

RP11-C/RP03

DISKLESS
MD-11-DZRPA-D

EP DZRPA-D-DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50
51	52	53	54	55
56	57	58	59	60
61	62	63	64	65
66	67	68	69	70
71	72	73	74	75
76	77	78	79	80
81	82	83	84	85
86	87	88	89	90
91	92	93	94	95
96	97	98	99	100

CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMNETS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
- 3.0 LOADING PROCEDURE
- 4.0 STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
 - 4.3 PROGRAM AND/OR OPERATOR ACTION
- 5.0 OPERATING PROCEDURE
 - 5.1 OPERATION SWITCH SETTINGS
 - 5.2 SUBROUTINE ABSTRACT
- 6.0 ERRORS
- 7.0 RESTRICTIONS
- 8.0 MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
- 9.0 PROGRAM DESCRIPTION
 - 9.1 DISKLESS TEST
 - 9.2 SWITCH TEST

70
71
72
73
74
75
76
77
78
79
80
81
82

03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

1.0 ABSTRACT

THE RP11C DISKLESS DIAGNOSTIC EXERCISES THE RP11C IN THE MAINTENANCE MODE. IT CONSISTS OF TWO SEGMENTS. THE FIRST SEGMENT VERIFIES THE LOGIC CONTAINED IN THE RP11C BY UTILIZING THE THREE MAINTENANCE REGISTERS WHICH SIMULATE THE SIGNALS PASSING BETWEEN THE RP11C AND THE RPO3. SEGMENT TWO OPERATES IN NORMAL MODE AND IS USED TO VERIFY THE SWITCHES CONTAINED ON THE RP11C AND THE RPO3.

2.0 REQUIREMENTS

2.1 EQUIPMENT

STANDARD PDP-11 CONFIGURATION RP11C DISK CONTROLLER. ONE OR MORE RPO3 DISK DRIVES (SWITCH TEST ONLY).

2.2 STORAGE

PROGRAM REQUIRES 4K OF STORAGE.

2.3 PRELIMINARY PROGRAMS

NONE.

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPE.

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1.1.

138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
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

4.2 STARTING ADDRESS

THE PROGRAM MAY BE STARTED AT ONE OF TWO LOCATIONS.

1. STARTING ADDRESS 200 RUNS THE DISKLESS DIAGNOSTIC. THE RP11C MUST HAVE THE MAINTENANCE SWITCH ENABLED. IF A DRIVE IS ON THE SYSTEM, IT MUST BE POWERED DOWN AND CIRCUIT BREAKER #1 LOCATED TO THE RIGHT OF THE DRIVE'S "SIGNAL IN" CABLE MUST BE TURNED OFF.
2. STARTING ADDRESS 250 RUNS THE RP11C/RP03 SWITCH TEST. AT LEAST ONE DRIVE MUST BE ON LINE.

4.3 PROGRAMS AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY USING THE ABS LOADER.
2. LOAD DESIRED STARTING ADDRESS.
3. SET SWITCHES (SEE SECTION 5.1.1).
4. PRESS START.
5. IF RUNNING THE DISKLESS TEST THE PROGRAM WILL MAKE 100 PASSES THRU THE TEST AND THEN TYPE "END OF PASS 1". THE PROGRAM WILL UPDATE THE PASS COUNT AND CONTINUE LOOPING. AFTER EACH 100 PASSES THE PASSCOUNT WILL BE TYPED OUT.
6. IF RUNNING THE SWITCH TEST, THE PROGRAM TYPES TEST COMPLETE WHEN FINISHED AND THEN HALTS.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AFTER LOADING THE STARTING ADDRESS, SELECT THE DESIRED SWITCHES.

187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242

5.1.1 SWITCH SETTINGS ARE:

- SW<15>=1 HALT ON ERROR
- SW<14>=1 LOOP ON ERROR
- SW<13>=1 INHIBIT PRINTOUT
- SW<12> NOT USED
- SW<11>=1 RING BELL ON ERROR

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED AT THE END OF EACH SUBTEST AND PROVIDES THE ABILITY TO LOOP ON AN ERROR. WHENEVER AN ERROR IS DETECTED, AN ERROR FLAG IS SET. THIS FLAG IS TESTED BY THE SCOPE ROUTINE. IF SET, AND LOOP ON ERROR SW<14> IS SET, THE PROGRAM WILL LOOP BACK AND REPEAT THE CONDITIONS CAUSING THE ERROR. PRIOR TO EACH SCOPE CALL THE LOOP ADDRESS IS MOVED INTO LOCATION LAD. ONCE THE PROGRAM STARTS LOOPING ON AN ERROR, IT WILL CONTINUE LOOPING EVEN THOUGH THE ERROR MAY BE INTERMITTENT. TO GO OUT OF THE LOOP RESET SW<14>.

5.2.2 HLT

THIS ROUTINE IS ENTERED UPON DETECTION OF AN ERROR. IT WILL TYPE THE PC OF THE ERROR AND ADDITIONAL ERROR INFORMATION. THIS ROUTINE TEST FOR HALT ON ERROR, INHIBIT TYPEOUTS, AND RING THE BELL. IT ALSO SETS THE ERROR FLAG USED BY THE SCOPE ROUTINE.

5.2.3 TRAP CATCHER

A "+2" - "HALT" SEQUENCE IS REPEATED FROM 0-776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR +2.

6.0 ERRORS

243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298

6.1 WHEN ERRORS ARE ENCOUNTERED, THE ADDRESS OF THE ERROR ALONG WITH THE CONTENTS OF RPDS, RPER AND RPCS ARE TYPED. BY REFERRING TO THE LISTING ADDITIONAL INFORMATION CAN BE FOUND REGARDING THE CAUSE OF THE ERROR IN THE COMMENTS. WHEN APPROPRIATE, ADDITIONAL INFORMATION IS TYPED OUT SUCH AS EXPECTED AND RECEIVED RESULTS OF AN OPERATION. ALL INFORMATION IS IN OCTAL.

ERROR MESSAGE FORMAT

PC = PC OF FAILURE
RPDS = CONTENTS OF RPDS
RPER = CONTENTS OF RPER
RPCS = CONTENTS OF RPCS

7.0 RESTRICTIONS

NONE.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE PROGRAM WILL MAKE 100 ITERATIONS THRU THE DISKLESS TEST TO COMPLETE ONE PASS. AT THE END OF EACH PASS, THE PASS COUNT IS TYPED OUT AND THE PROGRAM CONTINUES LOOPING.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500.

9.0 PROGRAM DESCRIPTION

9.1 DISKLESS TEST

THIS TEST UTILIZES THE RP11C MAINTENANCE REGISTERS TO CHECK THE LOGICAL RESPONSES OF THE RP11C. THE MAINTENANCE REGISTERS SIMULATE THE SIGNALS PASSING BETWEEN THE RP11C AND THE RPO3. TO RUN THIS TEST

H01

RP11C DISKLESS DIAGNOSTIC
DZRPAD.M01

MACY11 27(732) 16-SEP-76 15:23 PAGE 7

299
300

THE RP11C MAINTENANCE SWITCH MUST BE ENABLED. IF A
DRIVE EXISTS, IT MUST BE POWERED DOWN AND CIRCUIT

301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
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

BREAKER #1 LOCATED JUST TO THE RIGHT OF THE DRIVES'S
"SIGNAL IN" CABLE MUST BE TURNED OFF.

9.2 SWITCH TEST

THIS TEST FUNCTIONALLY CHECKS THE RESPONSES OF THE
SWITCHES FOUND ON THE RP11C AND THE RP03.

%

.LIST ME
.NLIST MC,MD,CND
.ABS
.TITLE FRONT END

;COPYRIGHT 1972,1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS

;GENERAL REGISTER ASSIGNMENTS

000000
000001
000002
000003
000004
000005
000006
000007

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7

;STATUS REGISTER (PSW) BIT ASSIGNMENTS

000001
000002
000004
000010
000020
000340
000300
000240
000200
000140
000100
000040

C=1 ;C BIT
V=2 ;V BIT
Z=4 ;Z BIT
N=10 ;N BIT
T=20 ;T BIT
PRI7=340 ;PRIORITY LEVEL 7
PRI6=300 ;PRIORITY LEVEL 6
PRI5=240 ;PRIORITY LEVEL 5
PRI4=200 ;PRIORITY LEVEL 4
PRI3=140 ;PRIORITY LEVEL 3
PRI2=100 ;PRIORITY LEVEL 2
PRI1=40 ;PRIORITY LEVEL 1

;VECTOR ADDRESSES

000004
000010
000014
000020
000024
000030

ERRVEC=4
RESVEC=10
TBITVEC=14
IOTVEC=20
PFVEC=24
EMTVEC=30

;ERROR VECTOR
;RESERVED INST VECTOR
;T BIT VECTOR
;IOT TRAP VECTOR
;POWER FAIL VECTOR
;EMT VECTOR

```

357      000034          TRAPVEC=34          ;TRAP VECTOR
358
359          ;REGISTER ADDRESSES
360      177776          PSW=177776          ;PROCESSOR STATUS REGISTER
361      177560          TKS=177560          ;KEYBOARD CSR
362      177562          TKB=177562          ;ADDR OF KEYBOARD BUFFER
363      177564          TPS=177564          ;TELEPRINTER CSR
364      177566          TPB=177566          ;TELEPRINTER BUFFER
365      177570          SWR=177570          ;CONSOLE SWITCH REGISTER
366      177570          DISPLAY=177570      ;CONSOLE DISPLAY REGISTER
367
368          ;INITIAL STACK POINTER
369      000500          STKPTR=500          ;PROGRAM STACK POINTER
370
371          ;BIT ASSIGNMENTS
372      100000          B15=100000
373      040000          B14=40000
374      020000          B13=20000
375      010000          B12=10000
376      004000          B11=4000
377      002000          B10=2000
378      001000          B9=1000
379      000400          B8=400
380      000200          B7=200
381      000100          B6=100
382      000040          B5=40
383      000020          B4=20
384      000010          B3=10
385      000004          B2=4
386      000002          B1=2
387      000001          B0=1
388
389          ;MEMORY MANAGEMENT REGISTER ASSIGNMENTS
390
391      177572          SR0=177572
392      172340          KIPAR0=172340
393      172342          KIPAR1=172342
394      172344          KIPAR2=172344
395      172346          KIPAR3=172346
396      172350          KIPAR4=172350
397      172352          KIPAR5=172352
398      172354          KIPAR6=172354
399      172356          KIPAR7=172356
400      172300          KIPDR0=172300
401      172302          KIPDR1=172302
402      172304          KIPDR2=172304
403      172306          KIPDR3=172306
404      172310          KIPDR4=172310
405      172312          KIPDR5=172312
406      172314          KIPDR6=172314
407      172316          KIPDR7=172316
408      000006          RW=6
409      000000          UP=00
410
411          ;INSTRUCTION EQUATES
412

```

```

413      104400      HLT=TRAP      ;HLT IS A TRAP TO THE ERROR ROUTINE
414
415      104000      SCOPE=EMT     ;SCOPE IS AN EMT TRAP
416

```

```

417      ;INDEX OF MACROS
418      .SCOPE
419      .SAVE
420      .REST
421      .ERROR
422      .PRINT
423      .DUMP
424      .RAND
425      .READ
426      .PACK

```

```

427      ;INDEX OF CALLS
428      SCOPE
429      SAVE
430      REST
431      HLT
432      PRINT
433      DUMP
434      DUMPF
435      SDUMP
436      SDUMPF
437      RAND
438      READ
439      PACK

```

```

458
459      .LIST      ME
460      .=200
461      000200    012707    002336    MOV      #START,PC      ;GO TO START OF TEST
462      001000    001000
463      001000    000000    ICNT:      0      ;CONTAINS PASS COUNT
464      001002    000000    LAD:      0      ;PROGRAM TRACE

```

```

465      ;SCOPE (EMT) SERVICE ROUTINE
466      ;THIS ROUTINE WILL LOOP IF AN ERROR OCCURED AND
467      ;LOOP ON ERROR SWITCH IS SET (BIT 14). IF LOOPING IS INDICATED
468      ;THE CONTENTS OF "LAD" EQUAL THE LOOP ADDRESS. IN ORDER

```

```

469 ;TO LOOP ON ERROR, BIT 14 OF THE SWITCH REGISTER MUST BE SET AND
470 ;LOCATION "ERRFLG" MUST BE NEGATIVE INDICATING AN ERROR. ONCE THE
471 ;LOOP IS INITIATED IT WILL CONTINUE UNTIL SWITCH 14 IS CLEARED.
472
473 001004 032737 040000 177570 SCOPES: BIT #B14,@#SWR ;LOOP ON ERROR?
474 001012 001403 BEQ 2$ ;BRANCH IF NO
475 001014 005767 000220 TST ERRFLG ;IS THERE AN ERROR?
476 001020 001003 BNE 1$ ;BRANCH IF YES
477 001022 005067 000212 2$: CLR ERRFLG ;RESET ERROR CONDITION
478 001026 000002 RTI ;EXIT
479 001030 016716 177746 1$: MOV LAD,(SP) ;MODIFY RETURN ADDRESS
480 001034 000002 RTI ;EXIT
481 ;ROUTINE TO SAVE REGISTERS ON THE STACK.
482 ;CALLED BY SAVE MACRO
483 001036 012667 000020 SAVES: MOV (SP)+,1$ ;SAVE RETURN PC
484 001042 010546 MOV R5,-(SP)
485 001044 010446 MOV R4,-(SP)
486 001046 010346 MOV R3,-(SP)
487 001050 010246 MOV R2,-(SP)
488 001052 010146 MOV R1,-(SP)
489 001054 010046 MOV R0,-(SP)
490 001056 016707 000000 MOV 1$,PC ;RETURN
491 001062 000000 1$: 0 ;CONTAINS RETURN ADDRESS
492 ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
493 ;CALLED BY REST MACRO
494 001064 012667 000020 RESTS: MOV (SP)+,1$ ;SAVE RETURN PC
495 001070 012600 MOV (SP)+,R0
496 001072 012601 MOV (SP)+,R1
497 001074 012602 MOV (SP)+,R2
498 001076 012603 MOV (SP)+,R3
499 001100 012604 MOV (SP)+,R4
500 001102 012605 MOV (SP)+,R5
501 001104 016707 000000 MOV 1$,PC ;RETURN
502 001110 000000 1$: 0 ;CONTAINS RETURN ADDR
503 ;ERROR SERVICE ROUTINE CALLED BY HLT
504 ;THIS ROUTINE WILL HALT ON ERROR, RING THE BELL, AND
505 ;TRANSFER CONTROL TO A USER SUPPLIED ROUTINE IF SPECIFIED
506 001112 005737 177570 ERROR: TST @#SWR ;HALT ON ERROR?
507 001116 100001 BPL 3$ ;BRANCH IF NO
508 001120 000000 HALT
509 001122 032737 004000 177570 3$: BIT #B11,@#SWR ;RING THE BELL?
510 001130 001403 BEQ 1$ ;BRANCH IF NO
511 001132 004567 000144 JSR R5,PRNTF$ ;FORCE PRINT THE MESSAGE
512 001136 001250 BELL
513 001140 032737 020000 177570 1$: BIT #B13,@#SWR ;SKIP TYPEOUT?
514 001146 001022 BNE 2$ ;BRANCH IF YES
515 001150 004567 000110 JSR R5,PRINT$ ;PRINT MESSAGE
516 001154 001252 ERRPC
517 001156 011667 000062 MOV (6),HLTADS ;GET ERROR PC+2
518 001162 162767 000002 000054 SUB #2,HLTADS ;MODIFY
519 001170 117767 000050 000044 MOVB @HLTADS,HLTCTS ;SAVE HLT ARGUMENT
520 001176 016767 000042 000356 MOV HLTADS,TTY
521 001204 004767 000134 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
522 001210 004767 012100 JSR PC,MSG ;GO TO USER ERROR ROUTINE
523 001214 005737 177570 2$: TST @#SWR ;HALT ON ERROR?
524 001220 100001 BPL 4$ ;BRANCH IF NO

```

```

525 001222 000000          HALT
526 001224 052767 100000 000006 4$: BIS      #B15,ERRFLG      ;SET ERROR FLAG
527 001232 005267 000010          INC      ERRORS        ;UPDATE ERROR COUNTER
528 001236 000002          RTI
529 001240 000000          ERRFLG: 0
530 001242 000000          HLTCTS: 0
531 001244 000000          HLTADS: 0          ;PC OF ERROR
532 001246 000000          ERRORS: 0          ;ERROR COUNT
533 001250 000007          BELL:  .ASCIZ  <7>
534 001252 005015 005015 041520 ERRPC:  .ASCIZ  <15><12><15><12>'PC= '
535 001260 020075 000
536 001264 001264          .EVEN
537          ;THIS ROUTINE WILL PRINT AN ASCIZ MESSAGE.
538          ;THE MESSAGE MUST TERMINATE IN 0
539 001264 032737 020000 177570 PRINTS: BIT      #B13,0#SWR      ;INHIBIT TYPEOUTS?
540 001272 001403          BEQ      PRNTFS        ;BRANCH IF NO
541 001274 062705 000002          ADD      #2,R5          ;UPDATE RETURN ADDR
542 001300 000205          RTS      R5
543 001302 105737 177564          PRNTFS: TSTB     0#TPS          ;WAIT FOR PRINTER TO FINISH
544 001306 100375          BPL      -4
545 001310 010546          MOV      R5,-(SP)
546 001312 062716 000002          ADD      #2,(SP)        ;ADJUST RETURN PC
547 001316 011505          MOV      (R5),R5        ;GET MESSAGE ADDR
548 001320 105715          1$:  TSTB     (R5)        ;CHECK FOR TERMINATOR
549 001322 001002          BNE      2$
550 001324 012605          MOV      (SP)+,R5        ;GET RETURN ADDR
551 001326 000205          RTS
552 001330 112537 177566          2$:  MOVB     (R5)+,0#TPB      ;PRINT CHARACTER
553 001334 105737 177564          TSTB     0#TPS          ;WAIT TILL DONE
554 001340 100375          BPL      -4
555 001342 000766          BR      1$
556          ;THIS ROUTINE TYPES A LOCATION IN OCTAL
557 001344 032737 020000 177570 PRINTR: BIT      #B13,0#SWR      ;INHIBIT TYPEOUT?
558 001352 001406          BEQ      PRINTA        ;BRANCH IF NO
559 001354 000207          RTS      PC
560 001356 032737 020000 177570 PRINTS: BIT      #B13,0#SWR      ;INHIBIT TYPEOUT?
561 001364 001405          BEQ      PRINTB        ;BRANCH IF NO
562 001366 000207          RTS      PC
563 001370 112767 000001 000140 PRINTA: MOVB     #1,.PR          ;SET ZERO FILL SWITCH
564 001376 000402          BR      .+6           ;SKIP
565 001400 005067 000132          PRINTB: CLR     .PR          ;SUPPRESS LEADING ZEROS
566 001404 112767 177772 000125 .PTIT: MOVB     #-6,.PR+1      ;SET COUNT
567 001412 010446          .PTIT: MOV     R4,-(SP)      ;SAVE R4
568 001414 012704 001540          MOV     #.PR+2,R4        ;SET POINTER TO FIRST CHARACTER
569 001420 105014          CLRB     (R4)          ;CLEAR FIRST BYTE
570 001422 000413          BR      .PRF          ;ROTATE FIRST BIT
571 001424 105014          .PRL: CLRB    (R4)        ;CLEAR BYTE OF CHAR
572 001426 032767 000100 000102 BIT      #100,.PR          ;BIT TYPING MODE
573 001434 001006          BNE     .PRF          ;YES SKIP 2 ROTATES
574 001436 006167 000120          ROL     TTY          ;ROTATE BIT INTO C
575 001442 106114          ROLB    (4)          ;PACK IT
576 001444 006167 000112          ROL     TTY
577 001450 106114          ROLB    (4)
578 001452 006167 000104          .PRF: ROL     TTY
579 001456 106114          ROLB    (4)
580 001460 105714          TSTB    (4)          ;IS IT ZERO

```

581	001462	001402		BEQ	+.6	;SKIP INC
582	001464	105267	000046	INCB	.PR	;SET FILL SWITCH
583	001470	105767	000042	TSTB	.PR	;CHECK FILL SWITCH
584	001474	001402		BEQ	+.6	;SKIP BITSET
585	001476	152724	000060	BISB	#'0,(4)+	;MAKE INTO ASCIZ CHAR
586	001502	105267	000031	INCB	.PR+1	;INC COUNT
587	001506	001346		BNE	.PRL	;REPEAT
588	001510	022704	001540	CMP	#.PR+2,R4	;EMPTY BUFFER
589	001514	001002		BNE	+.6	;SKIP IF NOT
590	001516	112724	000060	MOVB	#'0,(4)+	;LOAD ONE ZERO
591	001522	105014		CLRB	(4)	;NULL TERMINATOR
592	001524	004567	177534	JSR	R5,PRINT\$;PRINT MESSAGE
593	001530	001540		.PR+2		
594	001532	012604		MOV	(SP)+,R4	;RESTORE R4
595	001534	000207		RTS	PC	
596	001536	000012		.BLKW	12	
597	001562	000000				
598	001564					
599	001564	004767	177246	JSR	PC,SAVE\$;SAVE THE REGISTERS
600	001570	016700	000106	MOV	LONUM,R0	;SET R0 WITH LOW
601	001574	016701	000100	MOV	HINUM,R1	;SET R1 WITH HIGH
602	001600	012703	177771	MOV	#-7,R3	;SET SHIFT COUNT
603	001604	005002		CLR	R2	
604	001606	006300		ASL	R0	;SHIFT R0 LEFT AND
605	001610	006101		ROL	R1	;ROTATE CARRY INTO R1 AND
606	001612	006102		ROL	R2	;ROTATE CARRY INTO R2
607	001614	005203		INC	R3	;CHECK FOR DONE
608	001616	001373		BNE	1\$	
609	001620	066702	000056	ADD	LONUM,R2	;ADD # TO MAKE X 129
610	001624	005501		ADC	R1	;PROPOGATE CARRY
611	001626	066701	000046	ADD	HINUM,R1	;ADD # TO MAKE X 129
612	001632	005502		ADC	R2	;PROPOGATE CARRY
613	001634	062700	001057	ADD	#1057,R0	
614	001640	005501		ADC	R1	;PROPOGATE CARRY
615	001642	005502		ADC	R2	;PROPOGATE CARRY
616	001644	062701	047401	ADD	#47401,R1	
617	001650	005502		ADC	R2	
618	001652	062702	000006	ADD	#6,R2	
619	001656	060200		ADD	R2,R0	
620	001660	005501		ADC	R1	
621	001662	010067	000014	MOV	R0,LONUM	
622	001666	010167	000006	MOV	R1,HINUM	
623	001672	004767	177166	JSR	PC,REST\$;RESTORE THE REGISTERS
624	001676	000207		RTS	PC	
625						
626	001700	000000		HINUM:	0	
627	001702	000000		LONUM:	0	
628	001704	010346		READS:	MOV R3,-(6)	;SAVE R3
629	001706	012703	002014	1\$:	MOV #INPUT\$,R3	;GET BUFFER ADDR
630	001712	022703	002034	2\$:	CMP #INPUT\$+20,R3	;BUFFER FULL?
631	001716	001412			BEQ 4\$;YES..TYPE ?
632	001720	105737	177560		TSTB @#177560	;WAIT FOR A CHAR
633	001724	100375			BPL -4	
634	001726	113713	177562		MOVB @#177562,(3)	;GET CHAR
635	001732	142713	000200		BICB #200,(3)	;GET RID OF JUNK
636	001736	122713	000177		CMPB #177,(3)	;IS IT A RUBOUT?

```

637 001742 001004          BNE      3$          ;SKIP IF NO
638 001744          4$:          JSR      RS,PRINTS      ;PRINT MESSAGE
639 001744 004567 177314          BR      1$          ;CLEAR BUFFER AND START OVER
640 001750 002054          READMS          ;ECHO THE CHAR
641 001752 000755          BR      1$
642 001754 013737 177562 177566 3$:  MOV     @#TKB,@#TPB
643 001762 105737 177564          TSTB    @#TPS
644 001766 100375          BPL     .-4          ;WAIT FOR READY
645 001770 122723 000015          CMPB    #15,(3)+    ;CHECK FOR RETURN
646 001774 001346          BNE     2$          ;LOOP IF NOT RETURN
647 001776 105063 177777          CLRB   -1(3)       ;REMOVE THE RETURN
648 002002 004567 177256          JSR     RS,PRINTS      ;PRINT MESSAGE
649 002006 002060          READLS
650 002010 012603          MOV     (6)+,R3     ;RESTORE R3
651 002012 000207          RTS      PC         ;RETURN
652
653 002014 000020          INPUTS: .BLKW 20
654 002054 006477 000012          READMS: .ASCIZ '??'<15><12>
655 002060 000012          READLS: .ASCIZ <12>
656
657          ;TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
658          ;PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER
659
660          PACKS:
661 002062 004767 176750          JSR     PC,SAVES     ;SAVE THE REGISTERS
662 002066 005067 000242          CLR     NUMS
663 002072 005000          CLR     RO
664 002074 105760 002014          2$:    TSTB    INPUTS(RO)
665 002100 001402          BEQ     1$
666 002102 005200          INC     RO
667 002104 000773          BR      2$
668 002106 005300          1$:    DEC     RO
669 002110 004767 000166          JSR     PC,PACS      ;GET OCTAL CHAR
670 002114 016767 000212 000212          MOV     PK$,NUMS     ;PACK FIRST CHAR
671 002122 004767 000154          JSR     PC,PACS      ;GET OCTAL CHAR
672 002126 000241          CLC
673 002130 006167 000176          ROL     PK$
674 002134 006167 000172          ROL     PK$
675 002140 006167 000166          ROL     PK$
676 002144 056767 000162 000162          BIS    PK$,NUMS     ;PACK SECOND CHAR
677 002152 004767 000124          JSR     PC,PACS      ;GET OCTAL CHAR
678 002156 000241          CLC
679 002160 000367 000146          SWAB   PK$
680 002164 006067 000142          ROR     PK$
681 002170 006067 000136          ROR     PK$
682 002174 056767 000132 000132          BIS    PK$,NUMS     ;PACK THIRD CHAR
683 002202 004767 000074          JSR     PC,PACS      ;GET OCTAL CHAR
684 002206 000367 000120          SWAB   PK$
685 002212 000241          CLC
686 002214 006167 000112          ROL     PK$
687 002220 056767 000106 000106          BIS    PK$,NUMS     ;PACK FOURTH CHAR
688 002226 004767 000050          JSR     PC,PACS      ;GET OCTAL CHAR
689 002232 000367 000074          SWAB   PK$
690 002236 000241          CLC
691 002240 006167 000066          ROL     PK$
692 002244 006167 000062          ROL     PK$

```

FRONT END
DZRPAD.M01

MACY11 27(732) 16-SEP-76 15:23 PAGE 15

693	002250	006167	000056		RUL	PKS	
694	002254	006167	000052		ROL	PKS	
695	002260	056767	000046	000046	BIS	PKS, NUMS	;PACK FIFTH CHAR
696	002266	000402			BR	PKEX1\$	
697	002270	062706	000002		ADD	#2, SP	;MODIFY STACK
698	002274				PKEX1\$:		
699	002274	004767	176564		JSR	PC, RESTS	;RESTORE THE REGISTERS
700	002300	000207			RTS	PC	;EXIT
701							
702	002302	005700			PACS:	TST	RO
703	002304	100771				BMI	PKEX\$
704	002306	005067	000020			CLR	PKS
705	002312	116067	002014	000012		MOVB	INPUTS(RO),PKS ;GET INPUT CHAR
706	002320	005300				DEC	RO
707	002322	042767	177770	000002		BIC	#177770,PKS ;CLEAR UNWANTED BITS
708	002330	000207				RTS	PC
709							
710	002332	000000			PKS:	0	
711	002334	000000			NUMS:	0	


```

712 .TITLE RP11C DISKLESS DIAGNOSTIC
713 BACK=.
714 .=250
715 000250 012707 011102 MOV #SWTST,PC
716 002336 .=BACK
717 000254 VECTOR=254
718 000256 STATUS=256
719 002336 012706 000500 START: MOV #STKPTR,SP ;SETUP STACK
720 002342 005067 011250 CLR PASSCT ;CLEAR THE PASSCOUNT
721 002346 005067 176426 LOOP: CLR ICNT ;CLEAR THE ITERATION COUNT
722 002352 004767 010676 JSR PC,INIT ;INITIALIZE VECTORS
723
724 ;TEST LOADING AND READING OF ALL POSSIBLE BITS IN RP11C
725 ;REGISTERS USING A PATTERN OF WALKING 0'S (-2,-3,-5 ETC) AND
726 ;WALKING 1'S (1,2,4,10 ETC) ALSO TEST RESET ON EACH REGISTER
727
728
729 ;RPER-ERROR REGISTER
730
731 ERBT:
732 002356 004567 010430 JSR R5,BITST ;TEST BITS IN REGISTER
733 002362 176656 .WORD 176656 ;MASK=176656
734 002364 176712 .WORD 176712 ;REGISTER=RPER
735 002366 104403 HLT +3 ;INCORRECT DATA RECEIVED FROM RPER
736 002370 000207 RTS PC ;EXIT FROM ERROR CALL
737 002372 012737 176656 176712 2$: MOV #176656,#176712 ;SET MASK BITS IN THE REGISTER
738 002400 000005 RESET ;CLEAR REGISTER RPER
739 002402 005737 176712 TST #176712 ;DID RESET CLEAR THE REGISTER
740 002406 001406 BEQ 1$ ;BRANCH IF YES
741 002410 013767 176712 011070 MOV #176712,RECS
742 002416 005067 011062 CLR EXPS
743 002422 104403 HLT +3 ;REGISTER RPER DID NOT CLEAR WITH RESET
744 002424 012767 002372 176350 1$: MOV #2$,LAD ;SET UP ERROR LOOP ADDR
745 002432 104000 SCOPE
746 002434 012777 176656 011170 ERGO: MOV #176656,#RPER ;SET ALL POSSIBLE BITS IN RPER
747 002442 012777 000001 011164 MOV #1,#RPCS ;CLEAR RP11C
748 002450 017767 011156 011030 MOV #RPER,RECS ;GET CONTENTS OF RPER
749 002456 005767 011024 TST RECS ;DID GO CLEAR RPER?
750 002462 001403 BEQ 1$ ;BRANCH IF YES
751 002464 005067 011014 CLR EXPS
752 002470 104403 HLT +3 ;GO DID NOT CLEAR RPER
753 002472 012767 002434 176302 1$: MOV #ERGO,LAD ;SETUP ERROR LOOP ADDR
754 002500 104000 SCOPE ;LOOP ON TEST IF ERROR
755
756 ;RPCS-CONTROL STATUS REGISTER
757
758 CSBT: JSR R5,BITST ;TEST BITS IN REGISTER
759 002506 037576 .WORD 37576 ;MASK=37576
760 002510 176714 .WORD 176714 ;REGISTER=RPCS
761 002512 104403 HLT +3 ;INCORRECT DATA RECEIVED FROM RPCS
762 002514 000207 RTS PC ;EXIT FROM ERROR CALL
763 002516 012777 037576 011110 2$: MOV #37576,#RPCS ;SET MASK BITS IN REGISTER
764 002524 000005 RESET ;CLEAR THE SYSTEM
765 002526 017767 011102 010752 MOV #RPCS,RECS ;GET CONTENTS OF RPCS
766 002534 042767 000200 010744 BIC #B7,RECS ;CLEAR THE READY BIT
767 002542 001404 BEQ 1$ ;BRANCH IF RESULT IS ZERO

```

```

768 002544 005067 010734          CLR      EXPS
769 002550 104403          HLT      +3          ;RESET DID NOT CLEAR RPCS
770 002552 000405          BR       3$
771 002554 032777 000200 011052 1$:  BIT      #B7,DRPCS   ;IS READY SET?
772 002562 001001          BNE     3$
773 002564 104400          HLT
774 002566 012767 002516 176206 3$:  MOV      #2$,LAD    ;READY NOT SET IN RPCS AFTER RESET
775 002574 104000          SCOPE   ;SETUP LOOP ADDR
776
777          ;RPWC-WORD COUNT REGISTER
778
779 002576          RPBT:
780 002576 004567 010210          JSR      R5,BITST   ;TEST BITS IN REGISTER
781 002602 177777          .WORD   177777     ;MASK=177777
782 002604 176716          .WORD   176716     ;REGISTER=RPWC
783 002606 104403          HLT      +3        ;INCORRECT DATA RECEIVED FROM RPWC
784 002610 000207          RTS     PC         ;EXIT FROM ERROR CALL
785 002612 012737 177777 176716 2$:  MOV      #177777,#176716 ;SET MASK BITS IN THE REGISTER
786 002620 000005          RESET   ;CLEAR REGISTER RPWC
787 002622 005737 176716          TST     @#176716   ;DID RESET CLEAR THE REGISTER
788 002626 001406          BEQ     1$        ;BRANCH IF YES
789 002630 013767 176716 010650          MOV      @#176716,RECS
790 002636 005067 010642          CLR      EXPS
791 002642 104403          HLT      +3        ;REGISTER RPWC DID NOT CLEAR WITH RESET
792 002644 012767 002612 176130 1$:  MOV      #2$,LAD    ;SET UP ERROR LOOP ADDR
793 002652 104000          SCOPE
794
795          ;RPBA- BUS ADDRESS REGISTER
796
797 002654          BABT:
798 002654 004567 010132          JSR      R5,BITST   ;TEST BITS IN REGISTER
799 002660 177777          .WORD   177777     ;MASK=177777
800 002662 176720          .WORD   176720     ;REGISTER=RPBA
801 002664 104403          HLT      +3        ;INCORRECT DATA RECEIVED FROM RPBA
802 002666 000207          RTS     PC         ;EXIT FROM ERROR CALL
803 002670 012737 177777 176720 2$:  MOV      #177777,#176720 ;SET MASK BITS IN THE REGISTER
804 002676 000005          RESET   ;CLEAR REGISTER RPBA
805 002700 005737 176720          TST     @#176720   ;DID RESET CLEAR THE REGISTER
806 002704 001406          BEQ     1$        ;BRANCH IF YES
807 002706 013767 176720 010572          MOV      @#176720,RECS
808 002714 005067 010564          CLR      EXPS
809 002720 104403          HLT      +3        ;REGISTER RPBA DID NOT CLEAR WITH RESET
810 002722 012767 002670 176052 1$:  MOV      #2$,LAD    ;SET UP ERROR LOOP ADDR
811 002730 104000          SCOPE
812
813          ;RPCA-CYLINDER ADDRESS REGISTER
814
815 002732          CABT:
816 002732 004567 010054          JSR      R5,BITST   ;TEST BITS IN REGISTER
817 002736 000777          .WORD   777        ;MASK=777
818 002740 176722          .WORD   176722     ;REGISTER=RPCA
819 002742 104403          HLT      +3        ;INCORRECT DATA RECEIVED FROM RPCA
820 002744 000207          RTS     PC         ;EXIT FROM ERROR CALL
821 002746 012737 000777 176722 2$:  MOV      #777,#176722 ;SET MASK BITS IN THE REGISTER
822 002754 000005          RESET   ;CLEAR REGISTER RPCA
823 002756 005737 176722          TST     @#176722   ;DID RESET CLEAR THE REGISTER

```

```

824 002762 001406          BEQ      1$          ;BRANCH IF YES
825 002764 013767 176722 010514  MOV     @#176722,RECS
826 002772 005067 010506          CLR     EXPS
827 002776 104403          HLT     +3          ;REGISTER RPDA DID NOT CLEAR WITH RESET
828 003000 012767 002746 175774 1$:  MOV     #2$,LAD      ;SET UP ERROR LOOP ADDR
829 003006 104000          SCOPE
830
831          ;RPDA-DISK ADDRESS REGISTER
832
833 003010 004567 007776  DABT:  JSR     R5,BITST  ;TEST BITS IN REGISTER
834 003014 017417          .WORD   17417      ;MASK=17417
835 003016 176724          .WORD   176724      ;REGISTER=RPDA
836 003020 104403          HLT     +3          ;INCORRECT DATA RECEIVED FROM RPDA
837 003022 000207          RTS     PC          ;EXIT FROM ERROR CALL
838 003024 012777 017417 010614 2$:  MOV     #17417,@RPDA ;SET MASK BITS IN RPDA
839 003032 000005          RESET          ;CLEAR THE CONTROLLER
840 003034 017767 010606 010444  MOV     @RPDA,RECS  ;GET CONTENTS OF RPDA
841 003042 042767 000360 010436  BIC     #360,RECS  ;CLEAR UNWANTED BITS
842 003050 001403          BEQ     1$          ;BRANCH IF RESULT IS ZERO
843 003052 005067 010426          CLR     EXPS
844 003056 104403          HLT     +3          ;RESET DID NOT CLEAR RPDA
845 003060 012767 003024 175714 1$:  MOV     #2$,LAD      ;SETUP LOOP ADDR
846 003066 104000          SCOPE
847
848          ;WC REGISTER RAPID DATA TEST - A FAILURE HERE MAY NOT REPORT AS A
849          ;DROP OR PICKED BIT(S) IN ERROR TYPOUT, THIS INDICATES THAT THE REGISTER
850          ;DOES NOT SETTLE WITH THE PROPER DATA FAST ENOUGH
851
852 003070 004567 010062  RAPWC: JSR     R5,RAPBIS
853 003074 176716          .WORD   176716      ;REGISTER = RPWC
854 003076 104403          HLT     +3          ;INCORRECT DATA FROM RPWC
855 003100 000207          RTS     PC          ;EXIT FROM ERROR CALL
856 003102 005037 176716          CLR     @#176716    ;CLEAR RPWC
857 003106 012767 003070 175666  MOV     #RAPWC,LAD  ;IF LOOPING ON ERROR, SETUP FOR THIS TEST
858 003114 104000          SCOPE
859
860          ;BA REGISTER RAPID DATA TEST - A FAILURE HERE MAY NOT REPORT AS A DROP
861          ;OR PICKED BIT(S) IN ERROR TYPOUT, THIS INDICATES THAT THE REGISTER
862          ;DOES NOT SETTLE WITH THE PROPER DATA FAST ENOUGH.
863
864 003116 004567 010034  RAPBA: JSR     R5,RAPBIS
865 003122 176720          .WORD   176720      ;REGISTER = RPBA
866 003124 104403          HLT     +3          ;INCORRECT DATA FROM RPBA
867 003126 000207          RTS     PC          ;RETURN FROM ERROR CALL
868 003130 005037 176720          CLR     @#176720    ;CLEAR RPBA
869 003134 012767 003116 175640  MOV     #RAPBA,LAD  ;IF LOOPING ON ERROR, SETUP FOR THIS TEST
870 003142 104000          SCOPE
871
872          ;TEST THE ABILITY OF BIT 0 OF RPCS TO CLEAR THE CONTROLLER
873          ;WHEN THE COMMAND BITS ARE 0. ERR SYNC(1) AND
874          ;RESET C SHOULD GENERATE INIT L
875
876 003144 012777 177777 010466  RESWC: MOV     #-1,@RPWC ;SET RPWC
877 003152 004767 010332          JSR     PC,CLRP     ;CLEAR THE RP11C
878 003156 017767 010456 010322  MOV     @RPWC,RECS  ;GET CONTENTS OF RPWC
879 003164 005767 010316          TST     RECS        ;DID INIT L CLEAR RPWC?

```

880	003170	001403			
881	003172	005067	010306		
882	003176	104403			
883	003200	012767	003144	175574	1\$:
884	003206	104000			

BEG	1\$
CLR	EXPS
HLT	+3
MOV	#RESWC,LAD
SCOPE	

```

;BRANCH IF YES
;CLEAR EXPECTED RESULTS
;INIT L DID NOT CLEAR RPWC

```

```

885 .SBTTL ***RPDS FUNCTION TEST***
886
887 ;TEST THE SETTING OF SELECTED UNIT READY
888
889 003210 005077 010444 DSF1: CLR JRPMS
890 003214 005077 010410 CLR JRPDS
891 003220 052777 040000 010432 BIS #B14,JRPM3 ;SET MAINT UNIT READY
892 003226 032777 100000 010374 BIT #B15,JRPDS ;DID SELECTED UNIT READY SET?
893 003234 001001 BNE IS ;BRANCH IF SET
894 003236 104400 HLT ;SELECTED UNIT READY NOT SET
895 003240 012767 003210 175534 1$: MOV #DSF1,LAD
896 003246 104000 SCOPE
897
898 ;TEST SELECTED UNIT READY TO CLEAR
899
900 003250 005077 010404 DSF2: CLR JRPMS
901 003254 052777 040000 010376 BIS #B14,JRPM3 ;SET MAINT UNIT READY
902 003262 042777 040000 010370 BIC #B14,JRPM3 ;CLEAR MAINT UNIT READY
903 003270 032777 100000 010332 BIT #B15,JRPDS ;DID SELECTED UNIT READY CLEAR?
904 003276 001401 BEQ IS
905 003300 104400 HLT ;SELECTED UNIT READY DID NOT CLEAR
906
907 003302 012767 003250 175472 1$: MOV #DSF2,LAD
908 003310 104000 SCOPE
909
910 ;TEST THE SETTING OF SEEK INCOMPLETE
911
912
913 003312 000005 DSF4: RESET
914 003314 005077 010340 CLR JRPMS
915 003320 005077 010304 CLR JRPDS
916 003324 052777 002000 010326 BIS #B10,JRPM3 ;SET MAINT. SEEK INCOMPLETE
917 003332 032777 004000 010270 BIT #B11,JRPDS ;DID SEEK INCOMPLETE SET?
918 003340 001002 BNE IS
919 003342 104400 HLT ;SEEK INCOMPLETE DID NOT SET
920 003344 000421 BR 2$
921 003346 032777 000001 010256 1$: BIT #B0,JRPER ;DID DISK ERROR SET?
922 003354 001002 BNE 3$ ;BRANCH IF YES
923 003356 104400 HLT ;DISK ERROR DID NOT SET AFTER SUSI
924 003360 000413 BR 2$
925 003362 032777 040000 010244 3$: BIT #B14,JRPCS ;DID HARD ERROR SET?
926 003370 001002 BNE 4$ ;BRANCH IF SET
927 003372 104400 HLT ;HARD ERROR DID NOT SET AFTER SUSI
928 003374 000405 BR 2$
929 003376 032777 100000 010230 4$: BIT #B15,JRPCS ;DID ERROR SET?
930 003404 001001 BNE 2$ ;BRANCH IF SET
931 003406 104400 HLT ;ERROR DID NOT SET AFTER SUSI
932 003410 012767 003312 175364 2$: MOV #DSF4,LAD
933 003416 104000 SCOPE
934
935 ;TEST SEEK INCOMPLETE TO CLEAR
936
937 003420 005077 010234 DSF5: CLR JRPMS
938 003424 005077 010200 CLR JRPDS
939 003430 052777 002000 010222 BIS #B10,JRPM3 ;SET MAINT SEEK INCOMPLETE
940 003436 042777 002000 010214 BIC #B10,JRPM3 ;CLEAR
  
```

```

941 003444 032777 004000 010156 BIT #B11,ARPDS ;DID SEEK INCOMPLETE CLEAR?
942 003452 001401 BEQ 1$ ;BRANCH IF CLEAR
943 003454 104400 HLT ;SEEK INCOMPLETE DID NOT CLEAR
944 003456 012767 003420 175316 1$: MOV #DSF5,LAD
945 003464 104000 SCOPE
946
947 ;TEST THE SETTING OF FILE UNSAFE
948
949 003466 005077 010166 DSF7: CLR ARPMS
950 003472 005077 010132 CLR ARPDS
951 003476 052777 004000 010154 BIS #B11,ARPMS ;SET MAINT. FILE UNSAFE
952 003504 032777 001000 010116 BIT #B9,ARPDS ;DID FILE UNSAFE SET?
953 003512 001001 BNE 1$
954 003514 104400 HLT ;FILE UNSAFE DID NOT SET
955 003516 012767 003466 175256 1$: MOV #DSF7,LAD
956 003524 104000 SCOPE
957
958 ;TEST CLEARING OF FILE UNSAFE
959
960 003526 005077 010126 DSF10: CLR ARPMS
961 003532 005077 010072 CLR ARPDS
962 003536 052777 004000 010114 BIS #B11,ARPMS ;SET MAINT FILE UNSAFE
963 003544 042777 004000 010106 BIC #B11,ARPMS ;CLEAR
964 003552 032777 001000 010050 BIT #B9,ARPDS ;IS FILE UNSAFE CLEAR?
965 003560 001401 BEQ 1$
966 003562 104400 HLT ;FILE UNSAFE DID NOT CLEAR
967 003564 012767 003526 175210 1$: MOV #DSF10,LAD
968 003572 104000 SCOPE
969
970 ;TEST THE SETTING OF WRITE PROTECT
971
972 003574 005077 010060 DSF11: CLR ARPMS
973 003600 005077 010024 CLR ARPDS
974 003604 052777 100000 010046 BIS #B15,ARPMS ;SET MAINT READ ONLY
975 003612 032777 000400 010010 BIT #B8,ARPDS ;DID WRITE PROTECT SET?
976 003620 001001 BNE 1$
977 003622 104400 HLT ;WRITE PROTECT DID NOT SET
978 003624 012767 003574 175150 1$: MOV #DSF11,LAD
979 003632 104000 SCOPE
980
981 ;TEST THE CLEARING OF WRITE PROTECT
982
983 003634 005077 010020 DSF12: CLR ARPMS
984 003640 005077 007764 CLR ARPDS
985 003644 052777 100000 010006 BIS #B15,ARPMS ;SET MAINT READ ONLY
986 003652 042777 100000 010000 BIC #B15,ARPMS ;CLEAR
987 003660 032777 000400 007742 BIT #B8,ARPDS ;DID WRITE PROTECT CLEAR?
988 003666 001401 BEQ 1$
989 003670 104400 HLT ;WRITE PROTECT DID NOT CLEAR
990 003672 012767 003634 175102 1$: MOV #DSF12,LAD
991 003700 104000 SCOPE
992
993 ;TEST SETTING OF ATTENTION BITS
994
995 003702 012767 000001 007574 DSF13: MOV #1,EXPS ;INITIATE ATTENTION BIT PATTERN
996 003710 005077 007740 2$: CLR ARPMS

```

997	003714	005077	007740			CLR	QRPMS	
998	003720	012777	000377	007702		MOV	#377, QRPDS	; CLEAR ATTENTION BITS
999	003726	016777	007552	007720		MOV	EXPS, QRPMS	; SET MAINT ATTENTION BIT
1000	003734	126777	007544	007666		CMPB	EXPS, QRPDS	; DID THE ATTN BIT SET IN RPDS?
1001	003742	001404				BEQ	1\$; BRANCH IF OK
1002	003744	117767	007660	007534		MOVB	QRPDS, RECS	
1003	003752	104401				HLT	+1	; ATTENTION BIT DID NOT SET PROPERLY
1004	003754	012767	003710	175020	1\$:	MOV	#2\$, LAD	
1005	003762	104000				SCOPE		
1006	003764	016777	007514	007636	4\$:	MOV	EXPS, QRPDS	; CLEAR ATTENTION BIT
1007	003772	105777	007632			TSTB	QRPDS	; DID IT CLEAR?
1008	003776	001401				BEQ	3\$; BRANCH IF CLEAR
1009	004000	104400				HLT		; ATTENTION BIT DID NOT CLEAR
1010	004002	012767	003764	174772	3\$:	MOV	#4\$, LAD	
1011	004010	104000				SCOPE		
1012	004012	000241				CLC		
1013	004014	006167	007464			ROL	EXPS	; ROTATE PATTERN
1014	004020	032767	000400	007456		BIT	#88, EXPS	; END OF PATTERN?
1015	004026	001730				BEQ	2\$; BRANCH IF NO
1016	004030	012777	000377	007616	6\$:	MOV	#377, QRPMS	; SET ATTENTION BITS
1017	004036	000005				RESET		
1018	004040	105777	007564			TSTB	QRPDS	; DID RESET CLEAR ATTN BITS?
1019	004044	001401				BEQ	5\$	
1020	004046	104400				HLT		; RESET DID NOT CLEAR ATTENTION BITS
1021	004050	012767	004030	174724	5\$:	MOV	#6\$, LAD	
1022	004056	104000				SCOPE		
1023	004060	005077	007570		8\$:	CLR	QRPMS	
1024	004064	012777	000377	007562		MOV	#377, QRPMS	; SET ALL ATTENTION BITS
1025	004072	005077	007532			CLR	QRPDS	; ISSUE CLEAR RPDS
1026	004076	122777	000377	007524		CMPB	#377, QRPDS	; DID ATTENTION BITS REMAIN SET?
1027	004104	001410				BEQ	7\$; BRANCH IF YES
1028	004106	005067	007372			CLR	EXPS	
1029	004112	005067	007370			CLR	RECS	
1030	004116	117767	007506	007362		MOVB	QRPDS, RECS	; GET BAD DATA
1031	004124	104403				HLT	+3	; ATTENTION BITS CLEARED WITH A ZERO
1032	004126	012767	004060	174646	7\$:	MOV	#8\$, LAD	
1033	004134	104000				SCOPE		

```

1034 .SBTTL ***RPER FUNCTION TEST***
1035
1036 ;TEST THE SETTING OF WRITE PROTECT VIOLATION
1037
1038 004136 000005 ERF1: RESET
1039 004140 052777 100000 007512 BIS #B15,DRPM3 ;SET MAINT READ ONLY
1040 004146 012777 177777 007464 MOV #-1,DRPWC
1041 004154 112777 000003 007452 MOVB #3,DRPCS ;ISSUE A WRITE
1042 004162 012767 160001 007434 MOV #B15+B14+B13+B0,TPL
1043 004170 004767 007366 JSR PC,T3P ;GENERATE 3 TIMING PULSES
1044 004174 032777 100000 007430 BIT #B15,DRPER ;DID WRITE VIOLATION SET?
1045 004202 001002 BNE 1$
1046 004204 104400 HLT ;WRITE VIOLATION DID NOT SET
1047 004206 000413 BR 2$
1048
1049 004210 032777 040000 007416 1$: BIT #B14,DRPCS ;DID HARD ERROR SET AFTER WPV?
1050 004216 001002 BNE 3$
1051 004220 104400 HLT ;HARD ERROR DID NOT SET AFTER WPV
1052 004222 000405 BR 2$
1053 004224 032777 100000 007402 3$: BIT #B15,DRPCS ;DID ERROR SET AFTER WPV?
1054 004232 001001 BNE 2$
1055 004234 104400 HLT ;ERROR DID NOT SET AFTER WPV
1056 004236 012767 004136 174536 2$: MOV #ERF1,LAD
1057 004244 104000 SCOPE
1058
1059 ;TEST THE SETTING OF FILE UNSAFE VIOLATION
1060
1061 004246 000005 ERF2: RESET
1062 004250 005077 007356 CLR DRPER
1063 004254 052777 004000 007376 BIS #B11,DRPM3 ;SET MAINT FILE UNSAFE
1064 004262 012777 177777 007350 MOV #-1,DRPWC
1065 004270 112777 000003 007336 MOVB #3,DRPCS ;ISSUE WRITE
1066 004276 012767 064001 007320 MOV #B14+B13+B11+B0,TPL
1067 004304 004767 007252 JSR PC,T3P ;GENERATE 3 TIMING PULSES
1068 004310 032777 040000 007314 BIT #B14,DRPER ;DID FILE UNSAFE VIOLATION SET?
1069 004316 001002 BNE 1$
1070 004320 104400 HLT ;FILE UNSAFE VIOLATION DID NOT SET
1071 004322 000413 BR 2$
1072 004324 032777 040000 007302 1$: BIT #B14,DRPCS ;DID HARD ERROR SET AFTER FUV
1073 004332 001002 BNE 3$
1074 004334 104400 HLT ;HARD ERROR DID NOT SET AFTER FUV
1075 004336 000405 BR 2$
1076 004340 032777 100000 007266 3$: BIT #B15,DRPCS ;DID ERROR SET
1077 004346 001001 BNE 2$
1078 004350 104400 HLT ;ERROR DID NOT SET
1079 004352 012767 004246 174422 2$: MOV #ERF2,LAD
1080 004360 104000 SCOPE
1081
1082 ;NON-EXISTENT CYLINDER SHOULD NOT SET WITH LEGAL ADDRESSES
1083
1084 004362 005067 007116 ERF3: CLR EXPS ;START AT ADDR 0
1085 004366 004767 007116 2$: JSR PC,CLRP ;CLEAR THE CONTROLLER
1086 004372 005077 007234 CLR DRPER
1087 004376 016777 007102 007240 MOV EXPS,DRPCA ;LOAD CYLINDER ADDR
1088 004404 012777 177777 007226 MOV #-1,DRPWC
1089 004412 012777 000003 007214 MOV #3,DRPCS ;ISSUE A WRITE
  
```


1090	004420	012767	060001	007176	MOV	#B14+B13+80, TPL	
1091	004426	004767	007130		JSR	PC, T3P	; GENERATE 3 TIMING PULSES
1092	004432	032777	020000	007172	BIT	#B13, RPER	; DID NON-EXISTENT CYLINDER SET?
1093	004440	001411			BEQ	1\$	
1094	004442	104400			HLT		; NON-EXISTENT CYLINDER SET ON LEGAL ADDR
1095	004444	004567	174614		JSR	R5, PRINT\$; PRINT MESSAGE
1096	004450	013757			MES10		
1097	004452	016767	007026	175102	MOV	EXPS, TTY	
1098	004460	004767	174660		JSR	PC, PRINTR	; TYPE LOCATION WITH LEADING ZEROS
1099	004464	012767	004366	174310	MOV	#2\$, LAD	
1100	004472	104000			SCOPE		
1101	004474	005267	007004		INC	EXPS	; UPDATE CYLINDER ADDR
1102	004500	022767	000626	006776	CMP	#626, EXPS	; IS ADDR STILL LEGAL?
1103	004506	001327			BNE	2\$; BRANCH IF YES
1104							
1105							; TEST SETTING OF NON-EXISTENT CYLINDER WITH ILLEGAL CYL ADDR
1106							
1107	004510	004767	006774		ERF4: JSR	PC, CLRP	; CLEAR THE CONTROLLER
1108	004514	005077	007112		CLR	RPER	
1109	004520	016777	006760	007116	MOV	EXPS, RPCA	; LOAD CYLINDER ADDR
1110	004526	012777	177777	007104	MOV	#-1, RPPWC	
1111	004534	012777	000003	007072	MOV	#3, RPPCS	; ISSUE WRITE
1112	004542	012767	060001	007054	MOV	#B14+B13+80, TPL	
1113	004550	004767	007006		JSR	PC, T3P	; GENERATE 3 TIMING PULSES
1114	004554	032777	020000	007050	BIT	#B13, RPER	; DID NON-EXISTENT CYL SET?
1115	004562	001012			BNE	1\$	
1116	004564	104400			HLT		; NON-EXISTENT CYL DID NOT SET
1117	004566	004567	174472		JSR	R5, PRINT\$; PRINT MESSAGE
1118	004572	013757			MES10		
1119	004574	016767	006704	174760	MOV	EXPS, TTY	
1120	004602	004767	174536		JSR	PC, PRINTR	; TYPE LOCATION WITH LEADING ZEROS
1121	004606	000413			BR	2\$	
1122	004610	032777	040000	007016	BIT	#B14, RPPCS	; DID HARD ERROR SET AFTER NXC?
1123	004616	001002			BNE	3\$	
1124	004620	104400			HLT		; HARD ERROR DID NOT SET
1125	004622	000405			BR	2\$	
1126	004624	032777	100000	007002	BIT	#B15, RPPCS	; DID ERROR SET AFTER NXC
1127	004632	001001			BNE	2\$	
1128	004634	104400			HLT		; ERROR DID NOT SET
1129	004636	012767	004510	174136	MOV	#ERF4, LAD	
1130	004644	104000			SCOPE		
1131	004646	005267	006632		INC	EXPS	; UPDATE CYLINDER ADDR
1132	004652	022767	000700	006624	CMP	#700, EXPS	; PATTERN EXCEEDED?
1133	004660	001313			BNE	ERF4	
1134							
1135							; NON-EXISTENT TRACK SHOULD NOT SET WITH LEGAL ADDRESSES
1136							
1137	004662	000005			ERF5: RESET		
1138	004664	005067	006614		CLR	EXPS	; STARTING TRACK ADDR OF 0
1139	004670	005077	006752		CLR	RPPDA	
1140	004674	004767	006610		ERL5: JSR	PC, CLRP	; CLEAR THE CONTROLLER
1141	004700	005077	006726		CLR	RPER	
1142	004704	116777	006574	006736	MOV	EXPS, RPPDA1	; LOAD TRACK ADDRESS
1143	004712	012777	177777	006720	MOV	#-1, RPPWC	
1144	004720	012777	000003	006706	MOV	#3, RPPCS	; WRITE
1145	004726	012767	060001	006670	MOV	#B14+B13+80, TPL	

1146	004734	004767	006622		JSR	PC,T3P		;GENERATE 3 TIMING PULSES
1147	004740	032777	010000	006664	BIT	#B12,ARPER		;IS NXT SET?
1148	004746	001411			BEQ	1\$		
1149	004750	104400			HLT			;NXT SET ON LEGAL ADDR
1150	004752	004567	174306		JSR	R5,PRINT\$;PRINT MESSAGE
1151	004756	013757			MES10			
1152	004760	016767	006520	174574	MOV	EXPS,TTY		
1153	004766	004767	174352		JSR	PC,PRINTR		;TYPE LOCATION WITH LEADING ZEROS
1154	004772	012767	004674	174002	1\$:	MOV	#ERL5,LAD	
1155	005000	104000			SCOPE			
1156	005002	005267	006476		INC	EXPS		;INCREMENT TRACK ADDR
1157	005006	022767	000024	006470	CMP	#24,EXPS		;IS ADDR STILL LEGAL?
1158	005014	001327			BNE	ERL5		;BRANCH IF YES
1159								
1160								
1161								
1162								
1163	005016	004767	006466		ERF6:	JSR	PC,CLRP	;CLEAR THE CONTROLLER
1164	005022	005077	006604		CLR	ARPER		
1165	005026	116777	006452	006614	MOVB	EXPS,ARPD1		;LOAD TRACK ADDR
1166	005034	012777	177777	006576	MOV	#-1,ARPWC		
1167	005042	012777	000003	006564	MOV	#3,ARPCS		;WRITE
1168	005050	012767	060001	006546	MOV	#B14+B13+B0,TPL		
1169	005056	004767	006500		JSR	PC,T3P		;GENERATE 3 TIMING PULSES
1170	005062	032777	010000	006542	BIT	#B12,ARPER		;DID NXT SET?
1171	005070	001012			BNE	1\$		
1172	005072	104400			HLT			;NXT DID NOT SET
1173	005074	004567	174164		JSR	R5,PRINT\$;PRINT MESSAGE
1174	005100	013757			MES10			
1175	005102	016767	006376	174452	MOV	EXPS,TTY		
1176	005110	004767	174230		JSR	PC,PRINTR		;TYPE LOCATION WITH LEADING ZEROS
1177	005114	000413			BR	2\$		
1178	005116	032777	040000	006510	1\$:	BIT	#B14,ARPCS	;DID HARD ERROR SET AFTER NXT?
1179	005124	001002			BNE	3\$		
1180	005126	104400			HLT			;HARD ERROR DID NOT SET AFTER NXT
1181	005130	000405			BR	2\$		
1182	005132	032777	100000	006474	3\$:	BIT	#B15,ARPCS	;DID ERROR SET AFTER NXT?
1183	005140	001001			BNE	2\$		
1184	005142	104400			HLT			;ERROR DID NOT SET AFTER NXT
1185	005144	012767	005016	173630	2\$:	MOV	#ERF6,LAD	
1186	005152	104000			SCOPE			
1187	005154	005267	006324		INC	EXPS		;INCREMENT TRACK ADDR.
1188	005160	022767	000040	006316	CMP	#40,EXPS		;END OF PATTERN?
1189	005166	001313			BNE	ERF6		;BRANCH IF NO
1190								
1191								
1192								
1193	005170	005077	006452		ERF7:	CLR	ARPD1	
1194	005174	005067	006304		CLR	EXPS		;STARTING SECTOR ADDR OF 0
1195	005200	004767	006304		ERL7:	JSR	PC,CLRP	;CLEAR THE CONTROLLER
1196	005204	005077	006422		CLR	ARPER		
1197	005210	016777	006270	006430	MOV	EXPS,ARPD1		;LOAD SECTOR ADDR
1198	005216	012777	177777	006414	MOV	#-1,ARPWC		
1199	005224	012777	000003	006402	MOV	#3,ARPCS		;WRITE
1200	005232	012767	060001	006364	MOV	#B14+B13+B0,TPL		
1201	005240	004767	006316		JSR	PC,T3P		;GENERATE 3 TIMING PULSES

;NON-EXISTENT SECTOR SHOULD NOT SET WITH LEGAL SECTOR ADDRESSES

1202	005244	032777	004000	006360		BIT	#B11,ARPER	;DID NXS SET?
1203	005252	001411				BEQ	1\$	
1204	005254	104400				HLT		;NXS SET ON LEGAL ADDR
1205	005256	004567	174002			JSR	R5,PRINT\$;PRINT MESSAGE
1206	005262	013757				MES10		
1207	005264	016767	006214	174270		MOV	EXPS,TTY	
1208	005272	004767	174046			JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS
1209	005276	012767	005200	173476	1\$:	MOV	#ERL7,LAD	
1210	005304	104000				SCOPE		
1211	005306	005267	006172			INC	EXPS	;UPDATE SECTOR ADDR
1212	005312	022767	000012	006164		CMP	#12,EXPS	;IS ADDR STILL LEGAL?
1213	005320	001327				BNE	ERL7	;BRANCH IF YES
1214								
1215								
1216								
1217	005322	004767	006162			ERF10: JSR	PC,CLRP	;CLEAR THE CONTROLLER
1218	005326	005077	006300			CLR	ARPER	
1219	005332	016777	006146	006306		MOV	EXPS,ARPER	;LOAD SECTOR ADDR
1220	005340	012777	177777	006272		MOV	#-1,ARPER	
1221	005346	012777	000003	006260		MOV	#3,ARPCS	;WRITE
1222	005354	012767	060001	006242		MOV	#B14+B13+80,TPL	
1223	005362	004767	006174			JSR	PC,T3P	;GENERATE 3 TIMING PULSES
1224	005366	032777	004000	006236		BIT	#B11,ARPER	;DID NXS SET?
1225	005374	001012				BNE	1\$	
1226	005376	104400				HLT		;NXS DID NOT SET WITH ILLEGAL ADDR
1227	005400	004567	173660			JSR	R5,PRINT\$;PRINT MESSAGE
1228	005404	013757				MES10		
1229	005406	016767	006072	174146		MOV	EXPS,TTY	
1230	005414	004767	173724			JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS
1231	005420	000413				BR	2\$	
1232	005422	032777	040000	006204	1\$:	BIT	#B14,ARPCS	;DID HARD ERROR SET AFTER NXS?
1233	005430	001002				BNE	3\$	
1234	005432	104400				HLT		;HARD ERROR DID NOT SET AFTER NXS
1235	005434	000405				BR	2\$	
1236	005436	032777	100000	006170	3\$:	BIT	#B15,ARPCS	;DID ERROR SET AFTER NXS?
1237	005444	001001				BNE	2\$	
1238	005446	104400				HLT		;ERROR DID NOT SET AFTER NXS
1239	005450	012767	005322	173324	2\$:	MOV	#ERF10,LAD	
1240	005456	104000				SCOPE		
1241	005460	005267	006020			INC	EXPS	;UPDATE ADDRESS
1242	005464	022767	000020	006012		CMP	#20,EXPS	;IS PATTERN EXHAUSTED?
1243	005472	001313				BNE	ERF10	;BRANCH IF NO
1244								
1245								
1246								
1247	005474	005077	006132			ERF11: CLR	ARPER	
1248	005500	004767	006004			JSR	PC,CLRP	;CLEAR RP11
1249	005504	012777	000003	006122		MOV	#3,ARPCS	;ISSUE WRITE
1250	005512	012767	060001	006104		MOV	#B14+B13+80,TPL	
1251	005520	004767	006036			JSR	PC,T3P	;GENERATE 3 TIMING PULSES
1252	005524	032777	002000	006100		BIT	#B10,ARPER	;DID PROGRAM ERROR SET?
1253	005532	001002				BNE	1\$	
1254	005534	104400				HLT		;PROGRAM ERROR DID NOT SET
1255	005536	000413				BR	2\$;WITH WORD COUNT 0
1256	005540	032777	040000	006066	1\$:	BIT	#B14,ARPCS	;IS HARD ERROR SET?
1257	005546	001002				BNE	3\$	


```

1289
1290          .SBTTL  ***RPCS FUNCTION TEST***
1291
1292          ;CHECK ABILITY OF THE ATTENTION FLAGS TO INTERRUPT
1293
1294 005706 012767 000001 005570 CSF1:  MOV    #1,EXPS      ;STARTING TEST PATTERN
1295 005714 005077 005740 2S:    CLR    @RPM3
1296 005720 012737 000340 000256  MOV    @PRI7,@STATUS
1297 005726 012737 006020 000254  MOV    @1S,@VECTOR
1298 005734 004767 005550  JSR    PC,CLRP      ;CLEAR RP11
1299 005740 005037 177776  CLR    @PSW         ;CLEAR PRIORITY LEVEL
1300 005744 052777 020000 005662  BIS    @B13,@RPCS  ;ENABLE ATTENTION INTERRUPT
1301 005752 016777 005526 005674  MOV    EXPS,@RPM2   ;SET ATTENTION BIT AND
1302 005760 000240  NOP                                ;WAIT FOR INTERRUPT
1303 005762 000240  NOP
1304 005764 000240  NOP
1305 005766 012777 000340 172002  MOV    @PRI7,@PSW   ;LOCKOUT INTERRUPTS
1306 005774 104400  HLT                                ;NO INTERRUPT FROM ATTENTION BIT
1307 005776 004567 173262  JSR    RS,PRINTS    ;PRINT MESSAGE
1308 006002 013757  MES10
1309 006004 016767 005474 173550  MOV    EXPS,TTY
1310 006012 004767 173326  JSR    PC,PRINTR    ;TYPE LOCATION WITH LEADING ZEROS
1311 006016 000412  BR    3S
1312 006020 012706 000500 1S:    MOV    @STKPTR,SP   ;RESTORE STACK
1313 006024 016777 005454 005576  MOV    EXPS,@RPDS  ;CLEAR ATTENTION BIT
1314 006032 032777 020000 005574  BIT    @B13,@RPCS  ;DID AEI CLEAR?
1315 006040 001401  BEQ    3S
1316 006042 104400  HLT                                ;ATTENTION INTERRUPT ENABLE DID NOT
1317                                          ;CLEAR WHEN ATTENTION BIT WAS CLEARED
1318 006044 012767 005714 172730 3S:    MOV    @2S,LAD
1319 006052 104000  SCOPE
1320 006054 006167 005424  ROL    EXPS        ;SHIFT TEST PATTERN
1321 006060 032767 000400 005416  BIT    @B8,EXPS    ;PATTERN EXCEEDED?
1322 006066 001712  BEQ    2S          ;BRANCH IF NO
1323
1324
1325          ;TEST FOR TWO ATTENTION INTERRUPTS
1326
1327 006070 012737 000340 000256 TSTAT: MOV    @PRI7,@STATUS
1328 006076 012737 006150 000254  MOV    @1S,@VECTOR
1329 006104 004767 005400  JSR    PC,CLRP      ;CLEAR THE CONTROLLER
1330 006110 005037 177776  CLR    @PSW         ;LOWER PROCESSOR PRIORITY
1331 006114 052777 020000 005512  BIS    @B13,@RPCS  ;ENABLE ATTENTION INTERRUPT
1332 006122 012777 000003 005524  MOV    @3,@RPM2    ;SET ATTENTION BITS
1333 006130 000240  NOP
1334 006132 000240  NOP
1335 006134 000240  NOP
1336 006136 012737 000340 177776  MOV    @PRI7,@PSW   ;RAISE PROCESSOR PRIORITY
1337 006144 104400  HLT                                ;RP11C DID NOT INTERRUPT WITH ATTENTION
1338                                          ;BITS 0 AND 1 SET
1339 006146 000424  BR    2S
1340 006150 012706 000500 1S:    MOV    @STKPTR,SP   ;RESTORE STACK
1341 006154 012737 006220 000254  MOV    @2S,@VECTOR
1342 006162 005037 177776  CLR    @PSW
1343 006166 052777 020000 005440  BIS    @B13,@RPCS  ;ENABLE ATTENTION INTERRUPT
1344 006174 012777 000001 005426  MOV    @1,@RPDS    ;CLEAR ATTENTION BIT ZERO

```

```

1345 006202 000240      NOP
1346 006204 000240      NOP
1347 006206 000240      NOP
1348 006210 012737 000340 177776  MOV      #PRI7,2#PSW      ;RAISE PROCESSOR PRIORITY
1349 006216 104400      HLT          ;ATTENTION BIT 1 DID NOT INTERRUPT
1350 006220 012706 000500      2$: MOV      #STKPTR,SP      ;RESTORE THE STACK
1351 006224 012767 006070 172550  MOV      #TSTAT,LAD
1352 006232 104000      SCOPE
1353
1354
1355      ;TEST INTERRUPT ON READY
1356
1357 006234 000005      CSF2: RESET
1358 006236 012737 006276 000254  MOV      #2$,2#VECTOR      ;RETURN VECTOR
1359 006244 012737 000040 177776  MOV      #PRI1,2#PSW      ;ALLOW INTERRUPTS
1360 006252 052777 000100 005354  BIS      #B6,2RPCS        ;ENABLE INTERRUPT ON READY
1361 006260 000240      NOP
1362 006262 000240      NOP
1363 006264 000240      NOP
1364 006266 012737 000340 177776  MOV      #PRI7,2#PSW      ;LOCKOUT INTERRUPTS
1365 006274 104400      HLT          ;NO READY INTERRUPT
1366 006276 012706 000500      2$: MOV      #STKPTR,SP
1367 006302 012767 006234 172472  MOV      #CSF2,LAD
1368 006310 104000      SCOPE
1369
1370      ;CHECK THAT INTERRUPTS DON'T OCCUR WITHOUT INTERRUPT ENABLES
1371
1372 006312 000005      CSF3: RESET
1373 006314 012737 006352 000254  MOV      #1$,2#VECTOR      ;INTERRUPT VECTOR
1374 006322 112777 000377 005324  MOV      #377,2RPM2        ;SET ATTENTION BITS
1375 006330 005037 177776      CLR      2#PSW            ;ALLOW INTERRUPTS AND
1376 006334 000240      NOP                    ;WAIT AWHILE
1377 006336 000240      NOP
1378 006340 000240      NOP
1379 006342 012737 000340 177776  MOV      #PRI7,2#PSW      ;LOCKOUT INTERRUPTS
1380 006350 000403      BR          2$
1381 006352 012706 000500      1$: MOV      #STKPTR,SP      ;RESTORE THE STACK
1382 006356 104400      HLT          ;INTERRUPT RECEIVED WITHOUT
1383 006360 012767 006312 172414  2$: MOV      #CSF3,LAD      ;ANY CONTROLLER ENABLES
1384 006366 104000      SCOPE
1385
1386      ;TEST INTERRUPT AT PRIORITY LEVEL 2
1387
1388 006370 004767 005114      CSF4: JSR      PC,CLRP      ;CLEAR THE CONTROLLER
1389 006374 012737 006434 000254  MOV      #1$,2#VECTOR      ;SETUP TRAP VECTOR
1390 006402 052777 000100 005224  BIS      #B6,2RPCS        ;ENABLE RF11 INTERRUPT
1391 006410 012737 000100 177776  MOV      #PRI2,2#PSW      ;LOWER PROCESSOR PRIORITY
1392
1393 006416 000240      NOP
1394 006420 000240      NOP
1395 006422 000240      NOP
1396 006424 012737 000340 177776  MOV      #PRI7,2#PSW
1397 006432 104400      HLT          ;NO READY INTERRUPT AT PRIORITY 2
1398 006434 012706 000500      1$: MOV      #STKPTR,SP
1399 006440 012767 006370 172334  MOV      #CSF4,LAD
1400 006446 104000      SCOPE
  
```

```

1401
1402           ;TEST INTERRUPT AT PRIORITY LEVEL 3
1403
1404 006450 004767 005034 CSF5: JSR PC,CLRP ;CLEAR RF11
1405 006454 012737 006514 000254 MOV #1$,J#VECTOR ;SETUP INTERRUPT VECTOR
1406 006462 052777 000100 005144 BIS #B6,JRPCS ;ENABLE READY INTERRUPT
1407 006470 012737 000140 177776 MOV #PRI3,J#PSW ;LOWER PROCESSOR LEVEL
1408 006476 000240 NOP
1409 006500 000240 NOP
1410 006502 000240 NOP
1411 006504 012737 000340 177776 MOV #PRI7,J#PSW
1412 006512 104400 HLT ;NO INTERRUPT AT LEVEL 3
1413 006514 012706 000500 1$: MOV #STKPTR,SP ;RESTORE STACK
1414 006520 012767 006450 172254 MOV #CSF5,LAD
1415 006526 104000 SCOPE
1416
1417           ;TEST INTERRUPT AT PRIORITY LEVEL 4
1418
1419 006530 004767 004754 CSF6: JSR PC,CLRP ;CLEAR RF11
1420 006534 012737 006574 000254 MOV #1$,J#VECTOR ;SETUP INTERRUPT VECTOR
1421 006542 052777 000100 005064 BIS #B6,JRPCS ;ENABLE READY INTERRUPT
1422 006550 012737 000200 177776 MOV #PRI4,J#PSW ;LOWER PROCESSOR LEVEL
1423 006556 000240 NOP
1424 006560 000240 NOP
1425 006562 000240 NOP
1426 006564 012737 000340 177776 MOV #PRI7,J#PSW
1427 006572 104400 HLT ;NO READY INTERRUPT AT LEVEL 4
1428 006574 012706 000500 1$: MOV #STKPTR,SP ;RESTORE STACK
1429 006600 012767 006530 172174 MOV #CSF6,LAD
1430 006606 104000 SCOPE
1431
1432           ;TEST THAT PRIORITY LEVEL 5 LOCKS OUT INTERRUPTS
1433
1434 006610 004767 004674 CSF7: JSR PC,CLRP ;CLEAR RF11
1435 006614 012737 006654 000254 MOV #1$,J#VECTOR ;SETUP INTERRUPT VECTOR
1436 006622 052777 000100 005004 BIS #B6,JRPCS ;ENABLE INTERRUPT ON READY
1437 006630 012737 000240 177776 MOV #PRI5,J#PSW ;SET PRIORITY LEVEL TO 5
1438 006636 000240 NOP
1439 006640 000240 NOP
1440 006642 000240 NOP
1441 006644 012737 000340 177776 MOV #PRI7,J#PSW
1442 006652 000403 BR 2$
1443 006654 012706 000500 1$: MOV #STKPTR,SP ;RESTORE STACK
1444 006660 104400 HLT ;INTERRUPT RECEIVED AT PRIORITY LEVEL 5
1445 006662 012767 006610 172112 2$: MOV #CSF7,LAD
1446 006670 104000 SCOPE
1447
1448           ;TEST THAT PRIORITY LEVEL 6 LOCKS OUT INTERRUPTS
1449
1450 006672 004767 004612 CSF10: JSR PC,CLRP ;CLEAR RF11
1451 006676 012737 006736 000254 MOV #1$,J#VECTOR ;TRAP VECTOR
1452 006704 052777 000100 004722 BIS #B6,JRPCS ;ENABLE READY INTERRUPT
1453 006712 012737 000300 177776 MOV #PRI6,J#PSW ;SET PRIORITY LEVEL TO 6
1454 006720 000240 NOP
1455 006722 000240 NOP
1456 006724 000240 NOP

```

```

1457 006726 012737 000340 177776      MOV      #PRI7,@#PSW
1458 006734 000403                      BR        2$
1459 006736 012706 000500          1$:      MOV      #STKPTR,SP      ;RESTORE STACK
1460 006742 104400                      HLT                      ;INTERRUPT RECEIVED AT LEVEL 6
1461 006744 012767 006672 172030      2$:      MOV      #CSF10,LAD
1462 006752 104000                      SCOPE
1463
1464          ;TEST THAT PRIORITY LEVEL 7 LOCKS OUT INTERRUPTS
1465
1466 006754 004767 004530      CSF11:   JSR      PC,CLRP      ;CLEAR RF11
1467 006760 012737 007004 000254      MOV      #1$,@#VECTOR      ;SETUP VECTOR INTERRUPT
1468 006766 052777 000100 004640      BIS      #B6,@RPCS        ;ENABLE READY INTERRUPT
1469 006774 000240                      NOP
1470 006776 000240                      NOP
1471 007000 000240                      NOP
1472 007002 000403                      BR        2$
1473 007004 012706 000500          1$:      MOV      #STKPTR,SP
1474 007010 104400                      HLT                      ;INTERRUPT RECEIVED AT LEVEL 7
1475 007012 012767 006754 171762      2$:      MOV      #CSF11,LAD
1476 007020 104000                      SCOPE
1477
1478          ;TEST THE CLEARING AND SETTING OF READY (RPCS-B7)
1479
1480 007022 004767 004462      CSF12:   JSR      PC,CLRP      ;CLEAR RF11
1481 007026 012777 000003 004600      MOV      #3,@RPCS        ;ISSUE WRITE
1482 007034 105777 004574                      TSTB     @RPCS            ;IS READY SET?
1483 007040 100002                      BPL      1$              ;BRANCH IF NO
1484 007042 104400                      HLT                      ;READY DID NOT CLEAR WITH GO H
1485 007044 000411                      BR        2$
1486 007046 012767 060001 004550      1$:      MOV      #B14+B13+B0,TPL
1487 007054 004767 004502      JSR      PC,T3P          ;GENERATE 3 TIMING PULSES
1488 007060 105777 004550      TSTB     @RPCS          ;DID READY SET?
1489 007064 100401                      BMI      2$
1490 007066 104400                      HLT
1491 007070 012767 007022 171704      2$:      MOV      #CSF12,LAD
1492 007076 104000                      SCOPE
1493

```



```

1511          .SBTTL  ***RPDA FUNCTION TEST***
1512
1513 007164 004767 004320          DAF1: JSR    PC,CLRP          ;CLEAR RF11
1514 007170 012700 000001          MOV    #1,RO
1515 007174 004767 004372          JSR    PC,INDEXP        ;GENERATE 1 INDEX PULSE
1516 007200 052777 000400 004452  BIS    #88,DRPM3        ;GENERATE 1 SECTOR PULSE
1517 007206 042777 000400 004444  BIC    #88,DRPM3
1518 007214 052777 000400 004436  BIS    #88,DRPM3
1519 007222 042777 000400 004430  BIC    #88,DRPM3        ;FORCE SOT TO COUNT OF 1
1520 007230 012700 000001          MOV    #1,RO
1521 007234 004767 004332          JSR    PC,INDEXP        ;GENERATE 1 INDEX PULSE
1522 007240 052777 000400 004412  BIS    #88,DRPM3
1523 007246 042777 000400 004404  BIC    #88,DRPM3        ;CLEAR SOT
1524 007254 017700 004366          MOV    DRPDA,RO
1525 007260 006000          ROR    RO
1526 007262 006000          ROR    RO
1527 007264 006000          ROR    RO
1528 007266 006000          ROR    RO
1529 007270 042700 177760          BIC    #-17-1,RO        ;CLEAR UNWANTED BITS
1530 007274 005700          TST    RO                ;WAS SOT CLEARED?
1531 007276 001401          BEQ    IS
1532 007300 104400          HLT
1533 007302 012767 007164 171472  IS:   MOV    #DAF1,LAD        ;SOT DID NOT CLEAR WITH INDEX PULSE
1534 007310 104000          SCOPE
1535
1536          ;TEST ABILITY OF SOT TO COUNT
1537
1538 007312 004767 004172          DAF2: JSR    PC,CLRP          ;CLEAR RP11C
1539 007316 005067 004162          CLR    EXPS              ;EXPECT STARTING ADDR OF 0
1540 007322 012700 000001          MOV    #1,RO
1541 007326 004767 004240          JSR    PC,INDEXP        ;GENERATE ONE INDEX PULSE
1542 007332 052777 000400 004320  BIS    #88,DRPM3        ;GENERATE ONE SECTOR PULSE
1543 007340 042777 000400 004312  BIC    #88,DRPM3
1544 007346 017767 004274 004132  MOV    DRPDA,RECS
1545 007354 006067 004126          ROR    RECS
1546 007360 006067 004122          ROR    RECS
1547 007364 006067 004116          ROR    RECS
1548 007370 006067 004112          ROR    RECS
1549 007374 042767 177760 004104  BIC    #-17-1,RECS      ;CLEAR UNWANTED BITS
1550 007402 026767 004076 004076  CMP    EXPS,RECS        ;IS CONTENTS OF SOT CORRECT?
1551 007410 001401          BEQ    IS
1552 007412 104401          HLT    +1                ;CONTENTS OF SOT INCORRECT
1553 007414 012767 007312 171360  IS:   MOV    #DAF2,LAD
1554 007422 104000          SCOPE
1555 007424 005267 004054          INC    EXPS              ;UPDATE TEST ADDR
1556 007430 022767 000012 004046  CMP    #12,EXPS        ;ADDR EXHAUSTED?
1557 007436 001335          BNE    2S                ;BRANCH IF NO
1558
  
```

```

1559 .SBTTL *** SILO MEMORY TEST ***
1560
1561 ;THIS TEST CHECKS THE SILO MEMORY IN MAINTENANCE
1562 ;MODE. IF INREADY IS SET, DATA IS OUTPUT TO REGISTER
1563 ;776736 WHICH IS THE SILO MEMORY. AFTER THE DATA FILTERS
1564 ;THRU THE MEMORY, OUTREADY GOES TRUE. THE DATA IS READ
1565 ;BACK AND COMPARED.
1566
1567 007440 012704 177760 SILO: MOV #-20,R4 ;SET UP ITERATION COUNT FOR SILO TEST
1568 007444 004767 004040 SILO: JSR PC,CLRP ;CLEAR THE RP11C
1569 007450 012767 000001 004026 MOV #1,EXPS ;INITIATE FLOATING ONE PATTERN
1570 007456 005000 CLR R0 ;PATTERN FLOG
1571 007460 000005 1S: RESET
1572 007462 032777 004000 004162 BIT #B11,DRPM1 ;IS INREADY SET?
1573 007470 001002 BNE 6S ;BRANCH IF SET
1574 007472 104400 HLT ;SILO INREADY IS NOT SET AFTER CLEAR
1575 007474 000425 BR 3S
1576 007476 016777 004002 004156 6S: MOV EXPS,DRPSO ;LOAD DATA IN SILO
1577 007504 012701 000060 MOV #60,R1 ;WAIT FOR OUT READY TO SET
1578 007510 005301 8S: DEC R1
1579 007512 001376 BNE 8S
1580 007514 032777 010000 004130 2S: BIT #B12,DRPM1 ;IS OUTREADY SET?
1581 007522 001002 BNE 7S ;BRANCH IF SET
1582 007524 104400 HLT ;SILO OUTREADY IS NOT SET
1583 007526 000410 BR 3S
1584 007530 017767 004126 003750 7S: MOV DRPSO,RECS ;GET DATA BACK FROM SILO
1585 007536 026767 003742 003742 CMP EXPS,RECS ;IS DATA CORRECT?
1586 007544 001401 BEQ 3S
1587 007546 104403 HLT +3 ;DATA READ FROM SILO INCORRECT
1588 007550 012767 007460 171224 3S: MOV #1S,LAD
1589 007556 104000 SCOPE
1590 007560 005700 TST R0 ;ARE WE FLOATING A ONE?
1591 007562 001013 BNE 4S ;BRANCH IF NO
1592 007564 000241 CLC
1593 007566 006167 003712 ROL EXPS ;ROTATE PATTERN
1594 007572 103401 BCS 5S ;BRANCH IF PATTERN EXCEEDED
1595 007574 000731 BR 1S
1596 007576 012700 000001 5S: MOV #1,R0 ;SET PATTERN FLAG
1597 007602 012767 077777 003674 MOV #077777,EXPS ;FLOATING ZERO PATTERN
1598 007610 000723 BR 1S
1599 007612 000241 4S: CLC
1600 007614 006267 003664 ASR EXPS ;SHIFT FLOATING ZERO
1601 007620 052767 100000 003656 BIS #B15,EXPS
1602 007626 103714 BCS 1S
1603 ;ENSURE THAT THE SILO MEMORY CAN HOLD 64 DISCREET NUMBERS
1604 ;AT ONE TIME. AFTER LOADING THE 64 NUMBERS SIGNAL INREADY
1605 ;SHOULD CLEAR INDICATING THE SILO IS FULL. THE SILO IS THEN
1606 ;READ OUT EXPECTING SEQUENTIAL NUMBERS OF 1 THRU 100 OCTAL. AT
1607 ;THIS TIME OUT READY SHOULD CLEAR.
1608
1609 007630 004767 003654 SILO1: JSR PC,CLRP ;CLEAR THE RP11C
1610 007634 012767 000001 003642 MOV #1,EXPS ;INITIAL TEST PATTERN
1611 007642 032777 004000 004002 1S: BIT #B11,DRPM1 ;IS INREADY SET?
1612 007650 001002 BNE 2S ;BRANCH IF SET
1613 007652 104400 HLT ;SILO INREADY SHOULD BE SET
1614 007654 000454 BR 3S

```

1615	007656	016777	003622	003776	2\$:	MOV	EXPS,ARPSO	;LOAD PATTERN IN SILO
1616	007664	005267	003614			INC	EXPS	;UPDATE PATTERN
1617	007670	022767	000101	003606		CMP	#101,EXPS	;IS THE SILO FULL?
1618	007676	001361				BNE	1\$;BRANCH NOT FULL
1619	007700	032777	004000	003744		BIT	#B11,ARPM1	;DID INREADY CLEAR?
1620	007706	001402				BEQ	7\$;BRANCH IF YES
1621	007710	104400				HLT		;SILO INREADY DID NOT CLEAR
1622	007712	000435				BR	3\$;AFTER SILO WAS FILLED
1623	007714	012767	000001	003562	7\$:	MOV	#1,EXPS	;RESET PATTERN
1624	007722	032777	010000	003722	6\$:	BIT	#B12,ARPM1	;IS OUTREADY SET?
1625	007730	001002				BNE	4\$;BRANCH IF SET
1626	007732	104400				HLT		;SILO OUTREADY SHOULD BE SET
1627	007734	000424				BR	3\$	
1628	007736	017767	003720	003542	4\$:	MOV	ARPSO,RECS	;READ SILO MEMORY
1629	007744	026767	003534	003534		CMP	EXPS,RECS	;IS DATA CORRECT?
1630	007752	001402				BEQ	5\$;BRANCH IS EQUAL
1631	007754	104403				HLT	+3	;INCORRECT DATA RECEIVED FROM SILO
1632	007756	000413				BR	3\$	
1633	007760	005267	003520		5\$:	INC	EXPS	;UPDATE EXPECTED PATTERN
1634	007764	022767	000101	003512		CMP	#101,EXPS	;HAVE 6+ WORDS BEEN READ
1635	007772	001353				BNE	6\$;BRANCH IF NO
1636	007774	032777	010000	003650		BIT	#B12,ARPM1	;HAS SILO OUTREADY CLEARED
1637	010002	001401				BEQ	3\$;BRANCH IF CLEAR
1638	010004	104400				HLT		;SILO SHOULD BE EMPTY BUT
1639								;OUTREADY IS STILL SET
1640	010006	012767	007630	170766	3\$:	MOV	#SILO1,LAD	
1641	010014	104000				SCOPE		
1642	010016	005204				INC	R4	;INCREMENT ITERATION COUNT
1643	010020	001402				BEQ	SEEK	
1644	010022	000167	177416			JMP	SILOR	;LOOP 20 TIMES
1645								
1646								
1647								
1648								
1649								
1650	010026	000005				SEEK:	RESET	
1651	010030	005004				CLR	R4	
1652	010032	017767	003614	003446	10\$:	MOV	ARPM1,RECS	;GET CONTENTS OF RPM1
1653	010040	042767	174000	003440		BIC	#174000,RECS	;CLEAR UNWANTED BITS
1654	010046	001404				BEQ	1\$;BRANCH IF RESULT IS ZERO
1655	010050	005067	003430			CLR	EXPS	
1656	010054	104403				HLT	+3	;SOME BUS OUT SIGNALS TO THE
1657	010056	000553				BR	2\$;RPO3 ARE SET AFTER RESET
1658	010060	012777	000376	003556	1\$:	MOV	#376,ARPCA	;LOAD CYCL 376 INTO RPCA
1659	010066	005704				TST	R4	
1660	010070	001403				BEQ	4\$	
1661	010072	012777	000001	003544		MOV	#1,ARPCA	
1662	010100	112777	000017	003542	4\$:	MOVB	#17,ARPDA1	;LOAD TRACK ADDR
1663	010106	005704				TST	R4	;IS THIS FIRST PASS
1664	010110	001403				BEQ	3\$;BRANCH IF YES
1665	010112	112777	000020	003530		MOVB	#20,ARPDA1	;SET HIGH ORDER BIT OF TRACK ADDR
1666	010120	012777	000011	003506	3\$:	MOV	#11,ARPCS	;ISSUE SEEK COMMAND
1667	010126	012767	060001	003470		MOV	#B14+B13+B0,TPL	
1668	010134	012700	000005			MOV	#5,R0	
1669	010140	004767	003400			JSR	PC,TIMEP	;GENERATE 5 TIMING PULSES
1670	010144	017767	003502	003334		MOV	ARPM1,RECS	;GET CONTROL LINER FROM RPM1

;TEST THE OPERATION OF A SEEK COMMAND IN MAINTENANCE MODE.
 ;ISSUE A SEEK COMMAND AND CHECK THE SETTING OF SET CYLINDER
 ;RESET HEAD, SET HEAD SEEK START AND CAR BITS 0 THRU 7.
 ;ALL THESE SIGNALS ARE FOUND ON BUS OUT CONTROL.

1671	010152	042767	174000	003326		BIC	#174000, RECS	; CLEAR UNWANTED BITS
1672	010160	012767	000577	003316		MOV	#577, EXPS	; LOAD EXPECTED VALUE OF RPM1
1673	010166	005704				TST	R4	; IS THIS FIRST PASS
1674	010170	001403				BEQ	5\$; BRANCH IF YES
1675	010172	012767	000600	003304		MOV	#600, EXPS	
1676	010200	026767	003300	003300	5\$:	CMP	EXPS, RECS	; WERE CONTENTS OF RPM1 CORRECT?
1677	010206	001402				BEQ	6\$; BRANCH IF YES
1678	010210	104403				HLT	+3	; THE CONTROL SIGNAL SET CYLINDER
1679	010212	000475				BR	2\$; AND THE CONTENTS OF CAR SHOULD ; BE ON THE BUS OUT LINES
1680								
1681	010214	012700	000004		6\$:	MOV	#4, R0	
1682	010220	004767	003320			JSR	PC, TIMEP	; GENERATE 4 CLOCK PULSES
1683	010224	017767	003422	003254		MOV	DRPM1, RECS	; GET CONTENTS OF RPM1
1684	010232	042767	174000	003246		BIC	#174000, RECS	; CLEAR UNWANTED BITS
1685	010240	012767	002010	003236		MOV	#810+83, EXPS	; LOAD EXPECTED VALUE OF RPM1
1686	010246	026767	003232	003232		CMP	EXPS, RECS	; ARE CONTENTS OF RPM1 CORRECT?
1687	010254	001402				BEQ	7\$; BRANCH IF YES
1688	010256	104403				HLT	+3	; THE CONTROL SIGNAL RESET HEAD
1689	010260	000452				BR	2\$; SHOULD BE ON THE BUS OUT LINES
1690	010262	012700	000004		7\$:	MOV	#4, R0	
1691	010266	004767	003252			JSR	PC, TIMEP	; GENERATE 4 CLOCK PULSES
1692	010272	017767	003354	003206		MOV	DRPM1, RECS	; GET CONTENTS OF RPM1
1693	010300	042767	174000	003200		BIC	#174000, RECS	; CLEAR UNWANTED
1694	010306	012767	001360	003170		MOV	#1360, EXPS	; LOAD EXPECTED VALUE OF RPM1
1695	010314	005704				TST	R4	; IS THIS THE FIRST PASS
1696	010316	001403				BEQ	8\$; BRANCH IF YES
1697	010320	012767	001010	003156		MOV	#1010, EXPS	
1698	010326	026767	003152	003152	8\$:	CMP	EXPS, RECS	; ARE CONTENTS OF RPM1 CORRECT?
1699	010334	001402				BEQ	9\$	
1700	010336	104403				HLT	+3	; THE CONTROL SIGNAL SET HEAD AND
1701	010340	000422				BR	2\$; THE HEAD ADDRESS SHOULD BE ON ; THE BUS OUT LINES
1702								
1703	010342	012700	000004		9\$:	MOV	#4, R0	
1704	010346	004767	003172			JSR	PC, TIMEP	; GENERATE 4 CLOCK PULSES
1705	010352	017767	003274	003126		MOV	DRPM1, RECS	; GET CONTENTS OF RPM1
1706	010360	042767	174000	003120		BIC	#174000, RECS	; CLEAR UNWANTED BITS
1707	010366	012767	002004	003110		MOV	#810+82, EXPS	; LOAD EXPECTED BITS
1708	010374	026767	003104	003104		CMP	EXPS, RECS	; ARE CONTENTS OF RPM1 CORRECT?
1709	010402	001401				BEQ	2\$; BRANCH IS YES
1710	010404	104403				HLT	+3	; THE CONTROL SIGNAL SEEK START
1711								; SHOULD BE ON THE BUS OUT LINES
1712	010406	000005			2\$:	RESET		
1713	010410	012767	010032	170364		MOV	#10\$, LAD	
1714	010416	104000				SCOPE		
1715	010420	005704				TST	R4	; IS THIS FIRST PASS?
1716	010422	001003				BNE	SEEK1	; BRANCH IF NO
1717	010424	005204				INC	R4	
1718	010426	000167	177400			JMP	10\$; MAKE SECOND PASS
1719	010432	000240			SEEK1:	NOP		
1720								
1721								; ISSUE A RESTORE COMMAND AND CHECK THE GENERATION OF
1722								; THE SIGNAL RESTORE ON THE BUS OUT CONTROL LOGIC
1723								
1724	010434	000005			RESTOR:	RESET		
1725	010436	017767	003210	003042	3\$:	MOV	DRPM1, RECS	; GET CONTENTS OF RPM1
1726	010444	042767	174000	003034		BIC	#174000, RECS	; CLEAR UNWANTED BITS

```

1727 010452 001404          BEQ      1$          ;BRANCH IF RESULT IS YES
1728 010454 005067 003024   CLR      EXPS
1729 010460 104403          HLT      +3          ;SOME BUS OUT SIGNALS TO THE
1730 010462 000425          BR       2$          ;RPO3 ARE SET AFTER RESET
1731 010464 012777 000015 003142 1$:   MOV      #15,ARPCS   ;ISSUE HOME COMMAND
1732 010472 012700 000004   MOV      #4,RO
1733 010476 004767 003042   JSR      PC,TIMEP    ;GENERATE 4 TIMING PULSES
1734 010502 017767 003144 002776   MOV      ARPM1,RECS  ;GET CONTROL LINES FROM RPM1
1735 010510 042767 174000 002770   BIC      #174000,RECS ;CLEAR UNWANTED BITS
1736 010516 012767 002100 002760   MOV      #B6+B10,EXPS ;LOAD EXPECTED VALUE OF RPM1
1737 010524 026767 002754 002754   CMP      EXPS,RECS   ;WERE CONTENTS OF RPM1 CORRECT?
1738 010532 001401          BEQ      2$          ;BRANCH IF YES
1739 010534 104403          HLT      +3          ;ISSUED HOME COMMAND-EXPECTED
1740                                     ;RESTORE AND CONTROL TO BE SET
1741                                     ;ON BUS OUT LINES
1742 010536 000005          2$:   RESET
1743 010540 012767 010436 170234   MOV      #3$,LAD
1744 010546 104000          SCOPE
1745
1746                                     ;ISSUE A READ COMMAND WITH NO SEEK IMPLIED AND CHECK THE GENERATION
1747                                     ;OF THE SIGNAL READ IN THE BUS OUT LOGIC
1748
1749 010550 000005          READT: RESET
1750 010552 012777 177777 003060 2$:   MOV      #-1,ARPWC   ;LOAD WORD COUNT
1751 010560 012777 000017 003046   MOV      #17,ARPCS  ;ISSUE READ COMMAND
1752 010566 012700 000004   MOV      #4,RO
1753 010572 004767 002746   JSR      PC,TIMEP
1754 010576 012767 060401 003020   MOV      #B14+B13+B8+B0,TPL
1755 010604 012700 000003   MOV      #3,RO
1756 010610 004767 002730   JSR      PC,TIMEP    ;GENERATE TIMING PULSE
1757 010614 012767 060001 003002   MOV      #B14+B13+B0,TPL
1758 010622 017767 003024 002656   MOV      ARPM1,RECS  ;GET CONTENTS OF RPM1
1759 010630 042767 174000 002650   BIC      #174000,RECS ;CLEAR UNWANTED BITS
1760 010636 012767 002042 002640   MOV      #B10+B5+B1,EXPS ;LOAD EXPECTED VALUE OF RPM1
1761 010644 026767 002634 002634   CMP      EXPS,RECS   ;ARE CONTENTS OF RPM1 CORRECT
1762 010652 001401          BEQ      1$          ;BRANCH IF YES
1763 010654 104403          HLT      +3          ;ISSUED READ WITH NO SEEK COMMAND
1764                                     ;EXPECTED CONTROL, SELECT HEAD, AND READ
1765                                     ;SIGNALS ON THE BUS OUT LINES
1766 010656 000005          1$:   RESET
1767 010660 012767 010552 170114   MOV      #2$,LAD
1768 010666 104000          SCOPE
1769
1770                                     ;ISSUE A WRITE FORMAT COMMAND AND CHECK THE GENERATION OF THE
1771                                     ;WRITE SIGNAL IN THE BUS OUT LOGIC
1772
1773 010670 012777 177777 002742 WRITE: MOV      #-1,ARPWC   ;SET UP WORD COUNT
1774 010676 012777 000001 002742   MOV      #1,ARPOA   ;SELECT SECTOR ONE
1775 010704 012777 014003 002722   MOV      #14003,ARPCS ;ISSUE WRITE FORMAT COMMAND
1776 010712 012700 000023   MOV      #23,RO
1777 010716 004767 002622   JSR      PC,TIMEP    ;GENERATE 19 TIMING PULSES
1778 010722 012700 000001   MOV      #1,RO
1779 010726 004767 002640   JSR      PC,INDEXP   ;GENERATE INDEX PULSE
1780 010732 052777 000400 002720   BIS      #B8,ARPM3   ;GENERATE SECTOR PULSE
1781 010740 042777 000400 002712   BIC      #B8,ARPM3
1782 010746 052777 000400 002704   BIS      #B8,ARPM3   ;GENERATE SECTOR PULSE

```

1783	010754	012700	000003		MOV	#3,RO	
1784	010760	004767	002560		JSR	PC,TIMEP	;GENERATE 3 TIMING PULSES
1785	010764	017767	002662	002514	MOV	3RPM1,RECS	;GET CONTENTS OF RPM1
1786	010772	042767	174000	002506	BIC	#174000,RECS	;CLEAR UNWANTED BITS
1787	011000	012767	002061	002476	MOV	#B10+B5+B4+B0,EXPS	;LOAD EXPECTED VALUE OF RPM1
1788	011006	026767	002472	002472	CMP	EXPS,RECS	;ARE CONTENTS OF RPM1 CORRECT
1789	011014	001401			BEQ	1\$;BRANCH IF YES
1790	011016	104403			HLT	+3	;ISSUED A WRITE FORMAT COMMAND
1791							;AND EXPECTED CONTROL, SELECT HEADS, ERASE,
1792							;AND WRITE SIGNALS ON THE BUS OUT LOGIC
1793	011020	000005			1\$:	RESET	
1794	011022	012767	010670	167752	MOV	#WRITE,LAD	
1795	011030	104000			SCOPE		
1796	011032	005267	167742		INC	ICNT	;INCREMENT ITERATION COUNT
1797	011036	022767	000100	167734	CMP	#100,ICNT	;64 PASSES YET?
1798	011044	001402			BEQ	2\$;BRANCH IF YES
1799	011046	000167	171304		JMP	ERBT	
1800	011052	005267	002540		2\$:	INC	;INCREMENT PASS COUNT
1801	011056	004567	170202		JSR	RS,PRINT\$;PRINT MESSAGE
1802	011062	013666			MES1		
1803	011064	016767	002526	170470	MOV	PASSCT,TTY	
1804	011072	004767	170260		JSR	PC,PRINT\$;TYPE LOCATION-SUPRESS ZEROS
1805	011076	000167	171244		JMP	LOOP	

```

1806 .SBTTL ***SWITCH TEST***
1807
1808 ;THIS TEST IS CALLED WHEN IT IS DESIRED TO CHECK THE RPO3 AND
1809 ;THE RP11C SWITCHES. INITIALLY BEFORE STARTING THE TEST, THE
1810 ;ENABLE-DISABLE SWITCH SHOULD BE ENABLED. THE DRIVE SHOULD HAVE
1811 ;READ-WRITE ENABLED. THE WRITE LOCKOUT SWITCH SHOULD BE CLEARED
1812 ;AND FORMATTING AND MAINTENANCE DISABLED. TO START THE TEST
1813 ;LOAD 250 INTO PC AND START.
1814
1815 011102 012706 000500 SWTST: MOV #STKPTR,SP ;SETUP STACK
1816 011106 004767 002142 JSR PC,INIT ;INITIALIZE VECTORS
1817 011112
1818 011112 004567 170146 1$: JSR R5,PRINTS ;PRINT MESSAGE
1819 011116 012112 MES20 ;GET NUMBER OF DRIVES
1820 011120 004767 170560 JSR PC,READS ;INPUT MESSAGE
1821 011124 004767 170732 JSR PC,PACKS ;CONVERT INPUT TO A NUMBER
1822 011130 005767 171200 TST NUMS ;IS RESPONSE 0?
1823 011134 001766 BEQ 1$ ;BRANCH IF YES
1824 011136 162767 000001 171170 SUB #1,NUMS
1825 011144 022767 000010 171162 CMP #10,NUMS ;IS NUMBER TOO HIGH?
1826 011152 101757 BLOS 1$ ;BRANCH IF YES
1827 011154 016767 171154 002440 MOV NUMS,HUNIT ;SAVE HIGHEST UNIT NO.
1828 011162 005067 002432 CLR UNIT
1829 011166 004567 170072 JSR R5,PRINTS ;PRINT MESSAGE
1830 011172 012170 MES22
1831 011174
1832 011174 004567 170064 5$: JSR R5,PRINTS ;PRINT MESSAGE
1833 011200 012147 MES21 ;HAVE DRIVE DISABLED
1834 011202 016767 002412 170352 MOV UNIT,TTY
1835 011210 004767 170142 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
1836 011214 004767 170464 JSR PC,READS ;INPUT MESSAGE
1837 011220 012777 000001 002406 2$: MOV #1,ARPCS ;CLEAR THE CONTROLLER
1838 011226 116777 002366 002402 MOVB UNIT,ARPCS1 ;SELECT UNIT
1839 011234 032777 040000 002366 BIT #B14,ARPDS ;IS THE UNIT ON LINE?
1840 011242 001402 BEQ 4$ ;BRANCH IF NO
1841 011244 104400 HLT ;SELECTED UNIT ON LINE WITH THE
1842 011246 000425 BR 3$ ;DRIVE DISABLED
1843 011250
1844 011250 004567 170010 4$: JSR R5,PRINTS ;PRINT MESSAGE
1845 011254 012244 MES23 ;HAVE DRIVE ENABLED
1846 011256 016767 002336 170276 MOV UNIT,TTY
1847 011264 004767 170066 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
1848 011270 004767 170410 JSR PC,READS ;INPUT MESSAGE
1849 011274 012777 000001 002332 MOV #1,ARPCS ;CLEAR THE CONTROLLER
1850 011302 116777 002312 002326 MOVB UNIT,ARPCS1 ;RESELECT THE UNIT
1851 011310 032777 040000 002312 BIT #B14,ARPDS ;IS THE UNIT ON LINE?
1852 011316 001001 BNE 3$
1853 011320 104400 HLT ;SELECTED UNIT NOT ON LINE WITH
1854 ;THE DRIVE ENABLED
1855 011322 012767 011220 167452 3$: MOV #2$,LAD
1856 011330 104000 SCOPE
1857 011332 026767 002262 002262 CMP UNIT,HUNIT ;HAVE ALL UNITS BEEN TESTED?
1858 011340 001403 BEQ RDSW ;BRANCH IF YES
1859 011342 005267 002252 INC UNIT ;UPDATE UNIT
1860 011346 000712 BR 5$
1861 011350 000005 RDSW: RESET

```


1862	011352	005067	002242			CLR	UNIT	
1863	011356				5\$:			
1864	011356	004567	167702			JSR	R5,PRINT\$;PRINT MESSAGE
1865	011362	012264				MES24		;HAVE READ ONLY SET
1866	011364	016767	002230	170170		MOV	UNIT,TTY	
1867	011372	004767	167760			JSR	PC,PRINT\$;TYPE LOCATION-SUPRESS ZERCS
1868	011376	004767	170302			JSR	PC,READ\$;INPUT MESSAGE
1869	011402	012777	000001	002224	3\$:	MOV	#1,ARPCS	;CLEAR THE CONTROLLER
1870	011410	116777	002204	002220		MOV	UNIT,ARPCS1	;SELECT THE UNIT
1871	011416	032777	000400	002204		BIT	#88,ARPDS	;IS SELECTED UNIT WRITE PROTECTED
1872	011424	001002				BNE	1\$	
1873	011426	104400				HLT		;SELECTED UNIT WRITE PROTECT DID
1874	011430	000425				BR	2\$;NOT SET WITH READ ONLY SET
1875	011432				1\$:			
1876	011432	004567	167626			JSR	R5,PRINT\$;PRINT MESSAGE
1877	011436	012316				MES25		;CLEAR READ ONLY
1878	011440	016767	002154	170114		MOV	UNIT,TTY	
1879	011446	004767	167704			JSR	PC,PRINT\$;TYPE LOCATION-SUPRESS ZEROS
1880	011452	004767	170226			JSR	PC,READ\$;INPUT MESSAGE
1881	011456	012777	000001	002150		MOV	#1,ARPCS	;CLEAR THE CONTROLLER
1882	011464	116777	002130	002144		MOV	UNIT,ARPCS1	;RESELECT THE UNIT
1883	011472	032777	000400	002130		BIT	#88,ARPDS	;IS SELECTED UNIT WRITE PROTECTED
1884	011500	001401				BEQ	2\$	
1885	011502	104400				HLT		;SELECTED UNIT WRITE PROTECT SET
1886								;WITH DRIVE WRITE ENABLED
1887	011504	012767	011402	167270	2\$:	MOV	#3\$,LAD	
1888	011512	104000				SCOPE		
1889	011514	026767	002100	002100		CMP	UNIT,HUNIT	;HAVE ALL DRIVES BEEN TESTED?
1890	011522	001403				BEQ	4\$;BRANCH IF YES
1891	011524	005267	002070			INC	UNIT	;UPDATE UNIT
1892	011530	000712				BR	5\$	
1893	011532	005067	002062		4\$:	CLR	UNIT	
1894								
1895								;TEST THE SETTING OF WRITE LOCKOUT AND THE BOUNDARY SWITCHES ON THE
1896								;RP11C CONTROL PANEL.
1897								
1898	011536	000005			WRSW:	RESET		
1899	011540	012700	000002			MOV	#2,RO	;INITIATE SWITCH PATTERN
1900	011544	004567	167514			JSR	R5,PRINT\$;PRINT MESSAGE
1901	011550	012352				MES26		;HAVE WRITE LOCKOUT SET
1902	011552	004567	167506			JSR	R5,PRINT\$;PRINT MESSAGE
1903	011556	012425				MES27		
1904	011560	004767	170120			JSR	PC,READ\$;INPUT MESSAGE
1905	011564	005077	002054		3\$:	CLR	ARPCA	
1906	011570	032777	000400	002032		BIT	#88,ARPDS	;IS SELECTED UNIT WRITE PROTECT SET?
1907	011576	001002				BNE	1\$	
1908	011600	104400				HLT		;SELECTED UNIT WRITE PROTECT SHOULD
1909	011602	000410				BR	2\$;BE SET-RPCA EQUAL TO ZERO
1910	011604	012777	000002	002032	1\$:	MOV	#2,ARPCA	;PUT A TWO IN RPCA
1911	011612	032777	000400	002010		BIT	#88,ARPDS	;IS SELECTED UNIT WRITE PROTECTED?
1912	011620	001401				BEQ	2\$	
1913	011622	104400				HLT		;SELECTED UNIT WRITE PROTECT IS SET
1914								;WITH RPCA EQUAL TO TWO.
1915	011624	012767	011564	167150	2\$:	MOV	#3\$,LAD	
1916	011632	104000				SCOPE		
1917								

1974	012074	004567	167164			JSR	RS,PRINTS	;PRINT MESSAGE
1975	012100	012726				MES32		
1976	012102	004567	167156			JSR	RS,PRINTS	;PRINT MESSAGE
1977	012106	012746				MES33		
1978	012110	000000				HALT		
1979	012112	005015	047510	020127	MES20:	.ASCIZ	<15><12>/HOW MANY DRIVES ARE THERE?/	
1980	012120	040515	054516	042040				
1981	012126	044522	042526	020123				
1982	012134	051101	020105	044124				
1983	012142	051105	037505	000				
1984	012147	015	042012	051511	MES21:	.ASCIZ	<15><12>/DISABLE DRIVE /	
1985	012154	041101	042514	042040				
1986	012162	044522	042526	000040				
1987	012170	005015	042522	050123	MES22:	.ASCIZ	<15><12>/RESPOND TO ALL REQUEST WITH CR WHEN READY/	
1988	012176	047117	020104	047524				
1989	012204	040440	046114	051040				
1990	012212	050505	042525	052123				
1991	012220	053440	052111	020110				
1992	012226	051103	053440	042510				
1993	012234	020116	042522	042101				
1994	012242	000131						
1995	012244	005015	047105	041101	MES23:	.ASCIZ	<15><12>/ENABLE DRIVE /	
1996	012252	042514	042040	044522				
1997	012260	042526	000040					
1998	012264	005015	042523	020124	MES24:	.ASCIZ	<15><12>/SET READ ONLY ON DRIVE /	
1999	012272	042522	042101	047440				
2000	012300	046116	020131	047117				
2001	012306	042040	044522	042526				
2002	012314	000040						
2003	012316	005015	046103	040505	MES25:	.ASCIZ	<15><12>/CLEAR READ ONLY ON DRIVE /	
2004	012324	020122	042522	042101				
2005	012332	047440	046116	020131				
2006	012340	047117	042040	044522				
2007	012346	042526	000040					
2008	012352	005015	042523	020124	MES26:	.ASCIZ	<15><12>/SET WRITE LOCKOUT AND CLEAR THE BOUNDARY/	
2009	012360	051127	052111	020105				
2010	012366	047514	045503	052517				
2011	012374	020124	047101	020104				
2012	012402	046103	040505	020122				
2013	012410	044124	020105	047502				
2014	012416	047125	040504	054522				
2015	012424	000						
2016	012425	015	051412	044527	MES27:	.ASCIZ	<15><12>/SWITCHES ON THE RP11C SWITCH PANEL/	
2017	012432	041524	042510	020123				
2018	012440	047117	052040	042510				
2019	012446	051040	030520	041461				
2020	012454	051440	044527	041524				
2021	012462	020110	040520	042516				
2022	012470	000114						
2023	012472	005015	046120	041501	MES28:	.ASCIZ	<15><12>/PLACE THE FOLLOWING VALUES IN THE CYLINDER BOUNDARY SWITCHES:/	
2024	012500	020105	044124	020105				
2025	012506	047506	046114	053517				
2026	012514	047111	020107	040526				
2027	012522	052514	051505	044440				
2028	012530	020116	044124	020105				
2029	012536	054503	044514	042116				

2030	012544	051105	041040	052517	
2031	012552	042116	054522	051440	
2032	012560	044527	041524	042510	
2033	012566	035123	000		
2034	012571	015	044412	051516	MES29: .ASCIZ <15><12>/INSERT /
2035	012576	051105	020124	000	
2036	012603	015	041412	042514	MES30: .ASCIZ <15><12>/CLEAR THE BOUNDARY SWITCHES/
2037	012610	051101	052040	042510	
2038	012616	041040	052517	042116	
2039	012624	051101	020131	053523	
2040	012632	052111	044103	051505	
2041	012640	000			
2042	012641	015	051412	052105	MES31: .ASCIZ <15><12>/SET FOLLOWING VALUE IN THE DRIVE BOUNDARY SWITCHES/
2043	012646	043040	046117	047514	
2044	012654	044527	043516	053040	
2045	012662	046101	042525	044440	
2046	012670	020116	044124	020105	
2047	012676	051104	053111	020105	
2048	012704	047502	047125	040504	
2049	012712	054522	051440	044527	
2050	012720	041524	042510	000123	
2051	012726	005015	042524	052123	MES32: .ASCIZ <15><12>/TEST COMPLETE/
2052	012734	041440	046517	046120	
2053	012742	052105	000105		
2054	012746	005015	042522	042523	MES33: .ASCIZ <15><12>/RESET WRITE LOCKOUT....THANK YOU/
2055	012754	020124	051127	052111	
2056	012762	020105	047514	045503	
2057	012770	052517	027124	027056	
2058	012776	052056	040510	045516	
2059	013004	054440	052517	000	
2060					
2061	013012				.EVEN

```

2062 .SBTTL *** SUBROUTINES ***
2063
2064 ;FLOAT A ONE AND ZERO THRU A DESIGNATED REGISTER
2065
2066 013012 012567 000576 BITST: MOV (R5)+,MASK ;FETCH DATA MASK
2067 013016 012504 MOV (R5)+,R4 ;GET REGISTER UNDER TEST
2068 013020 010567 000602 MOV R5,LERR ;GET ERROR RETURN ADDR
2069 013024 062705 000004 ADD #4,R5 ;MODIFY RETURN ADDR.
2070 013030 012703 000001 MOV #1,R3 ;INITIALIZE DATA PATTERN
2071 013034 004767 000016 BTL1: JSR PC,BTL2 ;OUTPUT FLOATING ZERO
2072 013040 004767 000012 JSR PC,BTL2 ;OUTPUT FLOATING ONE
2073 013044 000241 CLC
2074 013046 006103 ROL R3 ;SHIFT PATTERN
2075 013050 005703 TST R3
2076 013052 001370 BNE BTL1
2077 013054 000205 RTS R5
2078 013056 005103 BTL2: COM R3 ;COMPLIMENT PATTERN
2079 013060 010367 000420 BTL3: MOV R3,EXPS
2080 013064 005167 000524 COM MASK ;AND MASK WITH PATTERN
2081 013070 046767 000520 000406 BIC MASK,EXPS
2082 013076 005167 000512 COM MASK
2083 013102 016714 000376 MOV EXPS,(R4) ;OUTPUT TO REGISTER
2084 013106 011467 000374 MOV (R4),RECS ;INPUT FROM REGISTER
2085 013112 005167 000476 COM MASK
2086 013116 046767 000472 000362 BIC MASK,RECS ;AND MASK AND RECEIVED DATA
2087 013124 005167 000464 COM MASK
2088 013130 026767 000350 000350 CMP EXPS,RECS ;IS DATA CORRECT
2089 013136 001402 BEQ BTN2
2090 013140 004777 000462 JSR PC,ALERR ;GO REPORT THE ERROR
2091 013144 012767 013060 165630 BTN2: MOV #BTL3,LAD
2092 013152 104000 SCOPE
2093 013154 000207 RTS PC
2094
2095 ;RAPID DATA TEST FOR W.C. AND B.A. REGISTERS
2096
2097 013156 012501 RAPBIS: MOV (R5)+,R1 ;GET REGISTER ADDRESS IN R1
2098 013160 012567 000442 MOV (R5)+,LERR ;GET ERROR RETURN ADDRESS
2099 013164 005725 TST (R5)+ ;MODIFY RETURN ADDRESS
2100 013166 012737 013234 000004 MOV #25,2#ERRVEC ;SETUP FOR END MEMORY TRAP
2101 013174 005000 CLR R0 ;SETUP START MEMORY LOCATION
2102 013176 011011 15: MOV (R0),(R1) ;MOVE MEMORY TO REGISTER
2103 013200 021120 CMP (R1),(R0)+ ;TEST FOR PROPER VALUE IN REGISTER
2104 013202 001775 BEQ 15
2105 013204 014067 000274 MOV -(R0),EXPS ;MOVE GOOD DATA TO EXPS FOR TYP0UT
2106 013210 011167 000272 MOV (R1),RECS ;MOVE REGISTER DATA TO RECS FOR ERROR TYP0UT
2107 013214 004777 000406 JSR PC,ALERR ;REPORT ERROR
2108 013220 012767 013176 165554 MOV #15,LAD ;LOOP ADDRESS FOR SCOPE
2109 013226 104000 SCOPE
2110 013230 005720 TST (R0)+ ;NO LOOP CONTINUE TEST
2111 013232 000761 BR 15
2112 013234 022626 25: CMP (SP)+,(SP)+ ;CLEAN OFF STACK
2113 013236 012737 000006 000004 MOV #ERRVEC+2,2#ERRVEC ;RESTORE TRAP CATCHER
2114 013244 012737 000340 177776 MOV #PRI7,2#PSW ;RESTORE NO INTERUPTS
2115 013252 000205 RTS R5 ;RETURN TO CALLER
2116
2117 ;INITALIZE TRAP VECTORS AND LOCKOUT TRAPS

```

```

2118
2119 013254 012767 001112 164552 INIT: MOV #ERROR,34 ;SETUP TRAP VECTOR
2120 013262 012767 000340 164546 MOV #PRI7,36
2121 013270 012767 001004 164532 MOV #SCOPE$,30 ;SETUP EMT VECTOR
2122 013276 012767 000340 164526 MOV #PRI7,32
2123 013304 012737 000340 177776 MOV #PRI7,3#PSW ;LOCKOUT INTERRUPTS
2124 013312 000207 RTS PC
2125
2126 ;TYPE ERROR MESSAGES
2127
2128 013314 032767 000002 165720 MSG: BIT #B1,HLTCTS ;TYPE ENTIRE MESSAGE?
2129 013322 001033 BNE 1$ ;BRANCH IF NO
2130 013324 004567 165734 JSR RS,PRINTS ;PRINT MESSAGE
2131 013330 013740 MES8
2132 013332 004567 165726 JSR RS,PRINTS ;PRINT MESSAGE
2133 013336 013727 MES2A
2134 013340 017767 000264 166214 MOV @RPOS,TTY
2135 013346 004767 165772 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
2136 013352 004567 165706 JSR RS,PRINTS ;PRINT MESSAGE
2137 013356 013705 MES1A
2138 013360 017767 000246 166174 MOV @RPER,TTY
2139 013366 004767 165752 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
2140 013372 004567 165666 JSR RS,PRINTS ;PRINT MESSAGE
2141 013376 013716 MES2
2142 013400 017767 000230 166154 MOV @RPCS,TTY
2143 013406 004767 165732 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
2144 013412 032767 000001 165622 1$: BIT #B0,HLTCTS ;TYPE EXPECTED RECEIVED?
2145 013420 001001 BNE 2$ ;BRANCH IF YES
2146 013422 000207 RTS PC
2147 013424 032767 000002 165610 2$: BIT #B1,HLTCTS
2148 013432 001403 BEQ 3$
2149 013434 004567 165624 JSR RS,PRINTS ;PRINT MESSAGE
2150 013440 014000 MES17
2151 013442 3$:
2152 013442 004567 165616 JSR RS,PRINTS ;PRINT MESSAGE
2153 013446 014020 MES18
2154 013450 016767 000030 166104 MOV EXPS,TTY
2155 013456 004767 165662 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
2156 013462 004567 165576 JSR RS,PRINTS ;PRINT MESSAGE
2157 013466 014033 MES19
2158 013470 016767 000012 166064 MOV RECS,TTY
2159 013476 004767 165642 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
2160 013502 000207 RTS PC
2161 013504 000000 EXP$: 0
2162 013506 000000 RECS: 0
2163
2164 ;ROUTINE TO CLEAR THE RP11C IN MAINT MODE
2165
2166
2167 013510 012700 000003 CLRP: MOV #3,R0
2168 013514 012777 000001 000112 MOV #1,@RPCS
2169 013522 012767 060001 000074 MOV #B14+B13+B0,TPL
2170 013530 004767 000010 JSR PC,TIMEP ;GENERATE 2 TIME PULSES
2171 013534 052777 020000 000116 BIS #B13,@RPM3 ;SET UNIT ON LINE
2172 013542 000207 RTS PC
2173
  
```

```

2174                                     ;GENERATE TIMMING PULSES
2175
2176 013544 016777 000054 000106 TIMEP: MOV    TPL, @RPM3      ;SET SPECIFIED MAINT BITS
2177 013552 005300          DEC    RO
2178 013554 005700          TST    RO                ;DONE?
2179 013556 001372          BNE   TIMEP            ;0=YES
2180 013560 000207          RTS    PC
2181
2182 013562 012700 000003  T3P:  MOV    #3, RO
2183 013566 000167 177752  JMP    TIMEP            ;GENERATE 3 TIMING PULSES
2184
2185                                     ;GENERATE AN INDEX PULSE
2186
2187 013572 052777 010000 000060 INDEXP: BIS    #B12, @RPM3    ;GENERATE AN INDEX PULSE
2188 013600 042777 010000 000052 BIC    #B12, @RPM3
2189 013606 005300          DEC    RO
2190 013610 001370          BNE   INDEXP          ;DONE?
2191 013612 000207          RTS    PC
2192
2193
2194 013614 000000  MASK:  0
2195 013616 000000  PASSCT: 0
2196 013620 000000  UNIT:    0
2197 013622 000000  HUNIT:  0
2198 013624 000000  TPL:    0
2199 013626 000000  LERR:   0
2200 013630 176710  RPDS:   176710
2201 013632 176712  RPER:   176712
2202 013634 176714  RPCS:   176714
2203 013636 176715  RPCS1:  176715
2204 013640 176716  RPWC:   176716
2205 013642 176720  RPBA:   176720
2206 013644 176722  RPCA:   176722
2207 013646 176724  RPDA:   176724
2208 013650 176725  RPDA1:  176725
2209 013652 176726  RPM1:   176726
2210 013654 176730  RPM2:   176730
2211 013656 176731  RPM21:  176731
2212 013660 176732  RPM3:   176732
2213 013662 176736  RPS0:   176736
2214 013664 176734  SUCA:   176734
2215
2216                                     ;MESSAGES USED BY THE PROGRAM
2217
2218 013666 005015 047105 020104 MES1:  .ASCIZ <15><12>/END OF PASS /
2219 013674 043117 050040 051501
2220 013702 020123 000
2221 013705 015 051012 042520 MES1A: .ASCIZ <15><12>/RPER= /
2222 013712 036522 000040
2223 013716 005015 050122 051503 MES2:  .ASCIZ <15><12>/RPCS= /
2224 013724 020075 000
2225 013727 015 051012 042120 MES2A: .ASCIZ <15><12>/RPDS= /
2226 013734 036523 000040
2227 013740 005015 052123 052101 MESB:  .ASCIZ <15><12>/STATUS ERROR/
2228 013746 051525 042440 051122
2229 013754 051117 000
  
```

2230						
2231	013757	015	052012	051505	MES10:	.ASCIZ <15><12>/TEST PATTERN= /
2232	013764	020124	040520	052124		
2233	013772	051105	036516	000040		
2234	014000	005015	047503	050115	MES17:	.ASCIZ <15><12>/COMPARE ERROR/
2235	014006	051101	020105	051105		
2236	014014	047522	000122			
2237						
2238	014020	005015	054105	042520	MES18:	.ASCIZ <15><12>/EXPECTED/
2239	014026	052103	042105	000		
2240	014033	015	051012	041505	MES19:	.ASCIZ <15><12>/RECEIVED /
2241	014040	044505	042526	020104		
2242	014046	000				
2243		014050				.EVEN
2244		000001				.END

KIPAR5=	172352	397#																		
KIPAR6=	172354	398#																		
KIPAR7=	172356	399#																		
KIPDR0=	172300	400#																		
KIPDR1=	172302	401#																		
KIPDR2=	172304	402#																		
KIPDR3=	172306	403#																		
KIPDR4=	172310	404#																		
KIPDR5=	172312	405#																		
KIPDR6=	172314	406#																		
KIPDR7=	172316	407#																		
LAC	001002	464#	479	744*	753*	774*	792*	810*	828*	845*	857*	869*	883*	895*						
		907#	932*	944*	955*	967*	978*	990*	1004*	1010*	1021*	1032*	1056*	1079*						
		1099#	1129*	1154*	1185*	1209*	1239*	1264*	1286*	1318*	1351*	1367*	1383*	1399*						
		1414#	1429*	1445*	1461*	1475*	1491*	1505*	1533*	1553*	1588*	1640*	1713*	1743*						
		1767#	1794*	1855*	1887*	1915*	1943*	1968*	2091*	2108*										
LERR	013626	2068#	2090	2098*	2107	2199#														
LONUM	001702	600	609	621*	627#															
LOOP	002346	721#	1805																	
MASK	013614	2066#	2080*	2081	2082*	2085*	2086	2087*	2194#											
MES1	013666	1802	2218#																	
MES1A	013705	2137	2221#																	
MES10	013757	1096	1118	1151	1174	1206	1228	1308	2231#											
MES17	014000	2150	2234#																	
MES18	014020	2153	2238#																	
MES19	014033	2157	2240#																	
MES2	013716	2141	2223#																	
MES2A	013727	2133	2225#																	
MES20	012112	1819	1979#																	
MES21	012147	1833	1984#																	
MES22	012170	1830	1987#																	
MES23	012244	1845	1995#																	
MES24	012264	1865	1998#																	
MES25	012316	1877	2003#																	
MES26	012352	1901	2008#																	
MES27	012425	1903	2016#																	
MES28	012472	1920	2023#																	
MES29	012571	1923	1958	2034#																
MES30	012603	1953	2036#																	
MES31	012641	1955	2042#																	
MES32	012726	1975	2051#																	
MES33	012746	1977	2054#																	
MES8	013740	2131	2227#																	
MSG	013314	522	2128#																	
N	= 000010	340#																		
NUMS	002334	662*	670*	676*	682*	687*	695*	711#	1822	1824*	1825	1827								
PACKS	002062	660#	1821																	
PACS	002302	669	671	677	683	688	702#													
PASSCT	013616	720*	1800*	1803	2195#															
PC	=%000007	334#	461*	490*	501*	521*	522*	559*	562*	595*	599*	623*	624*	651*						
		661*	669*	671*	677*	683*	688*	699*	700*	708*	715*	722*	736*	762*						
		784*	802*	820*	837*	855*	867*	877*	1043*	1067*	1085*	1091*	1098*	1107*						
		1113*	1120*	1140*	1146*	1153*	1163*	1169*	1176*	1195*	1201*	1208*	1217*	1223*						
		1230*	1248*	1251*	1269*	1274*	1298*	1310*	1329*	1388*	1404*	1419*	1434*	1450*						
		1466*	1480*	1487*	1498*	1513*	1515*	1521*	1538*	1541*	1568*	1609*	1669*	1682*						
		1691*	1704*	1733*	1753*	1756*	1777*	1779*	1784*	1804*	1816*	1820*	1821*	1835*						

RPDS	013630	890*	892	903	915*	917	938*	941	950*	952	961*	964	973*	975
		984*	987	998*	1000	1002	1006*	1007	1018	1025*	1026	1030	1313*	1344*
RPER	013632	1839	1851	1871	1883	1906	1911	1932	1938	1963	2134	2200#		
		746*	748	921	1044	1062*	1068	1086*	1092	1108*	1114	1141*	1147	1164*
RPM1	013652	1170	1196*	1202	1218*	1224	1247*	1252	1270*	1275	2138	2201#		
		1572	1580	1611	1619	1624	1636	1652	1670	1683	1692	1705	1725	1734
RPM2	013654	1758	1785	2209#										
RPM21	013656	996*	999*	1016*	1023*	1024*	1301*	1332*	1374*	2210#				
RPM3	013660	1499*	2211#											
		889*	891*	900*	901*	902*	914*	916*	937*	939*	940*	949*	951*	960*
		962*	963*	972*	974*	983*	985*	986*	997*	1039*	1063*	1295*	1516*	1517*
		1518*	1519*	1522*	1523*	1542*	1543*	1780*	1781*	1782*	2171*	2176*	2187*	2188*
		2212#												
RPS0	013662	1576*	1584	1615*	1628	2213#								
RPWC	013640	876*	878	1040*	1064*	1088*	1110*	1143*	1166*	1198*	1220*	1271*	1750*	1773*
		2204#												
RW	= 000006	408#												
RO	=%000000	327#	489	495*	600*	604*	613*	619*	621	663*	664	666*	668*	702
		705	706*	1514*	1520*	1524*	1525*	1526*	1527*	1528*	1529*	1530	1540*	1570*
		1590	1596*	1668*	1681*	1690*	1703*	1732*	1752*	1755*	1776*	1778*	1783*	1899*
		1925*	1926	1929*	1931	1946*	1947	2101*	2102	2103	2105	2110	2167*	2177*
		2178	2182*	2189*										
R1	=%000001	328#	488	496*	601*	605*	610*	611*	614*	616*	620*	622	1577*	1578*
		2097*	2102*	2103	2106									
R2	=%000002	329#	487	497*	603*	606*	609*	612*	615*	617*	618*	619		
R3	=%000003	330#	486	498*	602*	607*	628	629*	630	650*	2070*	2074*	2075	2078*
		2079												
R4	=%000004	331#	485	499*	567	568*	569*	571*	588	594*	1567*	1642*	1651*	1659
		1663	1673	1695	1715	1717*	2067*	2083*	2084					
R5	=%000005	332#	484	500*	511*	515*	541*	542*	545	547*	548	550*	551*	552
		592*	639*	648*	732*	758*	780*	798*	816*	833*	852*	864*	1095*	1117*
		1150*	1173*	1205*	1227*	1307*	1801*	1818*	1829*	1832*	1844*	1864*	1876*	1900*
		1902*	1919*	1922*	1952*	1954*	1957*	1974*	1976*	2066	2067	2068	2069*	2077*
		2097	2098	2099	2115*	2130*	2132*	2136*	2140*	2149*	2152*	2156*		
SAVES	001036	483#	599	661										
SCOPE	= 104000	415#	745	754	775	793	811	829	846	858	870	884	896	908
		933	945	956	968	979	991	1005	1011	1022	1033	1057	1080	1100
		1130	1155	1186	1210	1240	1265	1287	1319	1352	1368	1384	1400	1415
		1430	1446	1462	1476	1492	1506	1534	1554	1589	1641	1714	1744	1768
		1795	1756	1888	1916	1944	1969	2092	2109					
SCOPE\$	001004	473#	2121											
SEEK	010026	1643	1650#											
SEEK1	010432	1716	1719#											
SILO	007440	1567#												
SILOR	007444	1568#	1644											
SIL01	007630	1609#	1640											
SP	=%000006	333#	479*	483	484*	485*	486*	487*	488*	489*	494	495	496	497
		498	499	500	545*	546*	550	567*	594	697*	719*	1312*	1340*	1350*
		1366*	1381*	1398*	1413*	1428*	1443*	1459*	1473*	1815*	2112			
SRO	= 177572	391#												
START	002336	461	719#											
STATUS	= 000256	718#	1296*	1327*										
STKPTR	= 000500	369#	719	1312	1340	1350	1366	1381	1398	1413	1428	1443	1459	1473
		1815												
SUCA	013664	1500	2214#											
SWR	= 177570	365#	473	506	509	513	523	539	557	560				

ADC	610	612	614	615	617	620									
ADD	541	546	609	611	613	616	618	619	697	1937	2069				
ASL	604														
ASR	1600														
BCS	1594	1602													
BEQ	474	510	540	558	561	581	584	631	665	740	750	757	788	806	824
	842	880	904	942	965	988	1001	1008	1015	1019	1027	1093	1148	1203	1315
	1322	1502	1509	1531	1551	1586	1620	1630	1637	1643	1654	1660	1664	1674	1677
	1687	1696	1699	1709	1727	1738	1762	1789	1798	1823	1840	1858	1884	1890	1912
	1939	1948	1973	2089	2104	2148									
BIC	707	766	841	902	940	963	986	1517	1519	1523	1529	1543	1549	1653	1671
	1684	1693	1706	1726	1735	1759	1781	1786	2081	2086	2188				
BICB	635														
BIS	526	676	682	687	695	891	901	916	939	951	962	974	985	1039	1063
	1300	1331	1343	1360	1390	1406	1421	1436	1452	1468	1516	1518	1522	1542	1601
	1780	1782	2171	2187											
BISB	585														
BIT	473	509	513	539	557	560	572	771	892	903	917	921	925	929	941
	952	964	975	987	1014	1044	1049	1053	1068	1072	1076	1092	1114	1122	1126
	1147	1170	1178	1182	1202	1224	1232	1236	1252	1256	1260	1275	1279	1293	1314
	1321	1508	1572	1580	1611	1619	1624	1636	1839	1851	1871	1883	1906	1911	1932
	1938	1947	1963	1972	2128	2144	2147								
BLOS	1826														
BMI	703	1489													
BNE	476	514	549	573	587	589	608	637	646	772	893	918	922	926	930
	953	976	1045	1050	1054	1069	1073	1077	1103	1115	1123	1127	1133	1158	1171
	1179	1183	1189	1213	1225	1233	1237	1243	1253	1257	1261	1276	1280	1284	1557
	1573	1579	1581	1591	1612	1618	1625	1635	1716	1852	1872	1907	1933	1964	2076
	2129	2145	2179	2190											
BPL	507	524	544	554	633	644	1483								
BR	555	564	570	641	667	696	770	920	924	928	1047	1052	1071	1075	1121
	1125	1177	1181	1231	1235	1255	1259	1278	1282	1311	1339	1380	1442	1458	1472
	1485	1575	1583	1595	1598	1614	1622	1627	1632	1657	1679	1689	1701	1730	1842
	1860	1874	1892	1909	1935	2111									
CLC	672	678	685	690	1012	1592	1599	1924	1928	1945	1970	2073			
CLR	477	565	603	662	663	704	720	721	742	751	768	790	808	826	843
	856	868	881	889	890	900	914	915	937	938	949	950	960	961	972
	973	983	984	996	997	1023	1025	1028	1029	1062	1084	1086	1108	1138	1139
	1141	1164	1193	1194	1196	1218	1247	1270	1295	1299	1330	1342	1375	1497	1539
	1570	1651	1655	1728	1828	1862	1893	1905	2101						
CLRB	569	571	591	647											
CMP	588	630	1102	1132	1157	1188	1212	1242	1501	1550	1556	1585	1617	1629	1634
	1676	1686	1698	1708	1737	1761	1788	1797	1825	1857	1889	2088	2103	2112	
CMPB	636	645	1000	1026											
COM	2078	2080	2082	2085	2087										
DEC	668	706	1578	2177	2189										
EMT	415														
HALT	459	508	525	1978											
INC	527	607	666	1101	1131	1156	1187	1211	1241	1507	1555	1616	1633	1642	1717
	1796	1800	1859	1891											
INCB	582	586													
JMP	1644	1718	1799	1805	2183										
JSR	511	515	521	522	592	599	623	639	648	661	669	671	677	683	688
	699	722	732	758	780	798	816	833	852	864	877	1043	1067	1085	1091
	1095	1098	1107	1113	1117	1120	1140	1146	1150	1153	1163	1169	1173	1176	1195
	1201	1205	1208	1217	1223	1227	1230	1248	1251	1269	1274	1298	1307	1310	1329

RP11C DISKLESS DIAGNOSTIC MACY11 27(732) 16-SEP-76 15:23 PAGE 61
 DZRPAD.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.MACR	411	442	443	444	445	446	447	448	449	450	451	453	454	455	456
	457	719													
.MACRO	452														
.NLIST	319	459													
.PAGE	712	885	1034	1289	1494	1511	1559	1806	2062						
.REM	14														
.REPT	459														
.SBTTL	885	1034	1290	1494	1511	1559	1806	2062							
.TITLE	321	712													
.WORD	733	734	759	760	781	782	799	800	817	818	834	835	853	865	

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*.DZRPAD.SEG/SOL/CRF/PAGNUM+DZRPAD.M01,DZRPAD.P11
 RUN-TIME: 8 14 3 SECONDS
 RUN-TIME RATIO: 141/26=5.2
 CORE USED: 8K (15 PAGES)

