

B01

IDENTIFICATION

Product Code: MAINDEC-11-DZRCA-B-D
Replaces: MAINDEC-11-DZRCA-A-D
Product Name: RC11 STATIC TEST
Date Created: 31-MAR-73
Maintainer: Diagnostic Group
Author: Bob Brain/C CASWELL

COPYRIGHT (C) 1972, 1973
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS.

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
 - 2.1 Equipment
 - 2.2 Storage
 - 2.3 Preliminary Programs
3. LOADING PROCEDURE
4. STARTING PROCEDURE
 - 4.1 Worst case operation
5. OPERATING PROCEDURE
 - 5.1 Control switch settings
 - 5.2 Routine abstracts
6. ERROR REPORTS
7. MISCELLANEOUS
8. RUNTIMES

1. Abstract

The RC11 STATIC TEST is a series of small, simple tests designed to insure the basic operation of the RC11. This test used in conjunction with the RC11 DISK DATA and RC11 MULTI DISK assures the user of an error free system, when used in its entirety.

2. Requirements

2.1 Equipment

PDP-11 with RC11 and RS64

2.2 Storage

Program occupies from 0 to 11776

2.3 Preliminary Programs

Processor and memory tests

3. Loading procedure

Use standard procedure for ABS tapes

4. Starting procedure

4.1 Worst case disk test unit zero

- A) Set switch register equal to 200
- B) Depress load address
- C) Set switch register equal to zero
- D) Depress start

5. Operating procedure

5.1 Control switch settings

SW14	1 0	Delete typeouts Report message
SW11	1 0	Loop on test Continue to next test
SW10	1 0	Halt after error report Continue after error report

5.2 Routine abstracts

ADT1 - Test word access during a WRITE

In this test a one word WRITE is attempted on each address of track zero. If no access is accomplished within 100 milliseconds the error condition missed transfer should set. If this flag should fail to set, A program time out will be reported. If no control error occurs and address confirmed takes place. The routine then checks the address pointer for the correct terminating address.

ADT2X - Test word access during a READ

In this test a one word READ is attempted on each address of track zero. If no access is accomplished within 100 milliseconds the error condition missed transfer should set. If this flag should fail to set a program time out will be reported. If no control error occurs and address confirmed takes place, the routine then checks the address pointer register for the correct terminating address.

6. Error Reports

6.1 Static and address report.

ERROR	XXX	XXXXXX	XXXXXX
	A	B	C

A= Is the tag for the listing
B= What was expected (WORK1) optional
C= What was received (WORK)

When a report only contains one word the program was expecting zeros but received what was reported.

6.2 Error bits in RCCS and RCER

Bit layout of DCS register

	BIT15=	ERROR
	BIT14=	DATA ERROR
	BIT13=	ADDRESS ERROR
	BIT12=	WRITE LOCK ERROR
	BIT11=	NON-EXISTENT DISK
	BIT10=	WRITE CHECK ERROR
	BIT9=	INHIBIT INCREMENTING CA
	BIT8=	ABORT
	BIT7=	READY
	BIT6=	INTERRUPT ENABLE
	BIT5=	EXTENDED MEMORY 1 (XM1)
	BIT4=	EXTENDED MEMORY 0 (XM0)
	BIT3=	MAINTENANCE
	BIT2-1=	FUNCTION REGISTER
BIT 2	BIT 1	OPERATION
0	0	LOOK AHEAD
1	0	READ
0	1	WRITE
1	1	WRITE CHECK
	BIT0=	GO (WRITE ONLY BIT)

Note: If an error occurs, the following information is available to the user in the DAE:

BIT4=	ADDRESS NOT FOUND
BIT5=	DISK OVERFLOW
BIT6=	ADDRESS SYNC. ERROR
BIT7=	ADDRESS PARITY ERROR
BIT8=	B TRACK ERROR
BIT9=	A TRACK ERROR
BIT12=	NONEXISTENT MEMORY
BIT13=	DATA SYNC. ERROR
BIT14=	BLOCK CHECK ERROR
BIT15=	DATA LATE

6.3 PARITY ERROR
THIS MESSAGE IS REPORTED IF THE MEMORY DEFECTS A PURITY ERROR
DURING EXECUTION.

6.4 END
END

This message is reported at the end of one complete pass of
the disk system.

7. Miscellaneous

In some address tests the program depends on writing and
reading data correctly from the disk, and if it does not it
may report an address failure, when in fact it was a data
failure.

8. Runtime

Typeout will occur within 5 Min.

158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186

000000
000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000
000004
104403
104407
104405
000000

%
;TITLE MAINDEC-11-DZRCA-A RC11 STATIC TEST REPLACES D5JA-PB1
;COPYRIGHT 1971, 1972, DIGITAL EQUIPMENT CORP. MAYNARD, MASS.
;PROGRAM BY BOB BRAIN/C. CASWELL

.ENABL ABS
N= 0
BIT0= 1
BIT1= 2
BIT2= 4
BIT3= 10
BIT4= 20
BIT5= 40
BIT6= 100
BIT7= 200
BIT8= 400
BIT9= 1000
BIT10= 2000
BIT11= 4000
BIT12= 10000
BIT13= 20000
BIT14= 40000
BIT15= 100000
SCOPE= IOT
WRITE= TRAP+3
WRCHECK=TRAP+7
READ= TRAP+5
.= 0

;TRAP CATCHER 0 - 776

```

187      000200 000200      =200
188 000200 000167 000674      JMP      START
189
190      ;STATIC ROUTINES
191
192      000300      =300
193 000300 000167 000734      JMP      STAI1
194 000304 000167 001006      JMP      STAI2
195 000310 000167 001032      JMP      STAI3
196 000314 000167 001056      JMP      STAI4
197 000320 000167 001102      JMP      STAI5
198 000324 000167 001126      JMP      STAI6
199 000330 000167 001152      JMP      STAI7
200 000334 000167 001204      JMP      STAI10
201 000340 000167 001256      JMP      STAI11
202
203 000344 000167 001326      JMP      STAI12
204 000350 000167 001372      JMP      STAI13
205
206 000354 000167 001432      JMP      STAI14
207 000360 000167 001476      JMP      STAI15
208
209 000364 000167 001532      JMP      STAI16
210 000370 000167 001576      JMP      STAI17
211
212 000374 000167 006514      JMP      LATST
213 000400 000167 007062      JMP      CHKBLK
214 000404 000167 001626      JMP      STAI22
215
216 000410 000167 002464      JMP      STAI40
217
218 000414 000167 003336      JMP      STAI56
219
220 000420 000167 004174      JMP      STAI74
221 000424 000167 004322      JMP      STAI77
222 000430 000167 004502      JMP      STAI03
223 000434 000167 004656      JMP      ST105X
224 000440 000167 005060      JMP      NXMTSM
225 000444 000167 005360      JMP      STAI06
226
227 000450 000167 005504      JMP      STAI10
228
229 000454 000167 005630      JMP      STAI12
230
231 000460 000167 005754      JMP      STAI14
232
233 000464 000167 006050      JMP      ADT1
234
235
236 000470 000167 006232      JMP      ADT2X
237
238

```

```

:TEST RESET TO CONTROL REGISTER
:TEST RESET TO CURRENT ADDRESS REGISTER
:TEST RESET TO WORD COUNT REGISTER
:TEST RESET TO DISK ADDRESS REGISTER
:TEST RESET TO DISK EXT. ADDRESS REGISTER
:TEST RESET TO DATE BUFFER REGISTER
:TEST RESET TO MAINTENANCE REGISTER
:CAN WE SET W/R BITS IN RCCS REGISTER
:CAN WE CLEAR THE RCCS REGISTER
:USING DISK CLEAR.
:CAN WE SET ALL RCBA BITS
:CAN WE CLEAR ALL RCBA BITS
:USING DISK CLEAR
:CAN WE SET ALL RCWC BITS
:CAN WE CLEAR ALL RCWC BITS
:USING DISK CLEAR
:CAN WE SET 13 OF THE RCDA BITS
:CAN WE CLEAR ALL THE RCDA BITS
:USING DISK CLEAR
:TEST LOOK AHEAD
:BLOCK CHECK TEST
:EXECUTE A ONE WORD WRITE
:FOLLOWED BY A ONE WORD WRITE CONTINUE
:EXECUTE A ONE WORD WRITE CHECK
:FOLLOWED BY A ONE WORD WRITE CHECK CONTINUE
:EXECUTE A ONE WORD READ
:FOLLOWED BY A ONE WORD READ CONTINUE
:TEST TRACK INCREMENT
:TEST DISK INCREMENT
:TEST THAT NED RAISES ERROR FLAG
:CHECK RCBA INHIBIT
:TEST NON-EXISTENT MEMORY ERROR
:TEST THAT THE DISK WILL NOT TRAP
:AT PRIORITY 7
:TEST THAT THE DISK WILL NOT TRAP
:AT PRIORITY 6
:TEST THAT THE DISK WILL NOT TRAP
:AT PRIORITY 5
:TEST THAT THE DISK WILL TRAP
:AT PRIORITY 4
:ADDRESS TEST 1
:CHECK TIMING BY EXECUTING
:A ONE WORD WRITE
:ADDRESS TEST 2
:CHECK TIMING BY EXECUTING
:A ONE WORD READ

```

```

239          000650          .=-650
240
241          ;MAINTENANCE ROUTINES
242
243 000650 000167 007420      JMP      SELWC      ;LOAD WORD COUNT REG WITH SWR
244 000654 000167 007424      JMP      SELBA      ;LOAD CURRENT ADDRESS REG WITH SWR
245 000660 000167 007430      JMP      SELDA      ;LOAD DISK ADDRESS REG WITH SWR
246 000664 000167 007434      JMP      SELER      ;LOAD DISK EXT. ADDRESS REG WITH SWR
247 000670 000167 007440      JMP      SELRCDB    ;LOAD DATA BUFFER REG WITH SWR
248 000674 000167 007444      JMP      MOVLK      ;MOVE LOOK AHEAD INTO LIGHTS
249 000700 000167 007452      JMP      SELCS      ;LOAD FUNCTION REG WITH SWR
250 000704 000167 007516      JMP      STAMP      ;SELECT TRACKS STATICLY
251
252
253          ;RC11 DATA TEST
254          ;VECTORS USED IN PROGRAM
255
256          ;#1 LOC 210 DISK INTERRUPT
257          ;#2 LOC 30 EMT (TELETYPE OUTPUT)
258          ;#3 LOC 34 TRAP (DISK HANDLERS)
259
260          001000          .=-1000
261
262          ;I/O ADDRESS POINTERS
263
264 001000 177570      SWR:    177570      ;SWITCH REGISTER
265 001002 177776      PS:     177776      ;PROCESSOR STATUS REGISTER
266 001004 177566      TPB:    177566      ;TELETYPE REGISTERS
267 001006 177562      TKB:    177562
268 001010 177564      TPS:    177564
269 001012 177560      TKS:    177560
270 001014 033604      ODT:    33604
271
272          ;DISK I/O REGISTERS
273
274 001016 177446      RCCS:   177446      ;DISK CONTROL REGISTER 444
275 001020 177450      RCWC:   177450      ;WORD COUNT REGISTER
276 001022 177452      RCBA:   177452      ;CURRENT MEMORY ADDRESS REGISTER
277 001024 177442      RCDA:   177442      ;LOWER 16 BITS OF DISK ADDRESS
278 001026 177444      RCER:   177444      ;EXTENSION ADDRESS REGISTER 442
279 001030 177456      RCDB:   177456      ;DATA BUFFER REGISTER
280 001032 177454      RCMR:   177454      ;MAINTENANCE REGISTER
281 001034 177440      RCLA:   177440      ;LOOK AHEAD REGISTER
282 001036 000210      RCVEC:  210         ;INTERUPT VECTOR

```

```

283
284
285 001040 000000
286 001042 146723
287 001044 000000
288 001046 000000
289 001050 000000
290 001052 000000
291 001054 000000
292 001056 000000
293 001060 000000
294 001062 000000
295 001064 000000
296 001066 000000
297 001070 000000

```

;RC11 DEDICATE REGISTERS (MEMORY)

```

FLAG: 0 ;INTERNAL PROGRAM FLAG
RANNU: 146723 ;RANDOM NUMBER PRIME
WRDCT: 0 ;WORKING WORD COUNT
DMA: 0 ;WORKING RCDA
PATNU: 0 ;DATA PATTERN INDEX
BUF: 0 ;WORKING DATA BUFFER (OUT-IN)
TWRDCT: 0 ;TEMP WORD COUNT
TDMA: 0 ;TEMP RCDA
SWRDCT: 0 ;STANDARD WORD COUNT
ERCOUNT: 0 ;ERROR COUNT FOR MESSAGES.
SAVE: 0
SAV1: 0
PASS: 0

```

```

298
299
300
301 001072 000000
302 001074 000000
303 001076 000000

```

;RC11 WORK REGISTERS
;(CAN BE CHANGED IN ANY ROUTINE)

```

WORK: 0
WORK1: 0
WORK2: 0

```

```

304
305 001100 000005
306 001102 012706 001000
307 001106 012767 000006 176670
308 001114 005067 176666
309 001120 012767 000340 177064
310 001126 012767 010710 176674
311 001134 012767 000340 176670
312 001142 012767 010226 176664
313 001150 012767 000340 176660
314 001156 012777 000340 177616
315 001164 012767 010156 176626
316 001172 012767 000340 176622
317 001200 012767 011454 176616
318 001206 012767 000340 176606
319 001214 005067 177626
320 001220 012767 011242 176632
321 001226 052777 000100 177556
322 001234 004767 010036

```

```

START: RESET
MOV #1000,%6
MOV #6,4
CLR 6
MOV #340,212
MOV #EMTRP,30
MOV #340,32
MOV #DISK,34
MOV #340,36
MOV #340,3PS
MOV #LOOP,20
MOV #340,22
MOV #DOWN,24
MOV #340,22
CLR DMA
MOV #TTYH,60
BIS #100,3TKS
JSR %7,MAMF

```

```

;CLEAR THE WORLD
;SET UP STACK

;LOCK UP INTERRUPTS
;SET UP TTY POINTER
;LOCK UP INTERRUPTS
;SET UP DISK HANDLER POINTER
;LOCK UP INTERRUPTS
;LOCK UP INTERRUPT LEVELS
;SET UP FOR SCOPE LOOP
;LOCK UP PRIORITY
;SET PWR FAIL
;LOCK UP PRIORITY
;CLEAR RCDA REGISTERS

;SET UP PARITY SWITCHES

```

323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367

001240	000005		
001242	017767	177550	177622
001250	032767	177577	177614
001256	001405		
001260	012767	000000	177574
001266	004567	007256	
001272	105767	177574	
001276	001005		
001300	012767	000001	177554
001306	004567	007236	
001312	000004		
001314	001240		
001316	000005		
001320	017767	177476	177544
001326	001405		
001330	012767	000002	177524
001336	004567	007206	
001342	000004		
001344	001316		
001346	000005		
001350	017767	177444	177514
001356	001405		
001360	012767	000003	177474
001366	004567	007156	
001372	000004		
001374	001346		

```

;WE ARE NOW ENTERING THE STATIC TEST
;IF THE OPERATOR WOULD LIKE TO CHECK
;THE DISK REGISTERS PRIOR TO ENTERING THIS
;TEST WE HAVE SOME HANDY ROUTINES
;WHICH WOULD ALLOW YOU TO LOAD THESE
;REGISTERS UNDER SWITCH REGISTER CONTROL
;PLEASE REFERENCE THE STARTING ADDRESS
;TO SEE WHICH ROUTINE BEST SUITS YOUR
;PROBLEM.
;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
;TO CLEAR ALL THE DISK REGISTERS
;TEST CONTROL REGISTER

```

```

STAI1:  RESET
        MOV     @RCCS,WORK
        BIT     #177577,WORK
        BEQ     XSTAI1
ERR0:   MOV     #0,ERCOUNT
        JSR     %5,STAER
XSTAI1: TSTB   WORK
        BNE     LPST1
ERR1:   MOV     #1,ERCOUNT
        JSR     %5,STAER
LPST1:  SCOPE
        STAI1

```

```

;CLEAR THE WORLD
;FETCH CONTROL REGISTER
;IS IT CLEARED
;REGISTER OK (TEST READY)
;***** ERROR 0 *****
;REPORT STATIC ERROR
;TEST FOR READY
;BRANCH IF READY
;***** ERROR 1 *****
;REPORT READY NOT SET
;ENTER SCOPE LOOP

```

;TEST CURRENT ADDRESS REGISTER

```

STAI2:  RESET
        MOV     @RCBA,WORK
        BEQ     LPST2
ERR2:   MOV     #2,ERCOUNT
        JSR     %5,STAER
LPST2:  SCOPE
        STAI2

```

```

;CLEAR THE WORLD
;FETCH RCBA REGISTER
;REGISTER OK
;***** ERROR 2 *****
;REPORT ERROR
;ENTER SCOPE LOOP

```

;TEST WORD COUNT REGISTER

```

STAI3:  RESET
        MOV     @RCWC,WORK
        BEQ     LPST3
ERR3:   MOV     #3,ERCOUNT
        JSR     %5,STAER
LPST3:  SCOPE
        STAI3

```

```

;CLEAR THE WORLD
;FETCH RCWC REG.
;REGISTER OK
;***** ERROR 3 *****
;REPORT ERROR
;ENTER SCOPE LOOP

```



```

398 001506 000005          STAI7: RESET          ;CLEAR THE WORLD
399 001510 017767 177316 177354      MOV      @RCMR,WORK    ;FETCH MAINTENANCE REG
400 001516 032767 167377 177346      BIT      #167377,WORK
401 001524 001405          BEQ      LPST7          ;RCMR CLEARED
402 001526 012767 000007 177326      MOV      #7,ERCOUNT   ;***** ERROR 7 *****
403 001534 004567 007010          JSR      %5,STAER     ;REPORT ERROR
404 001540 000004          LPST7: SCOPE
405 001542 001506          STAI7
406
407          ;CAN WE SET THE FUNCTION BITS IN THE RCCS REG.
408          ;BITS 7,6,5,4,3,2&1
409
410 001544 012777 000340 177230      STAI10: MOV      #340,@PS      ;LOCK OUT INTERRUPTS
411 001552 012777 000176 177236      MOV      #176,@RCCS     ;SET DISK FUNCTION BITS
412 001560 017767 177232 177304      MOV      @RCCS,WORK    ;FETCH FUNCTION BITS
413 001566 022767 000376 177276      CMP      #376,WORK     ;ARE THE FUNCTION BITS SET
414 001574 001410          BEQ      LPST10       ;FUNCTION BITS SET
415 001576 012767 000376 177270      MOV      #376,WORK1    ;SET UP FOR ERROR REPORT
416 001604 012767 000010 177250      ERR10: MOV      #10,ERCOUNT ;***** ERROR 10 *****
417 001612 004567 006770          JSR      %5,STAER1    ;REPORT ERROR (ERROR IN FUNCTION BITS)
418 001616 000004          LPST10: SCOPE
419 001620 001544          STAI10
420
421          ;WILL DISK CLEAR CLEAR THE FUNCTION BITS
422
423 001622 012777 000340 177152      STAI11: MOV      #340,@PS      ;LOCK OUT INTERRUPTS
424 001630 012777 000176 177160      MOV      #176,@RCCS     ;SET DISK FUNCTION BITS
425 001636 005077 177154          CLR      @RCCS
426 001642 017767 177150 177222      MOV      @RCCS,WORK    ;FETCH CONTROL REG
427 001650 022767 000200 177214      CMP      #200,WORK     ;IS ONLY READY SET
428 001656 001405          BEQ      LPST11       ;REGISTER CLEARED
429 001660 012767 000011 177174      ERR11: MOV      #11,ERCOUNT ;***** ERROR 11 *****
430 001666 004567 006656          JSR      %5,STAER     ;REPORT ERROR
431 001672 000004          LPST11: SCOPE
432 001674 001622          STAI11

```

```

433                                     ;CAN WE SET ALL THE RCBA BITS
434
435 001676 012767 177776 177170 STAI12: MOV      #177776,WORK1 ;SET UP CURRENT ADDR. OF ALL ONES
436 001704 016777 177164 177110      MOV      WORK1,@RCBA ;LOAD RCBA
437 001712 017767 177104 177152      MOV      @RCBA,WORK ;FETCH RCBA
438 001720 026767 177150 177144      CMP      WORK1,WORK ;COMPARE FOR ALL BITS SET
439 001726 001405                BEQ      LPST12 ;ALL BITS SET?
440 001730 012767 000012 177124 ERR12: MOV      #12,ERCOUNT ;***** ERROR 12 *****
441 001736 004567 006644                JSR      %5,STAER1 ;REPORT ERROR
442 001742 000004                LPST12: SCOPE ;ENTER SCOPE LOOP
443 001744 001676                STAI12
444
445                                     ;WILL DISK CLEAR - CLEAR THE RCBA REGISTER
446
447 001746 012777 177777 177046 STAI13: MOV      #177777,@RCBA ;SET RCBA EQUAL TO ALL ONES
448 001754 005077 177042                CLR      @RCBA
449 001760 005777 177036                TST      @RCBA ;TEST FOR BIT0 SET IN RCBA (READ ONLY BIT)
450 001764 001410                BEQ      LPST13 ;WAS THE REST OF RCBA CLEARED?
451 001766 017767 177024 177076      MOV      @RCBS,WORK ;NO! FETCH RCBA REG.
452 001774 012767 000013 177060 ERR13: MOV      #13,ERCOUNT ;***** ERROR 13 *****
453 002002 004567 006542                JSR      %5,STAER ;REPORT ERROR
454 002006 000004                LPST13: SCOPE ;ENTER SCOPE LOOP
455 002010 001746                STAI13
456
457                                     ;CAN WE SET ALL BITS IN RCWC REGISTER
458
459 002012 012767 177777 177054 STAI14: MOV      #177777,WORK1 ;SET UP MAX. WORD COUNT
460 002020 016777 177050 176772      MOV      WORK1,@RCWC ;LOAD RCWC REGISTER
461 002026 026777 177042 176764      CMP      WORK1,@RCWC ;ARE ALL BITS SET
462 002034 001410                BEQ      LPST14 ;YES! EXIT
463 002036 017767 176756 177026      MOV      @RCWC,WORK ;NO! FETCH RCWC REG.
464 002044 012767 000014 177010 ERR14: MOV      #14,ERCOUNT ;***** ERROR 14 *****
465 002052 004567 006530                JSR      %5,STAER1 ;REPORT ERROR
466 002056 000004                LPST14: SCOPE ;ENTER SCOPE LOOP
467 002060 002012                STAI14

```

```

468                                     ;WILL DISK CLEAR-CLEAR THE WORD COUNT REGISTER
469
470 002062 012777 177777 176730 STAI15: MOV      #177777,@RCWC ;SET RCWC REGISTER EQUAL TO ALL ONES
471 002070 005077 176724          CLR      @RCWC
472 002074 017767 176720 176770      MOV      @RCWC,WORK ;FETCH WORD COUNT REGISTER
473 002102 001405          BEQ      LPST15 ;YES! EXIT
474 002104 012767 000015 176750 ERR15: MOV      #15,ERCOUNT ;***** ERROR 15 *****
475 002112 004567 006432          JSR      %5,STAER ;REPORT ERROR
476 002116 000004          LPST15: SCOPE ;ENTER SCOPE LOOP
477 002120 002062          STAI15
478
479                                     ;CAN WE SET ALL THE BITS IN THE RCDA REGISTER.
480
481 002122 012767 007777 176744 STAI16: MOV      #7777,WORK1 ;17777 WITH 4 DSKS
482 002130 016777 176740 176666      MOV      WORK1,@RCDA ;SET RCDA TO ALL ONES
483 002136 017767 176662 176726      MOV      @RCDA,WORK ;FETCH RCDA REGISTER
484 002144 026767 176724 176720      CMP      WORK1,WORK ;ARE ALL BITS SET
485 002152 001405          BEQ      LPST16 ;YES! EXIT
486 002154 012767 000016 176700 ERR16: MOV      #16,ERCOUNT ;***** ERROR 16 *****
487 002162 004567 006420          JSR      %5,STAER1 ;NOT ALL BITS SET REPORT ERROR
488 002166 000004          LPST16: SCOPE ;ENTER SCOPE LOOP
489 002170 002122          STAI16
490
491                                     ;CAN WE CLEAR THE RCDA REG. WITH DISK CLEAR.
492
493 002172 012777 177777 176624 STAI17: MOV      #177777,@RCDA ;SET RCDA TO ALL ONES
494 002200 005077 176620          CLR      @RCDA
495 002204 005777 176614          TST      @RCDA ;TEST FOR ZERO RCDA
496 002210 001410          BEQ      LPST17 ;YES EXIT
497 002212 017767 176606 176652      MOV      @RCDA,WORK ;NO BITS SET IN RCDA
498 002220 012767 000017 176634 ERR17: MOV      #17,ERCOUNT ;***** ERROR 17 *****
499 002226 004567 006316          JSR      %5,STAER ;REPORT ERROR
500 002232 000004          LPST17: SCOPE ;ENTER SCOPE LOOP
501 002234 002172          STAI17

```

```

502      ; DO ONE WORD WRITE FOLLOWED BY ONE WORD WRITE CONTINUE
503      ; EROR1=BUSY FAILED TO SET
504      ; EROR2=BUSY FAILED TO CLEAR
505      ; EROR3=CONTROL ERROR WHEN X-FERING DATA
506      ; EROR4=RCER INCREMENTED WHEN X-FERING DATA
507      ; EROR5=RCDA FAILED TO INCREMENT WHEN X-FERING DATA
508      ; EROR6=WORD COUNT FAILED TO OVERFLOW
509      ; EROR7=RCBA FAILED TO INCREMENT
510
511      ; * * * EXECUTE THE ONE WORD WRITE * * *
512
513      002236 005077 176562      STAI22: CLR      @RCDA      ; SET DSK ADDRESS TO 0
514      002242 012767 177777 007404      MOV      #177777,OUTBUF ; DATA TO BE X-FERED
515      002250 012777 011654 176544      MOV      @OUTBUF,@RCBA ; SET UP CURRENT ADDRESS
516      002256 012777 177777 176534      MOV      #-1,@RCWC      ; SET WORD COUNT TO -1
517      002264 012777 000003 176524      MOV      #3,@RCCS      ; GO WRITE
518      002272 105777 176520      TSTB    @RCCS      ; TEST FOR RDY=0
519      002276 100011      BPL     STAI23      ; RDY=0
520      002300 017767 176512 176564      MOV      @RCCS,WORK    ; BUSY NOT SET! FETCH RCCS
521      002306
522      002306 012767 000020 176546      EROR1:  MOV      #20,ERCOUNT ; ***** ERROR 20 *****
523      002314 004567 006230      ERR20:  JSR      %5,STAER1 ; REPORT ERROR
524      002320 000746      BR      STAI22      ; RESTART TEST
525      002322 005067 176544      STAI23: CLR      WORK
526      002326 005267 176540      INCWAT: INC      WORK      ; WAIT FOR BUSY=0
527      002332 105777 176460      TSTB    @RCCS      ; IS BUSY CLEARED
528      002336 100417      BMI     STAI24      ; FLAG CLEARED
529      002340 005767 176526      TST     WORK      ; HAVE WE WAITED LONG ENOUGH
530      002344 001370      BNE     INCWAT      ; RDY FAILED TO SET
531      002346 017767 176444 176516      MOV      @RCCS,WORK    ; FETCH CONTENTS OF RCCS REG
532      002354 052767 000002 176512      BIS     #2,WORK1     ; WANT RCCS SHOULD CONTAIN
533      002362
534      002362 012767 000021 176472      EROR2:  MOV      #21,ERCOUNT ; ***** ERROR 21 *****
535      002370 004567 006212      ERR21:  JSR      %5,STAER1 ; REPORT ERROR
536      002374 000720      BR      STAI22      ; RESTART TEST
537      002376 017767 176414 176466      STAI24: MOV      @RCCS,WORK ; FETCH CONTENTS OF RCCS REG
538      002404 005767 176462      TST     WORK      ; IS ERROR FLAG SET
539      002410 100012      BPL     STAI26      ; NO! X-FER OK
540      002412 012767 000202 176454      MOV      #202,WORK1   ; WHAT RCCS SHOULD CONTAIN
541      002420
542      002420 012767 000022 176434      EROR3:  MOV      #22,ERCOUNT ; ***** ERROR 22 *****
543      002426 004567 006154      ERR22:  JSR      %5,STAER1 ; REPORT ERROR
544      002432 000167 177600      JMP     STAI22      ; RESTART TEST

```

```

545 002436 017767 176362 176426 STAI26: MOV @RCDA,WORK ; WAS RCDA INCREMENTED BY 1
546 002444 022767 000001 176420 CMP #BIT0,WORK ; IS RCDA CORRECT
547 002452 001412 BEQ STAI27 ; RCDA OK
548 002454 012767 000001 176412 MOV #BIT0,WORK1 ; WHAT RCDA SHOULD CONTAIN
549 002462
550 002462 012767 000023 176372 EROR5: MOV #23,ERCOUNT ; ***** ERROR 23 *****
ERR23: JSR %5,STAER1 ; REPORT ERROR
551 002470 004567 006112 JMP STAI22 ; RESTART TEST
552 002474 000167 177536
553 002500 017767 176314 176364 STAI27: MOV @RCWC,WORK ; FETCH WORD COUNT
554 002506 001407 BEQ STAI30 ; WORD COUNT DID OVERFLOW
555 002510
556 002510 012767 000024 176344 EROR6: MOV #24,ERCOUNT ; ***** ERROR 24 *****
ERR24: JSR %5,STAER1 ; WORD COUNT FAILED TO OVERFLOW
557 002516 004567 006026 JMP STAI22 ; RESTART TEST
558 002522 000167 177510
559 002526 017767 176270 176336 STAI30: MOV @RCBA,WORK ; FETCH CURRENT ADDRESS
560 002534 012767 011656 176332 MOV #OUTBUF+2,WORK1 ; WHAT RCBA SHOULD EQUAL
561 002542 026767 176326 176322 CMP WORK1,WORK ; IS RCBA CORRECT
562 002550 001407 BEQ STAI31 ; YES EXECUTE CONTINUE
563 002552
564 002552 012767 000025 176302 EROR7: MOV #25,ERCOUNT ; ***** ERROR 25 *****
ERR25: JSR %5,STAER1 ; REPORT ERROR IN RCBA
565 002560 004567 006022 JMP STAI22 ; ERROR RESTART TEST
566 002564 000167 177446

```

567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604

002570 012777 177777 176222
002576 052777 000001 176212
002604 105777 176206
002610 100012
002612 017767 176200 176252
002620
002620 012767 000026 176234
002626 004567 005716
002632 000167 177400
002636 005067 176230
002642 105777 176150
002646 100415
002650 005267 176216
002654 001372
002656 017767 176134 176206
002664
002664 012767 000027 176170
002672 004567 005652
002676 000167 177334
002702 005777 176110
002706 100015
002710 017767 176102 176154
002716 017767 176104 176150
002724
002724 012767 000030 176130
002732 004567 005650
002736 000167 177274

: TEST WRITE: A WRITE CONTINUE WILL BE EXECUTED NOW
: WORD COUNT WILL BE SET TO -1 AGAIN

: EROR10=BUSY NOT SET BY GO
: EROR11=BUSY NOT CLEARED BY OVERFLOW
: EROR12=DISK ERROR OCCURED WHILE X-FERING
: EROR13=RCER INCREMENTED WHEN CONTINUE WAS EXECUTED
: EROR14=RCDA DID NOT EQUAL 2 AFTER SECOND X-FER
: EROR15=RCWC DID NOT OVERFLOW AT THE END OF X-FER
: EROR16=RCBA DID NOT EQUAL OUTBUF+2 AT END OF X-FER

```

STAI31: MOV      #177777, @RCWC      ; SET RCWC TO -1
        BIS      @BIT0, @RCCS      ; SET GO TO CONTINUE
        TSTB    @RCCS              ; TEST FOR RDY SET
        BPL     STAI32             ; BUSY WAS SET BY GO
        MOV     @RCCS, WORK        ; FETCH CONTENTS OF RCCS

EROR10:
ERR26:  MOV     #26, @ERCOUNT      ; ***** ERROR 26 *****
        JSR    %5, @STAER         ; REPORT BUSY NOT SET
        JMP    STAI22

STAI32: CLR     WORK
INCBUSY: TSTB  @RCCS              ; TST FOR RDY SET BY OVERFLO
        BMI    STAI33             ; READY SET CONTINUE
        INC    WORK               ; WAIT FOR RDY=1
        BNE    INCBUSY           ; GO WAIT FOR RDY
        MOV    @RCCS, WORK        ; FETCH CONTENTS OF RCCS

EROR11:
ERR27:  MOV     #27, @ERCOUNT      ; ***** ERROR 27 *****
        JSR    %5, @STAER         ; REPORT BUSY NOT CLEARED
        JMP    STAI22            ; RESTART ROUTINE

STAI33: TST     @RCCS              ; DID AN ERROR OCCUR WHILE X-FERING
        BPL    STAI35             ; NO CONTINUE
        MOV    @RCCS, WORK        ; YES! CONTENTS OF RCCS
        MOV    @RCER, WORK1       ; EXT ERROR BITS

EROR12:
ERR30:  MOV     #30, @ERCOUNT      ; ***** ERROR 30 *****
        JSR    %5, @STAER1        ; REPORT ERROR OCCURRED.
        JMP    STAI22            ; RESTART ROUTINE
    
```

605	002742	017767	176056	176122	STAI35:	MOV	RCDA,WORK	;DID RCDA INCREMENT ON CONTINUE
606	002750	012767	000002	176116		MOV	#2,WORK1	;WHAT RCDA SHOULD CONTAIN
607	002756	026767	176112	176106		CMP	WORK1,WORK	;IS RCDA CORRECT
608	002764	001407				BEQ	STAI36	;RCDA OK
609	002766				EROR14:			
610	002766	012767	000031	176066	ERR31:	MOV	#31,ERCOUNT	; ***** ERROR 31 *****
611	002774	004567	005606			JSR	%5,STAER1	;REPORT RCDA INCORRECT
612	003000	000167	177232			JMP	STAI22	;RE-START ROUTINE
613	003004	017767	176010	176060	STAI36:	MOV	RCWC,WORK	;FETCH WORD COUNT
614	003012	001407				BEQ	STAI37	;WORD COUNT OVERFLOWED
615	003014				EROR15:			
616	003014	012767	000032	176040	ERR32:	MOV	#32,ERCOUNT	; ***** ERROR 32 *****
617	003022	004567	005522			JSR	%5,STAER	;REPORT WORD COUNT FAILED TO CLEAR
618	003026	000167	177204			JMP	STAI22	;RESTART ROUTINE
619	003032	017767	175764	176032	STAI37:	MOV	RCBA,WORK	;FETCH RCBA
620	003040	012767	011660	176026		MOV	#OUTBUF+4,WORK1	;WHAT RCBA SHOULD EQUAL
621	003046	026767	176022	176016		CMP	WORK1,WORK	;IS RCBA CORRECT
622	003054	001407				BEQ	LPST22	;RCBA WAS CORRECT
623	003056				EROR16:			
624	003056	012767	000033	175776	ERR33:	MOV	#33,ERCOUNT	; ***** ERROR 33 *****
625	003064	004567	005516			JSR	%5,STAER1	;REPORT RCBA INCORRECT
626	003070	000167	177142			JMP	STAI22	;RESTART ROUTINE
627	003074	000004			LPST22:	SCOPE		;ENTER SCOPE LOOP
628	003076	002236				STAI22		

629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675

003100 005077 175720
003104 012767 177777 006542
003112 012777 011654 175702
003120 012777 177777 175672
003126 052777 000007 175662
003134 105777 175656
003140 100011
003142 017767 175650 175722
003150
003150 012767 000034 175704
003156 004567 005366
003162 000746
003164 005067 175702
003170 005267 175676
003174 105777 175616
003200 100417
003202 005767 175664
003206 001370
003210 017767 175602 175654
003216 052767 000006 175650
003224
003224 012767 000035 175630
003232 004567 005350
003236 000720
003240 017767 175552 175624
003246 100016
003250 026727 175616 102206
003256 001012
003260 012767 000206 175606
003266
003266 012767 000036 175566
003274 004567 005306
003300 000167 177574

```

: TEST WRITE CHECK
: DO A ONE WORD WRITE CHECK FOLLOWED BY
: A ONE WORD WRITE CHECK CONTINUE

: EROR17=BUSY FAILED TO SET
: EROR20=BUSY FAILED TO CLEAR
: EROR21=DISK ERROR WHEN X-FERING DATA
: EROR22=RCER INCREMENTED WHEN X-FERING DATA
: EROR23=RCDA FAILED TO INCREMENT WHEN X-FERING DATA
: EROR24=WORD COUNT FAILED TO OVERFLOW
: EROR25=RCBA FAILED TO INCREMENT

: * * *EXECUTE THE ONE WORD WRITE CHECK* * *

STAI40: CLR      RCDA           ;SET DSK ADDRESS TO 0
        MOV      #177777,OUTBUF ;DATA TO BE X-FERED
        MOV      #OUTBUF,RCBA   ;SET UP CURRENT ADDRESS
        MOV      #-1,RCMC       ;SET WORD COUNT TO -1
        BIS      #7,RCCS        ;GO WRITE CHECK
        TSTB     RCSS           ;TEST FOR READY
        BPL      STAI41         ;NOT READY
        MOV      RCSS,WORK      ;SET! FETCH RCCS

EROR17:
ERR34:  MOV      #34,ERCOUNT     ; ***** ERROR 34 *****
        JSR      %5,STAER        ;REPORT ERROR
        BR       STAI40         ;RESTART TEST

STAI41: CLR      WORK
WCWAT:  INC      WORK           ;WAIT FOR READY
        TSTB     RCSS
        BMI      STAI42         ;FLAG SET
        TST      WORK           ;HAVE WE WAITED LONG ENOUGH
        BNE     WCWAT          ;READY FAILED TO SET
        MOV      RCSS,WORK      ;FETCH CONTENTS OF RCCS REG
        BIS      #6,WORK1       ;WANT RCCS SHOULD CONTAIN

EROR20:
ERR35:  MOV      #35,ERCOUNT     ; ***** ERROR 35 *****
        JSR      %5,STAER1       ;REPORT ERROR
        BR       STAI40         ;RESTART TEST

STAI42: MOV      RCSS,WORK      ;FETCH CONTENTS OF RCCS REG
        BPL      STAI44         ;NO! X-FER OK
        CMP      WORK,#102206   ;DID RCWCE OCCUR
        MOV      #206,WORK1     ;WHAT RCCS SHOULD CONTAIN

EROR21:
ERR36:  MOV      #36,ERCOUNT     ; ***** ERROR 36 *****
        JSR      %5,STAER1       ;REPORT ERROR
        JMP      STAI40         ;RESTART TEST

```

MAINDEC-11-DZRC-A
DZRCAB.P11

RC11 STATIC TEST

REPLACES DSJA-PB1

MACY11 27(732) 10-SEP-76 11:53 PAGE 20

676	003304	017767	175514	175560	STAI44:	MOV	ARCDA,WORK	; WAS RCDA INCREMENTED BY 1?
677	003312	022767	000001	175552		CMP	#BIT0,WORK	; IS RCDA CORRECT
678	003320	001412				BEQ	STAI45	; RCDA OK
679	003322	012767	000001	175544		MOV	#BIT0,WORK1	; WHAT RCDA SHOULD CONTAIN
680	003330				EROR23:			
681	003330	012767	000037	175524	ERR37:	MOV	#37,ERCOUNT	; ***** ERROR 37 *****
682	003336	004567	005244			JSR	%5,STAER1	; REPORT ERROR
683	003342	000167	177532			JMP	STAI40	; RESTART TEST
684	003346	017767	175446	175516	STAI45:	MOV	ARCWC,WORK	; FETCH WORD COUNT
685	003354	001407				BEQ	STAI46	; WORD COUNT DID OVERFLOW
686	003356				EROR24:			
687	003356	012767	000040	175476	ERR40:	MOV	#40,ERCOUNT	; ***** ERROR 40 *****
688	003364	004567	005160			JSR	%5,STAER	; WORD COUNT FAILED TO OVERFLOW
689	003370	000167	177504			JMP	STAI40	; RESTART TEST
690	003374	017767	175422	175470	STAI46:	MOV	ARCBA,WORK	; FETCH CURRENT ADDRESS
691	003402	012767	011656	175464		MOV	#OUTBUF+2,WORK1	; WHAT RCBA SHOULD EQUAL
692	003410	026767	175460	175454		CMP	WORK1,WORK	; IS RCBA CORRECT
693	003416	001407				BEQ	STAI47	; YES EXECUTE CONTINUE
694	003420				EROR25:			
695	003420	012767	000041	175434	ERR41:	MOV	#41,ERCOUNT	; ***** ERROR 41 *****
696	003426	004567	005154			JSR	%5,STAER1	; REPORT ERROR IN RCBA
697	003432	000167	177442			JMP	STAI40	; ERROR RESTART TEST

```

698 ;A WRITE CHECK CONTINUE WILL BE EXECUTED NOW;WORD COUNT WILL BE SET TO -1 AGAIN
699
700 ;EROR26=BUSY WAS NOT SET BY GO
701 ;EROR27=BUSY NOT CLEARED BY OVERFLOW
702 ;EROR30=DISK ERROR OCCURRED WHILE X-FERING
703 ;EROR32=RCDA DID NOT EQUAL 2 AFTER SECOND X-FER
704 ;EROR33=RCWC DID NOT=0 AFTER X-FER
705 ;EROR34=RCBA DID NOT=OUTBUF+2 AT END OF X-FER
706
707 003436 012777 177777 175354 STAI47: MOV #177777,RCWC ;SET RCWC TO -1
708 003444 052777 000001 175344 BIS #BIT0,RCCS ;SET GO TO CONTINUE
709 003452 105777 175340 TSTB RCSS ;TEST FOR READY
710 003456 100012 BPL STAI50 ;
711 003460 017767 175332 175404 MOV RCSS,WORK ;FETCH CONTENTS OF RCCS
712 003466
713 003466 012767 000042 175366 EROR26: MOV #42,ERCOUNT ; ***** ERROR 42 *****
714 003474 004567 005050 ERR42: JSR %S,STAER ;REPORT BUSY NOT SET
715 003500 000167 177374 JMP STAI40
716 003504 005067 175362 STAI50: CLR WORK
717 003510 105777 175302 WCBUSI: TSTB RCSS ;TST FOR RDY SET BY OVERFLOW
718 003514 100415 BMI STAI51 ;BRANCH IF READY SET
719 003516 005267 175350 INC WORK ;WAIT FOR BUSY=0
720 003522 001372 BNE WCBUSI ;GO WAIT FOR RDY
721 003524 017767 175266 175340 MOV RCSS,WORK ;FETCH CONTENTS OF RCCS
722 003532
723 003532 012767 000043 175322 EROR27: MOV #43,ERCOUNT ; ***** ERROR 43 *****
724 003540 004567 005004 ERR43: JSR %S,STAER ;REPORT RDY NOT SET
725 003544 000167 177330 JMP STAI40 ;RESTART ROUTINE
726 003550 005777 175242 STAI51: TST RCSS ;DID AN ERROR OCCUR WHILE X-FERING
727 003554 100021 BPL STAI53 ;NO CONTINUE
728 003556 017767 175234 175306 MOV RCSS,WORK ;YES! CONTENTS OF RCCS
729 003564 026727 175302 102206 CMP WORK,#102206
730 003572 001012 BNE STAI53
731 003574 012767 000206 175272 MOV #206,WORK1
732 003602
733 003602 012767 000044 175252 EROR30: MOV #44,ERCOUNT ; ***** ERROR 44 *****
734 003610 004567 004772 ERR44: JSR %S,STAER1 ;REPORT ERROR OCCURRED
735 003614 000167 177260 JMP STAI40 ;RESTART ROUTINE

```

736	003620	017767	175200	175244	STAI53:	MOV	@RCDA,WORK	:	DID RCDA INCREMENT ON CONTINUE
737	003626	012767	000002	175240		MOV	#2,WORK1	:	WHAT RCDA SHOULD CONTAIN
738	003634	026767	175234	175230		CMP	WORK1,WORK	:	IS RCDA CORRECT
739	003642	001407				BEQ	STAI54	:	RCDA OK
740	003644				EROR32:				
741	003644	012767	000045	175210	ERR45:	MOV	#45,ERCOUNT	:	***** ERROR 45 *****
742	003652	004567	004730			JSR	%5,STAER1	:	REPORT RCDA INCORRECT
743	003656	000167	177216			JMP	STAI40	:	RE-START ROUTINE
744	003662	017767	175132	175202	STAI54:	MOV	@RCWC,WORK	:	FETCH WORD COUNT
745	003670	001407				BEQ	STAI55	:	WORD COUNT OVERFLOWED
746	003672				EROR33:				
747	003672	012767	000046	175162	ERR46:	MOV	#46,ERCOUNT	:	***** ERROR 46 *****
748	003700	004567	004644			JSR	%5,STAER	:	REPORT WORD COUNT FAILED TO CLEAR
749	003704	000167	177170			JMP	STAI40	:	RESTART ROUTINE
750	003710	017767	175106	175154	STAI55:	MOV	@RCBA,WORK	:	FETCH RCBA
751	003716	012767	011660	175150		MOV	#OUTBUF+4,WORK1	:	WHAT RCBA SHOULD EQUAL
752	003724	026767	175144	175140		CMP	WORK1,WORK	:	IS RCBA CORRECT
753	003732	001407				BEQ	LPST40	:	RCBA WAS CORRECT
754	003734				EROR34:				
755	003734	012767	000047	175120	ERR47:	MOV	#47,ERCOUNT	:	***** ERROR 47 *****
756	003742	004567	004640			JSR	%5,STAER1	:	REPORT RCBA INCORRECT
757	003746	000167	177126			JMP	STAI40	:	RESTART ROUTINE
758	003752	000004			LPST40:	SCOPE		:	ENTER SCOPE LOOP
759	003754	003100				STAI40			

```

760          ; TEST READ
761          ; ONE WORD READ FOLLOWED BY A ONE WORD
762          ; READ CONTINUE.
763
764          ; EROR35=BUSY WAS NOT SET BY GO
765          ; EROR36=BUSY NOT CLEARED BY OVERFLOW
766          ; EROR37=DISK ERROR OCCURRED WHILE X-FERING DATA
767          ; EROR41=RCDA FAILED TO INCREMENT WHEN X-FERING DATA
768          ; EROR42=WORD COUNT FAILED TO OVERFLOW
769          ; EROR43=RCBA FAILED TO INCREMENT
770
771          ; * * *EXECUTE THE ONE WORD READ* * *
772
773 003756 005077 175042 STAI56: CLR      @RCDA      ; SET DSK ADDRESS TO 0
774 003762 012767 177777 005664      MOV      #177777,OUTBUF ; DATA TO BE X-FERED
775 003770 012777 011654 175024      MOV      @OUTBUF,@RCBA ; SET UP CURRENT ADDRESS
776 003776 012777 177777 175014      MOV      #-1,@RCWC     ; SET WORD COUNT TO -1
777 004004 012777 000005 175004      MOV      #5,@RCCS     ; GO READ
778 004012 105777 175000      TSTB   @RCCS         ; TEST FOR BUSY=1
779 004016 100011      BPL     STAI57       ; BUSY SET
780 004020 017767 174772 175044      MOV      @RCCS,WORK   ; BUSY NOT SET! FETCH RCCS
781 004026
782 004026 012767 000050 175026 EROR35: MOV      #50,ERCOUNT ; ***** ERROR 50 *****
783 004034 004567 004510 ERR50: JSR      %5,STAER1 ; REPORT ERROR
784 004040 000746      BR       STAI56      ; RESTART TEST
785 004042 005067 175024 STAI57: CLR      WORK
786 004046 005267 175020 RDWAT: INC      WORK
787 004052 105777 174740      TSTB   @RCCS         ; WAIT FOR BUSY=0
788 004056 100417      BMI     STAI60      ; IS BUSY CLEARED
789 004060 005767 175006      TST    WORK          ; FLAG CLEARED
790 004064 001370      BNE     RDWAT        ; HAVE WE WAITED LONG ENOUGH
791 004066 017767 174724 174776      MOV      @RCCS,WORK   ; BUSY FAILED TO CLEAR
792 004074 052767 000005 174772      BIS     #5,WORK1     ; FETCH CONTENTS OF RCCS REG
793 004102 EROR36: BIS     #5,WORK1     ; WANT RCCS SHOULD CONTAIN
794 004102 012767 000051 174752 ERR51: MOV      #51,ERCOUNT ; ***** ERROR 51 *****
795 004110 004567 004472      JSR      %5,STAER1 ; REPORT ERROR
796 004114 000720      BR       STAI56      ; RESTART TEST
797 004116 017767 174674 174746 STAI60: MOV      @RCCS,WORK ; FETCH CONTENTS OF RCCS REG
798 004124 005767 174742      TST    WORK          ; IS ERROR FLAG SET
799 004130 100012      BPL     STAI62      ; NO! X-FER OK
800 004132 012767 000204 174734      MOV      #204,WORK1  ; WHAT RCCS SHOULD CONTAIN
801 004140 EROR37:
802 004140 012767 000052 174714 ERR52: MOV      #52,ERCOUNT ; ***** ERROR 52 *****
803 004146 004567 004434      JSR      %5,STAER1 ; REPORT ERROR
804 004152 000167 177600      JMP     STAI56      ; RESTART TEST

```

805	004156	017767	174642	174706	STAI62:	MOV	ARCDA,WORK	; WAS RCDA INCREMENTED BY 1
806	004164	022767	000001	174700		CMP	#BIT0,WORK	; IS RCDA CORRECT
807	004172	001412				BEQ	STAI63	; RCDA OK
808	004174	012767	000001	174672		MOV	#BIT0,WORK1	; WHAT RCDA SHOULD CONTAIN
809	004202				EROR41:			
810	004202	012767	000053	174652	ERR53:	MOV	#53,ERCOUNT	; ***** ERROR 53 *****
811	004210	004567	004372			JSR	%5,STAER1	; REPORT ERROR
812	004214	000167	177536			JMP	STAI56	; RESTART TEST
813	004220	017767	174574	174644	STAI63:	MOV	ARCWC,WORK	; FETCH WORD COUNT
814	004226	001407				BEQ	STAI64	; WORD COUNT DID OVERFLOW
815	004230				EROR42:			
816	004230	012767	000054	174624	ERR54:	MOV	#54,ERCOUNT	; ***** ERROR 54 *****
817	004236	004567	004306			JSR	%5,STAER	; WORD COUNT FAILED TO OVERFLOW
818	004242	000167	177510			JMP	STAI56	; RESTART TEST
819	004246	017767	174550	174616	STAI64:	MOV	ARCBA,WORK	; FETCH CURRENT ADDRESS
820	004254	012767	011656	174612		MOV	#OUTBUF+2,WORK1	; WHAT RCBA SHOULD EQUAL
821	004262	026767	174606	174602		CMP	WORK1,WORK	; IS RCBA CORRECT
822	004270	001407				BEQ	STAI65	; YES EXECUTE CONTINUE
823	004272				EROR43:			
824	004272	012767	000055	174562	ERR55:	MOV	#55,ERCOUNT	; ***** ERROR 55 *****
825	004300	004567	004302			JSR	%5,STAER1	; REPORT ERROR IN RCBA
826	004304	000167	177446			JMP	STAI56	; ERROR RESTART TEST

```

827 ;A READ CONTINUE WILL BE EXECUTED NOW;WORD COUNT WILL BE SET TO -1 AGAIN
828
829 ;EROR44=BUSY NOT SET BY GO
830 ;EBOR45=BUSY NOT CLEARED BY OVERFLOW
831 ;EROR46=DISK ERROR OCCURRED WHILE X-FERING
832 ;EROR50=RCDA DID NOT EQUAL 2 AFTER SECOND X-FER
833 ;EROR51=RCWC FAILED TO OVERFLOW ON READ CONTINUE
834 ;EROR52=RCBA DID NOT EQUAL OUTBUF+2 AFTER READ CONTINUE
835
836 004310 012777 177777 174502 STAI65: MOV #177777,@RCWC ;SET RCWC TO -1
837 004316 052777 000001 174472 BIS #BIT0,@RCCS ;SET GO TO CONTINUE
838 004324 105777 174466 TSTB @RCCS ;TEST FOR RDY=0
839 004330 100012 BPL STAI66 ;RDY SET
840 004332 017767 174460 174532 MOV @RCCS,WORK ;FETCH CONTENTS OF RCCS
841 004340
842 004340 012767 000056 174514 EROR44: MOV #56,ERCOUNT ; ***** ERROR 56 *****
843 004346 004567 004176 ERR56: JSR %5,STAER ;REPORT BUSY NOT SET
844 004352 000167 177400 JMP STAI56
845 004356 005067 174510 STAI66: CLR WORK
846 004362 105777 174430 INCRD: TSTB @RCCS ;TST FOR RDY SET BY OVERFLOW
847 004366 100415 BMI STAI67 ;RDY SET CONTINUE
848 004370 005267 174476 INC WORK ;WAIT FOR BUSY=0
849 004374 001372 BNE INCRD ;GO WAIT FOR BUSY
850 004376 017767 174414 174466 MOV @RCCS,WORK ;FETCH CONTENTS OF RCCS
851 004404
852 004404 012767 000057 174450 EROR45: MOV #57,ERCOUNT ; ***** ERROR 57 *****
853 004412 004567 004132 ERR57: JSR %5,STAER ;REPORT BUSY NOT CLEARED
854 004416 000167 177334 JMP STAI56 ;RESTART ROUTINE
855 004422 005777 174370 STAI67: TST @RCCS ;DID AN ERROR OCCUR WHILE X-FERING
856 004426 100015 BPL STAI71 ;NO CONTINUE
857 004430 017767 174362 174434 MOV @RCCS,WORK ;YES! CONTENTS OF RCCS
858 004436 017767 174364 174430 MOV @RCER,WORK1 ;EXT ERROR BITS
859 004444
860 004444 012767 000060 174410 EROR46: MOV #60,ERCOUNT ; ***** ERROR 60 *****
861 004452 004567 004130 ERR60: JSR %5,STAER1 ;REPORT ERROR OCCURRED
862 004456 000167 175554 JMP STAI22 ;RESTART ROUTINE

```

863	004462	017767	174336	174402	STAI71:	MOV	RCDA,WORK	:DID RCDA INCREMENT ON CONTINUE
864	004470	012767	000002	174376		MOV	#2,WORK1	:WHAT RCDA SHOULD CONTAIN
865	004476	026767	174372	174366		CMP	WORK1,WORK	:IS RCDA CORRECT
866	004504	001407				BEQ	STAI72	:RCDA OK
867	004506				ERORS0:			
868	004506	012767	000061	174346	ERR61:	MOV	#61,ERCOUNT	: ***** ERROR 61 *****
869	004514	004567	004066			JSR	%5,STAER1	:REPORT RCDA INCORRECT
870	004520	000167	177232			JMP	STAI56	:RESTART ROUTINE
871	004524	017767	174270	174340	STAI72:	MOV	RCWC,WORK	:FETCH WORD COUNT
872	004532	001407				BEQ	STAI73	:WORD COUNT OVERFLOWED
873	004534				ERORS1:			
874	004534	012767	000062	174320	ERR62:	MOV	#62,ERCOUNT	: ***** ERROR 62 *****
875	004542	004567	004002			JSR	%5,STAER	:REPORT WORD COUNT FAILED TO CLEAR
876	004546	000167	177204			JMP	STAI56	:RESTART ROUTINE
877	004552	017767	174244	174312	STAI73:	MOV	RCBA,WORK	:FETCH RCBA
878	004560	012767	011660	174306		MOV	OUTBUF+4,WORK1	:WHAT RCBA SHOULD EQUAL
879	004566	026767	174302	174276		CMP	WORK1,WORK	:IS RCBA CORRECT
880	004574	001407				BEQ	LPST56	:RCBA WAS CORRECT
881	004576				ERORS2:			
882	004576	012767	000063	174256	ERR63:	MOV	#63,ERCOUNT	: ***** ERROR 63 *****
883	004604	004567	003776			JSR	%5,STAER1	:REPORT RCBA INCORRECT
884	004610	000167	177142			JMP	STAI56	:RESTART ROUTINE
885	004614	000004			LPST56:	SCOPE		:ENTER SCOPE LOOP
886	004616	003756				STAI56		

919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959

004752 012777 003777 174044
004760 012777 177777 174032
004766 012777 011654 174026
004774 052777 000003 174014
005002 105777 174010
005006 100011
005010 017767 174002 174054
005016 012767 000066 174036
005024 004567 003520
005030 000750
005032 005067 174034
005036 105777 173754
005042 100415
005044 005267 174022
005050 001372
005052 017767 173740 174012
005060 012767 000067 173774
005066 004567 003456
005072 000167 177654
005076 022777 004000 173720
005104 001412
005106 017767 173712 173756
005114 012767 000070 173740
005122 004567 003422
005126 000167 177620
005132 000004
005134 004752

: IN THIS ROUTINE THE ABILITY OF THE CONTROL
: TO INCREMENT THE DISK NO. IS CHECKED

: ROUTINE PARAMETERS AT START
: RCWC=-1
: RCBA=#OUTBUF
: RCCS=#3

: PARAMETERS AT THE END OF ROUTINE
: RCDA=0
: RCWC=0
: RCBA=OUTBUF+2
: RCCS=0

STAI77: MOV #3777, @RCDA
MOV #177777, @RCWC
MOV #OUTBUF, @RCBA
BIS #3, @RCCS
TSTB @RCCS
BPL STAI00
MOV @RCCS, WORK
ERR66: MOV #66, @RCOUNT
JSR %5, @TAER
BR STAI77
STAI00: CLR WORK
INXDSK: TSTB @RCCS
BMI STAI01
INC WORK
BNE INXDSK
MOV @RCCS, WORK
ERR67: MOV #67, @RCOUNT
JSR %5, @TAER
JMP STAI77
STAI01: CMP #4000, @RCDA
BEQ LPST77
MOV @RCDA, WORK
ERR70: MOV #70, @RCOUNT
JSR %5, @TAER
JMP STAI77
LPST77: SCOPE
STAI77

: RCDA EQUALS ALL ONES
: WORD COUNT SET FOR ONE WORD
: CURRENT ADDRESS SET UP
: GO WRITE
: IS RDY CLEARED
: FETCH CONTROL RUG
: ***** ERROR 66 *****
: REPORT BUSY NOT SET
: RESTART ROUTINE
: IS X-FER COMPLETE
: TEST READY SHOULD BE SET
: GO CHECK AGAIN
: FETCH CONTENTS OF RCCS
: ***** ERROR 67 *****
: REPORT X-FER NOT COMPLETE
: RESTART ROUTINE
: INCREMENTED INTO DSK 1?
: YES
: FETCH RCDA ITS NOT CLEAR
: ***** ERROR 70 *****
: REPORT RCDA NOT CLEAR.
: RESTART ROUTINE
: ENTER SCOPE LOOP

```

960      ; IN THIS ROUTINE THE PROGRAM WILL GENERATE A
961      ; NON-EXISTENT DISK ERROR THIS WILL BE DONE
962      ; BY ATTEMPTING A 100 WORD WRITE ON DISK 3
963      ; THE LAST TRACK AND THE LAST WORD ON THE
964      ; LAST TRACK
965
966      ; IF DISK 3 IS THERE WE WILL RECEIVE
967      ; A NON-EXISTENT DISK ERROR WITH OVERFLOW
968      ; SET
969
970      ; IF DISK 3 IS NOT THERE WE WILL
971      ; JUST RECEIVE A NON-EXISTENT DISK ERROR
972
973 005136 012777 037777 173660 STA103: MOV      #37777, @RCDA      ; SET RCDA=TO ALL ONES
974 005144 012777 177700 173646      MOV      #177700, @RCWC     ; WORD COUNT=-2
975 005152 012777 011654 173642      MOV      @OUTBUF, @RCBA    ; CURRENT ADDRESS=OUTBUF
976 005160 052777 000003 173630      BIS      #3, @RCCS        ; GO WRITE
977 005166 005067 173700      CLR      WORK
978 005172 105777 173620      WAFBUS: TSTB @RCCS      ; IS RDY SET
979 005176 100414      BMI      STA104          ; YES EXIT
980 005200 005267 173666      INC      WORK            ; NO BUSY STILL SET
981 005204 001372      BNE      WAFBUS          ; GO WAIT FOR BUSY=0
982 005206 017767 173604 173656      MOV      @RCCS, WORK     ; FETCH CONTENTS OF RCCS
983 005214 012767 000071 173640 ERR71: MOV      #71, @RCOUNT    ; ***** ERROR 71 *****
984 005222 004567 003322      JSR      %5, @TAER       ; REPORT BUSY SET
985 005226 000743      BR       STA103          ; RESTART ROUTINE
986 005230 032777 004000 173560 STA104: BIT      #BIT11, @RCCS ; IS NCD SET
987 005236 001011      BNE      STA105          ; YES! IS ERROR SET
988 005240 017767 173552 173624      MOV      @RCCS, WORK     ; FETCH RCCS
989 005246 012767 000072 173606 ERR72: MOV      #72, @RCOUNT    ; ***** ERROR 72 *****
990 005254 004567 003270      JSR      %5, @TAER       ; REPORT NEED NOT SET
991 005260 000726      BR       STA103          ; RESTART ROUTINE
992 005262 005777 173530      STA105: TST      @RCCS    ; IS ERROR FLAG SET
993 005266 100411      BMI      LPSX103         ; ERROR IS SET
994 005270 017767 173522 173574      MOV      @RCCS, WORK     ; FETCH CONTENTS OF RCCS
995 005276 012767 000073 173556 ERR73: MOV      #73, @RCOUNT    ; ***** ERROR 73 *****
996 005304 004567 003240      JSR      %5, @TAER       ; REPORT ERROR NOT SET
997 005310 000712      BR       STA103          ; RESTART ROUTINE
998 005312 000004      LPSX103: SCOPE
999 005314 005136      STA103      ; ENTER SCOPE LOOP

```

```

1000 ;IN THIS TEST THE FEATURE OF CURRENT
1001 ;ADDRESS INHIBIT IS TESTED
1002
1003 ;DO A ONE WORD WRITE AND SEE
1004 ;IF RCBA INCREMENTED AFTER THE X-FER
1005
1006 005316 005077 173502 ST105X: CLR @RCDA ;SET DSK ADDRESS TO 0
1007 005322 012777 011654 173472 MOV @OUTBUF,@RCBA ;SET UP CURRENT ADDR
1008 005330 012777 177777 173462 MOV #-1,@RCWC ;SET WORD COUNT TO -1
1009 005336 012777 001000 173452 MOV @BIT9,@RCCS
1010 005344 105777 173446 TSTB @RCCS ;TEST FOR READY
1011 005350 100406 BMI LPX105 ;BRANCH IF READY SET
1012 005352 012767 000074 173502 ERR74: MOV #74,ERCOUNT ;***** ERROR 74 *****
1013 005360 004567 003164 JSR %5,STAER ;REPORT READY NOT SET
1014 005364 000754 BR ST105X ;TRY AGAIN
1015 005366 052777 000003 173422 LPX105: BIS #3,@RCCS ;GO WRITE
1016 005374 017767 173416 173470 MOV @RCCS,WORK
1017 005402 105767 173464 TSTB WORK ;TEST FOR NOT RDY
1018 005406 100006 BPL LPX106 ;BRANCH IF RDY=0
1019 005410 012767 000075 173444 ERR75: MOV #75,ERCOUNT ;***** ERROR 75 *****
1020 005416 004567 003126 JSR %5,STAER ;REPORT RDY NOT CLEARED BY CMD.
1021 005422 000735 BR ST105X ;LOOP ON ERROR
1022 005424 005067 173442 LPX106: CLR WORK
1023 005430 105777 173362 WATRDY: TSTB @RCCS
1024 005434 100411 BMI LPX107 ;BRANCH IF RDY SET
1025 005436 005267 173430 INC WORK ;WAIT FOR RDY
1026 005442 001372 BNE WATRDY
1027 005444 012767 000076 173410 ERR76: MOV #76,ERCOUNT ;***** ERROR 76 *****
1028 005452 004567 003072 JSR %5,STAER ;READY NEVER SET AFTER X-FER
1029 005456 000717 BR ST105X ;LOOP ON ERROR
1030 005460 012767 011654 173406 LPX107: MOV @OUTBUF,WORK1 ;WHAT RCBA SHOULD BE
1031 005466 017767 173330 173376 MOV @RCBA,WORK ;WHAT RCBA IS
1032 005474 026767 173372 173372 CMP WORK,WORK1 ;COMPARE
1033 005502 001406 BEQ LPX108 ;BRANCH IF EQUAL
1034 005504 012767 000077 173350 ERR77: MOV #77,ERCOUNT ;***** ERROR 77 *****
1035 005512 004567 003070 JSR %5,STAER1 ;REPORT THEY DID NOT CMP
1036 005516 000677 BR ST105X ;LOOP ON ERROR
1037 005520 000004 LPX108: SCOPE ;ENTER SCOPE LOOP
1038 005522 005316 ST105X

```

1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071

005524 005077 173274
005530 012777 000340 173244
005536 012777 177000 173254
005544 012777 177700 173250
005552 012777 000063 173236
005560 005067 173310
005564 017767 173226 173300
005572 005767 173274
005576 100422
005600 105767 173266
005604 100411
005606 005267 173262
005612 001364
005614 012767 000100 173240
005622 004567 002760
005626 000736
005630
005630 012767 000101 173224
005636 004567 002744
005642 000730
005644 017767 173156 173220
005652 032767 110000 173212
005660 001006
005662 012767 000102 173172
005670 004567 002712
005674 000713

; IN THIS ROUTINE THE ABILITY OF NON-EXISTENT
; MEMORY ERROR IS CHECK.

; TRANSFER TWO WORDS STARTING WITH THE
; LARGEST ADDRESSABLE MEMORY LOCATION IN THE
; PDP-11/20

NXMTSM: CLR @RCDA ; SET DSK ADDRESS TO 0
MOV #340,@PS ; LOCK UP PROCESSOR
MOV #-1000,@RCWC ; SET UP WORD COUNT
MOV #177700,@RCBA ; SET UP CURRENT ADDRESS
MOV #63,@RCCS ; ISSUE WRITE
CLR WORK1
INCNEM: MOV @RCCS,WORK
TST WORK ; TEST FOR ERROR
BMI TSTNEM ; BRANCH IF ERROR SET
TSTB WORK ; TEST FOR ERROR
BMI RDYERX ; REPORT NEM NOT SET
INC WORK1 ; WAIT FOR ERROR
BNE INCNEM
ERR100: MOV #100,ERCOUNT ; ***** ERROR 100 *****
JSR %5,STAER1 ; REPORT ERROR
BR NXMTSM ; LOOP ON ERROR
RDYERX: MOV #101,ERCOUNT ; ***** ERROR 101 *****
ERR101: JSR %5,STAER1 ; CONTROL NEVER COMPLETED WRITE
BR NXMTSM ; LOOP ON ERROR
TSTNEM: MOV @RCER,WORK
BIT #BIT15!BIT12,WORK
BNE CLRNEM ; BRANCH IF NEM ERROR SET
ERR102: MCV #102,ERCOUNT ; ***** ERROR 102 *****
JSR %5,STAER1 ; REPORT HWD NOT SET
BR NXMTSM ; LOOP ON ERROR

1072	005676	000005			CLRNM:	RESET		
1073	005700	022777	000200	173110		CMP	#200, @RCCS	: IS ERROR CLEARED
1074	005706	001412				BEQ	CKHWD	: ERROR FLAG CLEARED
1075	005710	012767	000103	173144	ERR103:	MOV	#103, ERCOUNT	: ***** ERROR 103 *****
1076	005716	017767	173074	173146		MOV	@RCCS, WORK	: ERROR NOT CLEARED BY DISK CLEAR
1077	005724	004567	002620			JSR	%5, STAER	: REPORT ERROR
1078	005730	000167	177570			JMP	NXMTSM	: LOOP ON ERROR
1079	005734	032777	040000	173054	CKHWD:	BIT	#BIT14, @RCCS	: IS HARD ERROR SET
1080	005742	001412				BEQ	CKNEM	: HWD CLEARED BY DISK CLEAR
1081	005744	012767	000104	173110	ERR104:	MOV	#104, ERCOUNT	: ***** ERROR 104 *****
1082	005752	017767	173040	173112		MOV	@RCCS, WORK	: REPORT HWD NOT CLEARED
1083	005760	004567	002564			JSR	%5, STAER	:
1084	005764	000167	177534			JMP	NXMTSM	: LOOP ON ERROR
1085	005770	032777	002000	173030	CKNEM:	BIT	#BIT10, @RCER	: IS NEM CLEARED BY DISK CLEAR
1086	005776	001412				BEQ	LPNEM	: NEM CLEARED BY DISK CLEAR
1087	006000	012767	000105	173054	ERR105:	MOV	#105, ERCOUNT	: ***** ERROR 105 *****
1088	006006	017767	173014	173056		MOV	@RCER, WORK	: REPORT NEM NOT CLEARED BY DISK CLEAR
1089	006014	004567	002530			JSR	%5, STAER	:
1090	006020	000167	177500			JMP	NXMTSM	: LOOP ON ERROR
1091	006024	000004			LPNEM:	SCOPE		: ENTER SCOPE LOOP
1092	006026	005524				NXMTSM		

```

1093      ; IN THIS ROUTINE THE PROGRAM WILL TEST
1094      ; THAT THE DISK WILL ONLY TRAP
1095      ; AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
1096      ; BY CLEARING THE DONE
1097      ; PROCESSOR OPERATING AT PRIORITY7
1098
1099 006030 012777 006130 173000 STA106: MOV      #INT106, @RCVEC ; SET UP INTERUPT VECTOR
1100 006036 012777 000340 172736      MOV      #340, @PS ; LOCK OUT ALL INTERRUPTS
1101 006044 005077 172754      CLR      @RCDA
1102 006050 012777 177777 172742      MOV      #177777, @RCWC ; SET WORD COUNT TO -1
1103 006056 012777 011654 172736      MOV      @OUTBUF, @RCBA ; LOAD CURRENT ADDRESS
1104 006064 052777 000103 172724      BIS      #103, @RCCS ; GO WRITE (INTERRUPT ENABLED)
1105 006072 005000      CLR      %0 ; WAIT FOR RDY
1106 006074 005200      INC      %0
1107 006076 001376      BNE     .-2
1108 006100 105777 172712      TSTB   @RCCS ; IS CONTROL STILL NOT RDY
1109 006104 100423      BMI     LP106 ; NO
1110 006106 017767 172704 172756      MOV      @RCCS, WORK ; YES!
1111 006114 012767 000106 172740 ERR106: MOV      #106, @RCOUNT ; ***** ERROR 106 *****
1112 006122 004567 002422      JSR     %5, @TAER ; REPORT CONTROL STILL BUSY
1113 006126 000414      BR      STA110
1114
1115      ; PROCESSOR SHOULD NOT TRAP TO INT106
1116      ; PROCESSOR PRIORITY IS LOCKED AT 7
1117
1118 006130      INT106:
1119 006130 012767 000107 172724 ERR107: MOV      #107, @RCOUNT ; ***** ERROR 107 *****
1120 006136 017767 172640 172726      MOV      @PS, WORK ; FETCH PROCESSOR PRIORITY
1121 006144 022626      CMP     (6)+, (6)+
1122 006146 004567 002376      JSR     %5, @TAER ; REPORT DISK INTERRUPTED
1123 006152 000572      BR      ADT1
1124 006154 000004      LP106: SCOPE
1125 006156 006030      STA106 ; ENTER SCOPE LOOP

```

```

1126      ; IN THIS ROUTINE THE PROGRAM WILL TEST
1127      ; THAT THE DISK WILL ONLY TRAP
1128      ; AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
1129      ; BY CLEARING THE DONE
1130      ; PROCESSOR OPERATING AT PRIORITY6
1131
1132 006160 012777 006260 172650 STA110: MOV      #INT110, @RCVEC ; SET UP INTERRUPT VECTOR
1133 006166 012777 000300 172606      MOV      #300, @PS ; LOCK OUT ALL INTERRUPTS ABOVE
1134 006174 005077 172624      CLR      @RCDA
1135 006200 012777 177777 172612      MOV      #177777, @RCWC ; SET WORD COUNT TO -1
1136 006206 012777 011654 172606      MOV      #OUTBUF, @RCBA ; LOAD CURRENT ADDRESS
1137 006214 052777 000103 172574      BIS      #103, @RCCS ; GO WRITE (INTERRUPT ENABLED)
1138 006222 005000      CLR      %0 ; WAIT FOR NOT BUSY
1139 006224 005200      INC      %0
1140 006226 001376      BNE      -2
1141 006230 105777 172562      TSTB    @RCCS ; IS CONTROL STILL BUSY
1142 006234 100423      BMI      LP110 ; NO
1143 006236 017767 172554 172626      MOV      @RCCS, WORK ; YES!
1144 006244 012767 000110 172610 ERR110: MOV      #110, @RCOUNT ; ***** ERROR 110 *****
1145 006252 004567 002272      JSR      %5, @TAER ; REPORT CONTROL STILL BUSY
1146 006256 000414      BR       STA112
1147
1148      ; PROCESSOR SHOULD NOT TRAP TO INT106
1149      ; PROCESSOR PRIORITY IS LOCKED AT 6
1150
1151 006260      INT110:
1152 006260 012767 000111 172574 ERR111: MOV      #111, @RCOUNT ; ***** ERROR 111 *****
1153 006266 017767 172510 172576      MOV      @PS, WORK ; FETCH PROCESSOR PRIORITY
1154 006274 022626      CMP      (6)+, (6)+
1155 006276 004567 002246      JSR      %5, @TAER ; REPORT DISK INTERRUPTED
1156 006302 000402      BR       STA112
1157
1158 006304 000004      LP110: SCOPE ; ENTER SCOPE LOOP
1159 006306 006160      STA110
    
```

```

1160      ; IN THIS ROUTINE THE PROGRAM WILL TEST
1161      ; THAT THE DISK WILL ONLY TRAP
1162      ; AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
1163      ; BY CLEARING THE DONE
1164      ; PROCESSOR OPERATING AT PRIORITYS
1165
1166 006310 012777 006410 172520 STA112: MOV    #INT112,@RCVEC ;SET UP INTERRUPT VECTOR
1167 006316 012777 000240 172456      MOV    #240,@PS      ;LOCK OUT ALL INTERRUPTS ABOVE
1168 006324 005077 172474      CLR    @RCDA
1169 006330 012777 177777 172462      MOV    #177777,@RCWC ;SET WORD COUNT TO -1
1170 006336 012777 011654 172456      MOV    #OUTBUF,@RCBA ;LOAD CURRENT ADDRESS
1171 006344 052777 000103 172444      BIS    #103,@RCCS    ;GO WRITE (INTERRUPT ENABLED)
1172 006352 005000      CLR    %0            ;WAIT FOR NOT BUSY
1173 006354 005200      INC    %0
1174 006356 001376      BNE    .-2
1175 006360 105777 172432      TSTB  @RCCS          ;IS CONTROL STILL BUSY
1176 006364 100423      BMI    LP112         ;NO
1177 006366 017767 172424 172476      MOV    @RCCS,WORK   ;YES!
1178 006374 012767 000112 172460 ERR112: MOV    #112,@RCOUNT ;***** ERROR 112 *****
1179 006402 004567 002142      JSR    %5,STAER     ;REPORT CONTROL STILL BUSY
1180 006406 000414      BR     STA114       ;RESTART ROUTINE

```

```

1181 ;PROCESSOR SHOULD NOT TRAP TO INT106
1182 ;PROCESSOR PRIORITY IS LOCKED AT 7
1183
1184 006410 INT112:
1185 006410 012767 000113 172444 ERR113: MOV #113,ERCOUNT ; ***** ERROR 113 *****
1186 006416 017767 172360 172446 MOV @PS,WORK ;FETCH PROCESSOR PRIORITY
1187 006424 022626 CMP (6)+,(6)+
1188 006426 004567 002116 JSR %5,STAER ;REPORT DISK INTERRUPTED
1189 006432 000402 BR STA114
1190
1191 006434 000004 LP112: SCOPE ;ENTER SCOPE LOOP
1192 006436 006310 STA112
1193
1194
1195
1196 ;THIS ROUTINE WILL TEST THE ABILITY OF THE DISK CONTROL
1197 ;TO TRAP AT BR5 WHEN THE DONE FLAG IS SET.
1198
1199 006440 012777 006532 172370 STA114: MOV #INT114,@RCVEC ;SET UP DISK TRAP VECTOR
1200 006446 012777 000200 172326 MOV #200,@PS ;SET PROCESSOR TO PRIORITY 4
1201 006454 005077 172344 CLR @RCDA ;CLEAR LOW ORDER ADDRESS BITS
1202 006460 012777 177777 172332 MOV #177777,@RCWC ;SET WORD COUNT TO -1
1203 006466 012777 011654 172326 MOV #OUTBUF,@RCBA ;LOAD CURRENT ADDRESS
1204 006474 052777 000103 172314 BIS #103,@RCCS ;WRITE (INTERRUPT ENABLE
1205 006502 005000 CLR %0
1206 006504 005200 INC %0 ;WAIT FOR INTERRUPT TO OCCUR
1207 006506 001376 BNE .-2
1208 006510 012767 000114 172344 ERR114: MOV #114,ERCOUNT ; ***** ERROR 114 *****
1209 006516 017767 172274 172346 MOV @RCCS,WORK ;FETCH CONTENTS OF RCCS
1210 006524 004567 002020 JSR %5,STAER ;REPORT INTERRUPT NO INTERRUPT
1211 006530 000403 BR ADT1 ;RESTART ROUTINE
1212
1213 006532 022626 INT114: CMP (6)+,(6)+ ;TRAP OK
1214 006534 000004 SCOPE ;ENTER SCOPE LOOP
1215 006536 006440 STA114

```

```

1216                                     ; * * * ADDRESS TEST 1 * * *
1217                                     ; EXECUTE A ONE WORD WRITE
1218                                     ; IF THE TIMING ON THE DISK IS CORRECT
1219                                     ; THE TERMINATING ADDRESS IN THE RCDA REGISTER
1220                                     ; WILL EQUAL THE ADDRESS +1 OF THE WORD
1221                                     ; THAT WAS WRITTEN
1222
1223                                     ; NOTE: DATA IS NOT CHECKED IN THIS TEST.
1224
1225 006540 012706 001000 ADT1:  MOV    #1000,%6      ; SET UP STACK
1226 006544 012767 000001 172272      MOV    #1,WRDCT    ; SET UP WORD COUNT
1227 006552 012767 011654 172272      MOV    #OUTBUF,BUF ; SET UP CURRENT ADDRESS
1228 006560 005067 172262              CLR    DMA
1229 006564 104403              WRADT: WRITE
1230 006566 105777 172224              TSTB  @RCCS      ; CHECK FOR READY
1231 006572 100375              BPL   #-4
1232 006574 005777 172216              TST  @RCCS      ; CHECK FOR ERROR
1233 006600 100011              BPL  CHKADT     ; BRANCH IF NO ERROR
1234 006602 017767 172210 172262      MOV    @RCCS,WORK
1235 006610 012767 000115 172244  ERR115: MOV    #115,ERCOUNT ; ***** ERROR 115 *****
1236 006616 004567 001726              JSR   %5,STAER  ; REPORT ERROR
1237 006622 000760              BR    WRADT     ; LOOP ON ERROR
1238 006624 016767 172216 172242  CHKADT: MOV    DMA,WORK1
1239 006632 005267 172236              INC   WORK1     ; WHAT RCDA SHOULD CONTAIN
1240 006636 017767 172162 172226      MOV    @RCDA,WORK
1241 006644 026767 172224 172220      CMP   WORK1,WORK ; IS RCDA CORRECT
1242 006652 001406              BEQ   INCDA     ; YES RCDA IS CORRECT
1243 006654 012767 000116 172200  ERR116: MOV    #116,ERCOUNT ; ***** ERROR 116 *****
1244 006662 004567 001720              JSR   %5,STAER1 ; REPORT RCDA NOT CORRECT
1245 006666 000736              BR    WRADT     ; LOOP ON ERROR
1246 006670 022767 004000 172176  INCDA: CMP    #4000,WORK1 ; IS IT THE LAST ADDR.
1247 006676 001405              BEQ   LPADT     ; LAST ADDRESS EXIT
1248 006700 016767 172170 172140      MOV    WORK1,DMA ; SET UP FOR NEXT ADDR.
1249 006706 000167 177652              JMP   WRADT     ; CHECK NEXT ADDRESS
1250 006712 032777 004000 172060  LPADT: BIT   #BIT11,@SWR ; LOOP ON TEST
1251 006720 001402              BEQ   .+6
1252 006722 000167 177612              JMP   ADT1     ; YES! BIT11 SET IN SWR

```

```

1253      ; * * * ADDRESS TEST 2X * * *
1254      ; EXECUTE A ONE WORD READ
1255      ; IF THE TIMING ON THE DISK IS CORRECT
1256      ; THE TERMINATING ADDRESS IN THE RCDA REGISTER
1257      ; WILL EQUAL THE ADDRESS +1 OF THE WORD
1258      ; THAT WAS READ
1259      ; NOTE: DATA IS NOT CHECK IN THIS TEST.
1260
1261 006726 012706 001000      ADT2X:  MOV      #1000,%6      ; SET UP STACK
1262 006732 012767 000001 172104      MOV      #1,WRDCT      ; SET UP WORD COUNT
1263 006740 012767 011664 172104      MOV      #INBUF,BUF    ; SET UP CURRENT ADDRESS
1264 006746 005067 172074      CLR      DMA
1265 006752 104405      ROADT:  READ
1266 006754 105777 172036      TSTB    @RCCS          ; CHECK FOR READY
1267 006760 100375      BPL     -4
1268 006762 005777 172030      TST     @RCCS          ; CHECK FOR ERROR
1269 006766 100011      BPL     XCHKDT        ; BRANCH IF NO ERROR
1270 006770 017767 172022 172074      MOV     @RCCS,WORK
1271 006776 012767 000117 172056  ERR117: MOV     #117,ERCOUNT    ; ***** ERROR 117 *****
1272 007004 004567 001540      JSR     %5,STAER      ; REPORT ERROR
1273 007010 000760      BR      ROADT         ; LOOP ON ERROR
1274 007012 016767 172030 172054  XCHKDT: MOV     DMA,WORK1
1275 007020 005267 172050      INC     WORK1         ; WHAT RCDA SHOULD CONTAIN
1276 007024 017767 171774 172040      MOV     @RCDA,WORK
1277 007032 026767 172036 172032      CMP     WORK1,WORK
1278 007040 001406      BEQ    ADDA           ; IS RCDA CORRECT
1279 007042 012767 000120 172012  ERR120: MOV     #120,ERCOUNT    ; YES RCDA IS CORRECT
1280 007050 004567 001532      JSR     %5,STAER1    ; ***** ERROR 120 *****
1281 007054 000736      BR      ROADT         ; REPORT RCDA NOT CORRECT
1282 007056 022767 000100 172010  ADDA:  CMP     #100,WORK1  ; LOOP ON ERROR
1283 007064 001405      BEQ    LPADT2        ; IS IT THE LAST ADDR.
1284 007066 016767 172002 171752      MOV     WORK1,DMA    ; LAST ADDRESS EXIT
1285 007074 000167 177652      JMP     ROADT        ; SET UP FOR NEXT ADDR
1286 007100 032777 004000 171672  LPADT2: BIT     #BIT11,@SWR  ; CHECK NEXT ADDRESS
1287 007106 001402      BEQ    +6            ; LOOP ON TEST
1288 007110 000167 177612      JMP     ADT2X        ; NO:
                          ; YES! BIT11 SET IN SWR

```

```

1289          ;LOOK AHEAD TEST
1290
1291 007114 012706 001000          LATST: MOV      #1000,%6          ;SET UP STACK
1292 007120 005077 171676          CLR      @RCBA          ;SET ADDRESS TO 0
1293 007124 005077 171674          CLR      @RCDA          ;SET DSK ADDRESS TO 0
1294 007130 012777 000001 171660 LA1:   MOV      #1,@RCCS          ;LOOK AHEAD
1295 007136 105777 171654          TSTB    @RCCS          ;DONE?
1296 007142 100375          BPL     .-4
1297 007144 017767 171646 171720 MOV     @RCCS,WORK
1298 007152 005767 171714          TST     WORK          ;ERROR?
1299 007156 100007          BPL     LA2
1300 007160 012767 000121 171674 ERR121: MOV    #121,ERCOUNT          ; ***** ERROR 121 *****
1301 007166 004567 001356          JSR     %5,STAER
1302 007172 000167 000150          JMP     LA0T
1303 007176 005777 171632          LA2:   TST     @RCLA          ;AT 0?
1304 007202 001352          BNE     LA1
1305 007204 005004          CLR     %4
1306 007206 005204          LA3:   INC     %4          ;%4=DSK ADR
1307 007210 012777 000001 171600 MOV     #1,@RCCS          ;LOOK AHEAD
1308 007216 105777 171574          TSTB    @RCCS          ;DONE?
1309 007222 100375          BPL     .-4
1310 007224 017767 171566 171640 MOV     @RCCS,WORK
1311 007232 005767 171634          TST     WORK          ;ERROR?
1312 007236 100007          BPL     LA4
1313 007240 012767 000122 171614 ERR122: MOV    #122,ERCOUNT          ; ***** ERROR 122 *****
1314 007246 004567 001276          JSR     %5,STAER
1315 007252 000167 000070          JMP     LA0T

```

```

1316 007256 017767 171552 171606 LA4:  MOV  @RCLA,WORK ;SAVE ADDRESS
1317 007264 005767 171602          TST  WORK ;ERROR BIT SET?
1318 007270 100007          BPL  LAS
1319 007272 012767 000123 171562 ERR123: MOV  @123,ERCOUNT ; ***** ERROR 123 *****
1320 007300 004567 001244          JSR  %5,STAER
1321 007304 000167 000036          JMP  LADT
1322 007310 020467 171556          LA5:  CMP  %4,WORK ;CHECK ADDRESS
1323 007314 001411          BEQ  LA6
1324 007316 010467 171552          MOV  %4,WORK1
1325 007322 012767 000124 171532 ERR124: MOV  @124,ERCOUNT ; ***** ERROR 124 *****
1326 007330 004567 001214          JSR  %5,STAER
1327 007334 000167 000006          JMP  LADT
1328 007340 022704 000077          LA6:  CMP  @77,%4 ;END?
1329 007344 001320          BNE  LA3
1330 007346 032777 004000 171424 LADT:  BIT  @BIT11,@SWR
1331 007354 001402          BEQ  .+6
1332 007356 000167 177532          JMP  LATST
1333
1334          ;WRITE THE WHOLE DISK WITH 4K CHUNKS OF CORE
1335
1336 007362 005227 177777          WRTDAT: INC  #-1 ;ONCE ONLY CODE
1337 007366 001037          BNE  CHKBLK ;SKIP IT
1338 007370 005077 171430          WR4:  CLR  @RCDA ;CLEAR DISK ADDRESS
1339 007374 005077 171422          WR1:  CLR  @RCBA ;CLEAR MEMORY ADDRESS
1340 007400 012777 170000 171412          MOV  #-10000,@RCMC ;LOAD WORD COUNT WITH 4K
1341 007406 012777 000003 171402          MOV  @3,@RCCS ;WRITE
1342 007414 032777 100200 171374 WR2:  BIT  @100200,@RCCS ;ERROR OR DONE?
1343 007422 001774          BEQ  WR2 ;NO - TEST AGAIN
1344 007424 100014          BPL  WR3 ;DONE!
1345 007426 017767 171364 171436          MOV  @RCCS,WORK ;LOAD STATUS FOR TYPING
1346 007434 017767 171366 171432          MOV  @RCER,WORK1 ;LOAD ERROR REG.
1347 007442 012767 000125 171412 ERR125: MOV  @125,ERCOUNT ; ***** ERROR 125 *****
1348 007450 004567 001132          JSR  5,STAER1 ;TYPE THE ERROR MESSAGE
1349 007454 000745          BR   WR4 ;TRY AGAIN
1350 007456 032777 003777 171340 WR3:  BIT  @3777,@RCDA ;END OF THE DISK?
1351 007464 001343          BNE  WR1 ;NO!

```

1352	007456	012706	001000		CHKBLK:	MOV	#1000,%6		;SET STACK
1353	007472	012777	000340	171302		MOV	#340,%PS		;LOCK OUT INTERRUPTS
1354	007500	005077	171320			CLR	ARCDA		;DA=0
1355	007504	012777	177740	171306	BLK1:	MOV	#-40,ARCWC		;1 BLOCK
1356	007512	012777	011674	171302		MOV	#DATAA,ARCBA		;SET DATA AREA
1357	007520	012777	000005	171270		MOV	#5,ARCCS		;READ
1358	007526	105777	171264			TSTB	ARCCS		;READY?
1359	007532	100375				BPL	-4		
1360	007534	005777	171256			TST	ARCCS		;ERROR?
1361	007540	100014				BPL	BLK2		
1362	007542	012767	000126	171312	ERR126:	MOV	#126,ERCOUNT		; ***** ERROR 126 *****
1363	007550	017767	171242	171314		MOV	ARCCS,WORK		
1364	007556	017767	171244	171310		MOV	ARCER,WORK1		
1365	007564	004567	001016			JSR	%5,STAER1		;REPORT ERROR
1366	007570	000736				BR	CHKBLK		
1367	007572	004567	000100		BLK2:	JSR	%5,BLKCHK		;GET BLOCK CHECK
1368	007576	026777	171270	171224		CMP	WORK,ARCDB		;SAME?
1369	007604	001411				BEQ	BLK3		
1370	007606	012767	000127	171246	ERR127:	MOV	#127,ERCOUNT		; ***** ERROR 127 *****
1371	007614	017767	171210	171252		MOV	ARCDB,WORK1		
1372	007622	004567	000760			JSR	%5,STAER1		;REPORT ERROR
1373	007626	000717				BR	CHKBLK		
1374	007630	022777	004000	171166	BLK3:	CMP	#4000,ARCDA		;END?
1375	007636	001322				BNE	BLK1		
1376	007640	032777	004000	171132		BIT	#BIT11,ASWR		
1377	007646	001307				BNE	CHKBLK		
1378	007650	104001			ENDX:	EMT+1			;REPORT END
1379	007652	011577				END			
1380	007654	013700	000042			MOV	#42,%0		;GET MONITOR ADDRESS
1381	007660	001404				BEQ	LXIT		;SKIP IF NO HOOK
1382	007662	004710			LOGICAL:	JSR	7,(0)		;GO TO MONITOR
1383	007664	000240				NOP			
1384	007666	000240				NOP			
1385	007670	000240				NOP			
1386	007672	000167	171342		LXIT:	JMP	STAIL		;RESTART TEST

```

1387          ;      BLOCK CHECK TEST
1388          ;      ENTRY VIA:      JSR      %5, BLKCHK
1389
1390 007676 010046          BLKCHK: MOV      %0, -(6)
1391 007700 010146          MOV      %1, -(6)
1392 007702 010246          MOV      %2, -(6)
1393 007704 010346          MOV      %3, -(6)
1394 007706 010446          MOV      %4, -(6)
1395 007710 012700 011674  MOV      #DATAA, %0      ;GET ADDRESS OF DATA WORD
1396 007714 012767 000040 000220  MOV      #32., WRDCT1      ;SET WORD COUNT
1397 007722 005004          CLR      %4
1398 007724 012001          WRDLOP: MOV     (0)+, %1      ;GET DATA WORD
1399 007726 012767 000020 000204  MOV      #16., BLKCT      ;SET BLOCK COUNT
1400 007734 006001          BLKLOP: ROR     %1      ;PUT BIT INTO C
1401 007736 006102          ROL      %2      ;PUT C INTO RIGHT MOST BIT (IN)
1402 007740 010403          MOV      %4, %3
1403 007742 006003          ROR      %3
1404 007744 006167 000174  ROL      .BIT0
1405 007750 006003          ROR      %3
1406 007752 006167 000170  ROL      .BIT1
1407 007756 010403          MOV      %4, %3
1408 007760 006103          ROL      %3
1409 007762 006103          ROL      %3
1410 007764 006167 000162  ROL      .BIT14
1411 007770 010267 000160  MOV      %2, .BIT15
1412 007774 066767 000144 000152  ADD      .BIT0, .BIT15
1413 010002 016767 000146 000140  MOV      .BIT15, .BIT13
1414 010010 066767 000136 000132  ADD      .BIT14, .BIT13
1415 010016 016767 000132 000120  MOV      .BIT15, .BIT0
1416 010024 066767 000116 000112  ADD      .BIT1, .BIT0
1417 010032 006004          ROR      %4
1418 010034 042704 120001  BIC      #120001, %4
1419 010040 032767 000001 000106  BIT      #1, .BIT15
1420 010046 001402          BEQ      .+6
1421 010050 052704 100000  BIS      #100000, %4
1422 010054 032767 000001 000066  BIT      #1, .BIT13
1423 010062 001402          BEQ      .+6
1424 010064 052704 020000  BIS      #20000, %4
1425 010070 032767 000001 000046  BIT      #1, .BIT0
1426 010076 001402          BEQ      .+6
1427 010100 052704 000001  BIS      #1, %4
1428 010104 005367 000030  DEC      BLKCT
1429 010110 001311          BNE      BLKLOP
1430 010112 005367 000024  DEC      WRDCT1
1431 010116 001302          BNE      WRDLOP
1432 010120 010467 170746  MOV      %4, WORK
1433 010124 012604          MOV      (6)+, %4
1434 010126 012603          MOV      (6)+, %3
1435 010130 012602          MOV      (6)+, %2
1436 010132 012601          MOV      (6)+, %1
1437 010134 012600          MOV      (6)+, %0
1438 010136 000205          RTS      %5

```

```

1439 010140 000000          BLKCT: 0
1440 010142 000000          WRDCT1: 0
1441 010144 000000          .BIT0: 0
1442 010146 000000          .BIT1: 0
1443 010150 000000          .BIT13: 0
1444 010152 000000          .BIT14: 0
1445 010154 000000          .BIT15: 0
1446
1447
1448          ;SCOPE LOOP ROUTINE
1449          ;IF BIT 11 SET LOOP ON TEST
1450
1451 010156 032777 004000 170614 LOOP: BIT      #BIT11, @SWR      ;TST FOR BIT 11
1452 010164 001402          BEQ      +6          ;BIT 11 NOT SET
1453 010166 013646          MOV      @ (6)+, -(6)
1454 010170 000002          RTI
1455 010172 005767 170672          TST      PASS      ;BIT 11 SET! LOOP ON TEST
1456 010176 001003          BNE      +10       ;TEST TO SET UP PASS COUNT
1457 010200 012767 000025 170662          MOV      #25, PASS ;PASS COUNT SET
1458 010206 005367 170656          DEC      PASS      ;SET UP PASS COUNT
1459 010212 001402          BEQ      +6          ;SUB. -1 EACH PASS
1460 010214 013646          MOV      @ (6)+, -(6)
1461 010216 000002          RTI
1462 010220 062716 000002          ADD      #2, (6)   ;LOOP ON TEST
1463 010224 000002          RTI              ;INDEX POINTER FOR NEXT TEST
1464
1465
1466          ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
1467          ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
1468          ;BYTE OF THE CONTROL REGISTER.
1469
1470 010226 012705 001026          DISK: MOV      #RCER, %S      ;SET UP TO LOAD DISK REG
1471 010232 016755 170610          MOV      DMA, @-(5)      ;LOAD WORD ADDRESS
1472 010236 016755 170610          MOV      BUF, @-(5)      ;SET UP CURRENT ADDRESS
1473 010242 016755 170576          MOV      WRDCT, @-(5)    ;LOAD WORD COUNT
1474 010246 005475 000000          NEG      @ (5)          ;SET UP TWO'S COMPLEMENT
1475 010252 011604          MOV      (6), %4
1476 010254 014467 170612          MOV      -(4), WORK
1477 010260 042767 177600 170604          BIC      #177600, WORK   ;MASK FUNCTION BITS
1478 010266 016755 170600          MOV      WORK, @-(5)    ;LOAD FUNCTION REG.
1479 010272 000002          RTI

```

```

1480 ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
1481 ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
1482
1483 ;WORD COUNT REGISTER
1484 010274 017777 170500 170516 SELWC: MOV @SWR,@RCWC ;MOV SWR INTO WORD COUNT REG
1485 010302 000774 BR SELWC
1486
1487 ;CURRENT ADDRESS REGISTER
1488 010304 017777 170470 170510 SELBA: MOV @SWR,@RCBA ;MOV SWR INTO CURRENT ADDR REG
1489 010312 000774 BR SELBA
1490
1491 ;DISK ADDRESS REGISTER
1492 010314 017777 170460 170502 SELDA: MOV @SWR,@RCDA ;MOV SWR INTO DISK ADDR REG
1493 010322 000774 BR SELDA
1494
1495 ;DISK ERROR REGISTER
1496 010324 017777 170450 170474 SELER: MOV @SWR,@RCER ;MOV SWR INTO DISK ADDR EXT REG
1497 010332 000774 BR SELER
1498
1499 ;DATA BUFFER REGISTER
1500 010334 017777 170440 170466 SELRCDB:MOV @SWR,@RCDB ; MOV SWR INTO DATA BUFFER
1501 010342 000774 BR SELRCDB
1502
1503
1504 ;LOOK AHEAD REGISTER
1505 010344 017700 170464 MOVLK: MOV @RCLA,%D ; ;FETCH LOOK AHEAD
1506 010350 000005 RESET ;DISPLAY IN LIGHTS
1507 010352 000005 RESET ;
1508 010354 000773 BR MOVLK ;
1509
1510 ;DISK CONTROL STATUS REGISTER
1511 010356 012777 000340 170416 SELCS: MOV #340,@PS ;LOCK UP INTERRUPTS
1512 010364 012777 177777 170426 MOV #177777,@RCWC ;SET WORD COUNT -1 WORD
1513 010372 012777 011654 170422 MOV @OUTBUF,@RCBA ;SET UP CURRENT ADDRESS
1514 010400 017777 170374 170410 MOV @SWR,@RCCS ;MOV SWR INTO CONTROL REG
1515 010406 032777 000001 170402 BIT @BIT0,@RCCS ;IS FUNCTION BITS SET
1516 010414 001760 BEQ SELCS ;FUNCTION BITS NOT SET
1517 010416 105777 170374 DKBUSY: TSTB @RCCS ;TEST FOR DISK READY
1518 010422 100375 BPL DKBUSY ;DISK STILL NOT READY
1519 010424 000754 BR SELCS ;DISK NOT BUSY SELECT NEW CR

```

```

1520 ; THIS ROUTINE ENABLES THE OPERATOR TO SELECT A TRACK STATICLY
1521 ; THE ROUTINE DOES A ONE WORD READ TO SELECT THE TRACK
1522 ; THE OPERATOR MAY CHANGE THE SWITCH REGISTER AT ANY TIME
1523 ; SWR4-0 EQUALS THE TRACK NUMBER
1524 ; SWR6-5 EQUALS THE DISK NUMBER
1525
1526 010426 000005 STAMP: RESET
1527 010430 017767 170344 170436 MOV @SWR,WORK1
1528 010436 017767 170336 170426 MOV @SWR,WORK
1529 010444 006367 170422 ASL WORK
1530 010450 006367 170416 ASL WORK
1531 010454 006367 170412 ASL WORK
1532 010460 006367 170406 ASL WORK
1533 010464 006367 170402 ASL WORK
1534 010470 006367 170376 ASL WORK
1535 010474 006367 170372 ASL WORK
1536 010500 016777 170366 170316 MOV WORK,@RCDA ;DISK ADDRESS REG LOADED
1537 010506 012777 011664 170306 MOV #INBUF,@RCBA ;LOAD CURRENT ADDRESS
1538 010514 012777 177777 170276 MOV #177777,@RCWC ;LOAD WORD COUNT
1539 010522 052777 000005 170266 BIS #5,@RCCS ;GO AND READ
1540 010530 105777 170262 CTBUSY: TSTB @RCCS ;TEST FOR CONTROL READY
1541 010534 100375 BPL CTBUSY ;WAIT FOR CONTROL READY
1542 010536 026777 170332 170234 SWRCHG: CMP WORK1,@SWR ;SWR HAS CHANGED
1543 010544 001330 BNE STAMP ;SWR HAS NOT CHANGED
1544 010546 000773 BR SWRCHG

```

```

1545 ;ROUTINE TO REPORT ERROR COUNT AND CONTENTS OF ONE REGISTER
1546
1547 010550 004567 000404 STAER: JSR %5,CONV ;CONVERT OCTAL TO ASCII
1548 010554 001072 WORK ;DATA TO BE CONVERTED
1549 010556 011570 MES6X ;ADDRESS OF MESSAGE
1550 010560 000006 6
1551 010562 004567 000372 JSR %5,CONV ;CONVERT OCTAL TO ASCII
1552 010566 001062 ERCOUNT ;ERROR COUNT TO BE CONVERTED
1553 010570 011547 ERNO ;ADDRESS OF MESSAGE
1554 010572 000003 3
1555 010574 104000 EMT +0 ;REPORT MESSAGE
1556 010576 011540 HED5
1557 010600 011565 MES6
1558 010602 177777 -1
1559 010604 000424 BR STAERX
1560
1561 ;ROUTINE TO REPORT ERROR COUNT AND THE CONTENTS OF TWO REGISTERS
1562
1563
1564
1565 010606 004567 000346 STAER1: JSR %5,CONV ;CONVERT OCTAL TO ASCII
1566 010612 001072 WORK ;DATA TO BE CONVERTED
1567 010614 011570 MES6X ;ADDRESS OF MESSAGE
1568 010616 000006 6
1569 010620 004567 000334 JSR %5,CONV ;CONVERT OCTAL TO ASCII
1570 010624 001074 WORK1 ;DATA TO BE CONVERTED
1571 010626 011556 MES5X ;ADDRESS OF MESSAGE
1572 010630 000006 6
1573 010632 004567 000322 JSR %5,CONV ;CONVERT OCTAL TO ASCII
1574 010636 001062 ERCOUNT ;ERROR COUNT TO BE CONVERTED
1575 010640 011547 ERNO
1576 010642 000003 3
1577 010644 104000 EMT +0 ;REPORT MESSAGE
1578 010646 011540 HED5
1579 010650 011553 MES5
1580 010652 011565 MES6
1581 010654 177777 -1
1582 010656 032777 000400 170114 STAERX: BIT #BIT8,JSWR ;ODT?
1583 010664 001403 BEQ XXX
1584 010666 000177 170122 JMP @ODT
1585 010672 000205 RTS %5
1586 010674 032777 002000 170076 XXX: BIT #BIT10,JSWR
1587 010702 001401 BEQ .+4
1588 010704 000000 HALT
1589 010706 000205 RTS %5 ;EXIT ROUTINE
1590

```

```

1591 ;ROUTINE TO DECODE EMT CALLS
1592 ;EMT+1=TYPE ONE LINE OF TEXT
1593 ;EMT+0=TYPE A SERIES OF LINES
1594
1595 010710 011600 EMTRP: MOV (6),%0
1596 010712 022740 104001 CMP #EMT+1,-(0) ;WAS THE CALL EMT+1
1597 010716 001103 BNE TYP5 ;NO! TYPE A SERIES OF LINES OF TEXT
1598 010720 000400 BR TYP ;YES TYPE ONE LINE OF TEXT
1599
1600 ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
1601
1602 010722 011600 TYP: MOV @%6,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1603 010724 062716 000002 ADD #2,@%6 ;SET UP EXIT.
1604 010730 011000 MOV @%0,%0 ;ADDRESS OF MESSAGE TO RD.
1605 010732 112067 000164 TYPA: MOVB (0)+,TYPDAT ;GET CHARACTER
1606 010736 122767 000100 000156 CMPB #100,TYPDAT ;CHECK FOR "3" CHARACTER
1607 010744 001005 BNE TYPC ;BRANCH IF NOT "3".
1608 010746 005067 000150 CLR TYPDAT ;OUTPUT NULL TO
1609 010752 004767 000030 JSR %7,TYPD ;CLEAR BUFFER
1610 010756 000002 RTI ;TERMINATOR CHAR. DONE. EXIT.
1611 010760 122767 000045 000134 TYPC: CMPB #45,TYPDAT ;CHECK FOR "%".
1612 010766 001442 BEQ TYPF ;BRANCH IF "%".
1613 010770 122767 000042 000124 CMPB #42,TYPDAT ;NOT "%". CHECK FOR "#".
1614 010776 001443 BEQ TYPG ;BRANCH IF "#".
1615 011000 004767 000002 JSR %7,TYPD ;TYPE CHAR IN TYPDAT
1616 011004 000752 BR TYPA
1617 011006 032777 040000 167764 TYPD: BIT #BIT14,@SWR
1618 011014 001026 BNE TYEXIT
1619 011016 116777 000100 167760 MOVB TYPDAT,@TPB ;OUTPUT CHARACTER TO PRINTER
1620 011024 105777 167760 TSTB @TPS ;WAIT FOR DONE FLAG.
1621 011030 100375 BPL -4
1622 011032 122767 000015 000062 CMPB #15,TYPDAT ;CHECK FOR CR
1623 011040 001003 BNE IS ;NO - SKIP
1624 011042 012767 000011 000054 MOV #9,NULL ;SET NULL COUNTER
1625 011050 005767 000050 IS: TST NULL ;TEST COUNTER
1626 011054 001406 BEQ TYEXIT ;ZERO - EXIT
1627 011056 005367 000042 DEC NULL ;OUTPUT NULL
1628 011062 112767 000000 000032 MOVB #0,TYPDAT ;ZERO OUTPUT
1629 011070 000746 BR TYPD ;OUTPUT NULL
1630 011072 000207 TYEXIT: RTS %7 ;EXIT
1631 011074 112767 000015 000020 TYPF: MOVB #15,TYPDAT ;MOVE CARRIAGE RETURN CODE TO TYPDAT
1632 011102 004767 177700 JSR %7,TYPD ;GO TYPE CHAR.
1633 011106 112767 000012 000006 TYPG: MOVB #12,TYPDAT ;MOVE LF CODE TO TYPDAT.
1634 011114 004767 177666 JSR %7,TYPD ;GO TYPE CHAR.
1635 011120 000704 BR TYPA
1636 011122 000000 TYPDAT: 0
1637 011124 000000 NULL: 0
1638
1639 ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
1640
1641 011126 011600 TYP5: MOV @%6,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1642 011130 062716 000002 ADD #2,@%6 ;UPDATE TO NEXT MESSAGE ADDRESS
1643 011134 011067 000014 MOV @%0,TYPSB ;ADDRESS OF MESSAGE TO TYPSB
1644 011140 022767 177777 000006 CMPB #-1,TYPSB ;CHECK FOR TERMINATOR
1645 011146 001001 BNE TYPSA ;BRANCH IF NOT TERMINATOR.
1646 011150 000002 RTI ;TERMINATOR. EXIT

```

K04

MAINDEC-11-DZRC-A
DZRCAB.P11

RC11 STATIC TEST

REPLACES DSJA-PB1

MACY11 27(732) 10-SEP-76 11:53 PAGE 48

1647 011152 104001
1648 011154 000000
1649 011156 000763

TYPSA: EMT +1
TYPSE: 0
BR TYPSE

;CALL ON TYP SUB TO TYPE MESSAGE
;ADDRESS OF MESSAGE GOES HERE
;GO PROCESS NEXT MESSAGE

```

1650 ;OCTAL TO ASCII CONVERT ROUTINE
1651 ;ENTER ROUTINE AS FOLLOWS
1652 ;JSR%5, CONV
1653 ;ADDR#=ADDRESS OF NUMBER TO BE CONVERTED
1654 ;ADDR BYTE=LSB OF WHERE ASCII IS GOING
1655 ;ASCII#=THE NUMBER OF ASCII CHAR. TO BE CONVERTED
1656
1657
1658 011160 013567 000054 CONV: MOV @5+, ACNVX ;VALUE OF # TO BE CONVERTED
1659 011164 012501 MOV (5)+,%1 ;ASCII ADDR
1660 011166 012502 MOV (5)+,%2 ;# OF ASCII CHAR
1661 011170 060201 ADD %2,%1
1662
1663 011172 016703 000042 ACVN: MOV ACNVX,%3
1664 011176 042703 177770 BIC #177770,%3 ;ISOLATE LEAST SIGNIFICANT OCTAL#
1665 011202 062703 000060 ADD #60,%3 ;SET UP ASCII#
1666 011206 110341 MOVB %3,-(1) ;STORE ASCII CHAR
1667 011210 042767 000007 000022 BIC #7,ACNVX
1668 011216 006067 000016 ROR ACNVX ;ROTATE OCTAL#
1669 011222 006067 000012 ROR ACNVX
1670 011226 006067 000006 ROR ACNVX
1671 011232 005302 DEC %2 ;-1 FROM ASCII CHAR COUNT
1672 011234 001356 BNE ACVN
1673 011236 000205 RTS %5 ;EXIT # CONVERTED
1674 011240 000000 ACNVX: 0 ;WORK REGISTER
1675
1676 011242 017767 167540 000024 TTYH: MOV @TKB, TTYB
1677 011250 042767 000200 000016 BIC #200, TTYB
1678 011256 022767 000003 000010 CMP #3, TTYB
1679 011264 001002 BNE +6
1680 011266 000167 166706 JMP 200
1681 011272 000002 RTI
1682
1683 011274 000000 TTYB: 0
1684
1685 ;ROUTINE TO SET ACTION ENABLE ON MA/MF PARITY MEMORIES
1686 ;CALL JSR PC, MAMF
1687
1688 172100 PARCSR= 172100
1689 000114 PARVEC= 114
1690 000004 ERRVEC= 4
1691 000006 SP= %6
1692
1693 011276 012737 011370 000114 MAMF: MOV #PARSRV, @PARVEC ;SET PARITY INTERRUPT VECTOR
1694 011304 012767 000340 166604 MOV #340, PARVEC+2 ;AND PRIORITY LEVEL 7 ON INTERRUPT
1695 011312 013746 000004 MOV @ERRVEC, -(SP) ;SAVE CURRENT ERROR VECTOR
1696 011316 013746 000006 MOV @ERRVEC+2, -(SP) ;AND PRIORITY LEVEL
1697 011322 012737 000006 000004 MOV #ERRVEC+2, @ERRVEC
1698 011330 012737 000002 000006 MOV #RTI, @ERRVEC+2
1699 011336 012700 172100 MOV #PARCSR,%0 ;GET FIRST CSR ADDRESS
1700 011342 012702 000001 MOV #1,%2
1701 011346 012720 000001 15: MOV #1,(0)+ ;SET ACTION ENABLE IF AVAILABLE
1702 011352 006302 ASL %2 ;SHIFT AVAILABILITY INDICATOR
1703 011354 103374 BCC 15
1704 011356 012637 000006 MOV (SP)+, @ERRVEC+2 ;RESTORE ERROR VECTOR
1705 011362 012637 000004 MOV (SP)+, @ERRVEC ;PRIORITY LEVEL AND INTERRUPT VECTOR

```

```

1706 011366 000207          RTS      %7
1707
1708
1709          ;PARITY ERROR SERVICE ROUTINE
1710          ;WHEN A PARITY ERROR IS DETECTED THE ROUTINE
1711          ;SEARCHES MEMORY FOR THE PARITY ERROR.  WHEN THE
1712          ;ERROR IS DETECTED THE PROGRAM HALTS WITH
1713          ;THE ADDRESS CAUSING THE ERROR IN RO.
1714          ;TO CONTINUE - PRESS CONTINUE.
1715 011370 104001          PARSRV:  EMT+1
1716 011372 011627          PARERR
1717 011374 012737 011422 000114          MOV      #2$,@#PARVEC ;REPOSITION THE PARITY ERROR INTERRUPT.
1718 011402 012737 011450 000004          MOV      #4$,@#ERRVEC ;SET TIMEOUT TRAP
1719 011410 005037 000006          CLR      @#ERRVEC+2
1720 011414 005000          CLR      %0
1721 011416 005720          1$:      TST      (0)+ ;SCAN MEMORY
1722 011420 000776          BR       1$
1723 011422 000000          2$:      HALT ;PARITY ERROR - ADDRESS
1724                                     ;CAUSING ERROR IS IN REGISTER 0
1725 011424 000005          3$:      RESET
1726 011426 012737 011370 000114          MOV      #PARSRV,@#PARVEC ;RESTORE PARITY VECTOR
1727 011434 012737 000006 000004          MOV      #ERRVEC+2,@#ERRVEC ;RESTORE TIMEOUT HALT.
1728 011442 004767 177630          JSR     %7,MAMF
1729 011446 000002          RTI
1730 011450 000000          4$:      HALT
1731 011452 000764          BR       3$ ;ERROR - PARITY ERROR NOT DETECTED ON SCAN
1732                                     ;4(SP) CONTAINS PC WHERE
1733 011454 012767 011472 166342          DOWN:  MOV      #UP,24 ;PARITY ERROR WAS ORIGINALLY DETECTED.
1734 011462 052777 000400 167326          BIS     #BIT8,@RCCS ;SET RETURN
1735 011470 000000          HALT ;ABORT
1736 011472 012767 011454 166324          UP:    MOV      #DOWN,24 ;RESET DOWN
1737 011500 012706 001000          MOV     #1000,%6 ;RESET STACK
1738 011504 012767 177324 167360          MOV     #-300.,WORK
1739 011512 000005          TIMCNT: RESET
1740 011514 005267 167352          INC     WORK ;TIMEOUT
1741 011520 001374          BNE    TIMCNT
1742 011522 104001          EMT     +1 ;PRINT MSG.
1743 011524 011605          PWRF
1744 011526 000167 167346          JMP     START ;START PROG.

```

```

1745 011532 000000          TEXTBUF: 0
1746 011534 000000          TSTCH: 0
1747
1748          ;ERROR MESSAGE HEADERS
1749
1750
1751 011536 040045          HEDSA: .EVEN
1752 011540 042445 051122 051117 HEDS:  .ASCII  /%a/
1753 011546      040          HED5:  .ASCII  /%ERROR /
1754 011547      000          ERNO:  .BYTE  0,0,0,'a
1755 011552      100
1756
1757          ;MESSAGE TRAILERS
1758
1759 011553      040 020040          MESS:  .ASCII  /,/,/,/,/,/,/,/
1760 011556      000      000      000 MESSX: .BYTE  0,0,0,0,0,0,'a
1761 011561      000      000
1762 011564      100
1763 011565      040 020040          MES6:  .ASCII  /,/,/,/,/,/,/,/
1764 011570      000      000      000 MES6X: .BYTE  0,0,0,0,0,0,'a
1765 011573      000      000
1766 011576      100
1767 011577      045 047105 077504          END:   .ASCII  /%END/<177>/a/
1768 011604      100
1769 011605      045 047520 042527          PWRF:  .ASCII  /%POWER HAS FAILEDa/
1770 011612 020122 040510 020123
1771 011620 040506 046111 042105
1772 011626      100
1773
1774 011627      045 042515 047515          PARERR: .ASCII  /%MEMORY PARITY ERRORa/
1775 011634 054522 050040 051101
1776 011642 052111 020131 051105
1777 011650 047522 040122
1778
1779 011654 000000 000000 000000          OUTBUF: .EVEN .WORD  0,0,0,0
1780 011662 000000
1781 011664 000000 000000 000000          INBUF:  .WORD  0,0,0,0
1782 011672 000000
1783 011674 000000          DATA:  0
1784          .=.+100
1785
1786          .END
000001

```


ERR123	007272	1319#
ERR124	007322	1325#
ERR125	007442	1347#
ERR126	007542	1362#
ERR127	007606	1370#
ERR13	001774	452#
ERR14	002044	464#
ERR15	002104	474#
ERR16	002154	486#
ERR17	002220	498#
ERR2	001330	354#
ERR20	002306	522#
ERR21	002362	534#
ERR22	002420	542#
ERR23	002462	550#
ERR24	002510	556#
ERR25	002552	564#
ERR26	002620	584#
ERR27	002664	594#
ERR3	001360	364#
ERR30	002724	602#
ERR31	002766	610#
ERR32	003014	616#
ERR33	003056	624#
ERR34	003150	652#
ERR35	003224	664#
ERR36	003266	673#
ERR37	003330	681#
ERR4	001410	373#
ERR40	003356	687#
ERR41	003420	695#
ERR42	003466	713#
ERR43	003532	723#
ERR44	003602	733#
ERR45	003644	741#
ERR46	003672	747#
ERR47	003734	755#
ERR5	001440	383#
ERR50	004026	782#
ERR51	004102	794#
ERR52	004140	802#
ERR53	004202	810#
ERR54	004230	816#
ERR55	004272	824#
ERR56	004340	842#
ERR57	004404	852#
ERR6	001470	392#
ERR60	004444	860#
ERR61	004506	868#
ERR62	004534	874#
ERR63	004576	882#
ERR64	004676	908#
ERR65	004730	914#
ERR66	005016	940#
ERR67	005060	949#
ERR7	001526	402#

ERR70	005114	955#			
ERR71	005214	983#			
ERR72	005246	989#			
ERR73	005276	995#			
ERR74	005352	1012#			
ERR75	005410	1019#			
ERR76	005444	1027#			
ERR77	005504	1034#			
FLAG	001040	285#			
HEDS	011540	1556	1578	1752#	
HEDSA	011536	1751#			
INBUF	011664	1263	1537	1781#	
INCBUS	002642	588#	591		
INCDA	006670	1242	1246#		
INCNEH	005564	1052#	1058		
INCRD	004362	846#	849		
INCMAT	002326	526#	530		
INT106	006130	1099	1118#		
INT110	006260	1132	1151#		
INT112	006410	1166	1184#		
INT114	006532	1199	1213#		
INXDSK	005036	944#	947		
LADT	007346	1302	1315	1321	1327
LATST	007114	212	1291#	1332	1330#
LA1	007130	1294#	1304		
LA2	007176	1299	1303#		
LA3	007206	1306#	1329		
LA4	007256	1312	1316#		
LA5	007310	1318	1322#		
LA6	007340	1323	1328#		
LOGICA	007662	1382#			
LOOP	010156	315	1451#		
LPAOT	006712	1247	1250#		
LPAOT2	007100	1283	1286#		
LPNEH	006024	1086	1091#		
LPST1	001312	343	346#		
LPST10	001616	414	418#		
LPST11	001672	428	431#		
LPST12	001742	439	442#		
LPST13	002006	450	454#		
LPST14	002056	462	466#		
LPST15	002116	473	476#		
LPST16	002166	485	488#		
LPST17	002232	496	500#		
LPST2	001342	353	356#		
LPST22	003074	622	627#		
LPST3	001372	363	366#		
LPST4	001422	372	375#		
LPST40	003752	753	758#		
LPST5	001452	382	385#		
LPST56	004614	880	885#		
LPST6	001502	391	394#		
LPST7	001540	401	404#		
LPST74	004746	912	917#		
LPST77	005132	953	958#		
LPSX10	005312	993	998#		

XSTAI1	001272	339	342#											
XXX	010674	1583	1586#											
.	= 011776	186#	187#	192#	239#	260#	1107	1140	1174	1207	1231	1251	1267	1287
		1296	1309	1331	1359	1420	1423	1426	1452	1456	1459	1587	1621	1679
		1784#												
.BIT0	010144	1404#	1412	1415*	1416*	1425	1441#							
.BIT1	010146	1406#	1416	1442#										
.BIT13	010150	1413*	1414*	1422	1443#									
.BIT14	010152	1410*	1414	1444#										
.BIT15	010154	1411*	1412*	1413	1415	1419	1445#							

ERROR	187*	340	344	354	364	373	383	392	402	416	429	440	452	464	474
	486	498	521	533	541	549	555	563	583	593	601	609	615	623	651
	663	672	680	686	694	712	722	732	740	746	754	781	793	801	809
	815	823	841	851	859	867	873	881	908	914	940	949	955	983	989
	995	1012	1019	1027	1034	1059	1062	1069	1075	1081	1087	1111	1118	1144	1151
	1178	1184	1208	1235	1243	1271	1279	1300	1313	1319	1325	1347	1362	1370	

ADD	1412	1414	1416	1462	1603	1642	1661	1665							
ASL	1529	1530	1531	1532	1533	1534	1535	1702							
BCC	1703														
BEQ	339	353	363	372	382	391	401	414	428	439	450	462	473	485	496
	547	554	562	608	614	622	678	685	693	739	745	753	807	814	822
	866	872	880	912	953	1033	1074	1080	1086	1242	1247	1251	1278	1283	1287
	1323	1331	1343	1369	1381	1420	1423	1426	1452	1459	1516	1583	1587	1612	1614
	1626														
BIC	1418	1477	1664	1667	1677										
BIS	321	532	579	647	662	708	792	837	936	976	1015	1104	1137	1171	1204
	1421	1424	1427	1539	1734										
BIT	338	400	986	1067	1079	1085	1250	1286	1330	1342	1350	1376	1419	1422	1425
	1451	1515	1582	1586	1617										
BMI	528	589	658	718	788	847	904	945	979	993	1011	1024	1054	1056	1109
	1142	1176													
BNE	343	530	591	660	670	720	730	790	849	906	947	981	987	1026	1058
	1068	1107	1140	1174	1207	1304	1329	1337	1351	1375	1377	1429	1431	1456	1543
	1597	1607	1618	1623	1645	1672	1679	1741							
BPL	519	539	581	598	649	668	710	727	779	799	839	856	938	1018	1231
	1233	1267	1269	1296	1299	1309	1312	1318	1344	1359	1361	1518	1541	1621	
BR	524	536	654	666	784	796	910	942	985	991	997	1014	1021	1029	1036
	1061	1065	1071	1113	1123	1146	1156	1180	1189	1211	1237	1245	1273	1281	1349
	1366	1373	1485	1489	1493	1497	1501	1508	1519	1544	1559	1598	1616	1629	1635
	1649	1722	1731												
CLR	308	319	425	448	471	494	513	525	587	643	655	716	773	785	845
	902	943	977	1006	1022	1046	1051	1101	1105	1134	1138	1168	1172	1201	1205
	1228	1264	1292	1293	1305	1338	1339	1354	1397	1608	1719	1720			
CMP	413	427	438	461	484	546	561	607	621	669	677	692	729	738	752
	806	821	865	879	911	952	1032	1073	1121	1154	1187	1213	1241	1246	1277
	1282	1322	1328	1368	1374	1542	1596	1644	1678						
CMPB	1606	1611	1613	1622											
DEC	1428	1430	1458	1627	1671										
EMT	1378	1555	1577	1596	1647	1715	1742								
HALT	187	1588	1723	1730	1735										
INC	526	590	656	719	786	848	905	946	980	1025	1057	1106	1139	1173	1206
	1239	1275	1306	1336	1740										
IOT	181														
JMP	188	193	194	195	196	197	198	199	200	201	203	204	206	207	209
	210	212	213	214	216	218	220	221	222	223	224	225	227	229	231
	233	236	243	244	245	246	247	248	249	250	544	552	558	566	586
	596	604	612	618	626	675	683	689	697	715	725	735	743	749	757
	804	812	818	826	844	854	862	870	876	884	916	951	957	1078	1084
	1090	1249	1252	1285	1288	1302	1315	1321	1327	1332	1386	1584	1680	1744	
JSR	322	341	345	355	365	374	384	393	403	417	430	441	453	465	475
	487	499	523	535	543	551	557	565	585	595	603	611	617	625	653
	665	674	682	688	696	714	724	734	742	748	756	783	795	803	811
	817	825	843	853	861	869	875	883	909	915	941	950	956	984	990
	996	1013	1020	1028	1035	1060	1064	1070	1077	1083	1089	1112	1122	1145	1155
	1179	1188	1210	1236	1244	1272	1280	1301	1314	1320	1326	1348	1365	1367	1372
	1382	1547	1551	1565	1569	1573	1609	1615	1632	1634	1728				
MOV	306	307	309	310	311	312	313	314	315	316	317	318	320	337	340
	344	352	354	362	364	371	373	381	383	390	392	399	402	410	411
	412	415	416	423	424	426	429	435	436	437	440	447	451	452	459
	460	463	464	470	472	474	481	482	483	486	493	497	498	514	515
	516	517	520	522	531	534	537	540	542	545	548	550	553	556	559
	560	564	578	582	584	592	594	599	600	602	605	606	610	613	616

N05

MAINDEC-11-DZRCAB-A
DZRCAB.P11

RC11 STATIC TEST
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

REPLACES DSJA-PB1

MACY11 27(732) 10-SEP-76 11:53 PAGE 67

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DZRCAB, DZRCAB.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DZRCAB.P11
RUN-TIME: 6 13 4 SECONDS
RUN-TIME RATIO: 33/24=1.3
CORE USED: 10K (19 PAGES)

