

.REM -

IDENTIFICATION  
-----

PRODUCT CODE: AC-E745I-MC  
PRODUCT NAME: CXRPBI0 RH11-RH70 SGL PT DSK  
PRODUCT DATE: FEBRUARY 1979  
MAINTAINER: DEC/X11 SUPPORT GROUP

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

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

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

COPYRIGHT (C) 1974,1979 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT  
\*\*\*\*\*

RPB IS AN IOMOD THAT EXERCISES RP04/5/6 DISK DRIVES ON AN RH11/RH70 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE ITY.

2. REQUIREMENTS  
\*\*\*\*\*

HARDWARE: 1 TO 8 RP04/5/6'S WITH AN RH11/RH70 CONTROLLER

STORAGE: RPB REQUIRES:

1. DECIMAL WORDS: 1466
2. OCTAL WORDS: 02672
3. OCTAL BYTES: 5564

3. PASS DEFINITION  
\*\*\*\*\*

ONE PASS OF THE RPB MODULE CONSISTS OF 1300 CYCLES OF THE BASIC TEST SEQUENCE (WRITE, WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS, AND DATA-CHECKS SAME.

4. EXECUTION TIME  
\*\*\*\*\*

ONE PASS OF RPB RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS  
\*\*\*\*\*

DEFAULT PARAMETERS:

DEVADR: 176700, VECTOR: 254, BRI: 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 SECTOR ADDRESS  
D. GET A DRIVE ADDRESS  
E. DO A WRITE == IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT  
F. DO A WRITE-CHECK == IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT  
G. DO A READ == IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT  
H. DO A DATA-CHECK == IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT  
I. IF END OF PASS, REPORT AND GO TO C  
J. IF END OF DRIVES, GO TO C ELSE GO TO D

8. OPERATION OPTIONS  
\*\*\*\*\*  
SR1 BIT0 SET(1):  
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR  
IS ASSUMED AND THE DRIVE IS DROPPED  
  
SR1 BIT0 CLEAR(0):  
IF THE RETRY LIMIT IS EXCEEDED, THE FUNCTION IS ABORTED AND  
THE TESTING CONTINUES  
  
SR1 BIT2 SET(1):  
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT  
  
SR1 BIT2 CLEAR(0):  
TYPE OUT DATA LATE ERRORS AND COUNT THEM  
  
SR1 BIT5 CLEAR (0) ;NORMAL FOR RP04 PACKS THAT ARE FORMATED  
;FOR 16 BIT MODE (PDP-11)  
SR1 BIT5 SET (1) ;FOR RP04 18 BIT FORMATED PACKS

9. NON-STANDARD PRINTOUTS  
\*\*\*\*\*

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

RHCS1 RHCW RHBA RPDA RHCS2 RPDS RPER1 RPAS  
RPLA RHDB RPMR RPDT RPSN RPOF RPDC RPCC  
RPER2 RPER3 RPEC1 RPEC2 RBAE RHCS3 XFER CNT

```

000000* IOMODX <RPBI >,176700,254,5,0,0,1300,,60,BUFIN,256,,1024,
000000* MODULE 150000,RPBI ,176700,254,5,0,0,1300,,60,BUFIN,256,,1024,
; .TITLE RPBI DEC/X11 SYSTEM EXERCISER MODULE
DDXCOM VERSION 6 23-MAY-78
.LIST BIN
;*****
000000* BEGIN;
000000* 050122 044502 040 MODNAM: ,ASCII /RPBI / ;MODULE NAME,
000005* 000 XFLAG: ,BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006* 176700 ADDR: 176700+0 ;1ST DEVICE ADDR,
000010* 000254 VECTOR: 254+0 ;1ST DEVICE VECTOR,
000012* 240 BR1: ,BYTE PRTY5+0 ;1ST BR LEVEL,
000013* 000 BR2: ,BYTE PRTY0+0 ;2ND BR LEVEL,
000014* 000001 DVID1: 0+1 ;DEVICE INDICATOR 1,
000016* 000000 SR1: OPEN ;SWITCH REGISTER 1,
000020* 000000 SR2: OPEN ;SWITCH REGISTER 2,
000022* 000000 SR3: OPEN ;SWITCH REGISTER 3,
000024* 000000 SR4: OPEN ;SWITCH REGISTER 4
;*****
000026* 150000 STAT: 150000 ;STATUS WORD,
000030* 001452 INIT: START ;MODULE START ADDR,
000032* 000252 SPOINT: MODSP ;MODULE STACK POINTER,
000034* 000000 PASCNT: 0 ;PASS COUNTER,
000036* 002424 ICNT: 1300, ;# OF ITERATIONS PER PASS=1300,
000040* 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
000042* 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044* 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046* 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050* 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052* 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054* 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056* 000000 CONFIG: ;RESERVED FOR MONITOR USE
000056* 000000 RES1: 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,
001000* 000000 CSPA: OPEN ;ADDR OF CURRENT CSR,
00102* 000000 SBADR: ;ADDR OF GOOD DATA, OR
00102* 000000 ACSR: OPEN ;CONTENTS OF CSR,
00104* 000000 WASADR: ;ADDR OF BAD DATA, OR
00104* 000000 ASTAT: OPEN ;STATUS REG CONTENTS,
00106* 000000 FRRTYP: ;TYPE OF ERROR
00106* 000000 ASB: OPEN ;EXPECTED DATA,
000110* 000000 AWAS: OPEN ;ACTUAL DATA,
000112* 001734 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
000114* 000000 WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
000116* 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000120* 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122* 000060 IDNUM: 60 ;MODULE IDENTIFICATION NUMBER=60
000124* 000370 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
;*****
000150* 000040 .REPT SPSIZ ;MODULE STACK STARTS HERE,
;*****
000152* .LIST
;*****
000152* .WORD 0
;*****
000152* .LIST
;*****
000152* .ENDR
;*****
000152* MODSP:
;*****

```

222 000252\* 000000  
 223 000254\* 000000  
 224 000256\* 000000  
 225 000260\* 000000  
 226 000262\* 000000  
 227 000264\* 000000  
 228 000266\* 000000  
 229 000270\* 000000  
 230 000272\* 000000  
 231 000274\* 000000  
 232  
 233  
 234  
 235  
 236 000276\* 000000  
 237 000300\* 000000  
 238 000302\* 000000  
 239 000304\* 000000  
 240 000306\* 000000  
 241 000310\* 000000  
 242 000312\* 000000  
 243 000314\* 000000  
 244 000316\* 000000  
 245  
 246 000320\* 000006  
 247 000322\* 000162  
 248 000324\* 000024  
 249 000326\* 133224  
 250 000330\* 132132  
 251  
 252  
 253  
 254 000332\* 000005  
 255 000334\* 000137  
 256 000336\* 000022  
 257 000340\* 114046  
 258 000342\* 113113  
 259  
 260  
 261 000344\* 000000  
 262 000346\* 000000  
 263 000350\* 000000  
 264 000352\* 000000  
 265 000354\* 000000  
 266 000356\* 000000  
 267 000360\* 000000  
 268 000362\* 000000  
 269 000364\* 000000  
 270 000366\* 000000  
 271 000370\* 000400

DLTCNT: 0 ;DATA LATE ERROR COUNTER  
 CLK: 0  
 BLK1: 0 ;DISK BLOCK COUNTER  
 CYL: 0 ;CALCULATED CYLINDER ADDRESS  
 CYLSAV: 0 ;MODIFIED CYLINDER ADDRESS FOR RP04/5  
 SEC: 0  
 CNT: 0 ;CYCLE COUNTER  
 FUNC: 0  
 ONCEE: 0  
 FERADR: 0  
 ;DO NOT CHANGE THE ORDER OF THE NEXT 15 LOCATIONS  
 ;NEEDED FOR MAP22 ROUTINE  
 PA19: 0  
 XMEM: 0  
 PA22: 0  
 EA22: 0  
 MBLKRV: 0 ;HOLDS 1024 WD BLKS PER TRACK  
 MBLKTR: 0 ;HOLDS BLKS PER CYLINDER  
 MODE: 0 ;HOLDS LAST SECTOR # THAT WILL FIT PER TRACK  
 MLOWCY: 0 ;BLKS THAT FIT IN FIRST 410 CYLINDERS (0-409)  
 MHICY: 0 ;BLKS THAT FIT IN LAST 405 CYLINDERS (410-814)  
 ;  
 T16: 6 ;THESE 5 LOC. GET MOVED TO ABOVE 5 IF 16 BIT MODE  
 114. ;1024 WORD BLOCKS/CYLINDER  
 20. ;SECTORS/TRACK  
 46740. ;140, X 114. (CYLINDER 0 - 409)  
 46170. ;405, X 114. (CYLINDER 410 - 814)  
 ;  
 ;\*\*\*\*\*  
 T18: 5 ;THESE 5 LOC GET MAPPED INTO FIRST 5 ABOVE IF 18 BIT MOD  
 95. ;1024 WORD (18 BIT) BLOCKS/CYLINDER  
 15. ;SECTORS/TRACK  
 39950. ;140, X 95. (CYLINDER 0 - 409)  
 38475. ;405, X 95. (CYLINDER 410 - 814)  
 ;\*\*\*\*\*  
 ZERO: 0  
 DSKADR: 0  
 DRIVE: 0  
 DRIVE: 0  
 BLKSAV: 0  
 TBUF: 0  
 WCNT1: 0  
 WCNT2: 0  
 UNITNO: 0  
 MOD1: 0  
 BUFIN: ,BLKW 256.

272 001370\* 000000  
 273 001372\* 000000  
 274 001374\* 000000  
 275 001376\* 000000  
 276 001378\* 000000  
 277 001400\* 000000  
 278 001402\* 000000  
 279 001404\* 000000  
 280 001406\* 000000  
 281 001410\* 000000  
 282 001412\* 000000  
 283 001414\* 000000  
 284 001416\* 000000  
 285 001420\* 000000  
 286 001422\* 000000  
 287 001424\* 000000  
 288 001426\* 000000  
 289 001430\* 000000  
 290 001432\* 000000  
 291 001434\* 000000  
 292 001436\* 000000  
 293 001440\* 000000  
 294 001442\* 000000  
 295 001444\* 000274  
 296 001446\* 000256  
 297 001450\* 177777

TABLE:  
 RHCS1: 0  
 RHWC: 0  
 RHBA: 0  
 RPA: 0  
 RHCS2: 0  
 RPDS: 0  
 RPER: 0  
 RPAS: 0  
 RPLA: 0  
 PHDB: 0  
 RPR: 0  
 RPDT: 0  
 RPS: 0  
 RPOF: 0  
 RPDC: 0  
 RPCC: 0  
 RPER2: 0  
 RPER3: 0  
 RPEC1: 0  
 RPEC2: 0  
 RHBAE: 0  
 RHCS3: 0  
 XFERAD: FERADP  
 XFERCT: CHT  
 177777

```

298 001452* 012767 002000 176436 START: MOV #1024,,WDFR ;1024, WORDS FROM MEM/ITERATION
299 001460* 012767 000400 176426 MOV #256,,WDT0 ;256, WORDS TO MEM/ITERATION
300 001466* 012767 000003 176424 MOV #3,INTR ;3 INTERRUPTS/ITERATION
301 001474* 005067 176566 CLR CNT ;ZERO END OF PASS TESTER
302 001500* 005067 176546 CLR DLTCNT ;CLEAR DATA LATE ERROR COUNTER
303 001504* 012767 010000 176654 MOV #BIT12,MOD1 ;SETUP FOR 16 BIT MODE (NORMAL)
304 001512* 012700 000320* MOV #T16,R0 ;GET TABLE OF VALUES FOR 16 BIT MODE
305 001516* 032767 000040 176272 BIT #BIT5,SR1 ;16 BIT MODE?
306 001524* 001404 BEQ 2$ ;YES
307 001526* 005067 176634 CLR MOD1 ;NO 18 BIT MODE, CLEAR FMT BIT
308 001532* 012700 000332* MOV #T18,R0 ;GET TABLE OF VALUES FOR 18 BIT MODE
309 001536* 012701 000306* 2$: MOV #MBLKR,R1 ;BEGIN OF TABLE ENTRIES
310 001542* 012702 000005 MOV #5,R2 ;NO. OF ENTRIES
311 001546* 012021 3$: MOV (R0)+,(R1)+ ;STORE AN ENTRY
312 001550* 005302 DEC R2 ;COUNT IT
313 001552* 001375 BNE 3$ ;DO MORE
314 001554* 012767 000007 176510 1$: MOV #7,ONCEE ;SET ONE TIME ONLY FLAGS
315 001562* 105067 003772 CLRB FLAG ; CLEAR FLAGS
316 001566* 16767 176222 176554 MOV DVID1,DVICE ; GET DRIVE INDICATOR
317 001574* 016767 176550 DAOST: MOV DVICE,DRIVE ; ALSO SAVE IT IN DRIVE
318 001602* 016706 176224 MOV SPOINT,R6 ;RESTORE STACK POINTER
319 001606* 012767 177777 176442 MOV #-1,BLK1 ; INITIALIZE BLOCK COUNTER
320 001614* 012767 177777 176542 MOV #-1,UNITNO ; INITIALIZE DEVICE COUNTER
321 001622* 122737 000011 000041 CMPR #11,#41 ;IF RP IS THE LOAD MEDIUM THEN
322 001630* 001021 BNE 3$ ;BEGIN
323 001632* 113700 000040 MOVB #40,R0 ; GET LOAD-DRIVE NUMBER
324 001636* 012701 000001 MOV #1,R1 ; INITIALIZE DRIVE MASK
325 001642* 105700 5$: TSTR R0 ; WHILE R0>0 DO
326 001644* 001403 BEQ 6$ ; BEGIN
327 001646* 006301 ASL R1 ; SHIFT DRIVE MASK TO NEXT DRIVE
328 001650* 105300 DECR R0 ; DOWNCOUNT DRIVE #
329 001652* 000773 BF 5$ ; END
330 001654* 130167 176472 6$: RTR R1,DRIVE ; IF LOAD-DRIVE IS SELECTED, THEN
331 001660* 001405 BEQ 3$ ; BEGIN
332 001662* 113767 000040 176474 MOVB #40,UNITNO ; DEVICE NUMBER TO DROP
333 001670* 004767 001422 JSR PC,DROP ; DROP LOAD-DEVICE
334 ; END
335 ;END
336 001674* 004767 003000 3$: JSR PC,SETUP ;SET REGISTER ADDRESSES
337 001700* 004767 002704 JSR PC,REZET ;
338 001704* 000400 BR CT ; CONTINUE
339 001706* 012767 177777 176450 CT: MOV #-1,UNITNO ;INIT UNITNO AGAIN
340 001714* 016767 176430 176430 MOV DVICE,DRIVE ;SETUP TO FIND DRIVES
341 001722* 005767 176422 TST DVICE ; DROP THE MODULE ?
342 001726* 001002 BNE RESTR ; NO
343 001730* 000167 000250 JMP FINI ; YES DROP MODULE
344
345 001734* 104415 000000* 000124* RESTR: GETPAS,BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
346 001742* 016767 176164 176412 MOV RBUFVSZ,WCNT2 ; SAVE READ BUFFER SIZE
347 001750* 005467 176406 NEG WCNT2 ; GET THE 2'S COMPLEMENT
348 001754* 016767 176276 176372 STRT: MOV BLK1,BLKSAV ;INIT BLOCK COUNTER
349 001762* 104414 000000* 176364 STRT1: GWBVS, BEGIN ;GET WRITE BUFFER INFORMATION
350 001766* 016767 176150 176364 MOV WRBUFVSZ,WCNT1 ; SAVE WRITE BUFFER SIZE
351 001774* 005467 176360 2$: NEG WCNT1 ; GET THE 2'S COMPLEMENT

```

```

354 002000* 004767 002174 NEXT: JSR PC,FDUNIT ; FIND UNIT #
355 002004* 016777 176354 177366 MOV UNITNO,MRHCS2 ;GET UNIT # SO BOTH PORTS
356 ; WILL BE LOOKING FOR SAME DRIVE
357 002012* 005767 176332 TST DVICE ; ANY DRIVES LEFT ?
358 002016* 001002 BNE 1$ ; YES
359 002020* 000167 000160 JMP FINI ; NO DROP MODULE
360 002024* 132767 000010 003526 1$: BTR #BIT3,FLAG ;MORE DRIVES ON SYS?
361 002032* 001401 BEQ ABW ;YES
362 002034* 000747 ER STRT ;YES CORRECT SETUP FOR DUEL PORT RETURN
363 002036* 004567 000506 ABW: JSR R5,UPDATT ;UPDATE DSK ADDR
364 002042* 004567 002412 JSR R5,READY ; IS DRIVE READY ?
365 002046* 000402 BR 1$ ; YES, CONTINUE
366 002050* 004767 002202 JSR PC,NOTRDY ; NOT READY, GO WAIT UNTIL IT IS
367 002054* 005067 003464 1$: CLR TPY1 ; ZERO RETRY COUNTERS
368 002060* 005067 003462 CLR TPY3 ;
369 002064* 005067 003460 CLR TPY5 ;
370 002070* 005067 003456 CLR TPY7 ;
371 002074* 005067 003454 CLR TPY9 ;
372 002100* 005067 003452 CLR TPY11 ;
373 002104* 004567 000250 GOS: JSR R5,WRITE ; WRITE SOME DATA
374 002110* 000167 000100 JMP RETRY1 ; IF ERRORS, TRY IT AGAIN
375 002114* 132767 000004 003436 BTR #BIT2,FLAG ; DISK OVERFLOW ?
376 002122* 001410 BEQ GOA ; NO, CONTINUE
377 002124* 142767 000004 003426 DAOCL: BICR #BIT2,FLAG ; YES, CLEAR OVERFLOW FLAG
378 002132* 012767 177777 176116 MOV #-1,BLK1 ; RESET BLOCK #
379 002140* 000167 177610 JMP STRT ; CONTINUE
380 002144* 004567 000304 GOA: JSR R5,WRITCK ; WRITE-CHECK THE DATA
381 002150* 000167 000072 JMP RETRY2 ; IF ERRORS, TRY AGAIN
382 002154* 004567 000400 GOB: JSR R5,READ ; READ THE DATA WRITTEN
383 002160* 000167 000114 JMP RETRY3 ; IF ERRORS, TRY AGAIN
384 002164* 104412 000000* 000126* CDATA6,BEGIN,RBUFP6 ; REQUEST FOR MONITOR TO CHECK DATA
385 002172* 002174* +2 ; IF ERROR, CONTINUE
386 002174* CYCLE: ENDITS,BEGIN ;SIGNAL END OF ITERATION,
387 002174* 104413 000000* ;MONITOR SHALL TEST END OF PASS
388 ; NO, CONTINUE
389 002200* 000167 177574 JMP NEXT
390 002204* FINI: ENDS,BEGIN ; DROP THE MODULE
391 002204* 104410 000000* ;
392 ;-----
393 002210* 000167 177564 NXT1: JMP NEXT ;GET NEXT DRIVE
394 002214* 105267 003324 RETRY1: INCB TRY1 ; COUNT THE RETRYS
395 002220* 122767 000003 003316 CMPB #3,TRY1 ; LIMIT EXCEEDED
396 002226* 001402 BEQ 1$ ; YES
397 002230* 000167 177650 JMP GOS ; NO RETRY
398 002234* 1$: MSGN6,BEGIN,EXCED1 ;ASCII MESSAGE CALL WITH COMMON HEADER
399 002234* 104403 000000* 005466* JMP NEXTA ; GO TO NEXT DRIVE
400 002242* 000167 000064 RETRY2: INCB TRY2 ; COUNT THE RETRYS
401 002246* 105267 003273 003265 CMPB #3,TRY2 ; LIMIT EXCEEDED
402 002252* 122767 000003 003265 BEQ 1$ ; YES
403 002260* 001402 177656 JMP GOA ; NO RETRY
404 002262* 000167 177656 1$: MSGN6,BEGIN,EXCED2 ;ASCII MESSAGE CALL WITH COMMON HEADER
405 002266* 104403 000000* 005474* JMP NEXTA ; GO TO NEXT DRIVE
406 002274* 000167 000032 RETRY3: INCB TRY3 ; COUNT THE RETRYS
407 002274* 105267 003242 003234 CMPB #3,TRY3 ; LIMIT EXCEEDED
408 002300* 105267 003242 003234
409 002304* 122767 000003

```

```

410 002312* 001402          BEQ     18      ; YES
411 002314* 000167 177634   JMP     GOB     ; NO RETRY
412 002320*                   ;
413 002320* 104403 000000* 005502* 18: MSGN$,BEGIN,EXCED3 ;ASCII MESSAGE CALL WITH COMMON HEADER
414 002326* 000167 000000   JMP     NEXTA   ; GO TO NEXT DRIVE
  
```

```

415 002332* 032767 000001 175456 NEXTA: BIT     #BIT0,SRI      ; DROP THE DRIVE
416 002340* 001405          BEQ     18      ; NO, SKIP TO NEXT DRIVE
417 002342* 004767 000750   JSR     PC,DROP  ; YES, DROP OFFENDING DRIVE
418 002346* 104403 000000* 005524* MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
419 002354* 000167 177420   18:    JMP     NEXT      ; GO ON TO NEXT DRIVE
420                               ; -----
421
422
423                               ; MACRO LINEUP EABITS ; LINE UP EA BITS FOR RHCS1
424 LINEUP EABITS ; LINE UP EA BITS FOR RHCS1
425 ;
426 ;
427 ;
428 ;
429 ;
430 ;
431 ;
432 ;
433 ;
  
```



```
434 ; ----- RP DISK DRIVERS -----
435
436 002360* 012767 000161 175702 WRITE: MOV #161,FUNC ; LOAD WRITE FUNCTION
437 002366* 016777 175766 176776 MOV WCNT1,0RHWC ; LOAD WORD COUNT
438 002374* 016777 175534 176772 MOV WBUFA,0RHBA ; LOAD BUFFER ADDRESS
439 002402* 016777 175740 176766 MOV DSKADR,0RPDA ; LOAD DISK ADDRESS
440 002410* 004767 001316 JSR PC,RP06CK ; CHECK FOR RP06 DRIVE
441 002414* 016777 175640 177002 MOV CYL,0RPDC ; LOAD CYLINDER ADDRESS
442 002422* 016767 175634 175630 MOV CYLSAV,CYL ; RESTORE CALCULATED CYLINDER ADDRESS
443 ; LINEUP WBUFEA ; LINE UP EA BITS FOR RHCS1
444 002450* 000167 000306 JMP GOGO ; CONTINUE
445 002454* 012767 000151 175606 WRITCK: MOV #151,FUNC ; LOAD WRITE-CHECK FUNCTION
446 002462* 016777 175672 176702 MOV WCNT1,0RHWC ; LOAD WORD COUNT
447 002470* 016777 175440 176676 MOV WBUFA,0RHBA ; LOAD BUFFER ADDRESS
448 002476* 016777 175644 176672 MOV DSKADR,0RPDA ; LOAD DISK ADDRESS
449 002504* 004767 001422 JSR PC,RP06CK ; CHECK FOR RP06 DRIVE
450 002510* 016777 175544 176706 MOV CYL,0RPDC ; LOAD CYLINDER ADDRESS
451 002516* 016767 175540 175534 MOV CYLSAV,CYL ; RESTORE CALCULATED CYLINDER ADDRESS
452 ; LINEUP WBUFEA ; LINE UP EA BITS FOR RHCS1
453 002544* 000167 000212 JMP GOGO ; CONTINUE
454 002550* 004767 001116 UPDAT: JSR PC,BLOCK
455 002554* 016700 175476 MOV BLK1,R0
456 002560* 004767 001172 JSR PC,CONVRT
457 002564* 000205 RTS R5
458
459 002566* 012767 000171 175474 READ: MOV #171,FUNC ; LOAD READ FUNCTION
460 002574* 016777 175562 176570 MOV WCNT2,0RHWC ; LOAD WORD COUNT
461 002602* 016777 175320 176564 MOV RBUFA,0RHBA ; LOAD BUFFER ADDRESS
462 002610* 016777 175532 176560 MOV DSKADR,0RPDA ; LOAD DISK ADDRESS
463 002616* 004767 001310 JSR PC,RP06CK ; CHECK FOR RP06 DRIVE
464 002622* 016777 175432 176574 MOV CYL,0RPDC ; LOAD CYLINDER ADDRESS
465 002630* 016767 175426 175422 MOV CYLSAV,CYL ; RESTORE CALCULATED CYLINDER ADDRESS
466 ; LINEUP RBUFEA ; LINE UP EA BITS FOR RHCS1
467 002656* 000441 BR GOGO ; CONTINUE
```

```
468 002660* 016777 175500 176512 CLEAR: MOV UNITNO,0PHCS2 ; LOAD UNIT ADDRESS
469 002666* 012777 000011 176474 MOV #11,0RHCS1 ; ISSUE A DRIVE CLEAR
470 002674* 000240 NOP ; WAIT
471 002676* 000240 NOP ; FOR DRIVE CLEAR TO FINISH
472 002700* 104407 000000* BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR....
473 002704* 104407 000000* BREAKS,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
474 002710* 012777 000021 176452 MOV #21,0RHCS1 ; ISSUE A PACK ACK
475 002716* 105777 176446 1S: TST #RHCS1 ; FUNCTION DONE ?
476 002722* 100405 BMI 2S ; YES, CONTINUE
477 002724* 104407 000000* BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR....
478 002730* 104407 000000* BREAKS,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
479 002734* 000770 BR 1S ; NO, WAIT TILL DONE
480 002736* 012777 177777 176442 2S: MOV #1,0RPAS ; CLEAR AS BIT
481 002744* 012777 040000 176416 MOV #RIT14,0RHCS1 ; CLEAR ANY CONTROLLER ERRORS
482 002752* 016777 175410 176442 MOV MCD1,0RPOF ; SET BIT FOR 11 FORMAT
483 002760* 000205 RTS R5 ; RETURN
484
485 002762* 016777 175376 176410 GOGO: MOV UNITNO,0RHCS2 ; LOAD UNIT SELECT
486 002770* 012777 003116* 175012 MOV #INTRPT,0VECTOR ; SET INTERRUPT ENTRY POINTER
487 002776* 032767 001000 175052 BIT #ADDR22,RES1 ; 11/70?
488 003004* 001434 BEQ 1S ; NO
489 003006* 017767 176362 175262 MOV 0RHBA,PA18 ; GET 18 BIT ADDR
490 003014* 006267 175260 ASR XMEM ; SHIFT EA BITS TO POSITION 4,5
491 003020* 006267 175254 ASR XMEM
492 003024* 006267 175250 ASR XMEM
493 003030* 006267 175244 ASR XMEM
494 003034* 104416 000000* 000276* MAP22S, BEGIN,PA18 ; GET 22-BIT ADDR FROM 18-BIT ADDR
495 003042* 016777 175234 176324 MOV PA22,0RHBA ; LOAD BA REG
496 003050* 016777 175230 176362 MOV EA22,0RHBAE ; LOAD BAE REG
497 003056* 042767 000034 175220 BIC #34,EA22 ; CLEAR UNWANTED BITS
498 003064* 000367 175214 SWAB EA22 ; LOAD INTO BITS 8,9
499 003070* 016767 175210 175202 MOV EA22,XMEM ; LOAD XMEM TO SET INTO FUNCTION CODE
500 003076* 056767 175176 175164 1S: BIS XMEM,FUNC ; LOAD EXTENDED MEMORY BITS
501 003104* 016777 175160 176256 MOV FUNC,0RHCS1 ; EXECUTE THE FUNCTION
502 003112* 104400 000000* EXITS,BEGIN ; EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
503
504 003116* ;
505 ;
506 003116* 000004 000000* 003124* FIRCS,BEGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
507 ;
508 ;
509 003124* 004567 000224 1S: JSR R5,ERRORS ; GO CHECK FOR ERRORS
510 003130* 000205 RTS R5 ; ERRORS DETECTED, RETURN
511 003132* 005725 TST (R5)+ ; NO ERRORS, SKIP RETRY
512 003134* 005725 TST (R5)+
513 003136* 000205 RTS R5 ; RETURN OK
```

514 003140 016700 175112 ROOM: MOV BLK1,R0 ; SAVE THE CURRENT BLOCK NUMBER
515 003144 016703 174772 MOV WBUF\$Z,R3 ; GET THE TRANSFER SIZE
516 003150 132767 000002 002402 BITB #BIT1,FLAG ; PLENTY OF ROOM LEFT ?
517 003156 001451 48 BEQ ; YES, CONTINUE
518 003160 142767 000001 002372 BICB #BIT0,FLAG ; CLEAR 32K INDICATOR
519 003166 016701 175124 MOV MHICY,R1 ; LOAD MAX. NUMBER OF BLOCKS
520 003172 162701 000003 SUB #3,R1
521 003176 005002 CLR R2 ; ZERO REG. 2
522 003200 160001 SUB R0,R1 ; GET NUMBER OF BLOCKS LEFT ON DISK
523 003202 022701 000100 CMP #64,,R1 ; MORE THAN 64 BLOCKS LEFT ?
524 003206 003435 BLE 48 ; YES
525 003210 022701 000040 CMP #32,,R1 ; MORE THAN 32K LEFT ?
526 003214 003005 BGT 18 ; NO, CONTINUE
527 003216 152767 000001 002334 BICB #BIT0,FLAG ; YES, SET THE INDICATOR
528 003224 162701 000040 SUB #32,,R1 ; SUBTRACT 32K WORTH OF BLOCKS
529 003230 005701 18: TST R1 ; ANY BLOCKS LEFT ON DISK ?
530 003232 003425 BLE 56 ; NO, RETURN OK
531 003234 062702 002000 ADD #1024,,R2 ; GET TOTAL # OF WORDS LEFT
532 003240 005301 DEC R1 ; ALL BLOCKS ADDED IN ?
533 003242 003372 BGT 18 ; NO, KEEP ADDING
534 003244 005703 TST R3 ; REQUEST LARGER THAN 32K ?
535 003246 100007 BPL 28 ; NO, GO CHECK THAT CONDITION
536 003250 042703 100000 BIC #BIT15,R3 ; YES, GET RID OF 32K
537 003254 132767 000001 002276 BITB #BIT0,FLAG ; MORE THAN 32K LEFT ?
538 003262 001411 BEQ 56 ; NO, RETURN OK
539 003264 000404 BR 36 ; YES, GO COMPARE
540 003266 132767 000001 002264 28: BITB #BIT0,FLAG ; MORE THAN 32K LEFT ?
541 003274 001002 BNE 48 ; YES, PLENTY OF ROOM, ERROR
542 003276 020203 36: CMP R2,R3 ; ENOUGH ROOM FOR THE TRANSFER ?
543 003300 002402 BLT 56 ; NO, RETURN OK
544
545 003302 005725 40: TST (R5)+ ; YES, MUST BE A REAL ERROR
546 003304 000205 PTS R5 ; RETURN, ERROR
547 003306 152767 000004 002244 58: BICB #BIT2,FLAG ; SET OVERFLOW FLAG
548 003314 000205 RTS R5 ; RETURN OK
549
550 ;
551 003316 012701 000001 DROP: MOV #1,R1 ; INITIALIZE DROP PICKER
552 003322 016700 175036 MOV UNITNO,R0 ; GET THE DRIVE NUMBER
553 003326 001403 BFG 28 ; IF DRIVE 0 GO DROP IT
554 003330 006301 18: ASL R1 ; POINT TO NEXT DRIVE
555 003332 005300 DEC R0 ; IS THIS THE ONE ?
556 003334 001375 BNE 18 ; NO, LOOK AGAIN
557 003336 040167 175006 28: BIC R1,DEVICE ; DROP THE DRIVE
558 ;\*\*\*\*\*
559 ;CONVERT UNITNO TO ASCII AND
560 ;STORE AT ADR1
561 003342 104420 000000 000364 OTOAs,BEGIN,UNITNO,ADR1
562 003350 005534
563 ;\*\*\*\*\*
564 003352 000207 RTS PC ; RETURN
565 ;

566 003354 005777 176010 ERRORS: TST @RHCS1 ; ATTENTION OR ERROR ?
567 003360 100402 BMI 28 ; YES
568 003362 000167 000300 JMP RESYNC ; NO GO ON TO NEXT FUNCTION
569 003366 032777 001000 176010 228: BIT #BIT9,@RPFR1 ; ADDRESS OVERFLOW ?
570 003374 001403 BEQ 18 ; NO, CONTINUE
571 003376 004567 177536 JSR R5,ROOM ; YES, IS IT A REAL ERROR ?
572 003402 000531 BP RESYNC ; NO, CONTINUE
573 003404 032777 002000 175770 18: BIT #BIT10,@RPDS ; DID LBT SET?
574 003412 001125 BNE RESYNC ; YES
575 003414 004767 001152 JSR PC,ERSUB1 ; LOAD ERROR INFORMATION
576 003420 005777 175754 TST @RHCS2 ; IS THIS A DATA LATE ERROR?
577 003424 100012 BPL 118 ; NO
578 003426 005267 174620 INC DLTCNT ;ADD 1 TO DATA LATE COUNTER
579 003432 032767 000004 174356 BIT #BIT2,SR1 ;TYPE ERROR AND COUNT IT?
580 003440 001107 BNE 88 ;NO
581 003442 104403 000000 005510 MSGNS,BEGIN,DLTERR ;ASCII MESSAGE CALL WITH COMMON HEADER
582 003450 000440 BR 98 ;CONT
583 003452 032777 002000 175710 118: BIT #BIT13,@RHCS1 ; MASSBUS CONTROL PARITY ERROR ?
584 003460 001043 BNE 36 ; YES
585 003462 032777 000400 175710 BIT #BIT8,@RHCS2 ; MASSBUS DATA PARITY ERROR ?
586 003470 001043 BNE 48 ; YES
587 003472 032777 040000 175670 BIT #BIT14,@RHCS1 ; TRANSFER ERROR ?
588 003500 001015 BNE 28 ; YES
589 003502 032777 040000 175672 BIT #BIT14,@RPDS ; ANY DRIVE ERRORS ?
590 003510 001036 BNE 56 ; YES
591 003512 005777 175670 TST @RPAS ; ANY ATTENTIONS ACTIVE ?
592 003516 001033 BNE 56 ; YES, CONTINUE
593 003520 005067 174362 CLR EPRTYP ;UNKNOWN ERROR
594 ;\*\*\*\*\*
595 003524 104405 000000 001370 HDRS,BEGIN,TABLE ; SPECIAL CONDITION SET BUT NO REASON FOUND
596 ;\*\*\*\*\*
597 003532 000452 BR 88 ; RETURN
598 003534
599 003534 104403 000000 005452 28: MSGNS,BEGIN,TRERR ;ASCII MESSAGE CALL WITH COMMON HEADER
600 003542 032777 062000 175630 BIT #BIT14|BIT13|BIT10,@RHCS2 ;SOFT ERROR?
601 003550 001416 BEQ 56 ;HARD ERROR
602 003552 012767 000030 174326 96: MOV #30,ERRTYP
603 ;\*\*\*\*\*
604 003560 104406 000000 001370 SOFERS,BEGIN,TABLE ;TRANSFER ERROR
605 ;\*\*\*\*\*
606 003566 000434 BR 88
607 003570
608 003570 104403 000000 005456 38: MSGNS,BEGIN,MCPERR ;ASCII MESSAGE CALL WITH COMMON HEADER
609 003576 000403 BR 58 ; GO DUMP REGISTERS
610 003600
611 003600 104403 000000 005462 48: MSGNS,BEGIN,MDPERR ;ASCII MESSAGE CALL WITH COMMON HEADER
612 003606 005777 175574 58: TST @RPAS ; ANY ATTENTIONS ACTIVE ?
613 003612 001402 BEQ 68 ; NO, CONTINUE
614 003614 004767 000566 JSR PC,WHO ; YES, FIND OUT WHICH DRIVE IT IS
615 003620 016700 175566 MOV RHDB,R0 ; SAVE ADDRESS OF DATA BUFFER
616 003624 105777 175550 TSTB @RHCS2 ; CAN DATA BUFFER BE READ ?
617 003630 100403 BMI 78 ; YES, CONTINUE
618 003632 012767 000344 175552 MOV #ZERO,RHDB ; NO, LOAD ADDRESS OF ZERO
619 003640 012767 000030 174240 78: MOV #30,ERRTYP ;ERROR DURING DATA XFER
620 ;\*\*\*\*\*
621 003646 104405 000000 001370 HDRS,BEGIN,TABLE ; DUMP RH11 AND RP REGISTERS

```
622  
623 003654* 010067 175532 ;*****  
624 003660* 004567 176774 ; RESTORE DATA BUFFER ADDRESS  
625 003664* 000205 ; GO CLEAR OUT ERRORS  
; ERRORS DETECTED, RETURN
```

```
626 003666* 005725  
627 003670* 000205  
628  
629  
630  
631 ;THIS ROUTINE DETERMINES IF THERE IS ENOUGH ROOM ON THE DISK TO  
632 ;DO ANOTHER TRANSFER. IF NOT, PROGRAM GOES TO RESYNC TO  
633 ;ALLOW BOTH PROCESSORS TO RESYNC AND TO RESTART.  
634 003672* 005267 174360  
635 003676* 132767 000002 001654  
636 003704* 001012  
637 003706* 026767 174402 174342  
638 003714* 101017  
639 003716* 005067 174334  
640 003722* 152767 000002 001630  
641 003730* 000411  
642  
643 003732* 026767 174360 174316 1$: CMP MHCY,BLK1 ; BLOCK # IN WITHIN RANGE ?  
644 003740* 101005 ; YES, RETURN  
645 003742* 005067 174310 ; NO, RESET BLOCK #  
646 003746* 142767 000002 001604 ; SET FLAG TO LOWER RANGE FOR NEW PACK CYCLE  
647 003754* 000207 2$: RTS PC ; RETURN
```



```
734 004406* 017701 174774 WHO: MOV 0RPAS,R1 ; GET THE ATTENTION SUMMARY
735 004412* 017704 174762 MOV 0RHCS2,R4 ; SAVE THE STATUS REGISTER
736 004416* 012702 000001 MOV #BIT0,R2 ; SET POINTER TO DRIVE 0
737 004422* 005003 CLR R3 ; ZERO THE DRIVE COUNTER
738 004424* 030201 101 BIT R2,R1 ; FIND IT ?
739 004426* 001006 BNE 20 ; YES, CONTINUE
740 004430* 005203 INC R3 ; NO, INCREMENT THE DRIVE COUNTER
741 004432* 006302 ASL R2 ; SET POINTER TO NEXT DRIVE
742 004434* 032702 000400 BIT #BIT0,R2 ; ALL DONE ?
743 004440* 001771 BEQ 10 ; NO, GO AGAIN
744 004442* 000207 RTS PC ; SOMEBODY LIED -- NO ATTENTIONS SET
745 004444* 042704 000007 201 BIC #7,R4 ; CLEAR OUT OLD UNIT NUMBER
746 004450* 050304 BIS R3,R4 ; LOAD THE NEW UNIT NUMBER
747 004452* 010477 174722 MOV R4,0RHCS2 ; RESTORE THE STATUS REGISTER
748 004456* 000207 RTS PC ; RETURN
; -----
749
750
751 004460* 016777 173700 174712 READY: MOV UNITNO,0RHCS2 ; LOAD UNIT ADDRESS
752 004466* 017700 174710 MOV 0RPDS,R0 ; SAVE STATUS IN R0
753 004472* 105700 TSTR R0 ; DRIVE READY ?
754 004474* 100022 BPL 10 ; NO
755 004476* 032700 000100 BIT #BIT6,R0 ; VOLUME VALID ?
756 004502* 001417 BEQ 10 ; NO
757 004504* 032700 000400 BIT #BIT8,R0 ; DRIVE PRESENT ?
758 004510* 001414 BEQ 10 ; NO
759 004512* 032700 000400 BIT #BIT11,R0 ; WRITE LOCKED ?
760 004516* 001011 BNE 10 ; YES
761 004520* 032700 010000 BIT #BIT12,R0 ; MEDIUM ON LINE ?
762 004524* 001406 BEQ 10 ; NO
763 004526* 032700 040000 BIT #BIT14,R0 ; ANY ERRORS ?
764 004532* 001003 BNE 10 ; YES
765 004534* 005700 TST R0 ; ATTENTION SET ?
766 004536* 100401 BPL 10 ; YES
767 004540* 000205 RTS R5 ; RETURN READY
768 004542* 005725 101 TST (R5)+ ; SKIP INSTRUCTION FOLLOWING CALL
769 004544* 000205 RTS R5 ; RETURN AS NOT READY
; -----
770
771
772
773 004546* 014167 173334 ERSUB2: MOV -(R1),ASB ; LOAD THE DATA
774 004552* 010167 173324 MOV R1,SBADR ; LOAD ADDRESS OF DATA WRITTEN
775 004556* 014267 173326 MOV -(R2),AWAS ; LOAD THE DATA
776 004562* 010267 173316 MOV R2,WASADR ; LOAD ADDRESS OF DATA READ
777 004566* 005721 TST (R1)+ ; RESET REG. 1
778 004570* 005722 TST (R2)+ ; RESET REG. 2
779
780 004572* 016767 174572 173300 ERSUB1: MOV RHCS1,CSRA ; LOAD ADR OF CURRENT CSR
781 004600* 017767 174564 173274 MOV 0RHCS1,ACSR ; LOAD CONTENTS OF CURRENT CSR
782 004606* 000207 PUS PC ; RETURN
```

```
783
784
785
786
787
788
789 004610* 012777 000040 174562 REZET: MOV #BITS,0RHCS2 ; ISSUE AN RH11 INIT
790 004616* 012777 177777 174562 MOV #-1,0RPAS ; CLEAR ALL ATA BITS
791 004624* 012767 077777 173422 MOV #77777,CLK ; SET THE TIMER
792 004632* 105777 174532 101 TSTR 0RHCS1 ; CONTROLLER READY ?
793 004636* 100417 BMI 20 ; YES, CONTINUE
794 004640* 104407 000000* BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR....
795 004644* 104407 000000* BREAKS,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
796 004650* 005367 173400 DEC CLK ; WAIT SOME MORE ?
797 004654* 001366 BNE 10 ; YES
798 004656* 005067 173466 CLR DVICE ; NO, SET TO DROP THE MODULE
799 004662* 012767 000003 173216 MOV #3,ERRTYP ; CONTROLLER NOT READY
800 ;*****
801 004670* 104405 000000* 001370* HRDERS,BEGIN,TABLE ; CONTROLLER NOT READY
802 ;*****
803 004676* 000207 201 RTS PC ; RETURN
804 ; -----
```

```

005 004700 016700 173102 SETUP: MOV ADDR,R0 ; GET DEVICE ADDRESS
006 004704 010067 174460 MOV R0,RHCS1 ; GENERATE REGISTER ADDRESSES
007 004710 005720 TST (R0)+
008 004712 010067 174454 MOV R0,RHWC
009 004716 005720 TST (R0)+
010 004720 010067 174450 MOV R0,RHBA
011 004724 005720 TST (R0)+
012 004726 010067 174444 MOV R0,RPDA
013 004732 005720 TST (R0)+
014 004734 010067 174440 MOV R0,RHCS2
015 004740 005720 TST (R0)+
016 004742 010067 174434 MOV R0,RPDS
017 004746 005720 TST (R0)+
018 004750 010067 174430 MOV R0,RPER1
019 004754 005720 TST (R0)+
020 004756 010067 174424 MOV R0,PPAS
021 004762 005720 TST (R0)+
022 004764 010067 174420 MOV R0,RPPLA
023 004770 005720 TST (R0)+
024 004772 010067 174414 MOV R0,RHDB
025 004776 005720 TST (R0)+
026 005000 010067 174410 MOV R0,RPMR
027 005004 005720 TST (R0)+
028 005006 010067 174404 MOV R0,RPDT
029 005012 005720 TST (R0)+
030 005014 010067 174400 MOV R0,RPSN
031 005020 005720 TST (R0)+
032 005022 010067 174374 MOV R0,RPOF
033 005026 005720 TST (R0)+
034 005030 010067 174370 MOV R0,RPDC
035 005034 005720 TST (R0)+
036 005036 010067 174364 MOV R0,RPCC
037 005042 005720 TST (R0)+
038 005044 010067 174360 MOV R0,RPER2
039 005050 005720 TST (R0)+
040 005052 010067 174354 MOV R0,RPER3
041 005056 005720 TST (R0)+
042 005060 010067 174350 MOV R0,RPEC1
043 005064 005720 TST (R0)+
044 005066 010067 174344 MOV R0,RPEC2
045 005072 032767 001000 172756 BIT #ADDR22,RES1 ;11/707
046 005100 001400 BEQ 18 ;NO
047 005102 005720 TST (R0)+
048 005104 010067 174330 MOV R0,RHBAE
049 005110 005720 TST (R0)+
050 005112 010067 174324 MOV R0,RHCS3
051 005116 016700 172666 1$: MOV VECTOR,R0 ; GET VECTOR ADDRESS
052 005122 012720 001754 MOV #STRT,(R0)+ ; SET POINTER JUST IN CASE
053 005126 016710 172660 MOV# R0,(R0) ; SET PRIORITY
054 005132 000207 2$: RTS PC ; RETURN
    
```

```

055 005134 020040 051124 047101 MES1: .ASCIZ * TRANSFER ERROR%*
056 005142 043123 051105 020040
057 005150 051105 047522 022522
058 005156 0000
059 005157 0000 046440 051501 MES2: .ASCIZ * MASSBUS PARITY ERROR%*
060 005164 041123 051525 020040
061 005172 040520 044522 054524
062 005200 020040 051105 047522
063 005206 022522 0000
064 005211 0000 046440 051501 MES3: .ASCIZ * MASSBUS DATA PARITY ERROR%*
065 005216 041123 051525 020040
066 005224 040504 040524 020040
067 005232 040520 044522 054524
068 005240 020040 051105 047522
069 005246 022522 0000
070 005251 0000 042040 044522 MES4: .ASCIZ * DRIVE *
071 005256 042526 020040 0000
072 005263 0000 042040 047522 MES5: .ASCIZ * DROPPED%*
073 005270 050120 042105 000045
074 005276 051040 052105 054522 MES6: .ASCIZ * RETRY EXCEEDED%*
075 005304 042440 041530 042505
076 005312 042504 022504 0000
077 005317 0000 020040 051127 MES7: .ASCIZ * WRITE*
078 005324 052111 000105
079 005330 020040 053440 044522 MES8: .ASCIZ * WRITE-CHECK*
080 005336 042524 041455 042510
081 005344 045503 0000
082 005347 0000 020040 042522 MES9: .ASCIZ * READ*
083 005354 042101 0000
084 005357 0000 040504 040524 MES10: .ASCIZ * DATA LATE ERROR%*
085 005364 046040 052101 020105
086 005372 051105 047522 022522
087 005400 0000
088 005401 0000 051104 053111 MES11: .ASCIZ * DRIVE NOT READY%*
089 005406 020105 047516 020124
090 005414 042522 042101 022531
091 005422 0000
092 005423 0000 047503 046125 MES12: .ASCIZ * COULD NOT GET DRIVE%*
093 005430 020104 047516 020124
094 005436 042507 020124 051104
095 005444 053111 022505 0000
096 005452 005452* .EVEN
097 005452 005134* TRERR: MES1
098 005454 177777 177777
099 005456 005157* MCPERR: MES2
000 005460 177777 177777
001 005462 005211* MDPERR: MES3
002 005464 177777 177777
003 005466 005317* EXCED1: MES7
004 005470 005276* MES6
005 005472 177777 177777
006 005474 005330* EXCED2: MES8
007 005476 005276* MES6
008 005500 177777 177777
009 005502 005347* EXCED3: MES9
010 005504 005276* MES6
    
```







RHBA	001374R	275#	438#	447*	461*	489	495*	810*								
RHBAE	001440R	293#	496#	848*												
RHCS1	001370R	273#	469#	474*	475	481*	501*	566	583	587	712	728*	780	781		
		792	806#													
RHCS2	001400R	277#	355#	468*	485*	576	585	600	616	735	747*	751*	789*	814*		
RHCS3	001442R	294#	850#													
RHDB	001412R	282#	615	618*	623*	824*										
RHWC	001372R	274#	437*	446*	460*	808*										
ROOM	003140R	514#	571													
RPAS	001406R	280#	480*	591	612	734	790*	820*								
RPCC	001426R	288#	711#	836*												
RPDA	001376R	276#	439*	448*	462*	812*										
RPDC	001424R	287#	441*	450*	464*	834*										
RPDS	001402R	278#	573	589	752	816*										
RPDT	001416R	284#	683	685	828*											
RPEC1	001434R	291#	842*													
RPEC2	001436R	292#	844*													
RPER1	001404R	279#	569	818*												
RPER2	001430R	289#	838*													
RPER3	001432R	290#	840*													
RPLA	001410R	281#	822*													
RPMR	001414R	283#	826*													
RPOF	001422R	286#	482*	832*												
RPSN	001420R	285#	830*													
RP06CK	004132R	440	449	463	682*											
RSTRT	000112R	199#														
SBADR	000102R	192#	774*													
SEC	000264R	227#	649*	669*	674	676*	677									
SETUP	004700R	336	805#													
SOFcnt	000042R	175#														
SOFEP6=	104406	222#	604													
SOFPAS	000046R	177#														
SPOINT	000032R	171#	318													
SPSIZ =	000040	1#	215													
SR1	000016R	164#	305	415	579											
SR2	000020R	165#														
SR3	000022R	166#														
SR4	000024R	167#														
START	001452R	170	298#													
STAT	000026R	169#														
STR1	001754R	349#	362	379	852											
STRT1	001762R	350#														
SVR0	000062R	184#														
SVR1	000064R	185#														
SVR2	000066R	186#														
SVR3	000070R	187#														
SVR4	000072R	188#														
SVR5	000074R	189#														
SVR6	000076R	190#														
SYScnt	000052R	179#														
TABLE	001370R	272#	595	604	621	726	801									
TBUF	000356R	266#														
TOUT	005520R	917#														
TREPR	005452R	599	897#													
TRK	005561R	650#	664*	672	678	942*										
TRPDFD=	000022	222#														

TRY1	005544R	367#	394*	395	929*										
TRY10	005555R	938#													
TRY11	005556R	372#	939#												
TRY12	005557R	940#													
TRY2	005545R	401#	402	930#											
TRY3	005546R	368#	408*	409	931*										
TRY4	005547R	932#													
TRY5	005550R	369#	933#												
TRY6	005551R	934#													
TRY7	005552R	370#	935#												
TRY8	005553R	936#													
TRY9	005554R	371#	937#												
T16	000320R	246#	304												
T18	000332R	250#	308												
UNITNO	000364R	269#	320*	332*	339*	355	468	485	552	561	694*	696	702*	751	
UPDATT	002550R	363	454#												
VECTOR	000010R	160#	486*	851											
WASADR	000104R	194#	776*												
WBUFEA	000136R	209#	444	453											
WBUFPA	000134R	208#	438	447											
WBUFRO	000140R	210#													
WBUFSZ	000142R	211#	352	515											
WCNT1	000360R	267#	352*	353*	437	446									
WCNT2	000362R	268#	347*	348*	460										
WDFR	000116R	201#	298*												
WDTO	000114R	200#	299*												
WHO	004406R	614	734#												
WRITCK	002454R	380	445#												
WRITE	002360R	373	436#												
XFERAD	001444R	295#													
XFERCT	001446R	296#													
XFLAG	000005R	158#													
XMEM	000300R	237#	444*	453*	467*	490*	491*	492*	493*	499*	500				
ZERO	000344R	261#	618												
.	= 005564R	271#	385	896#	925#	928#	944#								

. ABS. 000006 000  
 005564 001

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

XRPBI0,XRPBI0/SOL/CRF:SYM=DDXCOM,XRPBI0  
 RUN=TIME: 1 2,4 SECONDS  
 RUN=TIME RATIO: 18/5=3,4  
 CORE USED: 7K (13 PAGES)