN OK
463 001562 003010 000000 BLT SS ; NO RETURN OK
464 001566 003010 000000 RTS (R5)+ ; MUST BE A REAL ERROR
465 001570 152767 000020 002043 6S: RISR #BIT4,FLAG ; RETURN INDICATING THE ERROR
466 001574 006205 000000 RTS PC ; SET OVERFLOW FLAG
467 001578 006205 000000 RTS PC ; RETURN OK
-----

```



```

469
470 001576 014167 176304 ERSUB2: MOV -(R1),ASB ; LOAD THE DATA
471 001602 010167 176274 MOV R1,ASADR ; LOAD ADDRESS OF DATA WRITTEN
472 001605 014267 176276 MOV -(R2),AWAS ; LOAD THE DATA
473 001612 010267 176276 MOV R2,WASADR ; LOAD ADDRESS OF DATA READ
474 001616 005261 ; RESET REG. 1
475 001620 005222 ; RESET REG. 2
477 001622 016767 001576 176250 ERSUB1: MOV RFCS,CSRA ; LOAD ADDR. OF CURRENT CSR
478 001630 017767 001570 176244 MOV MRFC,ACSR ; LOAD CONTENTS OF CURRENT CSR
479 001636 000267 ; RTS PC ; RETURN
480
481
482
483
484
485
486 001640 005777 001560 ERRORS: TST @RFCS ; ANY ERRORS ?
487 001644 100016 BPL 2S ; NO, RETURN
488 001656 004000 001550 BIT #BIT1,@RFCS ; NON-EXISTENT DISK?
489 001659 002707 BR 1S ; NO CONTINUE
490 001666 004567 JSR RS,ROOM ; YES, MAKE SURE ENOUGH ROOM FOR TRANSFER
491 001669 004767 JSR PC,ERSUB1 ; YES, DONT REPORT AS ERROR
492 001670 004767 BIT #BIT1,@RFCS ; HARD ERROR ?
493 001676 001005 BR 4S ; YES, GO REPORT
494 001700 000417 JSR RS,CLEAR ; GO CLEAR OUT ANY ERRORS
495 001700 004567 BNE 4S ; MUST BE SOFT ERROR
496 001706 004567 JSR PC,CLEAR ; GO CLEAR OUT ANY ERRORS
497 001710 000205 TST (R5)+ ; SKIP RETRY
498 001712 005567 CLR ERRTYP ; RETURN OK
499 ; ***** UNKNOWN ERROR *****
500 ; *****
501 001716 104405 000000 003424 HRDERS,BEGIN,TABLE ; *****
502 ; *****
503 001724 004567 JSR RS,CLEAR ; GO CLEAR OUT ERRORS
504 001736 000401 001701 DISB #BIT0,FLAG ; SET HARD ERROR DROP FLAG
505 001740 003277 000200 001466 BR 4S ; RETURN
506 001746 001411 BEQ 6S ; DATA REQUEST LATE?
507 001750 005267 INC DTCNT ; INCREMENT ERROR COUNTER
508 001754 032767 00004 176034 BIT #BIT2,SRI ; TYPE OUT ERROR?
509 ; *****
510 001762 001014 BNE MSGNS,BEGIN,DLTERR ; NO
511 001764 104403 000000 003624 MOV #2,ERRTYP ; DATA LATE
512 001772 012767 000002 176106 ; *****
513 ; *****
514 002000 104406 000000 003424 SOFDS,BEGIN,TABLE ; *****
515 ; *****
516 002006 152767 000002 001623 DISB #BIT1,FLAG ; SET SOFT ERROR DROP FLAG
517 002014 000205 RTS R5 ; RETURN, ERRORS
518 ; *****

```

```

519
520 002016 016700 175764 SETUP: MOV ADDR,R0 ; GET DEVICE ADDRESS
521 002022 010079 001376 TST (R0)+ ; GENERATE CONTROLLER REGS. ADDRESSES
522 002026 005720 MOV NO,RFWC
523 002030 010067 001372 TST (R0)+
524 002034 005720 MOV NO,RFWA
525 002036 010067 001366 TST (R0)+
526 002042 005720 MOV NO,RFDA
527 002044 010067 001362 TST (R0)+
528 002050 005720 MOV NO,RFDB
529 002054 010067 001356 TST (R0)+
530 002058 005720 MOV NO,RFDB
531 002060 010067 001352 TST (R0)+
532 002064 005720 MOV NO,RFMR
533 002072 005720 TST (R0)+
534 002074 010067 001342 MOV VECTOR,R0 ; GET THE VECTOR ADDRESS
535 002100 016700 175704 MOV R1,(R0)+ ; SET POINTER JUST IN CASE
536 002110 116710 175676 MOV#BRI,(R0) ; SET PRIORITY
537
538 002114 000207 1S: RTS PC ; RETURN
539 ; *****
540
541
542
543
544
545
546
547 002116 005067 000252 CONVRT: CLR DSK1 ; RESET DISK ADDRESSES
548 002122 005067 000250 CLR DSK2
549 002126 006200 ASR R0 ; DIVIDE BLOCK # BY 2 TO GET TRACK #
550 002130 133033 BCC 1S ; 1ST TIME ON THIS TRACK ?
551 002132 012767 002000 000234 MOV #1024,DSK1 ; 2ND TIME STEP TO NEXT 1024 WORDS
552 002140 012701 006011 1S: MOV #R1,R1 ; SHIFT 6 PLACES TO THE LEFT
553 002144 005301 2S: ASL R0 ; TO LINE UP TRACK ADDRESS
554 002148 005301 DEC R1 ; DONE ?
555 002150 003375 BGT 2S ; NO, KEEP SHIFTING
556
557 002152 005300 ASL R0 ; YES, GET MOST SIGNIFICANT BIT FOR RFAE
558 002154 010003 BCC 3S ; BIT SET ?
559 002156 012767 000002 000212 MOV #BIT1,DSK2 ; YES, SET IT FOR RFAE
560 002164 006300 ASL R0 ; YES, SET IT FOR RFAE
561 002166 133002 BCC 4S ; GET NEXT SIGNIFICANT BIT FOR RFAE
562 002170 133002 3S: ASL R0 ; BIT SET ?
563 002174 005067 000202 INCL DSK2 ; YES, SET IT FOR RFAE
564 002200 000207 MOV R0,DSK1 ; GET LEAST SIG. BITS OF TRACK ADR
565 ; *****

```

```

566
567
568 002202 012767 077777 001212 READY: MOV #77777,CLK ; SET THE TIMER
569 002210 165777 001210 1S: TSTR #RPCS ; CONTROLLER READY ?
570 002214 100001 BPL ZS ; NO, CONTINUE
571 002216 000205 RTS RS ; YES, RETURN READY
572
573 002220 004567 176736 2S: JSR R5,CLEAR ; GO CLEAR OUT ANY ERRORS
574 002224 104405 000000 BREAKS,REGIN ; REPORT RETURN TO MONITOR.
575 002230 104407 000000 BREAKS,REGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
576 002234 005367 001162 DEC CLK ; WAIT SOME MORE ?
577 002238 011363 RNE IS ; YES
578 002242 012767 000003 175636 MOV #3,ERRTYP ; CONTROLLER NOT READY CODE
579 ***** ; *****
580 002250 104405 000000 003424 HRDERS,REGIN,TABLE ; CONTROLLER NOT READY
581 ***** ; *****
582 002256 005367 000116 CLR OVICE ; SET TO DROP THE MODULE
583 002262 005725 TST (P5)+ ; SKIP INSTRUCTION FOLLOWING CALL
584 002264 000205 RTS RS ; RETURN
585 ;
586 -----
587
588 002266 004567 176670 REZET: JSR R5,CLEAR ; EXECUTE CONTROLLER RESET
589 002272 004567 177704 JSR R5,READY ; IS CONTROLLER READY ?
590 002276 000401 RR IS ; YES, CONTINUE
591 002300 000207 RTS PC ; NO, RETURN ERROR
592
593 002302 004767 177034 1S: JSR PC,DRVADR ; GET A DRIVE ADDRESS
594 002306 132767 000310 BITR #R13,FLAG ; ALL DRIVES DONE ?
595 002310 010123 BNE ZS ; YES, RETURN
596 002316 010777 000064 MOV DRVSFT,RFAB ; NO, LOAD DISK ADDRESS REG.
597 002324 032777 004000 BIT DRYT11,RFPCS ; DRIVE EXIST ?
598 002332 001763 BEO IS ; YES, CONTINUE
599 002340 012767 176714 JSR PC,DROP ; NO, DROP THE DRIVE
600 ***** ; *****
601 ***** ; *****
602 ***** ; *****
603 ***** ; *****
604 002346 104405 000000 000000 HRDERS,REGIN,NULL ; NON EXISTENT DEVICE
605 ***** ; *****
606 ***** ; *****
607 ***** ; *****
608 ***** ; *****
609 002354 104403 000000 003614 MSGNS,REGIN,DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
610 002362 000747 BPC ; WAKE SURE ALL GET CHECKED
611 002364 000207 RTS PC ; RETURN
612 ;
  
```

```

609
610
611
612
613 002366 000000 DLTCNT: C
614 002370 000000 FUNC: C
615 002374 000000 XMEM: C
616 002376 000000 DSK1: C
617 002380 000000 DSK2: C
618 002384 000000 OVICE: C
619 002388 000000 DRIVE: C
620 002392 000000 DRVSFT: C
621 002396 000000 BLK1: C
622 002400 000000 BLK2: C
623 002404 000000 TSP: C
624 002408 000000 WCNT1: C
625 002412 000000 WCNT2: C
626 002416 000000 CLK: C BLKW 256.
627 002420 000000 BUFIN: C
628 002424 000000 TABLE: C
629 002428 000000 RFCS: C
630 002432 000000 RFW: C
631 002436 000000 RFD: C
632 002440 000000 RFD2: C
633 002444 000000 RFD3: C
634 002448 000000 RFD4: C
635 002452 000000 RFD5: C
636 002456 000000 RFD6: C
637 002460 000000 RFD7: C
638 002464 000000 RFD8: C
639 002468 177777
640
641
642 003446 020040 051104 053111 MES3: .ASCIZ " DRIVE "
643 003450 020040 051104 050117 MES4: .ASCIZ " DROPPED"
644 003454 042520 032534 000 MES5: .ASCIZ " RETRY EXCEEDED"
645 003458 040 042522 051124
646 003462 020111 042105 027563
647 003466 020111 042105 000045
648 003470 053440 044822 042524 MES6: .ASCIZ " WRITE"
649 003474 000
650 003478 000
651 003482 026505 051127 052111 MES7: .ASCIZ " WRITE-CHECK"
652 003486 044103 041505
653 003490 000113
654 003494 051040 040505 000104 MES8: .ASCIZ " READ"
655 003498 042946 052101 020101 MES9: .ASCIZ " DATA REQUEST LATE"
656 003502 042522 052101 051505
657 003506 042124 040514 042524
658 003510 000045
659
660 003572 003514 EXCED1: EVEN
661 003574 003473 MES8
662 003576 177777 MES5
663 003578 003573 EXCED2: MES7
664 003580 003473 MES5
  
```


SVR4	000072R	200#				
SVR5	000074R	201#				
SVR6	000076R	202#				
SVSCNT	000052R	191#				
TABLE	003424R	501#	514	580	629#	
TBUF	002414R	624#				
TRPDFD=	000072R	234#				
TRV1	003826R	234#				
TRY2	003641R	326*	318*	319	680#	
TRY3	003642R	327	327	681#		
VECTOR	000010R	285*	334*	335	682#	
WASADR	000104R	172#	380*	537		
WBUFEA	000136R	221#	365	370		
WBUFEA	000134R	220#	364	369		
WBUFRQ	000140R	223#				
WBUFSZ	000142R	223#				
WCNT1	002416R	271*	271	440		
WCNT2	002420R	272*	272*	368	625#	
WDFR	000116R	266#	267*	363		
WDTD	000114R	213#	235*	373	626#	
WRITCK	001076R	212#	234*			
WRITE	001044R	298	267#			
YFLAG	000005R	291	362#			
XMEM	002372R	170#				
.	= 003844R	365*	370*	375*	383	615#
		303	627#	675#	683#	

. ABS. 000000 000
 003844 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

XRFAGO XRFAGO/SQL(CRF:SYN=DDXCOM,XRFAGO)
 RUN-TIME: 3 SECONDS
 RUN-TIME RATIO: 15/4=3.8
 CORE USED: 7K (13 PAGES)