

RK611
RK06, RK07

RK611 DU PORT LGC
CZR6GCO

AH-E034C-MC
FICHE 1 OF 2

APR 1982
COPYRIGHT © 78-82
MADE IN USA



The main body of the document is a large grid of data. Each cell in the grid contains a small, dense table of information, likely representing a detailed technical specification or a data log. The text within these cells is extremely small and difficult to read, but the overall structure is a regular grid of approximately 15 columns and 20 rows of smaller tables.

RK611
RK06, RK07

RK611 DU PORT LGC
CZR6GCO

AH-E034C-MC
FICHE 2 OF 2

APR 1982
COPYRIGHT © 78-82
MADE IN USA



Table with multiple columns and rows of data, likely a technical specification or data log. The text is very faint and difficult to read, but appears to be organized in a grid format.



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

.REM %

IDENTIFICATION

PRODUCT CODE: AC-E033C-MC
PRODUCT NAME: CZR6GCO RK611 DU PORT LGC
DATE: JANUARY 1982
MAINTAINER: STORAGE SYTEMS SOFTWARE TEST APPLICATIONS
AUTHOR: B. T. LEBLANC

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1977,1982 BY DIGITAL EQUIPMENT CORPORATION

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 C 1
PAGE 3

SEQ 0002

REVISION HISTORY

41
42
43
44
45
46
47
48
49
50

REVISION	CHANGES	DATE
CZR6GCO	IMPLEMENTED XXDP LOAD MEDIA OPTION	JAN 82

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

TABLE OF CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
 - 2.1 HARDWARE
 - 2.2 PRELIMINARY TESTING & PROGRAMS
 - 2.3 RESTRICTIONS & OPERATOR ACTIONS
- 3.0 PROGRAM CONSIDERATIONS
 - 3.1 PDP-11 FAMILY COMPATIBILITY
 - 3.2 XXDP
 - 3.3 ACT/APT
 - 3.3.1 APT ETABLE DEFINITIONS
 - 3.4 MEMORY MANAGEMENT
 - 3.5 PARITY CHECK ENABLED
 - 3.6 BAD SECTORS
 - 3.7 EXECUTION TIME
 - 3.8 FAULT ISOLATION
 - 3.9 ERROR CORRECTION & FAILURE RATE ANALYSIS
 - 3.10 DEFAULT UNIBUS ADDRESSES & VECTORS
- 4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS
 - 4.1 PROGRAM LOADING
 - 4.2 STARTING LOCATIONS
 - 4.3 CONSOLE SWITCH REGISTERS
 - 4.4 SOFTWARE SWITCH REGISTER
 - 4.5 INPUT DIALOGUE
 - 4.6 PROGRAM EXAMPLE
 - 4.7 HALTING THE PROGRAM
- 5.0 DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION
 - 5.1 GENERAL
 - 5.2 TEST DESCRIPTIONS
- 6.0 ERROR REPORTING
 - 6.1 ERROR INTERPRETATION
 - 6.2 ERROR PRINTOUT EXAMPLE
- 7.0 DUAL PROCESSOR-DUAL CONTROLLER TESTING

99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154

1.0 ABSTRACT

THE RK06 DUAL PORT LOGIC PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE DUAL PORT OPTION IS FUNCTIONING PROPERLY.

BOTH PORTS OF THE RK06 ARE CABLED TO THE SAME RK611 BY A STANDARD CABLE & THE DUAL PORT TEST SWITCH IS ENABLED ON THE DUAL PORT MODULE. THIS ARRANGEMENT ALLOWS THE DUAL PORT LOGIC TO BE TESTED FROM ONE PDP-11/RK611 TO A MAXIMUM OF 4 DRIVES.

THIS PROGRAM WILL TEST RK06/RK07 DRIVES WITHOUT OPERATOR INPUTS

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

PDP-11
CONSOLE TELETYPE
16K MEMORY
KW11-L OR KW11-P CLOCK
RK06 UNIBUS CONTROLLER (RK611)
1 TO 4 RK06/RK07 DRIVES
FORMATTED DISKPACKS

2.2 PRELIMINARY TESTING & PROGRAMS

1. THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFULLY ON BOTH PORTS.
2. THE RK06 DRIVE DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFULLY.
3. THE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC SHOULD FIRST RUN SUCCESSFULLY.

2.3 RESTRICTIONS & OPERATOR ACTION

TO TEST THE RK06 DUAL PORT OPTION WITH THIS PROGRAM, THE DUAL PORT TEST SWITCH MUST BE ENABLED ON THE DUAL PORT MODULE.

THE CABLE FROM THE RK611 IS DAISY CHAINED TO BOTH PORTS OF RK06 UNDER TEST. ENABLING THE SWITCH ON THE DUAL PORT MODULE, GROUNDS BIT 0 OF THE UNIT SELECT LINES GOING TO THE PORT B INTERFACE & TIMING MODULE (M7706).

THE EFFECT OF THIS IS THAT BIT 0 UNIT SELECT IS COMPLEMENTED

155
156
157
158
159
160
161
162
163
164
165
156
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210

ON PORT B AND THE DRIVE APPEARS AS 2 SEPARATE UNITS TO THE RK611. EACH PORT OF THE RK06 WILL RESPOND TO A DIFFERENT DRIVE ADDRESS.

THE ADDRESS OF EACH PORT WILL DEPEND ON THE DRIVE ADDRESS. FOR THIS REASON, THE RESTRICTION IS MADE THAT ONLY EVEN NUMBERED UNIT SELECT PLUGS BE USED. IN THIS WAY PORT 'A' WILL RESPOND TO THE DRIVE ADDRESS, & PORT 'B' WILL RESPOND TO ADDRESS+1 (THE ADDRESS DEVELOPED BY THE SWITCH).

BECAUSE OF THE ABOVE CONSIDERATIONS, A MAXIMUM OF 4 DRIVES CAN BE TESTED BY THIS PROGRAM, WHICH WILL 'SEE' 8 DRIVES.

ANY OTHER DRIVES ON THE SYSTEM WHICH HAS ANY ADDRESS IN CONFLICT WITH EITHER OF THE TEST ADDRESSES MUST BE DESELECTED, BOTH PORTS SWITCHED OFF.

A FURTHER REQUIREMENT IS THE SYSTEM MUST HAVE EITHER A KW11-P OR KW11-L. IF NEITHER IS PRESENT AN ERROR MESSAGE WILL BE TYPED AND THE PROGRAM WILL JUMP TO THE END OF PASS.

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

THIS PROGRAM CAN BE USED BY THE PDP-11/04,05,10,20, 34,35,40,45,50,55 & 70.

IT IS COMPATABLE WITH THE LSI-11 INSTRUCTION SET AND CAN TEST THE RK06 ONLY IF THE DRIVE CONTROLLER FOR THE LSI-11 IS DESIGNED TO BE DIAGNOSTICALLY COMPATABLE WITH THE RK611.

3.2 XXDP

THIS PROGRAM SHOULD NOT BE CHAINED BY XXDP.

CHAIN MODE OPERATION (MONITOR)

BY DEFINITION, ANY PROGRAM THAT REQUIRES OPERATOR INTERVENTION SHOULD NOT BE CHAINED.

IN THIS CASE, OPERATOR INTERVENTION IS REQ'D TO ENABLE THE DUAL PORT TEST SWITCH & DAISY CHAINING BOTH PORTS TO THE SAME RK611.

DUMP MODE OPERATION (MANUAL)

1. INPUT DIALOGUE IF STARTED FROM 220.
2. IF THE LOADING MEDIUM ON DRIVE 0 IS AN RK06, IT WILL BE

TESTED. THE OPERATOR IS FIRST GIVEN A MESSAGE TO REPLACE
THE PACK ON DRIVE 0 WITH A SCRATCH PACK & TYPE <CR>
WHEN DONE.

3.3 ACT/APT

THIS PROGRAM IS ACT COMPATIBLE.

HOWEVER, IT SHOULD BE RUN ONLY IN DUMP MODE.

AUTOMATIC MODE (MONITOR)

BY DEFINITION ANY PROGRAM THAT REQUIRES OPERATOR INTERVENTION
SHOULD NOT BE RUN IN THE AUTO MODE.

DUMP MODE (MANUAL): INPUT DIALOGUE IF STARTED FROM 220.

3.3.1 APT ETABLE DEFINITIONS

THE FOLLOWING DEFINITIONS ARE VALID FOR SPECIFYING APT ENVIRONMENTAL
TABLE (ETABLE) ENTRIES. VIA RUNNING THE APT UTILITY PROGRAM 'TSP':

1. SOFTWARE ENVIRONMENT:
=1 IF APT SCRIPT MODE
=0 IF STANDALONE MODE
2. ENVIRONMENT MODE:BYTE
BIT 7 = 1 ETABLE DOES SIZING
= 0 PROGRAM DOES SIZING
BIT 6 = 1 SPOOL MESSAGES TO APT IF SCRIPT MODE
= 0 DON'T SPOOL TO APT
BIT 5 = 1 SUPPRESS CONSOLE OUTPUT
= 0 ALLOW CONSOLE OUTPUT
BITS 4-0 NOT USED
3. SWITCH 1 (SOFTWARE SWITCH REGISTER)
IF ENVIRONMENT MODE BIT 7 (SIZING BIT) IS SET TO 1, THE SOFTWARE
SWITCH REGISTER WILL BE USED, INSTEAD OF THE HARDWARE CONSOLE
SWITCH REGISTER. REGARDLESS OF WHICH ONE IS USED, ALL BITS
DEFINED IN SECTIONS 4.3 & 4.4 (SWITCH REGISTER OPTIONS) MAY USED
WHEN RUNNING IN STANDALONE MODE.
IN APT SCRIPT MODE, HOWEVER, BIT 14 (LOOP ON TEST) MUST ALWAYS
BE SET TO 0.
4. SWITCH 2 (USER SWITCH REGISTER)
NOT USED
5. CPU OPTIONS:
NOT USED
6. MEMORY TYPES 1-4 AND MAX MEMORY ADDRESSES
NOT USED

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
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
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322

7. INTERRUPT VECTOR 1:
USED WHEN ENVIRONMENT MODE BIT 7=1. DEFAULT = 210
8. BUS PRIORITY 1:
USED WHEN ENVIRONMENT MODE BIT 7=1. DEFAULT = 5
9. INTERRUPT VECTOR 2:
NOT USED
10. BUS PRIORITY 2:
NOT USED
11. BASE ADDRESS:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 177440
12. DEVICE MAP:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. EACH BIT SET TO
1 IN BITS 0-7 WILL SELECT THE CORRESPONDING DRIVE TO BE
TESTED. BITS 8-15 ARE NOT USED.

NOTE: IN THIS PROGRAM, ONLY EVEN NUMBERED DRIVES CAN BE
TESTED (0,2,4,6)

13. CONTROLLER DESCRIPTORS:
NOT USED.
14. DRIVE DESCRIPTOR CODES (IN WORDS):
NOT USED

3.4 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT USED.

3.5 PARITY CHECK ENABLED

IF THE MEMORY PARITY CHECK OPTION IS AVAILABLE ON THE SYSTEM,
THE PROGRAM WILL RUN WITH MEMORY CHECK ENABLED.

3.6 BAD SECTOR

THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR
INFORMATION CONTAINED ON CYLINDER 410, HEAD 2. PRINTOUTS
OF DATA ERRORS DUE TO BAD SECTORS/TRACKS WILL BE MASKED OUT.

3.7 EXECUTION TIME

THE EXECUTION TIME IS APPROX. 2.5 MIN. BASED ON THE PDP 11/50.

3.8 FAULT ISOLATION

323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378

TO BE DETERMINED.

3.9 ERROR CORRECTION AND FAILURE RATE ANALYSIS

THIS PROGRAM WILL NOT DO ERROR CORRECTION OR FAILURE RATE ANALYSIS.

3.10 DEFAULT UNIBUS ADDRESSES & VECTORS

THE FOLLOWING IS A LIST OF ALL DEFAULT ADDRESSES & VECTORS OF ALL HARDWARE TO BE USED & THEIR MEMORY ADDRESSES WHERE THEY CAN BE CHANGED.

	LOCATION	DEFAULT CONTENTS
RK06-RK07 BUSS ADDRESS	1264	177440
CONTROLLER INTERRUPT VECTOR	1314	210
CONTROLLER PRIORITY	1316	240
P-CLOCK STATUS REG	1320	172540
P-CLOCK SET BUFFER	1322	172542
P-CLOCK READ BUFFER	1324	172544
L-CLOCK STATUS REG	1326	177546
L-CLOCK INTERRUPT VECTOR	1330	100
P-CLOCK INTERRUPT VECTOR	1332	104
TTY KB STATUS REG	1144	177560
TTY KB BUFFER	1146	177562
TTY PRINTER STATUS REG	1150	177564
TTY PRINTER BUFFER	1152	177566

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING

THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING STANDARD PROCEDURE FOR ABSOLUTE LOADER TAPES; OR FROM ANY MEDIA SUPPORTED BY XXDP.

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

4.1.2 SET SWITCH REGISTERS AS DESIRED (SEE SEC 4.3).

4.1.3 INSTALL CALBE(S) & SET DRIVES TO BE TESTED IN THE 'LOAD' CONDITION & WITH BOTH PORTS SELECTED & WRITE LOCK DISABLED. DRIVES NOT TO BE TESTED MUST HAVE BOTH PORTS DESELECTED. ALSO, THE DUAL PORT TEST SWITCH ON THE DUAL PORT MODULE MUST BE ENABLED.

NOTE: THE DRIVE WILL NOT RESPOND TO THE 'START SPINDLE' COMMAND IF THE RUN/STOP SWITCH IS IN THE 'STOP'

POSITION.

379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434

4.1.4 PRESS 'START'

THE PROGRAM WILL IDENTIFY ITSELF AND WILL BEGIN A DIALOGUE WITH THE OPERATOR TO DETERMINE DRIVES TO BE TESTED (SEE SEC 4.5).

THE PROGRAM BEGINS TESTING ONLY THOSE DRIVES SPECIFIED BY THE INPUT DIALOGUE. IF A SPECIFIED DRIVE CANNOT BE FOUND BY THE PROGRAM IT WILL BE FLAGGED AS AN ERROR THAT THE DRIVE WAS NOT AVAILABLE. THEN BEGINNING WITH THE LOWEST NUMERICAL DRIVE AND PROCEEDING IN SEQUENTIAL ORDER, ALL VALID DRIVES WILL BE TESTED. ONE PASS THROUGH THE TEST SEQUENCE WILL BE

PERFORMED ON EACH DRIVE BEFORE MOVING TO THE NEXT DRIVE IN SEQUENCE. THE DRIVE TO BE TESTED WILL BE TYPED AT THE BEGINNING OF EACH PASS. 'END OF PASS' WILL BE TYPED AFTER TESTING ALL DRIVES.

4.2 STARTING LOCATIONS

LOCATION 200 - STARTING ADDRESS TO DEFAULT THE BUSS ADDRESS & THE CONTROLLER INTERRUPT VECTOR & TEST ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION.

NOTE: THE DRIVE PRESENT CONDITION IS:

- A. HEADS MANUALLY LOADED
- B. BOTH PORTS SELECTED
- C. WRITE LOCK DISABLED
- D. DRIVE READY INDICATOR ON

LOCATION 220 - STARTING ADDRESS TO INPUT TESTING PARAMETERS VIA THE INPUT DIALOGUE. BUSS ADDRESS & CONT. INTERRUPT VECTOR INPUTTED ONLY ON 1ST PASS.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

THE PROGRAM WILL DETERMINE WHETHER THE DRIVE IS AN RK06/RK07 WITHOUT OPERATOR INPUTS.

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

SWITCH FUNCTION

435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490

15 HALT ON ERROR
14 LOOP ON TEST
13 INHIBIT ERROR TYPEOUT
12 BYPASS DRIVE AFTER 20 ERRORS

11 INHIBIT ITERATION
10 BELL ON ERROR
9 LOOP ON ERROR
8 LOOP ON TEST IN SW<07:00>

4.3.1 SW<15>

THE PROGRAM HALTS ON ENCOUNTERING AN ERROR, AFTER TYPING OUT THE ERROR MESSAGE AND PERTINENT INFORMATION. PRESSING "CONTINUE" CONTINUES OPERATION OF THE PROGRAM.

4.3.2 SW<14>

THE PROGRAM LOOPS ON THE TEST THAT IS BEING EXECUTED WHEN THE SWITCH IS PUT ON. THIS SWITCH IS NORMALLY USED ALONG WITH SW15.

4.3.3 SW<13>

THIS SWITCH INHIBITS ALL ERROR MESSAGES. NORMALLY USED WHEN LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9). WITH SW<13>SET, SW<15> SHOULD NOT BE SET

4.3.4 SW<12>

THIS SWITCH BYPASSES A GIVEN DRIVE AFTER 20 ERRORS HAVE BEEN DETECTED.

4.3.5 SW<11>

EACH TEST WILL BE EXECUTED ONLY ONCE. NORMALLY AFTER THE FIRST PASS, EACH SUBTEST IS ITERATED A NUMBER OF TIMES (USUALLY 50, 5 IN SOME CASES). SETTING THIS SWITCH INHIBITS ITERATIONS, SO THAT QUICK PASSES CAN BE MADE.

4.3.6 SW<10>

RINGS A BELL ON ERROR. USEFUL WHEN ERROR TYPEOUT IS INHIBITED.

4.3.7 SW<09>

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP FOR

491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546

ERRORS. IF THE PROGRAM DETECTS AN ERROR, IT WILL LOOP BACK TO THE BEGINNING OF TEST.

4.3.8 SW<08>

THIS SWITCH IS USED TO SELECT A PARTICULAR TEST (AS PER SW<00-7>) FOR EXECUTION AND SUBSEQUENT LOOPING. THUS IF TEST 15 IS TO BE SELECTED THE SWITCH SETTING WOULD BE 000415. IT SHOULD BE NOTED THAT BEFORE SELECTING & LOOPING TEST 15, ALL THE PREVIOUS TESTS (1-14) WILL BE EXECUTED.

4.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW =

EACH TIME SWITCH SETTING ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROS ARE NOT REQUIRED. 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

4.5 INPUT DIALOGUE

THE DIALOGUE WILL BE DONE INTERACTIVELY. THE PROGRAM WILL REQUEST A PARAMETER BY CONSOLE TYPEOUT. THE PARAMETER MAY THEN BE ENTERED AS SPECIFIED BELOW OR ALLOWED TO DEFAULT BY A CARRIAGE RETURN. UNRECOGNIZED OR ILLEGAL RESPONSES WILL BE ECHOED BACK FOLLOWED BY '?'. THE PROPER RESPONSE MAY THEN BE ENTERED.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.4.

547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602

4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

THE DEFAULT RESPONSE IS CARRIAGE RETURN TO TEST ALL DRIVES
IN THE 'DRIVE PRESENT' CONDITION.

THE OPERATOR CAN ALSO TYPE IN THE SPECIFIC DRIVE NUMBERS
TO BE TESTED, SEPARATED BY COMMAS & TERMINATED BY A CARRIAGE
RETURN.

E.G. DRIVES TO BE TESTED (EVEN NOS. ONLY): 0,4

NOTES: 1. FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

2. SEE 'RESTRICTIONS & OPERATOR ACTION' IN SECTION 2.3.

4.5.2 BUS ADDRESS

THE REQUEST WILL BE:

TYPE IN BUSS ADDRESS IF NOT 177440

THE DEFAULT IS A CARRIAGE RETURN

4.5.3 CONTROLLER INTERRUPT VECTOR

THE REQUEST WILL BE:

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210

THE DEFAULT IS A CARRIAGE RETURN.

4.5.4 EXAMPLE OF PROGRAM DIALOGUE

THE EXAMPLE SHOWN IS FOR A PROGRAM STARTED AT ADDRESS 220.
ALL OPERATOR RESPONSES ARE UNDERLINED.

UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC

CZR6GCO

DRIVES TO BE TESTED: 0,4<CR>

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>

603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>

WILL TEST DRIVES:
0
4

DRIVE 0

(THE REST IS IDENTICAL TO THE EXAMPLE SHOWN IN 4.6 BELOW)

4.6 PROGRAM EXAMPLE

THE FOLLOWING IS AN EXAMPLE OF A PROGRAM STARTED AT THE
DEFAULT ADDRESS (200) & WITH 2 DRIVES ON THE LINE.

UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC

CZR6GCO

WILL TEST DRIVES:
0
4

DRIVE 0

DRIVE SERIAL NO. AAA
CARTRIDGE SERIAL NO. BBB

DRIVE 4

DRIVE SERIAL NO. CCC
CARTRIDGE SERIAL NO. DDD

END PASS #1

WILL TEST DRIVES:
0
4

DRIVE 0

DRIVE 4

END PASS # 2

(ETC)

THE ABOVE ASSUMES NO ERRORS DETECTED.
THE NUMBER OF PASSES IS DETERMINED BY ACT/APT/XXDP

NOTES: 1. FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714

2. SEE 'RESTRICTIONS & OPERATOR ACTION' IN SECTION 2.3.

4.7 HALTING THE PROGRAM

THE PROGRAM PROVIDES A METHOD OF HALTING ITSELF SUCH THAT THE CARTRIDGE AND/OR DRIVE IS NOT LEFT IN AN UNDETERMINED STATE; IE: HEADS UNLOADED OR INVALID FORMAT.

TO PROPERLY HALT, TYPE CONTROL-C (^C) ON THE CONSOLE.

IF HEADS ARE LOADED, THE PROGRAM WILL:

1. ECHO ^C
2. TYPE "CPU HALTED"
3. HALT THE PROGRAM

IF HEADS ARE NOT LOADED, THE PROGRAM WILL:

1. ECHO ^C
2. TYPE 'HALT PENDING, PLEASE WAIT'
3. WILL LOAD HEADS
4. TYPE 'CPU HALTED'
5. HALT THE PROGRAM

NOTES:

1. OPERATING THE 'CONTINUE' SWITCH ON THE CPU CONSOLE WILL RETURN THE PROGRAM TO TEST 1 WHERE TESTING WILL BEGIN WITH THE 1'ST DRIVE AGAIN.

5.0 DUAL PORT DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION

5.1 GENERAL

A. BASIC CONTROLLER TESTS, SIZING & SETUP

THESE TESTS DO BASIC CONTROLLER REGISTER REFERENCE TESTS, CHECKS OPERATOR INPUTS AGAINST DRIVES ON THE LINE OR DEFAULTS TO TEST ALL DRIVES SEEN ON THE LINE WITHIN THE RESTRICTIONS DESCRIBED IN SECTION 2.3. CHECKS ARE MADE ON THE EXISTENCE OF EITHER AN L OR P CLOCK.

B. DUAL PORT TESTS

THESE TESTS VERIFY THE ABILITY OF THE DRIVE TO OPERATE IN THE DUAL PORT MODE. RELEASE, TIMEOUT & PROPER INTERACTION BETWEEN THE PORTS ARE VERIFIED.

C. METHOD TO DETERMINE THAT THE DRIVE IS IN NEUTRAL OR SEIZED.

THE PROGRAM DOES A 'SELECT DRIVE' COMMAND TO PORT 'A'. IF MESSAGE A0 RETURNS WITH THE 'DRIVE AVAILABLE' BIT NOT SET, IT ASSUMES PORT 'B' HAS SEIZED THE DRIVE.

715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770

IF MESSAGE A0 RETURNS WITH THE 'DRIVE AVAILABLE' BIT SET, IT ASSUMES THE DRIVE WAS IN NEUTRAL OR ALREADY SEIZED BY PORT 'A'.

AFTER A TIMEOUT OF 1 SECOND BY PORT 'A', ASSUMING NO FURTHER COMMANDS, THE DRIVE SHOULD BECOME AVAILABLE TO PORT 'B'.

THERE IS NO SPECIFIC METHOD TO VERIFY THE DRIVE IS IN NEUTRAL BECAUSE THE ACT OF DOING A SELECT DRIVE COMMAND TO CHECK FOR NEUTRAL SEIZES THE DRIVE & TAKES IT OUT OF NEUTRAL.

'DSC' & 'ATTN' DO NOT ASSERT AT THE END OF A TIMEOUT IF THERE IS NO PREVIOUS PORT REQUEST MADE.

THEY WILL ASSERT, HOWEVER, IF A PORT REQUESTED WHILE SEIZED BY THE OTHER PORT & A TIMEOUT OR RELEASE OCCURS ON THE OTHER PORT.

5.2 TEST DESCRIPTIONS

BASIC CONTROLLER TESTS, SIZING & SETUP

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS CAN BE ACCESSED. THE INABILITY TO BE ACCESSED WILL RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE. ANY ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER TESTS AND JUMPING TO 'END OF PASS'

TEST 2 SIZE THE BUSS

THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE MANUAL MODE.
EVERY EVEN NUMBERED DRIVE (0,2,4,6) IS ADDRESSED. CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE DRIVE WILL BE TESTED AS AN RK06. IF SET, THE PROGRAM WILL BYPASS TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET INDICATING THE OTHER PORT IS ACCESSED.
IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS RK07.

TEST 3 VERIFY OPERATOR DRIVE SELECTIONS

THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT DEFAULTED. EVERY EVEN NUMBERED DRIVE IS ADDRESSED & CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06 IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH

771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826

NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF
NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFORMATION TO
VERIFY IT WAS NOT SPECIFIED.
IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS RK07.

TEST 4 FIND NEXT DRIVE TO BE TESTED

THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
ADDRESS IN '\$UNIT' & \$TMP4 IS SET TO CDT IF DRIVE IS RK07.
THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
THE DRIVE WHOSE ADDRESS IS IN '\$UNIT'.

DUAL PORT TESTS

TEST 5 TEST PORT 'A' SEIZE & TIMEOUT

VERIFY THAT THE DRIVE CAN BE SEIZED & THAT THE PORT
TIMEOUT RELEASES THE DRIVE.

- A. SET VOLUME VALID FOR BOTH PORTS & DO A RECAL COMMAND
- B. A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'A'.
THE PROGRAM VERIFIES THE DRIVE HAS BEEN SEIZED BY 'DRIVE
AVAILABLE' SET.
- C. A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'B' THE
PROGRAM VERIFIES THAT 'DRIVE AVAILABLE' IS NOT SET
FOR PORT 'B' & THAT CERR IS SET.
- D. VERIFY THAT FOR ALL MESSAGES REQUESTED THRU PORT 'B', (MSG
A0-A3, B0-B3) THAT MESSAGE 0 ALWAYS RETURNS FROM PORT B
WHILE PORT 'A' IS SEIZED.
- E. WAIT FOR THE PORT TIMEOUT TO OCCUR ON PORT 'A' BY
CONTINUOUSLY CHECKING FOR ATTN ON PORT B. AFTER
ATTN-B IS REC'D, A DRIVE SELECT COMMAND IS ISSUED
THRU PORT B & 'DRIVE AVAILABLE' IS CHECKED TO
BE SET IN MESSAGE A0.

MEASURE THE DURATION OF THE TIMEOUT & TYPE THE VALUE
FOR THE FIRST PASS ONLY.
- F. VERIFY THAT ONLY PORT 'B' GETS 'DSC' & 'ATTN'.
- G. VERIFY THE DRIVE CLEAR COMMAND CLEARS 'DSC' & 'ATTN'
ON PORT 'B' BUT DOES NOT RELEASE THE DRIVE FROM PORT 'B'.

TEST 6 TEST PORT 'B' SEIZE & TIMEOUT

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882

TEST 7 PRINT DRIVE SERIAL NUMBER

THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A3
IN BCD ON THE 1'ST PASS ONLY.
IT ALSO TESTS THAT THE SERIAL # READ THRU BOTH PORTS
ARE THE SAME.

TEST 10 TEST PORT 'A' COMMAND SEIZE & ATTENTION

VERIFY THE OPERATION OF 'DSC' & 'ATTN' BITS AFTER A COMMAND.

- A. ISSUE A SEEK COMMAND TO CYLINDER 10 THRU PORT 'A'.
- B. VERIFY SEIZURE & THAT 'DSC' & 'ATTN' SETS FOR PORT 'A'
ONLY AFTER THE SEEK HAS COMPLETED.
- C. VERIFY 'ATTN' REMAINS SET BEYOND TIMEOUT
- D. VERIFY A DRIVE CLEAR COMMAND RESETS 'DSC' & 'ATTN'
& DOES NOT RELEASE THE DRIVE FROM PORT 'A'.

TEST 11 TEST PORT 'B' COMMAND SEIZE & ATTENTION

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'
BUT THE SEEK IS TO CYLINDER 0.

TEST 12 TEST RESET PORT 'A' ATTENTION BY DRIVE CLEAR COMMAND

VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF
THE SEIZING PORT

- A. SET EACH PORT'S ATTENTION BIT BY PERFORMING SEEK
COMMANDS TO CYLINDER 0 & ALLOWING TIMEOUTS.
- B. SEIZE THE DRIVE THRU PORT 'A' & ISSUE A DRIVE CLEAR COMMAND
VERIFY THAT 'DSC' & 'ATTN' FOR PORT 'A' HAVE BEEN CLEARED
- C. SEIZE THE DRIVE THRU PORT 'B' & VERIFY 'DSC' & 'ATTN'
HAVE NOT CLEARED

TEST 13 TEST RESET PORT 'B' ATTENTION BY DRIVE CLEAR COMMAND

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 14 TEST RELEASE, DRIVE SEIZED BY PORT 'A'

- A. SEIZURE THE DRIVE THRU PORT 'A'

883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938

B. ISSUE A RELEASE USING RKCS2 THRU PORT 'A'

C. VERIFY PORT 'B' CAN ACCESS THE DRIVE IMMEDIATELY &
THAT NEITHER PORT SEES 'DSC' OR 'ATTN'

TEST 15 TEST RELEASE, DRIVE SEIZED BY PORT 'B'

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 16 TEST RELEASE FROM PORT 'A' WITH PORT 'B' REQUESTING

A. PORT 'A' SEIZES THE DRIVE & DOES A SEEK TO SELF COMMAND.
THE PROGRAM VERIFIES 'DSC' & 'ATTN' ON PORT 'A' ONLY
ON COMPLETION

B. PORT 'B' TRIES TO ACCESS THE DRIVE. THE PROGRAM VERIFIES
DRIVE NOT AVAILABLE

C. A RELEASE BY PORT 'A' IS ISSUED. VERIFY PORT 'B' CAN
ACCESS THE DRIVE IMMEDIATELY & THAT 'DSC' & 'ATTN'
ARE SEEN ON PORT 'B' .

D. VERIFY PORT 'A' 'DSC' & 'ATTN' REMAINS SET AFTER RELEASE

E. THE PROGRAM ISSUES A DRIVE CLEAR COMMAND TO PORT 'B'
& VERIFIES 'DSC' & 'ATTN' RESETS.

F. THE PROGRAM THEN VERIFIES THAT PORT 'B' DOES NOT SEE
FURTHER (MULTIPLE) ATTENTIONS FROM WHAT WOULD HAVE BEEN
NORMAL TIMEOUT FROM PORT 'A'.

TEST 17 TEST RELEASE FROM PORT 'B' WITH PORT 'A' REQUESTING

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 20 TEST RELEASE FROM REQUESTING PORT 'B' INHIBITS 'ATTN'

A. PORT 'A' SEIZES THE DRIVE

B. PORT 'B' ATTEMPTS TO SEIZE THE DRIVE

C. PORT 'B' & PORT 'A' RELEASE THE DRIVE, IN THAT ORDER

D. THE PROGRAM VERIFIES THAT NEITHER PORT 'A' OR 'B' ATTENTION
BITS SET

TEST 21 TEST RELEASE FROM REQUESTING PORT 'A' INHIBITS 'ATTN'

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 22 TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'

939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994

VERIFY THAT A RELEASE ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE IS SEIZED BY THE OTHER PORT

- A. SEIZE THE DRIVE THRU PORT 'A'.
- B. ISSUE A RELEASE THRU PORT 'B' & VERIFY DRIVE STILL SEIZED BY PORT 'A'.

TEST 23 TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 24 TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE FROM PORT 'A'

- A. ISSUE A SEEK COMMAND TO CYL 10 FROM PORT 'A' & AN IMMEDIATE RELEASE TO PORT A.
- B. VERIFY THE DRIVE IS AVAILABLE TO PORT 'B' & PORT 'B' RECEIVES 'ATTN'
- C. VERIFY PORT 'A' DOES NOT RAISE ATTN WHEN THE SEEK IS COMPLETED.

TEST 25 COMMAND & IMMEDIATE RELEASE FROM PORT 'B'

THE PREVIOUS TEST IS REPEATED FOR PORT 'B' BUT THE SEEK IS TO CYLINDER 0.

TEST 26 TEST TIMEOUT RETRIGGER THRU PORT 'A'

VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.

- A. PORT 'A' SEIZES THE DRIVE
- B. THE PROGRAM WAITS 500MS & RE-SEIZES THE DRIVE THRU PORT 'A'
- C. PORT 'B' ATTEMPTS TO SEIZE THE DRIVE & THE PROGRAM VERIFIES THAT FULL TIMEOUT TOOK PLACE FROM STEP 'B' ABOVE.

TEST 27 TEST TIMEOUT RETRIGGER THRU PORT 'B'

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 30 TEST PORT 'A' TIMER INHIBIT

- A. PORT 'A' SEIZES THE DRIVE
- B. PORT 'B' ATTEMPTS TO SEIZE THE DRIVE

- C. PORT 'A' RELEASES THE DRIVE
- D. PORT 'A' ATTEMPTS TO GET THE DRIVE BACK.

THE PROGRAM VERIFIES THAT PORT 'A' CANNOT ACCESS
THE DRIVE FOR APPROX 1 SEC

TEST 31 TEST PORT 'B' TIMER INHIBIT

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 32 TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'A'

VERIFY THAT THE UNLOAD COMMAND THRU A PORT, SEIZES THAT
PORT FOR AS LONG HAS HEADS ARE UNLOADED & RELEASE IS NOT
ISSUED.

- A. ISSUE AN UNLOAD COMMAND THRU PORT 'A'.
VERIFY DRIVE UNLOADS & ATTENTION IS SET.
- B. DELAY FOR 5 SECONDS & VERIFY DRIVE
NOT AVAILABLE TO PORT 'B' TO INSURE TIMERS INHIBITED.
- C. ISSUE A RELEASE FROM PORT 'A'. VERIFY DRIVE BECOMES
AVAILABLE TO PORT 'B'
- D. LOAD HEADS FROM PORT 'B' & VERIFY 'ATTN-B' AT COMPLETION

TEST 33 TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'B'

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 34 TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'A'

VERIFY THAT THE RECAL COMMAND THRU A PORT, SEIZES THAT
PORT FOR AS LONG AS THE RECAL IS IN PROGRESS & A
RELEASE IS NOT ISSUED.

- A. ISSUE A RECAL COMMAND FROM THE LAST CYL THRU PORT 'A'
- B. VERIFY PORT 'B' CANNOT SEIZE THE DRIVE UNTIL
PORT 'A' RECEIVES ATTN.
THIS INSURES THAT THE TIMERS WERE INHIBITED.

TEST 35 TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'B'

THE PREVIOUS TEST IS REPEATED FOR PORT 'B'

TEST 36 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 ^{1 2}PAGE 22

SEQ 0021

1051
1052

THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.

1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108

THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE FACTORY DURING MANF. ALL BAD SECTOR INFO (BSE) WILL BE STORED AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.

SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE

IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,

A MESSAGE WILL BE TYPED INDICATING THAT ALL FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED. THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.

TEST 37 DATA TESTS

VERIFY UNIQUE DATA CAN BE WRITTEN THRU EITHER PORT & READ BACK CORRECTLY THRU BOTH PORTS.

- A. ALL 1'S ARE WRITTEN THRU PORT 'A' ON CYL 0, SECTOR 0, TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.
- B. ALL 1'S ARE WRITTEN THRU PORT 'B' ON CYL 10, SECTOR 0, TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.
- C. THE PROGRAM CHECKS THAT CYL 0 WAS NOT OVERWRITTEN BY READING & VERIFYING ALL 0'S THRU PORT 'B'.

TEST 40 ALTERNATING SEEK INTERACTION TEST

THIS TEST VERIFIES THAT THERE ARE NO TIMING INTERACTION PROBLEMS BETWEEN SEEKS FROM BOTH PORTS.

- A. PORT 'A' SEIZES THE DRIVE & SEEKS TO CYLINDER 0 & RELEASES THE DRIVE AFTER 'ATTN' IS RECEIVED. THE PROGRAM VERIFIES THAT UNTIL ATTN IS REC'D, PORT 'B' SEES 'CONTROLLER ERROR' & 'DRIVE NOT AVAILABLE'.
- B. PORT 'B' SEIZES THE DRIVE & SEEKS TO THE LAST CYL & RELEASES THE DRIVE AFTER 'ATTN' IS RECEIVED. THE PROGRAM VERIFIES THAT UNTIL ATTN IS REC'D, PORT 'A' SEES 'CONTROLLER ERROR' & 'DRIVE NOT AVAILABLE'.
- C. THE ABOVE IS REPEATED FOR A PATTERN OF CONVERGING SEEKS TOWARD THE CENTER OF THE CARTRIDGE.
- D. THE PROGRAM VERIFIES MULTIPLE ATTENTIONS OR ERRORS DO NOT OCCUR AS A RESULT OF TIMING PROBLEMS.

1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164

6.1 ERROR INTERPRETATION

WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN. MSG A(00), MSG B(01), RKER, RKBA...ETC, INDICATE THE CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF ERROR.

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS ADVISED TO LOOK UP THE PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY

VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION OF WHAT IS BEING CHECKED IN THE TEST IS GIVEN AT THE BEGINNING OF EVERY TEST. ALL THE NUMBERS GIVEN WITH ERROR MESSAGES ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLE:

MESSAGE A0 ERROR
AFTER START SPINDLE CMD & FWD SET

	TEST NO.	PC				
	000014	016530				
	EXPECT					
A0	B0	A1	B1	A2	B2	B3
030144	100000	013704	000001			
	ACTUAL					
	140144	100000	101744	000001		
RKCS1	RKCS2	RKASOF	RKER	RKDS	RKDC	
040200	000100	010000	000000	000000	000000	

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR MESSAGE REGISTERS A0, B0, A1 & B1.

MESSAGES A2, B2 & B3 WILL BE TYPED OUT ONLY AS REQUIRED IF THE CYLINDER DIFFERENCE/OFFSET, CYLINDER ADDRESS & HEAD & SECTOR INFORMATION IS A VARIABLE PARAMETER OF THE TEST.

7.0 DUAL PROCESSOR-DUAL CONTROLLER TESTING

1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193

THIS PROGRAM AS DESCRIBED, TESTS THRU A SINGLE CPU & RK611
IN A SYNCHRONOUS MANNER.

TO RUN IN A DYNAMIC MANNER, I.E.: DUAL CPU/DUAL RK611
WITH BOTH PORTS CONTENDING FOR THE DRIVE ASYNCHRONOUSLY,
THE FOLLOWING PROGRAMS SHOULD BE USED:

CZR6M SUBSYSTEM VERIFICATION PART 1 TEST 22
CZR6P PERFORMANCE EXERCISER

EACH OF THE ABOVE PROGRAMS REQUIRES A SPECIAL STARTING ADDRESS
TO ENTER THE DUAL PORT TESTING MODE.

TEST 22 IN THE SUBSYSTEM VERIFICATION PROGRAM CAN BE SET UP
SO THAT ONE PORT BECOMES THE WRITER OF A SPECIFIED DATA PATTERN
OVER A SPECIFIED RANGE OF HEADS & CYLINDERS, WHILE THE
OTHER PORT BECOMES THE READER.
THE ROLES OF EACH PORT CAN THEN BE REVERSED THRU ANOTHER SETUP.

THERE ARE MANY POSSIBLE WAYS OF DYNAMICALLY TESTING DUAL
PORT BY THE ABOVE TEST.
THE OPERATOR IS URGED TO READ THE PROGRAM DOCUMENT.

THE PERFORMANCE EXERCISER SHOULD BE USED TO RANDOMLY
TEST ALL POSSIBLE DUAL PORT OPERATIONS.
THIS PROGRAM WILL MOST SIMULATE ACTUAL ON-LINE DUAL PORT OPERATION.
THE OPERATOR IS URGED TO READ THE PROGRAM DOCUMENT.
[END OF SPECIFICATION]

%

1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239

167400
000001

```
: *** PGM REV 030 ***  
: 29-SEP-77 PATCH ALL ERROR LOGS  
  
.NLIST CND,MC,MD  
.LIST ME  
.ENABL ABS,AMA  
  
;DEFINE SYSMAC MACROS  
  
$SWR= 167400 ;DEFINE SWITCHES 15,14,13,11,10,9,8  
$TN= 1 ;SET FIRST TEST NO. TO 1  
  
.TITLE CZR6GCO RK611 DU PORT LGC  
:*COPYRIGHT (C) 1976,1982  
:*DIGITAL EQUIPMENT CORP.  
:*MAYNARD, MASS. 01754  
:*  
:*PROGRAM BY GARY PAPAIZAN  
:*  
:*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
:*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.  
:*  
  
.SBTTL OPERATIONAL SWITCH SETTINGS  
:*  
:* SWITCH USE  
:* -----  
:* 15 HALT ON ERROR  
:* 14 LOOP ON TEST  
:* 13 INHIBIT ERROR TYPEOUTS  
:* 12 ABORT DRIVE AFTER 20 ERRORS  
:* 11 INHIBIT ITERATIONS  
:* 10 BELL ON ERROR  
:* 9 LOOP ON ERROR  
:* 8 LOOP ON TEST IN SWR<7:0>  
  
.SBTTL SUMMARY OF STARTING LOCATIONS  
:*  
:*  
:* 200 DEFAULT PARAMETERS  
:* 220 INPUT PARAMETERS  
:* 240 ODT11  
:*
```

```
1240 .SBTTL BASIC DEFINITIONS
1241
1242 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
1243 001100 STACK= 1100
1244 .EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
1245 .EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
1246
1247 ;*MISCELLANEOUS DEFINITIONS
1248 000011 HT= 11 ;;CODE FOR HORIZONTAL TAB
1249 000012 LF= 12 ;;CODE FOR LINE FEED
1250 000015 CR= 15 ;;CODE FOR CARRIAGE RETURN
1251 000200 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
1252 177776 PS= 177776 ;;PROCESSOR STATUS WORD
1253 .EQUIV PS,PSW
1254 177774 STKLMT= 177774 ;;STACK LIMIT REGISTER
1255 177772 PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
1256 177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
1257 177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
1258
1259 ;*GENERAL PURPOSE REGISTER DEFINITIONS
1260 000000 R0= %0 ;;GENERAL REGISTER
1261 000001 R1= %1 ;;GENERAL REGISTER
1262 000002 R2= %2 ;;GENERAL REGISTER
1263 000003 R3= %3 ;;GENERAL REGISTER
1264 000004 R4= %4 ;;GENERAL REGISTER
1265 000005 R5= %5 ;;GENERAL REGISTER
1266 000006 R6= %6 ;;GENERAL REGISTER
1267 000007 R7= %7 ;;GENERAL REGISTER
1268 000006 SP= %6 ;;STACK POINTER
1269 000007 PC= %7 ;;PROGRAM COUNTER
1270
1271 ;*PRIORITY LEVEL DEFINITIONS
1272 000000 PR0= 0 ;;PRIORITY LEVEL 0
1273 000040 PR1= 40 ;;PRIORITY LEVEL 1
1274 000100 PR2= 100 ;;PRIORITY LEVEL 2
1275 000140 PR3= 140 ;;PRIORITY LEVEL 3
1276 000200 PR4= 200 ;;PRIORITY LEVEL 4
1277 000240 PR5= 240 ;;PRIORITY LEVEL 5
1278 000300 PR6= 300 ;;PRIORITY LEVEL 6
1279 000340 PR7= 340 ;;PRIORITY LEVEL 7
1280
1281 ;*'SWITCH REGISTER' SWITCH DEFINITIONS
1282 100000 SW15= 100000
1283 040000 SW14= 40000
1284 020000 SW13= 20000
1285 010000 SW12= 10000
1286 004000 SW11= 4000
1287 002000 SW10= 2000
1288 001000 SW09= 1000
1289 000400 SW08= 400
1290 000200 SW07= 200
1291 000100 SW06= 100
1292 000040 SW05= 40
1293 000020 SW04= 20
1294 000010 SW03= 10
1295 000004 SW02= 4
```

```
1296      000002      SW01= 2
1297      000001      SW00= 1
1298      .EQUIV      SW09,SW9
1299      .EQUIV      SW08,SW8
1300      .EQUIV      SW07,SW7
1301      .EQUIV      SW06,SW6
1302      .EQUIV      SW05,SW5
1303      .EQUIV      SW04,SW4
1304      .EQUIV      SW03,SW3
1305      .EQUIV      SW02,SW2
1306      .EQUIV      SW01,SW1
1307      .EQUIV      SW00,SW0
1308
1309      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1310      100000      BIT15= 100000
1311      040000      BIT14= 40000
1312      020000      BIT13= 20000
1313      010000      BIT12= 10000
1314      004000      BIT11= 4000
1315      002000      BIT10= 2000
1316      001000      BIT09= 1000
1317      000400      BIT08= 400
1318      000200      BIT07= 200
1319      000100      BIT06= 100
1320      000040      BIT05= 40
1321      000020      BIT04= 20
1322      000010      BIT03= 10
1323      000004      BIT02= 4
1324      000002      BIT01= 2
1325      000001      BIT00= 1
1326      .EQUIV      BIT09,BIT9
1327      .EQUIV      BIT08,BIT8
1328      .EQUIV      BIT07,BIT7
1329      .EQUIV      BIT06,BIT6
1330      .EQUIV      BIT05,BIT5
1331      .EQUIV      BIT04,BIT4
1332      .EQUIV      BIT03,BIT3
1333      .EQUIV      BIT02,BIT2
1334      .EQUIV      BIT01,BIT1
1335      .EQUIV      BIT00,BIT0
1336
1337      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1338      000004      ERRVEC= 4          ;; TIME OUT AND OTHER ERRORS
1339      000010      RESVEC= 10         ;; RESERVED AND ILLEGAL INSTRUCTIONS
1340      000014      TBITVEC=14        ;; "T" BIT
1341      000014      TRTVEC= 14         ;; TRACE TRAP
1342      000014      BPTVEC= 14         ;; BREAKPOINT TRAP (BPT)
1343      000020      IOTVEC= 20         ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
1344      000024      PWRVEC= 24         ;; POWER FAIL
1345      000030      EMTVEC= 30         ;; EMULATOR TRAP (EMT) **ERROR**
1346      000034      TRAPVEC=34        ;; "TRAP" TRAP
1347      000060      TKVEC= 60          ;; TTY KEYBOARD VECTOR
1348      000064      TPVEC= 64          ;; TTY PRINTER VECTOR
1349      000240      PIRQVEC=240        ;; PROGRAM INTERRUPT REQUEST VECTOR
1350
1351      .SBTTL      RK06 CONTROLLER REGISTER DEFINITION
```

```
1352
1353
1354
1355      000000      RKCS1= 0      ;CONTROL AND STATUS REGISTER 1
1356      000002      RKWC= 2      ;WORD COUNT REGISTER
1357      000004      RKBA= 4      ;BUS ADDRESS REGISTER
1358      000006      RKDA= 6      ;DESIRED TRACK SECTOR REGISTER
1359      000010      RKCS2= 10     ;CONTROL AND STATUS REGISTER 2
1360      000012      RKDS= 12     ;DRIVE STATUS REGISTER
1361      000014      RKER= 14     ;ERROR REGISTER
1362      000016      RKASOF= 16    ;ATTENTION SUMMARY AND OFFSET REGISTER
1363      000020      RKDC= 20     ;DESIRED CYLINDER REGISTER
1364      000024      RKDB= 24     ;DATA BUFFER
1365      000026      RKMR1= 26     ;MAINTENANCE REGISTER 1
1366      000034      RKMR2= 34     ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1367      000036      RKMR3= 36     ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1368      000030      RKECPS= 30    ;ECC POSITION INFORMATION
1369      000032      RKECPT= 32    ;ECC PATTERN INFORMATION
1370
1371      .SBTTL CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1372
1373      ; DRIVE COMMANDS
1374
1375      000001      SELDRV= 1      ;SELECT DRIVE (GET STATUS)
1376      000003      PACK= 3      ;PACK ACKNOWLEDGE
1377      000005      CLEAR= 5      ;DRIVE CLEAR
1378      000007      UNLOAD= 7     ;UNLOAD
1379      000011      SRTSPL= 11    ;START SPINDLE
1380      000013      RECAL= 13    ;RECALIBRATE
1381      000015      OFFSET= 15   ;OFFSET
1382      000017      SEEK= 17     ;SEEK
1383      000021      RDDATA= 21    ;READ DATA
1384      000023      WRDATA= 23    ;WRITE DATA
1385      000025      RDHEAD= 25    ;READ HEADER
1386      000027      WRHEAD= 27    ;WRITE HEADER AND DATA
1387      000031      WRTCHK= 31    ;WRITE CHECK
1388
1389      000001      GO= BIT0      ;GO BIT
1390      000100      IE= BIT6      ;INTERRUPT ENABLE
1391      000200      RDY= BIT7     ;CONTROLLER READY
1392      000400      BA16= BIT8    ;BUS ADDRESS BIT 16
1393      001000      BA17= BIT9    ;BUS ADDRESS BIT 17
1394      002000      CDT= BIT10    ;CONTROLLER DRIVE TYPE (0=RK06,1=RK07)
1395      004000      CTO= BIT11    ;CONTROLLER TIMEOUT
1396      010000      CFMT= BIT12   ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1397      020000      DCPAR= BIT13  ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1398      040000      DI= BIT14    ;DRIVE INTERRUPT
1399      100000      CERR= BIT15   ;CONTROLLER ERROR
1400      100000      CCLR= BIT15   ;CONTROLLER CLEAR
1401
1402      .SBTTL CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1403
1404      000007      DRVMSK= 7      ;MASK FOR DRIVE SELECTION CODE
1405      000010      RLS= BIT3      ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1406      000020      BAI= BIT4      ;BUS ADDRESS INCREMENT INHIBIT
1407      000040      SCLR= BIT5      ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
```

1408	000100	IR=	BIT6	:INPUT READY
1409	000200	OR=	BIT7	:OUTPUT READY
1410	000400	UFE=	BIT8	:UNIT FIELD ERROR
1411	001000	MDS=	BIT9	:MULTIPLE DRIVE SELECT
1412	002000	PGE=	BIT10	:PROGRAMMING ERROR
1413	004000	NEM=	BIT11	:NON-EXISTENT MEMORY
1414	010000	NED=	BIT12	:NON-EXISTENT DRIVE
1415	020000	UPE=	BIT13	:UNIBUS PARITY ERROR
1416	040000	WCE=	BIT14	:WRITE CHECK ERROR
1417	100000	DLT=	BIT15	:DATA LATE ERROR
1418				
1419		.SBTTL	ERROR REGISTER BIT DEFINITION (RKER:14)	
1420				
1421	000001	ILF=	BIT0	:ILLEGAL FUNCTION CODE
1422	000002	SKI=	BIT1	:SEEK INCOMPLETE
1423	000004	NXF=	BIT2	:NON-EXECUTABLE FUNCTION
1424	000010	DRPAR=	BIT3	:DRIVE DETECTED SERCON PARITY ERROR
1425	000020	FMTE=	BIT4	:FORMAT ERROR
1426	000040	DTYPE=	BIT5	:DRIVE TYPE ERROR
1427	000100	ECH=	BIT6	:ECC HARD
1428	000200	BSE=	BIT7	:BAD SECTOR ERROR
1429	000400	HVRC=	BIT8	:HEADER VRC ERROR
1430	001000	COE=	BIT9	:CYLINDER ADDRESS OVERFLOW ERROR
1431	002000	IDAE=	BIT10	:INVALID DISK ADDRESS ERROR: HEAD/CYL
1432	004000	WLE=	BIT11	:WRITE LOCK ERROR
1433	010000	DTE=	BIT12	:DRIVE TIMING ERROR
1434	020000	OPI=	BIT13	:OPERATION (SEARCH) INCOMPLETE
1435	040000	UNS=	BIT14	:DRIVE UNSAFE
1436	100000	DCK=	BIT15	:DATA CHECK
1437				
1438		.SBTTL	STATUS REGISTER BIT DEFINITION (RKDS:12)	
1439				
1440	000001	DRA=	BIT0	:DRIVE AVAILABLE (CONTROLLER IS SET IF : THIS BIT IS RESET)
1441				
1442	000004	OFST=	BIT2	:DRIVE OFFSET
1443	000010	ACLO=	BIT3	:AC LOW
1444	000020	DCLO=	BIT4	:DC LOW
1445	000040	DROT=	BIT5	:DRIVE OFF TRACK
1446	000100	VV=	BIT6	:VOLUME VALID
1447	000200	DRDY=	BIT7	:DRIVE READY
1448	000400	DDT=	BIT8	:DRIVE TYPE (0=RK06,1=RK07)
1449	004000	WRL=	BIT11	:WRITE LOCK
1450	020000	PIP=	BIT13	:POSITIONING IN PROGRESS
1451	040000	DSC=	BIT14	:DRIVE STATUS CHANGE
1452	100000	SVAL=	BIT15	:STATUS VALID
1453				
1454		.SBTTL	MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)	
1455				
1456	000017	MESMSK=	17	:MESSAGE MASK
1457	000020	PAT=	BIT4	:FORCE EVEN PARITY ON SERCON MESSAGE LINES
1458	000040	DMD=	BIT5	:DIAGNOSTIC MODE
1459	000100	MSP=	BIT6	:MAINTENANCE SECTOR PULSE
1460	000200	MIND=	BIT7	:MAINTENANCE INDEX
1461	000400	MCLK=	BIT8	:MAINTENANCE CLOCK
1462	001000	MERD=	BIT9	:MAINTENANCE ENCODED READ DATA
1463	002000	MEWD=	BIT10	:MAINTENANCE ENCODED WRITE DATA

1464	004000	PCA= BIT11	:PRECOMPENSATION ADVANCE
1465	010000	PCD= BIT12	:PRECOMPENSATION DELAY
1466	020000	ECCW= BIT13	:ECC WORD IS BEING READ OR WRITTEN
1467	040000	WRTGAT= BIT14	:WRITE GATE
1468	100000	RDGATE= BIT15	:READ GATE
1469			
1470		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1471			
1472	000040	D.DRA= BIT5	:DRIVE AVAILABLE
1473	000100	D.VV= BIT6	:VOLUME VALID
1474	000200	D.DRDY= BIT7	:DRIVE READY
1475	000400	D.DDT= BIT8	:DRIVE TYPE (0=RK06,1=RK07)
1476	001000	D.FORM= BIT9	:DRIVE FORMAT
1477	002000	D.OFF= BIT10	:CFFSET ON
1478	004000	D.WRL= BIT11	:WRITE LOCK
1479	010000	D.SPIN= BIT12	:SPINDLE ON
1480	020000	D.PIP= BIT13	:POSITIONING IN PROGRESS
1481	040000	D.DSC= BIT14	:DRIVE STATUS CHANGE
1482			
1483		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1484			
1485	000020	D.SSP= BIT4	:SERVO SIG PRESENT
1486	000040	D.HDHM= BIT5	:HEADS HOME
1487	000100	D.BRHM= BIT6	:BRUSHES HOME
1488	000200	D.DOOR= BIT7	:DOOR INTERLOCKED
1489	000400	D.CART= BIT8	:CARTRAGE INTERLOCK
1490	001000	D.SPOK= BIT9	:SPEED OK
1491	002000	D.FWD= BIT10	:FORWARD
1492	004000	D.REV= BIT11	:REVERSE
1493	010000	D.LOAD= BIT12	:HEADS LOADING
1494	020000	D.RTZ= BIT13	:RETURN TO ZERO
1495	040000	D.UNLD= BIT14	:HEADS UNLOADING
1496			
1497		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1498			
1499	000040	D.IDAE= BIT5	:INVALID DISK ADDRESS ERROR:HEAD/CYL
1500	000100	D.ACLO= BIT6	:AC LOW
1501	000200	D.FLT= BIT7	:DRIVE FAULT
1502	000400	D.ILF= BIT8	:ILLEGAL FUNCTION CODE
1503	001000	D.PAR= BIT9	:DRIVE DETECTED SERCON PARITY ERROR
1504	002000	D.SKI= BIT10	:SEEK INCOMPLETE
1505	004000	D.WLE= BIT11	:WRITE LOCK ERROR
1506	010000	D.SPLS= BIT12	:SPEED LOSS
1507	020000	D.DROT= BIT13	:DRIVE OFF TRACK
1508	040000	D.UNS= BIT14	:R/W UNSAFE
1509			
1510		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1511			
1512	000020	D.SECT= BIT4	:SECTOR ERROR
1513	000040	D.WCUR= BIT5	:WRITE CURRENT AND NO WRITE GATE
1514	000100	D.WGAT= BIT6	:WRITE GATE AND NO TRANSISTIONS
1515	000200	D.HDFL= BIT7	:HEAD FAULT
1516	000400	D.MHD= BIT8	:MULTIPLE HEAD SELECT
1517	001000	D.XERR= BIT9	:INDEX ERROR
1518	002000	D.TIB= BIT10	:TRIBIT ERROR
1519	004000	D.PLO= BIT11	:PLO ERROR

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 F 3 PAGE 32
DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)

SEQ 0031

1520	010000	D.NMOV= BIT12	:SEEK AND NO MOTION
1521	020000	D.LIMD= BIT13	:LIMIT DETECT ON SEEK
1522	040000	D.SUNS= BIT14	:SERVO UNSAFE
1523			
1524		.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)	
1525			
1526	000007	M.DRV= 7	:DRIVE CODE, ALL BYTES
1527	077770	M.SER= 77770	:DRIVE SERIAL #, BYTE 11
1528			
1529		.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)	
1530			
1531	000003	M.ID= 3	:BYTE ID, ALL BYTES
1532	040000	M.ALGN= BIT14	:ALIGN SIGN, BYTE 10
1533	000760	M.SECT= 760	:SECTOR COUNT, BYTE 11
1534	007000	M.HEAD= 7000	:HEAD DECODE, BYTE 11
1535	100000	M.PAR= BIT15	:PARITY, MESS A/B, ALL BYTES

```
1536  
1537  
1538  
1539      000000  
1540  
1541  
1542  
1543      000174  
1544 000174 000000  
1545 000176 000000  
1546  
1547 000200 000137 010050  
1548      000220  
1549 000220 000137 010040  
1550  
1551      000240  
1552 000240 000137 067350  
1553  
1554  
1555  
1556  
1557  
1558      000244  
1559      000046  
1560 000046 042772  
1561      000052  
1562 000052 100000  
1563      000244  
1564      001000  
1565  
1566  
1567  
1568  
1569  
1570      001000  
1571      000024  
1572 000024 000200  
1573      000044  
1574 000044 001000  
1575      001000  
1576  
1577  
1578  
1579  
1580 001000  
1581 001000 000000  
1582 001002 001210  
1583 001004 000430  
1584 001006 001130  
1585 001010 001130  
1586 001012 000042  
1587  
1588  
1589  
1590  
1591
```

```
.SBTTL TRAP CATCHER  
      .=0  
;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A "+2,HALT"  
;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS  
;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS  
      .=174  
DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER  
SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER  
.SBTTL STARTING ADDRESS(ES)  
      JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM  
      .=220  
      JMP PARSRT      ;INPUT PARAMETERS  
      .=240  
      JMP 0.ODT       ;ENTER ODT11  
  
.SBTTL ACT11 HOOKS  
;*****  
;HOOKS REQUIRED BY ACT11  
      $SVPC=.          ;SAVE PC  
      .=46  
      $ENDAD           ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP  
      .=52  
      .WORD 100000     ;;2)SET LOC.52 TO 100000  
      .=$$VPC         ;; RESTORE PC  
      .=1000  
.SBTTL APT PARAMETER BLOCK  
;*****  
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  
;*****  
      .$X=.           ;;SAVE CURRENT LOCATION  
      .=24           ;;SET POWER FAIL TO POINT TO START OF PROGRAM  
      200            ;;FOR APT START UP  
      .=44           ;;POINT TO APT INDIRECT ADDRESS PNTR.  
      $APTHDR        ;;POINT TO APT HEADER BLOCK  
      .=.$X          ;;RESET LOCATION COUNTER  
;*****  
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
;INTERFACE SPEC.  
$APTHD:  
$HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
$MBADR: .WORD $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)  
$STSTM: .WORD 280.   ;;RUN TIM OF LONGEST TEST  
$PASTM: .WORD 600.   ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
$UNITM: .WORD 600.   ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT  
      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)  
  
.LIST MD  
;
```

```
1592 ;USE LOOP X TO OMIT SUBCLR
1593 ;
1594 .MACRO LOOP A
1595 SCOP1
1596 MOV #STACK,SP ;RESTORE STK PTR
1597
1598 .IF B A
1599 JSR PC,SUBCLR
1600 ERROR 24 ;CERR AFTER SCLR
1601
1602 .ENDC
1603 .ENDM LOOP
1604
1605 ;
1606 ;THIS MACRO FILLS EXPECTED MSG A0, B0, A1, B1, A2, B2 & B3 WITH STANDARD BITS
1607 ;A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1608 ;NOTE: A CAN BE ANY BIT COMBINATION DESIRED & INCLUDES D.DRA
1609 ;
1610 .MACRO F.EAB A
1611
1612 MOV #<A!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
1613 CLR E.B0 ;EXPECTED MSG B0
1614 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
1615 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
1616 CLR E.A2 ;EXPECTED MSG A2
1617 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
1618 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
1619 .ENDM F.EAB
1620
1621 ;
1622 ;THIS MACRO ASSUMES DRIVE MSG A0, B0, A1, B1 WILL ALWAYS BE TESTED
1623 ;USE A,C,D,E FOR MSG A0, B0, A1, B1 ERROR NUMBERS RESP.
1624 ;USE G=T.A2 TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
1625 ;USE H=T.B2 TO READ MSG B2 & PUT INFO INTO 'CYLADD'
1626 ; I=T.B3 TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEAD'
1627 ;
1628 ;USE F=<ERROR DESCRIPTION>
1629 ;
1630 .MACRO CHECK A,C,D,E,F,G,H,I
1631
1632 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
1633 .WORD G!H!I ;& MSGS SPECIFIED HERE
1634 ERROR A ;MSG A0 ERROR F
1635 ERROR C ;MSG B0 ERROR
1636 ERROR D ;MSG A1 ERROR
1637 ERROR E ;MSG B1 ERROR
1638 .ENDM CHECK
1639
1640 ;
1641 ;A=CYL DIFF/OFFSET ERROR #
1642 ;B=CYL ADDR ERROR #
1643 ;C=<ERROR DESCRIPTION>
1644 ;
1645 .MACRO CWD2 A,B,C,?D,?E
1646 MOV #2,RKMR1(R5)
```

```
1648 JSR PC,GSTAT
1649 TST CYLDIF ;SEE IF MSG A2=0
1650 BEQ D ;BR IF YES
1651 ERROR A ;MSG A2 NOT CLEARED C
1652 D: TST CYLADD ;SEE IF MSG B2=0
1653 BEQ E ;BR IF YES
1654 ERROR B ;MSG B2 NOT CLEARED C
1655 E:
1656 .ENDM CWD2
1657
1658 .MACRO STDER1 B
1659 F.EAB <D.DRA!B>
1660 CHECK 165,166,167,170,<AFTER TIMEOUT>,0,0,0
1661 .ENDM STDER1
1662
1663 .MACRO STDER2 B
1664 F.EAB <D.DRA!B>
1665 CHECK 211,212,213,214,<AFTER RELEASE ISSUED>,0,0,0
1666 .ENDM STDER2
1667
1668
1669 .MACRO DRCLR C,?A
1670
1671 MOV #CCLR,RKCS1(R5)
1672 MOV $UNIT,RKCS2(R5) ;DRIVE#
1673 ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
1674 MOV #CLEAR,HCS1
1675 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
1676 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
1677 JSR PC,TSTATN ;TEST FOR ATTN
1678 BR A
1679 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1680 A:
1681 .IF B C
1682 F.EAB D.DRA
1683 CHECK 33,34,35,36,<AFTER DRV CLEAR CMD>,T.A2,T.B2,0
1684 .ENDC
1685
1686 .ENDM DRCLR
1687
1688
1689 ;
1690 ;USE CALIB X TO OMIT CHECKING MSGS A0, B0, A1, B1, A2 & B2
1691 ;
1692 .MACRO CALIB A,?C
1693
1694 MOV #CCLR,RKCS1(R5)
1695 MOV $UNIT,RKCS2(R5)
1696 ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
1697 MOV #RECAL,HCS1
1698 JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
1699 ERROR 124 ;RDY NOT SET AFTER RECAL CMD
1700
1701 MOV #1,RKMR1(R5) ;SELECT WORD 1
1702 JSR PC,GSTAT
1703 BIT #D.RTZ,HMR2
```

```
1704      BNE      C
1705      ERROR   70          ;RTZ NOT SET DURING RECAL CMD
1706      MOV     T10,TEMP2  ;SETUP TIMEOUT
1707      JSR     PC,FATT1    ;FIND ATTN
1708      ERROR   55          ;NO ATTN AFTER RECAL CMD
1709      .IF B
1710      A
1711      F.EAB   <D,DSC!D,DRA>
1712      CHECK  221,66,222,67,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
1713      CWD2   47,50,<AFTER RECAL CMD>
1714      .ENDC
1715      DRCLR
1716      .ENDM   CALIB
1717
1718      ;
1719      ;QUICK START SPINDLE
1720      ;
1721      .MACRO  QKSRT
1722
1723      JSR     PC,SUBCLR
1724      ERROR   24          ;CERR AFTER SCLR
1725
1726      MOV     #SRTSPL,HCS1
1727      JSR     PC,DOCMD    ;DO START SPINDLE CMD & GET CONTR RDY
1728      ERROR   121        ;RDY NOT SET AFTER ST SPIN CMD.
1729
1730      MOV     T100,TEMP2  ;SETUP TIMEOUT
1731      JSR     PC,FATT1    ;FIND ATTN
1732      ERROR   74          ;NO ATTN AFTER ST SPIN CMD.
1733
1734      CLR     UNLD
1735
1736      .ENDM   QKSRT
1737
1738      ;
1739      ; A=WRHEAD/<CFMT!WRHEAD>
1740      ; USE WRHDR   <A>,X TO OMIT CHECKING A0, B0, A1 & B1
1741      ;
1742      .MACRO  WRHDR   A,C,?D
1743
1744      MOV     #<A>,HCS1
1745      JSR     PC,DATCMD   ;DO DATA X FOR CMD & GET CONTR RDY
1746      ERROR   200        ;NO RDY AFTER WRITE HEADER CMD
1747      JSR     PC,GSTAT    ;GET FRESH STATUS
1748      BIT     #CERR,HCS1
1749      BEQ    D
1750      ERROR   201        ;CERR AFTER WRITE HEADER CMD
1751      TYPE   ,MSG21      ;ABORTING BALANCE OF TESTS
1752      JMP    $EOP
1753
1754      D:
1755      .IF B   C
1756      F.EAB   D,DRA
1757      CHECK  37,40,41,42,<AFTER WRITE HEADER CMD>,0,0,0
1758      .ENDC
1759      .ENDM   WRHDR
```

1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815

```

: A=RDHEAD/<CFMT!RDHEAD>
: USE RDHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1
:
: MACRO RDHDR A,C,?D,?E
      MOV #RHTAB,R0
      MOV #<A>,HCS1
      JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
      ERROR 171 ;NO RDY AFTER READ HEADER CMD
      BIT #CERR,HCS1
      BEQ D
      ERROR 174 ;CERR AFTER READ HEADER CMD
D:    MOV RKDB(R5),(R0)+ ;1'ST WORD FROM SILO TO RHTAB
      MOV RKDB(R5),(R0)+ ;2'ND WORD
      MOV RKDB(R5),(R0)+ ;3'RD WORD

      BIT #DLT,RKCS2(R5)
      BEQ E
      JSR PC,GSTAT
      ERROR 173 ;DLT AFTER READ HEADER CMD
E:
: .IF B C
      F.EAB D.DRA
      CHECK 301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
: .ENDC
: .ENDM RDHDR

: MACRO HDCHK3 ?A
      RDHDR RDHEAD
      CMP RH1AB,TOCYL ;CHECK WORD 0 ONLY, CYL#
      BEQ A ;BR IF SAME
      ERROR 51 ;WRONG CYL# ON HEADER
A:
: .ENDM HDCHK3

: A=TOCYL/FRCYL , B=HEAD# , C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
:
: MACRO HDTBL A,B,C
      MOV A,CALADD ;SETUP
      MOV #B,HEAD ;TO FILL
      MOV #C,FORMAT ;HEADER
      JSR PC,FHDTAB ;TABLE
: .ENDM HDTBL
;
```

```
1816 ;QUICK SEEK. ENTER WITH CYL# IN RKDC
1817 ;
1818 ;MACRO QKSEEK ?A
1819
1820 MOV #SEEK,HCS1
1821 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
1822 ERROR 131 ;NO RDY AFTER SEEK CMD
1823
1824 MOV T50000,TEMP1 ;SETUP TIMEOUT
1825 JSR PC,FATT2 ;FIND ATTN
1826 ERROR 132 ;NO ATTN AFTER SEEK CMD
1827
1828 BIT #CERR,HCS1
1829 BEQ A
1830 ERROR 210 ;CERR AFTER SEEK CMD
1831
1832 A: F.EAB <D.DSC!D.DRA>
1833 CHECK 161,162,163,164,<AFTER SEEK CMD>,0,0,0
1834
1835 .ENDM QKSEEK
1836
1837 ;
1838 ;A=WRDATA/<CFMT!WRDATA>
1839 ;C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
1840 ;D=ADDR TO JMP TO BYPASS TEST
1841 ;E: IF BLANK WILL CHECK A0, B0, A1 & B1 AT THE END OF WRITING
1842 ;E: IF NON BLANK WILL OMIT CHECKING A0 THRU B1
1843 ;
1844 ;
1845 ;
1846 .MACRO WDATA A,C,D,E,?F,?G,?H,?I
1847
1848 MOV #<A>,HCS1
1849 JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
1850 ERROR 11 ;NO RDY AFTER WRITE DATA CMD
1851 JSR PC,GSTAT ;GET FRESH STATUS
1852 BIT #CERR,HCS1
1853 BEQ I ;BR IF NO ERRORS
1854
1855 BIT #BSE,HER ;SEE IF BAD SECTOR FLAG
1856 BEQ G ;BR IF NO
1857 JSR PC,TRUERR ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
1858 BR H ;RETURN HERE IF NO
1859
1860 INC SECTOR ;RETURN HERE IF YES
1861 CMP SECTOR,#10. ;ARE 10 CONSEC. SECTORS BAD
1862 BNE F ;BR IF NO
1863 ERROR 46 ;ABORTING TEST DETECTED 10 BAD SECTORS
1864 JMP D ;BYPASS TEST
1865 F: MOV #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
1866 JMP C
1867 G: ERROR 12 ;CERR WITH WRITE DATA CMD
1868 F.EAB D.DRA
1869 CHECK 52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1870 TYPE ,MSG21 ;ABORTING BALANCE OF TESTS
1871 JMP $EOP
```

```
1872 H: ERROR 43 ;BAD SECTOR NOT LISTED IN TABLE
1873 I:
1874 .IF B E
1875 F.EAB D.DRA
1876 CHECK 52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1877 .ENDC
1878 .ENDM WDATA
1879
1880
1881 ;
1882 ;A=RDDATA/<CFMT!RDDATA>
1883 ;USE RDATA <A>,X TO OMIT CHECKING A0, B0, A1 & B1
1884 ;
1885
1886 .MACRO RDATA A,C,?D,?E,?F,?G,?H
1887
1888 MOV #<A>,HCS1
1889 JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
1890 ERROR 13 ;NO RDY AFTER READ DATA CMD
1891 JSR PC,GSTAT ;GET FRESH STATUS
1892 BIT #CERR,HCS1
1893 BEQ G
1894 BIT #BSE,HER ;SEE IF BAD SECTOR
1895 BEQ E
1896 ERROR 65 ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1897 BR H
1898
1899 E: BIT #DCK,HER ;SEE IF DATA CHECK ERROR
1900 BEQ F
1901 ERROR 21 ;DATA CHECK ERROR AFTER READ CMD (ECC)
1902 BR H
1903
1904 F: ERROR 14 ;CERR AFTER READ DATA CMD.
1905
1906 H: F.EAB D.DRA
1907 CHECK 54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
1908 TYPE ,MSG21 ;ABORTING BAL OF TESTS
1909 JMP $EOP
1910
1911 G:
1912 .IF B C
1913 F.EAB D.DRA
1914 CHECK 54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
1915 .ENDC
1916 .ENDM RDATA
1917
1918 ;
1919 ;A=WRTCHK/<CFMT!WRTCHK>
1920 ;C=EXPECTED DATA FOR TYPEOUT
1921 ;USE WRCHK <A>,DATA0,X TO OMIT CHECKING A0, B0, A1 & B1
1922 ;
1923
1924 .MACRO WRCHK A,C,D,?E,?F
1925
1926 MOV #<A>,HCS1
1927 JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
```



```
1928          ERROR 15          ;NO RDY AFTER WRITE CHECK CMD
1929          JSR   PC,GSTAT    ;GET FRESH STATUS
1930          BIT   #CERR,HCS1
1931          BEQ   F
1932          BIT   #WCE,HCS2    ;SEE IF WRITE CHECK ERROR
1933          BEQ   E
1934          MOV   RKDB(R5),WD1  ;ACTUAL WORD FOR PRINTOUT
1935          MOV   C,WD2        ;EXPECTED WORD FOR TYPEOUT
1936          ERROR 16          ;WCE AFTER WRITE CMD
1937          BR    F
1938
1939          E:    ERROR 22          ;CERR AFTER WRITE CHECK CMD
1940          F.EAB D.DRA
1941          CHECK 57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1942          TYPE  ,MSG21        ;ABORTING BALANCE OF TESTS
1943          JMP   $EOP
1944
1945          F:
1946          .IF   B             D
1947          F.EAB D.DRA
1948          CHECK 57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
1949          .ENDC
1950          .ENDM  WRCHK
1951
1952
1953          .MACRO EOPGM
1954
1955          SCOPE
1956          CLR   $ESCAPE
1957          MOV   #1,$TIMES
1958          MOV   #STACK,SP
1959          INC   $DEVCT        ;INCR COUNT FOR # DRIVES CHECKED
1960          CMP   DRIVS,$DEVCT  ;ARE ALL DRIVES PRESENT TESTED?
1961          BEQ   $EOP1+2      ;BR IF YES
1962          JMP   NUDRV        ;ELSE TEST NEXT DRIVE PRESENT
1963          $EOP1: SCOPE
1964          .ENDM  EOPGM
1965
1966
1967          .MACRO QKPACK ?A
1968          MOV   #CCLR,RKCS1(R5)
1969          MOV   $UNIT,RKCS2(R5)
1970          ADD   UNITB,RKCS2(R5)
1971          MOV   #PACK,HCS1
1972          JSR   PC,DOCMD      ;DO PACK CMD & GET CONTR RDY
1973          ERROR 116         ;CONTR NOT RDY AFT PACK CMD
1974
1975          BIT   #D.VV,HMR2
1976          BNE   A
1977          ERROR 27          ;VV NOT SET AFTER PACK CMD
1978          A:
1979          .ENDM  QKPACK
1980
1981
1982
1983          ;
```

1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022

```
:MACRO TO RELEASE PORT A OR B
:RLS PORT A: RELEAS 0,1,A,B (A,B,C,D IF USED WITH MACROS BELOW)
:RLS PORT B: RELEAS 1,0,B,A (B,A,D,C IF USED WITH MACROS BELOW)
:
.MACRO RELEAS W,X,Y,Z,?J
MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT Y
MOV #W,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'Y,MSG19A
ADD #RLS,RKCS2(R5) ;RELEASE PORT Y
MOV #SELDRV,HCS1
JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;NO RDY AFTER SEL DRV CMD

MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT Z
MOV #X,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'Z,MSG19A
MOV #SELDRV,HCS1
JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;NO RKY AFTER SEL DRV CMD

BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT Z
BNE J ;BR IF YES
ERROR 71 ;PORT Z NOT AVAIL AFTER PORT Y RLS

J:
.ENDM RELEAS

.MACRO SETP A,C
MOV #A,UNITB ;SETUP PORT C
MOVB #'C,MSG19A
MOV TIMER,COUNT
JSR PC,TMO ;DO TIMEOUT

JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
```

```
2023
2024          JSR      PC,DRAV          ;SEE IF DRIVE AVAIL
2025          ERROR   45                ;PORT C NOT AVAIL AFTER TMO
2026          .ENDM   SETP
2027
2028          .MACRO  REPTST
2029          : *
2030          : *      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
2031          : *
2032          .ENDM   REPTST
2033
2034          .MACRO  REPTA
2035          : *
2036          : *      THE PREVIOUS TEST IS REPEATED FOR PORT 'B',
2037          : *      BUT THE SEEK IS TO CYLINDER 0
2038          : *
2039          .ENDM   REPTA
2040
2041          .MACRO  HEADER
2042          :
2043          :A & B=0 FOR PORT A OR 1 FOR PORT B
2044          :C & D=A FOR PORT A OR B FOR PORT B
2045          :
2046          .ENDM   HEADER
2047
2048          .MACRO  SUNIT
2049          MOV      $UNIT,RKCS2(R5)
2050          ADD     UNITB,RKCS2(R5)
2051          .ENDM   SUNIT
2052
2053          :
2054          :A & B=0 FOR PORT A OR 1 FOR PORT B
2055          :C & D=A FOR PORT A OR B FOR PORT B
2056          :
2057          .MACRO  TST5      A,B,C,D,E
2058          SETP   A,C
2059          MOV    #5$, $ESCAPE
2060          F.EAB  D.DRA
2061          CHECK  77,100,101,102,<AFTER TIMEOUT>,0,0,0
2062          MOV    #B,UNITB          ;SELECT PORT D BEFORE TIMEOUT OR RELEASE
2063          MOV    #'D,MSG19A        ;SETUP ERROR MSG FOR PORT D
2064          CLR    $ESCAPE
2065          JSR    PC,DRAV          ;SEE IF DRIVE AVAIL
2066          BR     1$                ;BR IF NOT AVAIL
2067          ERROR  103              ;PORT D AVAIL BEFORE TMO OR RELEASE
2068
2069          1$:   BIT    #CERR,HCS1
2070          BNE   6$
2071          ERROR  130              ;CERR NOT SET AFTER SEL DRIVE & DRIVE NOT AVAIL
2072          6$:   MOV    #<D.SPIN!D.VV>,E.A0
2073          MOV    E.A0,E.A1          ;MSG A & B SHOULD ALWAYS RETURN SAME
2074          BIS    E.DDT,E.A1
2075          CLR    E.B0                ;WORD 0 FOR PORT D
2076          CLR    E.B1
2077          MOV    #5$, $ESCAPE
2078          CHECK  104,105,106,107,<WHILE PORT D UNAVAILABLE>,0,0,0
```

```
2079
2080          TST      $PASS
2081          BEQ      8$          ;BR IF FIRST PASS
2082          JMP      5$          ;ELSE EXIT TEST
2083      8$:    CLR      $ESCAPE
2084          MOV      #CCLR,RKCS1(R5)
2085          MOV      #A,UNITB    ;SETUP FOR PORT C AGAIN
2086          MOVB    #'C,MSG19A
2087          MOV      #360,COUNT  ;SETUP 4 SEC TIMEOUT
2088          JSR      PC,CLKON    ;TURN ON CLOCK
2089
2090          JSR      PC,DRAV
2091          ERROR    45          ;PORT C NOT AVAIL AFTER TIMEOUT
2092
2093          MOV      #B,UNITB    ;SELECT PORT D BEFORE TIMEOUT OR RELEASE
2094          MOVB    #'D,MSG19A
2095          JSR      PC,DRAV    ;SEE IF PORT D DRIVE AVAIL
2096          BR       3$          ;BR IF NOT AVAIL
2097          ERROR    103        ;PORT D AVAIL BEFORE TMO OR RELEASE
2098
2099      3$:    MOV      #CCLR,RKCS1(R5)
2100          MOV      $UNIT,R4
2101          ADD     UNITB,R4
2102          JSR      PC,FATT3
2103          ERROR    110        ;NO ATTN ON PORT D TO ALLOW SEIZE
2104
2105          JSR      PC,CLKOF
2106          JSR      PC,DRAV    ;SEE IF PORT D DRIVE AVAIL
2107          ERROR    45          ;PORT D NOT AVAIL
2108
2109          MOV      #5$, $ESCAPE
2110          F.EAB    <D.DSC!D.DRA>
2111          CHECK   77,100,101,102,<AFTER TIMEOUT>,0,0,0
2112          CLR      $ESCAPE
2113          MOV      #A,UNITB    ;SETUP FOR PORT C
2114          MOVB    #'C,MSG19A
2115          JSR      PC,TSTATN   ;TEST FOR ATTN ON PORT C
2116          BR       4$          ;PORT C ATTN SET W/O REQUEST PENDING
2117          ERROR    111
2118
2119      4$:    MOV      #5$, $ESCAPE
2120          MOV      #B,UNITB    ;SETUP FOR PORT D
2121          MOVB    #'D,MSG19A
2122          DRCLR
2123
2124          MOV      #360,R1
2125          SUB     COUNT,R1     ;R1-COUNT=R1
2126          MULT    #17.,R1     ;MULT BY 16.66 MS
2127          TYPE    ,E          ;PORT TIMEOUT
2128          MOV      R1,-(SP)    ;PUSH BINARY ONTO STACK
2129          JSR      PC,$SB2D    ;CONVERT TO ASCII
2130          JSR      PC,$SUPRS   ;TYPE IT
2131          TYPE    ,MSG22      ;MS
2132
2133      5$:    CLR      $ESCAPE
2134          JSR      PC,CLKOF
```

2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190

```
.ENDM TST5

:
:A & B=0 FOR PORT A OR 1 FOR PORT B
:C & D=A FOR PORT A OR B FOR PORT B
:
.MACRO TST10 A,B,C,D,E,F,?G
  SETP A,C
  MOV #F,RKDC(R5) ;SEEK TO CYL F
  QKSEEK
  JSR PC,RDCYLA ;READ CYL ADDR
  CMP CYLADD,#F ;SEE IF CYL F
  BEQ G ;BR IF YES
  MOV #E,FRCYL
  MOV #F,TOCYL
  MOV #F,CALDIF
  MOV #F,RKDC(R5) ;REFRESH RKDC
  ERROR 224 ;DID NOT SEEK TO CYL F

G: MOV #150,COUNT
  JSR PC,TMO ;DO 2.5 SEC TIMEOUT
  JSR PC,FATT2
  ERROR 112 ;ATTN CLEARED AFTER TMO
  JSR PC,DRAV
  ERROR 113 ;PORT C NOT AVAIL AFTER DRIVE CLEAR CMD
.ENDM TST10

:
:A & B=0 FOR PORT A OR 1 FOR PORT B
:C & D=A FOR PORT A OR B FOR PORT B
:
.MACRO TST12 A,B,C,D
  SETP A,C
  QKSEEK
  MOV TIMER,COUNT
  JSR PC,TMO ;DO 1.5 SEC TIMEOUT
  MOV #B,UNITB ;SETUP PORT D
  MOVB #'D,MSG19A
  JSR PC,DRAV
  ERROR 45 ;PORT D NOT AVAIL AFTER TMO

  STDER1 0
  QKSEEK
  MOV TIMER,COUNT
  JSR PC,TMO
  MOV #A,UNITB ;SETUP PORTC
  MOVB #'C,MSG19A
  JSR PC,DRAV
  ERROR 45 ;PORT C NOT AVAIL AFTER TMO

  STDER1 D.DSC
  DRCLR
  MOV TIMER,COUNT
  JSR PC,TMO ;DO 1.5 SEC TIMEOUT
  MOV #B,UNITB ;SETUP PORT D
```

```
2191          MOVB    #'D,MSG19A
2192          JSR     PC,TSTATN
2193          ERROR   114          ;ATTN RESET ON PORT D AFTER DR CLR CMD
2194                                     ;ON PORT C
2195
2196          STDER1  D.DSC
2197 .ENDM      TST12
2198
2199          ;
2200          ;A & B=0 FOR PORT A OR 1 FOR PORT B
2201          ;C & D=A FOR PORT A OR B FOR PORT B
2202          ;
2203          .MACRO  TST14  A,B,C,D
2204                  SETP    A,C
2205                  RELEAS  A,B,C,D
2206                  JSR     PC,TSTATN
2207                  BR     1$
2208          1$:     ERROR   115          ;ATTN SET IN PORT D AFTER RLS OF PORT C
2209                  F.EAB  D.DRA
2210                  CHECK  133,134,135,136,<AFTER RELEASED TO PORT D>,0,0,0
2211
2212          MOV     #A,UNITB          ;ADDRESS PORT C
2213          MOVB   #'C,MSG19A
2214          JSR     PC,TSTATN
2215          SKIP   R,<GOTO NEXT TST>
2216          ERROR   115          ;ATTN SET AFTER RLS ISSUED
2217          STDER2  D.DRA
2218 .ENDM      TST14
2219
2220          ;
2221          ;A & B=0 FOR PORT A OR 1 FOR PORT B
2222          ;C & D=A FOR PORT A OR B FOR PORT B
2223          ;
2224          .MACRO  TST16  A,B,C,D
2225                  RELEAS  B,A,D,C
2226                  STDER2  D.DRA
2227                  JSR     PC,SUBCLR
2228                  ERROR   24          ;CERR AFTER SCLR
2229                  QKSEEK
2230
2231          MOV     #B,UNITB          ;SETUP PORT D
2232          MOVB   #'D,MSG19A
2233          JSR     PC,DRAV          ;SEE IF DRIVE AVAIL
2234          BR     1$
2235          ERROR   103          ;PORT D AVAIL BEFORE TMO OR RELEASE
2236
2237          1$:     BIT     #CERR,HCS1
2238                  BNE    2$
2239                  ERROR   130          ;CERR NOT SET AFTER SEL DRIVE CMD
2240                                     ;& NO DRA
2241          2$:     JSR     PC,TSTATN
2242                  BR     3$
2243                  ERROR   115          ;ATTN SET IN PORT D AFTER RLS FROM PORT C
2244
2245          3$:     MOV     #CCLR,RKCS1(R5)
2246                  RELEAS  A,B,C,D
```

```
2247 JSR PC,TSTATN
2248 ERROR 122 ;NO ATTN IN PORT D AFTER RLS FROM PORT C
2249
2250 STDER2 <D.DSC!D.DRA>
2251 RELEAS B,A,D,C
2252 JSR PC,TSTATN
2253 ERROR 123 ;ATTN CLEARED IN PORT C AFT RLS FROM PORT D
2254
2255 STDER2 <D.DSC!D.DRA>
2256 RELEAS A,B,C,D
2257 DRCLR
2258 MOV TIMER,COUNT
2259 JSR PC,TMO ;DO 1.5 SEC TIMEOUT ON PORT D
2260 JSR PC,TSTATN
2261 SKIP R,<GOTO NEXT TST>
2262 ERROR 144 ;MULT ATTN ON PORT D
2263 .ENDM TST16
2264
2265
2266
2267 :A & B=0 FOR PORT A OR 1 FOR PORT B
2268 :C & D=A FOR PORT A OR B FOR PORT B
2269
2270 .MACRO TST20 A,B,C,D
2271 RELEAS B,A,D,C
2272 STDER2 D.DRA
2273 JSR PC,SUBCLR
2274 ERROR 24 ;CERR AFTER SCLR
2275
2276 MOV #B,UNITB ;SETUP PORT D
2277 MOVB #'D,MSG19A
2278 JSR PC,DRAV ;SEE IF DRIVE AVAIL
2279 BR 1$
2280 ERROR 103 ;PORT D AVAIL BEFORE TMO OR RELEASE
2281
2282 1$: BIT #CERR,HCS1
2283 BNE 2$
2284 ERROR 130 ;CERR NOT SET BY NO DRA
2285
2286 2$: JSR PC,TSTATN
2287 BR 3$
2288 ERROR 115 ;ATTN SET IN PORT D AFTER RLS FROM PORT C
2289
2290 3$: MOV #CCLR,RKCS1(R5)
2291 RELEAS B,A,D,C
2292 RELEAS A,B,C,D
2293 MOV TIMER,COUNT
2294 JSR PC,TMO ;DO 1.5 SEC TIMEOUT ON PORT D
2295 JSR PC,TSTATN
2296 BR 4$
2297 ERROR 115 ;ATTN SET ON PORT D AFTER RLS FROM PORT D
2298
2299 4$: STDER2 0
2300 MOV #A,UNITB ;SETUP PORT C
2301 MOVB #'C,MSG19A
2302 JSR PC,TSTATN
```

```
2303          SKIP      R,<GOTO NEXT TST>
2304          ERROR    115          ;ATTN SET ON PORT C, AFTER RLS FROM PORT C
2305          .ENDM    TST20
2306
2307
2308
2309          ;A & B=0 FOR PORT A OR 1 FOR PORT B
2310          ;C & D=A FOR PORT A OR B FOR PORT B
2311
2312          .MACRO   TST22      A,B,C,D
2313          SETP      A,C
2314          RELEAS   B,A,D,C
2315          STDER2   D.DRA
2316          .ENDM    TST22
2317
2318
2319          ;A & B=0 FOR PORT A OR 1 FOR PORT B
2320          ;C & D=A FOR PORT A OR B FOR PORT B
2321
2322          .MACRO   TST24      A,B,C,D,E
2323          RELEAS   B,A,D,C
2324          JSR      PC,SUBCLR
2325          ERROR    24          ;CERR AFTER SCLR
2326
2327          MOV      #E,RKDC(R5)
2328          MOV      #SEEK,HCS1
2329          JSR      PC,DOCMD          ;DO SEEK CMD & GET CONTR READY
2330          ERROR    131          ;NO RDY AFTER SEEK CMD
2331          RELEAS   A,B,C,D
2332          MOV      T50000,TEMP1
2333          JSR      PC,FATT2
2334          ERROR    152          ;NO ATTN ON PORT D AFTER SEEK & RLS FROM PORT C
2335
2336          F.EAB    <D.DRA!D.DSC>
2337          CHECK    145,146,147,150,<AFTER SEEK & RLS FROM PORT C>,.0,0,0
2338          RELEAS   B,A,D,C
2339          JSR      PC,TSTATN
2340          BR       1$
2341          ERROR    160          ;ATTN ON PORT C AFTER SEEK & RLS FROM PORT C
2342
2343          1$:      F.EAB    D.DRA
2344          CHECK    145,146,147,150,<AFTER SEEK & RLS FROM PORT C>,.0,0,0
2345          .ENDM    TST24
2346
2347
2348          ;A & B=0 FOR PORT A OR 1 FOR PORT B
2349          ;C & D=A FOR PORT A OR B FOR PORT B
2350
2351          .MACRO   TST26      A,B,C,D
2352          JSR      PC,SUBCLR
2353          ERROR    24          ;CERR AFTER SCLR
2354          RELEAS   B,A,D,C
2355          JSR      PC,SUBCLR
2356          ERROR    24          ;CERR AFTER SCLR
2357
2358          MOV      #30.,COUNT
```



```
2359 JSR PC,TMO ;DO 500MS TIMEOUT
2360 JSR PC,DRAV ;RE-SEIZE DRIVE THRU PORT C
2361 ERROR 45 ;PORT C NOT AVAIL AFTER TIMEOUT
2362 STDER1 0
2363 MOV #-1,COUNT
2364 JSR PC,CLKON ;TURN ON CLOCK
2365 MOV #B,UNITB ;SETUP PORT D
2366 JSR PC,DRAV ;SEE IF DRV AVAIL
2367 BR 1$ ;BR IF NO
2368 ERROR 103 ;PORT C AVAIL BEFORE TMO OR RLS
2369 1$: MOVB #'D,MSG19A
2370 MOV $UNIT,R4
2371 ADD UNITB,R4
2372 JSR PC,FATT3
2373 ERROR 110 ;NO ATTN ON PORT D TO ALLOW SEIZE
2374
2375 JSR PC,CLKOF ;TURN CLOCK OFF
2376 COM COUNT ;GET ACTUAL COUNT OF TIMEOUT
2377 CMP COUNT,#35. ;COMPARE COUNT AGAINST APPROX 1 SEC
2378 SKIP GE,<GO TO NEXT TEST>
2379 ERROR 153 ;TIMEOUT DID NOT RE-TRIGGER
2380 .ENDM TST26
2381
2382
2383
2384 ;A & B=0 FOR PORT A OR 1 FOR PORT B
2385 ;C & D=A FOR PORT A OR B FOR PORT B
2386
2387 .MACRO TST30 A,B,C,D
2388 JSR PC,SUBCLR
2389 ERROR 24 ;CERR AFTER SCLR
2390 RELEAS B,A,D,C
2391 JSR PC,SUBCLR
2392 ERROR 24 ;CERR AFTER SCLR
2393
2394 MOV #B,UNITB ;SETUP FOR PORT D
2395 MOVB #'D,MSG19A
2396 JSR PC,DRAV ;PORT D TRIES TO SEIZE THE DRIVE
2397 BR 1$ ;BR IF NOT AVAIL
2398 ERROR 103 ;PORT D AVAIL BEFORE TMO OR RELEASE
2399
2400 1$: MOV #CCLR,RKCS1(R5)
2401 RELEAS A,B,C,D
2402 MOV #A,UNITB ;SETUP FOR PORT C
2403 MOVB #'C,MSG19A
2404 JSR PC,DRAV
2405 BR 2$
2406 ERROR 103 ;PORT C AVAIL BEFORE TMO OR RELEASE
2407
2408 2$: MOV #CCLR,RKCS1(R5)
2409 MOV #-1,COUNT
2410 JSR PC,CLKON
2411 MOV $UNIT,R4
2412 ADD UNITB,R4
2413 JSR PC,FATT3
2414 ERROR 110 ;NO ATTN ON PORT C TO ALLOW SEIZE
```

2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470

```
      JSR      PC,CLKOF      ;TURN CLOCK OFF
      COM      COUNT        ;GET ACTUAL COUNT OF TIMEOUT
      CMP      COUNT,#35.   ;COMPARE AGAINST APPROX 1 SEC
      SKIP    GE,<GO TO NEXT TST>
      ERROR   153          ;TIMEOUT DID NOT RE-TRIGGER
.ENDM  TST30

:
:A & B=0 FOR PORT A OR 1 FOR PORT B
:C & D=A FOR PORT A OR B FOR PORT B
:
.MACRO TST32  A,B,C,D
      RELEAS  B,A,D,C
      JSR     PC,SUBCLR
      ERROR   24          ;CERR AFTER SCLR

      INC     UNLD        ;USED FOR VALID HALT
      MOV     #2$, $ESCAPE
      MOV     #UNLOAD,HCS1 ;UNLOAD CMD
      BIS     $TMP4,HCS1
      MOV     HCS1,RKCS1(R5)
      MOV     T10,TEMP1   ;SETUP TIMEOUT
      JSR     PC,FATT2    ;FIND ATTN
      ERROR   73          ;NO ATTN AFTER UNLD CMD

      MOV     #300.,COUNT
      JSR     PC,TMO      ;DO 5 SEC DELAY
      MOV     #B,UNITB   ;SETUP FOR PORT D
      MOVB    #'D,MSG19A
      JSR     PC,DRAV
      BR      1$
      ERROR   155        ;PORT D AVAIL BEFORE RLS WHEN UNLOADED
                          ;UNLOAD DID NOT INHIBIT TIMERS

1$:   MOV     #CCLR,RKCS1(R5)
2$:   RELEAS  A,B,C,D
      CLR     $ESCAPE
      QKSRT
      CLR     UNLD
.ENDM  TST32

:
:A & B=0 FOR PORT A OR 1 FOR PORT B
:C & D=A FOR PORT A OR B FOR PORT B
:
.MACRO TST32A A,B,C,D
      RELEAS  B,A,D,C
      JSR     PC,SUBCLR
      ERROR   24          ;CERR AFTER SCLR

      MOV     #5$, $ESCAPE
      MOV     LC,RKDC(R5) ;SEEK TO LAST CYL
      QKSEEK
      DRCLR
```

```
2471          MOV      #RECAL,HCS1      ;RECAL COMMAND
2472          BIS      $TMP4,HCS1
2473          MOV      HCS1,RKCS1(R5)
2474          MOV      $UNIT,R4
2475          ADD      UNITB,R4
2476          1$:      BITB     ATTN(R4),RKASOF+1(R5) ;SEE IF ATTN SET
2477          BNE      3$              ;BR IF YES
2478
2479          MOV      #B,UNITB          ;SETUP FOR PORT D
2480          MOVB     #'D,MSG19A
2481          JSR      PC,DRAV          ;SEE IF DRV AVAIL
2482          BR       2$              ;RETURN HERE IF NO
2483          ERROR    177              ;PORT D AVAIL
2484
2485          2$:      MOV      #CCLR,RKCS1(R5) ;RECAL DID NOT INHIBIT TIMERS
2486          MOV      #A,UNITB
2487          MOVB     #'C,MSG19A      ;SETUP PORT C
2488          JSR      PC,DRAV          ;SEE IF DRV AVAIL
2489          ERROR    203              ;PORT C NOT REMAIN AVAIL DURING RECAL
2490          BR       1$
2491
2492          3$:      JSR      PC,GSTAT
2493          BIT      #D.PIP,HMR2      ;SEE IF ANY MOTION
2494          BEQ      4$              ;BR IF NO
2495          ERROR    72              ;PIP SET AFTER ATTN REC'D FROM RECAL
2496          4$:      MOV      #120,COUNT
2497          JSR      PC,TMO           ;DO 2 SEC DLY
2498          MOV      #B,UNITB          ;SETUP FOR PORT D
2499          MOVB     #'D,MSG19A
2500          JSR      PC,DRAV          ;SEE IF DRV NOW AVAIL
2501          ERROR    45              ;PORT D NOT AVAIL AFTER TMO
2502
2503          5$:      CLR      $ESCAPE
2504          .ENDM    TST32A
2505
2506          .NLIST  MD
2507
2508
```

```
2509          .SBTTL  COMMON TAGS
2510
2511          ::*****
2512          :*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
2513          :*USED IN THE PROGRAM.
2514
2515          001100          .=1100
2516 001100          SCMTAG:          ;;START OF COMMON TAGS
2517 001100 000000          .WORD          0
2518 001102 000          $TSTNM: .BYTE          0          ;;CONTAINS THE TEST NUMBER
2519 001103 000          $ERFLG: .BYTE          0          ;;CONTAINS ERROR FLAG
2520 001104 000000          $ICNT:  .WORD          0          ;;CONTAINS SUBTEST ITERATION COUNT
2521 001106 000000          $LPADR: .WORD          0          ;;CONTAINS SCOPE LOOP ADDRESS
2522 001110 000000          $LPERR: .WORD          0          ;;CONTAINS SCOPE RETURN FOR ERRORS
2523 001112 000000          $ERTTL: .WORD          0          ;;CONTAINS TOTAL ERRORS DETECTED
2524 001114 000          $ITEMB: .BYTE          0          ;;CONTAINS ITEM CONTROL BYTE
2525 001115 001          $ERMAX: .BYTE          1          ;;CONTAINS MAX. ERRORS PER TEST
2526 001116 000000          $ERRPC: .WORD          0          ;;CONTAINS PC OF LAST ERROR INSTRUCTION
2527 001120 000000          $GDADR: .WORD          0          ;;CONTAINS ADDRESS OF 'GOOD' DATA
2528 001122 000000          $BDADR: .WORD          0          ;;CONTAINS ADDRESS OF 'BAD' DATA
2529 001124 000000          $GDDAT: .WORD          0          ;;CONTAINS 'GOOD' DATA
2530 001126 000000          $BDDAT: .WORD          0          ;;CONTAINS 'BAD' DATA
2531 001130 000000          .WORD          0          ;;RESERVED--NOT TO BE USED
2532 001132 000000          .WORD          0
2533 001134 000          $AUTOB: .BYTE          0          ;;AUTOMATIC MODE INDICATOR
2534 001135 000          $INTAG: .BYTE          0          ;;INTERRUPT MODE INDICATOR
2535 001136 000000          .WORD          0
2536 001140 177570          $SWR:  .WORD          DSWR          ;;ADDRESS OF SWITCH REGISTER
2537 001142 177570          $DISPLAY: .WORD          DDISP          ;;ADDRESS OF DISPLAY REGISTER
2538 001144 177560          $TKS:  177560          ;;TTY KBD STATUS
2539 001146 177562          $TKB:  177562          ;;TTY KBD BUFFER
2540 001150 177564          $TPS:  177564          ;;TTY PRINTER STATUS REG. ADDRESS
2541 001152 177566          $TPB:  177566          ;;TTY PRINTER BUFFER REG. ADDRESS
2542 001154 000          $NULL:  .BYTE          0          ;;CONTAINS NULL CHARACTER FOR FILLS
2543 001155 002          $FILLS: .BYTE          2          ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
2544 001156 012          $FILLC: .BYTE          12          ;;INSERT FILL CHARS. AFTER A 'LINE FEED'
2545 001157 000          $TPFLG: .BYTE          0          ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
2546 001160 000000          $TMP0: .WORD          0          ;;USER DEFINED
2547 001162 000000          $TMP1: .WORD          0          ;;USER DEFINED
2548 001164 000000          $TMP2: .WORD          0          ;;USER DEFINED
2549 001166 000000          $TMP3: .WORD          0          ;;USER DEFINED
2550 001170 000000          $TMP4: .WORD          0          ;;USER DEFINED
2551 001172 000000          $TMP5: .WORD          0          ;;USER DEFINED
2552 001174 000000          $TIMES: 0          ;;MAX. NUMBER OF ITERATIONS
2553 001176 000000          $ESCAPE: 0          ;;ESCAPE ON ERROR ADDRESS
2554 001200 177607 000377          $BELL:  .ASCIZ  <207><377><377>          ;;CODE FOR BELL
2555 001204 077          $QUES:  .ASCII  /?/          ;;QUESTION MARK
2556 001205 015          $CRLF:  .ASCII  <15>          ;;CARRIAGE RETURN
2557 001206 000012          $LF:   .ASCIZ  <12>          ;;LINE FEED
2558          ::*****
2559          .SBTTL  APT MAILBOX-ETABLE
2560
2561          ::*****
2562          .EVEN
2563 001210          $MAIL:          ;;APT MAILBOX
2564 001210 000000          $MSGTY: .WORD  AMSGTY  ;;MESSAGE TYPE CODE
```

2565	001212	000000	\$FATAL: .WORD	AFATAL	::FATAL ERROR NUMBER
2566	001214	000000	\$TESTN: .WORD	ATESTN	::TEST NUMBER
2567	001216	000000	\$PASS: .WORD	APASS	::PASS COUNT
2568	001220	000000	\$DEVCT: .WORD	ADEVCT	::DEVICE COUNT
2569	001222	000000	\$UNIT: .WORD	AUNIT	::I/O UNIT NUMBER
2570	001224	000000	\$MSGAD: .WORD	AMSGAD	::MESSAGE ADDRESS
2571	001226	000000	\$MSGLG: .WORD	AMSGLG	::MESSAGE LENGTH
2572	001230		\$ETABLE:		::APT ENVIRONMENT TABLE
2573	001230	000	\$ENV: .BYTE	AENV	::ENVIRONMENT BYTE
2574	001231	000	\$ENVM: .BYTE	AENVM	::ENVIRONMENT MODE BITS
2575	001232	000000	\$SWREG: .WORD	ASWREG	::APT SWITCH REGISTER
2576	001234	000000	\$USWR: .WORD	AUSWR	::USER SWITCHES
2577	001236	000000	\$CPUOP: .WORD	ACPUOP	::CPU TYPE,OPTIONS
2578			::*		BITS 15-11=CPU TYPE
2579			::*		11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
2580			::*		11/70=06,PDQ=07,Q=10
2581			::*		BIT 10=REAL TIME CLOCK
2582			::*		BIT 9=FLOATING POINT PROCESSOR
2583			::*		BIT 8=MEMORY MANAGEMENT
2584	001240	000	\$MAMS1: .BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
2585	001241	000	\$MTYP1: .BYTE	AMTYP1	::MEM. TYPE,BLK#1
2586			::*		MEM.TYPE BYTE -- (HIGH BYTE)
2587			::*		900 NSEC CORE=001
2588			::*		300 NSEC BIPOLAR=002
2589			::*		500 NSEC MOS=003
2590	001242	000000	\$MADR1: .WORD	AMADR1	::HIGH ADDRESS,BLK#1
2591			::*		MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
2592	001244	000	\$MAMS2: .BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTE
2593	001245	000	\$MTYP2: .BYTE	AMTYP2	::MEM. TYPE,BLK#2
2594	001246	000000	\$MADR2: .WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
2595	001250	000	\$MAMS3: .BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
2596	001251	000	\$MTYP3: .BYTE	AMTYP3	::MEM. TYPE,BLK#3
2597	001252	000000	\$MADR3: .WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
2598	001254	000	\$MAMS4: .BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
2599	001255	000	\$MTYP4: .BYTE	AMTYP4	::MEM. TYPE,BLK#4
2600	001256	000000	\$MADR4: .WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
2601	001260	000000	\$VECT1: .WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
2602	001262	000000	\$VECT2: .WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
2603	001264	177440	\$BASE: .WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
2604	001266	000000	\$DEVN: .WORD	ADEVN	::DEVICE MAP
2605	001270	000000	\$CDW1: .WORD	ACDW1	::CONTROLLER DESCRIPTION WORD#1
2606	001272	000000	\$CDW2: .WORD	ACDW2	::CONTROLLER DESCRIPTION WORD#2
2607	001274	000000	\$DDW0: .WORD	ADDW0	::DEVICE DESCRIPTOR WORD#0
2608	001276	000000	\$DDW1: .WORD	ADDW1	::DEVICE DESCRIPTOR WORD#1
2609	001300	000000	\$DDW2: .WORD	ADDW2	::DEVICE DESCRIPTOR WORD#2
2610	001302	000000	\$DDW3: .WORD	ADDW3	::DEVICE DESCRIPTOR WORD#3
2611	001304	000000	\$DDW4: .WORD	ADDW4	::DEVICE DESCRIPTOR WORD#4
2612	001306	000000	\$DDW5: .WORD	ADDW5	::DEVICE DESCRIPTOR WORD#5
2613	001310	000000	\$DDW6: .WORD	ADDW6	::DEVICE DESCRIPTOR WORD#6
2614	001312	000000	\$DDW7: .WORD	ADDW7	::DEVICE DESCRIPTOR WORD#7
2615	001314		\$ETEND:		
2616			.MEXIT		
2617		177440	ABASE=	177440	::DEFAULT BUSS ADDRESS
2618	001314	000210	RKVEC:	210	::DEFAULT CONTROLLER INTERRUPT VECTOR
2619	001316	000240	RKPRI:	PR5	::PRIORITY
2620	001320	172540	PKS:	172540	::P-CLOCK STATUS REG

2621	001322	172542	PKSB:	172542	:P-CLOCK SET BUFFER
2622	001324	172544	PKRB:	172544	:P-CLOCK READ BUFFER
2623	001326	177546	LKS:	177546	:L-CLOCK STATUS REG.
2624					
2625	001330	000100	LCVEC:	100	:L-CLOCK INTERRUPT VECTOR
2626	001332	000104	PCVEC:	104	:P-CLOCK INTERRUPT VECTOR.
2627					
2628		000114	MEMVEC=	114	:MEMORY PARITY VECTOR
2629		172100	MEMBAS=	172100	:MEMORY PARITY OPTION (CSR START ADDR
2630	001334	000000	TRAPPC:	0	:PC FOR MEM CHECK ENABLE TRAP
2631					
2632	001336	000000	PARAM:	0	:1 FOR 220 START, NO DEFAULT
2633	001340	000000	FTITLE:	0	:FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
2634					
2635	001342	000000	DRVPTR:	0	:CONTAINS THE POINTER TO THE DRIVE FLAG
2636					: (DRIV0-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
2637		000040	SPBAR=	40	:SPACEBAR
2638	001344	000000	FRCYL:	0	:FROM CYLINDER
2639	001346	000000	TOCYL:	0	:TO CYLINDER
2640	001350	000000	CCYL:	0	:CURRENT CYL, USED IN N SQUARE TEST
2641	001352	000000	PCYL:	0	:PREV CYL., USED IN N SQUARE TEST
2642	001354	000000	CALDIF:	0	:CALC CYL DIFF USED IN N SQUARE TEST
2643	001356	000000	CYLDIF:	0	:CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
2644	001360	000000	CYLADD:	0	:CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
2645	001362	000000	CALADD:	0	:CYL ADDR USED IN FHDTAB ROUTINE
2646					
2647	001364	000074	HZ:	60.	:60 FOR 60 CPS
2648					:50 FOR 50 CPS
2649	001366	000000	COUNT:	0	:LOADED TO 50 OR 60 TO COUNT TO 1 SEC
2650					:OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
2651	001370	000000	SEC:	0	:SECOND COUNTER
2652	001372	000000	TIMUP:	0	:FLAG TO INDICATE TIME IS UP
2653	001374	000000	SECTOR:	0	:SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
2654					
2655	001376	000001	T1:	1	:TIMEOUT CONSTANTS
2656	001400	000012	T10:	10.	
2657	001402	000062	T50:	50.	
2658	001404	000764	T500:	500.	
2659	001406	010000	T100:	10000	
2660	001410	011610	T5000:	5000.	
2661	001412	141520	T50000:	50000.	
2662					
2663					
2664	001414	000000	WD1:	0	:ACTUAL HEADER/DATA WORD
2665	001416	000000	WD2:	0	:EXPECTED DATA WORD
2666	001420	000000	HEAD:	0	:HEAD NUMBER
2667	001422	000000	HEADA:	0	:HEAD # FROM H.B3, RT. JUSTIFIED
2668	001424	000000	HD1:	0	:SHIFTED HEAD# FOR FORMATTER ROUTINE
2669	001426	000000	FORMAT:	0	:FORMAT TYPE
2670	001430	000000	FMT1:	0	:SHIFTED FORMAT FOR FORMATTER ROUTINE
2671	001432	000000	WDCNT:	0	:WORD COUNT
2672					
2673	001434	000000	DATA0:	0	:ALL 0'S
2674	001436	052525	DATA01:	52525	:0101 PATT
2675	001440	177777	DATA1:	177777	:ALL 1'S
2676	001442	133467	DPAT1:	133467	


```
2733
2734 005352 000000      TEMP1: 0          ;TEMPORARY STORAGE.
2735 005354 000000      TEMP2: 0
2736 005356 000000      TEMP3: 0
2737 005360 000000      TEMP4: 0
2738 005362 000000      TEMP5: 0
2739
2740      ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
2741
2742 005364 000000      H.A0: 0
2743 005366 000000      H.B0: 0
2744 005370 000000      H.A1: 0
2745 005372 000000      H.B1: 0
2746 005374 000000      H.A2: 0
2747 005376 000000      H.B2: 0
2748 005400 000000      H.A3: 0
2749 005402 000000      H.B3: 0
2750
2751      ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.
2752
2753 005404 000000      E.A0: 0
2754 005406 000000      E.B0: 0
2755 005410 000000      E.A1: 0
2756 005412 000000      E.B1: 0
2757 005414 000000      E.A2: 0
2758 005416 000000      E.B2: 0
2759 005420 000000      E.A3: 0
2760 005422 000000      E.B3: 0
2761
2762      ;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.
2763
2764      000001      T.A2=BIT0          ;TEST MSG A2 IF SET
2765      000002      T.B2=BIT1
2766      000004      T.B3=BIT2
2767
2768      ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
2769
2770 005424 000000      DDUMP: 0          ;FLAG - SET WHEN IN DDP DUMP MODE
2771 005426 000000      DDPCH: 0         ;FLAG - SET WHEN IN DDP CHAIN MODE
2772 005430 000000      ACT11: 0        ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
2773 005432 000000      PPTP: 0         ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
2774 005434 000000      DRIVS: 0        ;CONTAINS THE NUMBER OF DRIVES PRESENT
2775
2776      ;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
2777      ;IS PRESENT AND IS TO BE TESTED.
2778      ;ONLY DRIVES 0,2,4,6 CAN BE TESTED.
2779
2780 005436 000000      DRIV0: 0         ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
2781 005440 000000      DRIV1: 0         ;FOR DRIVE 1
2782 005442 000000      DRIV2: 0         ;FOR DRIVE 2
2783 005444 000000      DRIV3: 0         ;FOR DRIVE 3
2784 005446 000000      DRIV4: 0         ;FOR DRIVE 4
2785 005450 000000      DRIV5: 0         ;FOR DRIVE 5
2786 005452 000000      DRIV6: 0         ;FOR DRIVE 6
2787 005454 000000      DRIV7: 0         ;FOR DRIVE 7
2788
```


CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 D 5
APT MAILBOX-ETABLE PAGE 56

SEQ 0055

2789 005456 000000
2790 005460 000000
2791 005462 000000
2792 005464 000000
2793
2794 005466 000132

LCLKF: 0
PCLKF: 0
SIZFLG: 0
UNITB: 0
TIMER: 90.

:L-CLOCK FLAG PRESENT FLAG
:P-CLOCK FLAG PRESENT FLAG
:SET IF DEFAULT DO SIZING IN TEST 1
:0 FOR PORT A: DRIVES 0,2,4,6
:1 FOR PORT B: DRIVES 1,3,5,7
:TIMER FOR 1.5 SEC

2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850

005470

005470 057175
005472 062624
005474 065576
005476 066400

005500 057414
005502 062624
005504 065576
005506 066400

005510 057435
005512 062624
005514 065576
005516 066400

005520 057456
005522 062624
005524 065576
005526 066400

005530 000000
005532 000000
005534 000000
005536 000000

005540 057722
005542 062624
005544 065576
005546 066400

005550 057776
005552 062624
005554 065576
005556 066400

.SBTTL ERROR POINTER TABLE
:*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
:*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
:*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
:*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
:*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

:* EM ::POINTS TO THE ERROR MESSAGE
:* DH ::POINTS TO THE DATA HEADER
:* DT ::POINTS TO THE DATA
:* DF ::POINTS TO THE DATA FORMAT

\$ERRTB:
:ERROR 1 :DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
EM2
DH1
DT1
DF1
:ERROR 2 :DETECTED MDS
EM5
DH1
DT1
DF1
:ERROR 3 :DETECTED UFE
EM6
DH1
DT1
DF1
:ERROR 4 :DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
EM7
DH1
DT1
DF1
:ERROR 5
0
0
0
0
:ERROR 6 :DR NOT PRESENT BUT SPECIFIED BY OPERATOR
EM9
DH1
DT1
DF1
:ERROR 7 :ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
EM10
DH1
DT1
DF1

2851			:ERROR 10	
2852	005560	060061	EM11	:DRA & NED BOTH SET
2853	005562	062624	DH1	
2854	005564	065576	DT1	
2855	005566	066400	DF1	
2856			:ERR 11	
2857	005570	060125	EM12	:NO RDY
2858	005572	064366	DH27	:AFTER WRITE DATA CMD
2859	005574	065576	DT1	
2860	005576	066530	DF10	
2861			:ERR 12	
2862	005600	060521	EM21	:CERR SET
2863	005602	064366	DH 7	
2864	005604	065576	DT1	
2865	005606	066530	DF10	
2866			:ERR 13	
2867	005610	060125	EM12	:NO RDY
2868	005612	064336	DH26	:AFTER READ DATA CMD
2869	005614	065576	DT1	
2870	005616	066530	DF10	
2871			:ERR 14	
2872	005620	060521	EM21	:CERR SET
2873	005622	064336	DH26	
2874	005624	065576	DT1	
2875	005626	066530	DF10	
2876			:ERR 15	
2877	005630	060125	EM12	:NO RDY
2878	005632	064516	DH32	:AFTER WRITE CHECK CMD
2879	005634	065576	DT1	
2880	005636	066530	DF10	
2881			:ERR 16	
2882	005640	062304	EM80	:WRITE CHECK ERROR SET
2883	005642	064516	DH32	:AFTER WRITE CHECK CMD
2884	005644	065710	DT6	
2885	005646	066420	DF3	
2886			:ERR 17	
2887	005650	060205	EM14	:UNEXP MEM PARITY TRAP
2888	005652	063262	DH8	:TEST #, TRAP PC
2889	005654	065636	DT3	
2890	005656	066414	DF2	
2891			:ERR 20	
2892	005660	062343	EM82	:READ DATA NOT COMPARE WITH WRITE DATA
2893	005662	064336	DH26	:AFTER READ DATA CMD
2894	005664	065710	DT6	
2895				
2896	005666	066420	DF3	
2897			:ERR 21	
2898	005670	062415	EM83	:DATA CHECK ERROR
2899	005672	064336	DH26	
2900	005674	065576	DT1	
2901				
2902	005676	066530	DF10	
2903			:ERR 22	
2904	005700	060521	EM21	:CERR SET
2905	005702	064516	DH32	:AFTER WRITE CHECK CMD
2906	005704	065576	DT1	

2907	005706	066530		DF10	
2908			:ERR 23	EM18	:MSG B0 ERROR
2909	005710	060436		DH27	:AFTER WRITE DATA CMD
2910	005712	064366		DT13	
2911	005714	066134		DF21	
2912	005716	066660			
2913			:ERROR 24	EM21	:CERR SET
2914	005720	060521		DH21	:AFTER SCLR
2915	005722	064136		DT1	
2916	005724	065576		DF10	
2917	005726	066530			
2918			:ERR 25	EM20	:MSG B1 ERROR
2919	005730	060500		DH27	
2920	005732	064366		DT13	
2921	005734	066134		DF21	
2922	005736	066660			
2923			:ERR 26	EM18	
2924	005740	060436		DH26	:AFTER READ DATA CMD
2925	005742	064336		DT13	
2926	005744	066134		DF21	
2927	005746	066660			
2928			:ERROR 27	EM24	:VOL VALID NOT SET
2929				DH19	:AFTER PACK CMD
2930	005750	060750		DT1	
2931	005752	064060		DF10	
2932	005754	065576			
2933	005756	066530			
2934			:ERR 30	EM20	:MSG B1 ERROR
2935	005760	060500		DH26	:AFTER READ DATA CMD.
2936	005762	064336		DT13	
2937	005764	066134		DF21	
2938	005766	066660			
2939			:ERR 31	EM18	:MSG B0 ERROR
2940	005770	060436		DH32	:AFTER WRITE CHECK CMD
2941	005772	064516		DT13	
2942	005774	066134		DF21	
2943	005776	066660			
2944			:ERR 32	EM20	:MSG B1 ERROR
2945	006000	060500		DH32	
2946	006002	064516		DT13	
2947	006004	066134		DF21	
2948	006006	066660			
2949			:ERR 33	EM17	:A0 ERROR
2950	006010	060415		DH22	:AFTER DRIVE CLR CMD
2951	006012	064164		DT13	
2952	006014	066134		DF21	
2953	006016	066660			
2954			:ERR 34	EM18	:B0 ERROR
2955	006020	060436		DH22	
2956	006022	064164		DT13	
2957	006024	066134		DF21	
2958	006026	066660			
2959			:ERR 35	EM19	
2960	006030	060457		DH22	
2961	006032	064164		DT13	
2962	006034	066134			

2963	006036	066660		DF21	
2964			:ERR 36	EM20	
2965	006040	060500		DH22	
2966	006042	064164		DT13	
2967	006044	066134		DF21	
2968	006046	066660			
2969			:ERR 37	EM17	:A0 ERROR
2970	006050	060415		DH39	:AFTER WRITE HEADER CMD
2971	006052	065004		DT13	
2972	006054	066134		DF21	
2973	006056	066660			
2974			:ERR 40	EM18	
2975	006060	060436		DH39	
2976	006062	065004		DT13	
2977	006064	066134		DF21	
2978	006066	066660			
2979			:ERR 41	EM19	
2980	006070	060457		DH39	
2981	006072	065004		DT13	
2982	006074	066134		DF21	
2983	006076	066660			
2984			:ERR 42	EM20	
2985	006100	060500		DH39	
2986	006102	065004		DT13	
2987	006104	066134		DF21	
2988	006106	066660			
2989			:ERR 43	EM26	:BSE ERROR IN WRITE CMD NOT ON BSE TABLE
2990	006110	061055		DH27	:AFTER WRITE DATA CMD
2991	006112	064366		DT1	
2992	006114	065576		DF10	
2993	006116	066530			
2994			:ERR 44	EM15	:WCE AT CYL 411,TRK 2, SEC 21
2995	006120	060243		DH1	
2996	006122	062624		DT1	
2997	006124	065576		DF4	
2998	006126	066444			
2999			:ERR 45	EM31	:PORT NOT AVAIL
3000	006130	061450		DH5	:AFTER TIMEOUT
3001	006132	063026		DT1	
3002	006134	065576		DF10	
3003	006136	066530			
3004			:ERR 46	EM25	:DETECTED 10 BAD SECTORS
3005	006140	061003		DH27	:AFTER WRITE DATA CMD.
3006	006142	064366		DT1	
3007	006144	065576		DF10	
3008	006146	066530			
3009			:ERROR 47	EM39	:CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3010	006150	061761		DH17	:AFTER RECAL CMD
3011	006152	064007		DT14	
3012	006154	066214		DF22	
3013	006156	066714			
3014			:ERROR 50	EM40	:CYL ADDR IN RKMR3 NOT CLEARED
3015	006160	062030		DH17	:AFTER RECAL CMD
3016	006162	064007		DT14	
3017	006164	066214		DF22	
3018	006166	066714			

Line No.	Code 1	Code 2	Error Code	Description
3019			:ERR 51	
3020	006170	062543	EM93	:WRONG CYL# IN HEADER WORD (MISPOSITION)
3021	006172	064313	DH25	:AFTER SEEK CMD
3022	006174	066070	DT9	
3023	006176	066634	DF20	
3024			:ERR 52	
3025	006200	060415	EM17	:MSG A0 ERROR
3026	006202	064366	DH27	:AFTER WRITE DATA CMD
3027	006204	066134	DT13	
3028	006206	066660	DF21	
3029			:ERR 53	
3030	006210	060457	EM19	:MSG A1 ERROR
3031	006212	064366	DH27	
3032	006214	066134	DT13	
3033	006216	066660	DF21	
3034			:ERR 54	
3035	006220	060415	EM17	:MSG A0 ERROR
3036	006222	064336	DH26	:AFTER READ DATA CMD
3037	006224	066134	DT13	
3038	006226	066660	DF21	
3039			:ERROR 55	
3040	006230	060163	EM13	:NO ATTN
3041	006232	064007	DH17	:AFTER RECAL CMD
3042	006234	065576	DT1	
3043	006236	066530	DF10	
3044			:ERR 56	
3045	006240	060457	EM19	:MSG A1 ERROR
3046	006242	064336	DH26	
3047	006244	066134	DT13	
3048	006246	066660	DF21	
3049			:ERR 57	
3050	006250	060415	EM17	:MSG A0 ERROR
3051	006252	064516	DH32	:AFTER WRITE CHECK CMD
3052	006254	066134	DT13	
3053	006256	066660	DF21	
3054			:ERR 60	
3055	006260	060457	EM19	:MSG A1 ERROR
3056	006262	064516	DH32	
3057	006264	066134	DT13	
3058	006266	066660	DF21	
3059			:ERR 61	
3060	006270	060415	EM17	:A0 ERROR
3061	006272	064437	DH30	:AFTER READ HEADER CMD
3062	006274	066134	DT13	
3063	006276	066660	DF21	
3064			:ERR 62	
3065	006300	060436	EM18	
3066	006302	064437	DH30	
3067	006304	066134	DT13	
3068	006306	066660	DF21	
3069			:ERR 63	
3070	006310	060457	EM19	
3071	006312	064437	DH30	
3072	006314	066134	DT13	
3073	006316	066660	DF21	
3074			:ERR 64	

3075	006320	060500		EM20	
3076	006322	064437		DH30	
3077	006324	066134		DT13	
3078	006326	066660		DF21	
3079			:ERR 65		
3080	006330	061134		EM27	:DETECTED BSE IN READ BUT NOT IN WRITE CMD.
3081	006332	062624		DH1	
3082	006334	065576		DT1	
3083	006336	066400		DF1	
3084			:ERR 66		
3085	006340	060436		EM18	:BO ERROR
3086	006342	064007		DH17	:AFTER RECAL CMD
3087	006344	066134		DT13	
3088	006346	066660		DF21	
3089			:ERR 67		
3090	006350	060500		EM20	
3091	006352	064007		DH17	
3092	006354	066134		DT13	
3093	006356	066660		DF21	
3094			:ERR 70		
3095	006360	062236		EM74	:RTZ NOT SET
3096	006362	065073		DH41	:DURING RECAL CMD
3097	006364	065576		DT1	
3098	006366	066530		DF10	
3099			:ERR 71		
3100	006370	061450		EM31	:PORT NOT AVAIL
3101	006372	064550		DH35	:AFTER RLS
3102	006374	065576		DT1	
3103	006376	066530		DF10	
3104			:ERR 72		
3105	006400	061427		EM30	:PIP SET
3106	006402	062762		DH4	:AFTER ATTN REC'D FROM RECAL
3107	006404	065576		DT1	
3108	006406	066530		DF10	
3109			:ERR 73		
3110	006410	060163		EM13	:NO ATTN
3111	006412	064033		DH18	:AFTER UNLOAD CMD
3112	006414	065576		DT1	
3113	006416	066530		DF10	
3114			:ERR 74		
3115	006420	060163		EM13	:NO ATTN
3116	006422	063356		DH10	:AT END OF HEAD LOADING
3117	006424	065576		DT1	
3118	006426	066530		DF10	
3119			:ERR 75		
3120	006430	060543		EM22	:NO DRIVS IN \$DEV
3121	006432	062624		DH1	
3122	006434	065576		DT1	
3123	006436	066400		DF1	
3124			:ERR 76		
3125	006440	060650		EM23	:NO DRIVS ON BUSS
3126	006442	062624		DH1	
3127	006444	065576		DT1	
3128	006446	066400		DF1	
3129			:ERR 77		
3130	006450	060415		EM17	:AO ERROR

3131	006452	063026	DH5	
3132	006454	066134	DT13	
3133	006456	066660	DF21	
3134				:ERR 100
3135	006460	060436	EM18	
3136	006462	063026	DH5	
3137	006464	066134	DT13	
3138	006466	066660	DF21	
3139				:ERR 101
3140	006470	060457	EM19	
3141	006472	063026	DH5	
3142	006474	066134	DT13	
3143	006476	066660	DF21	
3144				:ERR 102
3145	006500	060500	EM20	
3146	006502	063026	DH5	
3147	006504	066134	DT13	
3148	006506	066660	DF21	
3149				:ERR 103
3150	006510	061473	EM32	
3151	006512	063577	DH14	
3152	006514	065576	DT1	
3153	006516	066530	DF10	
3154				:ERR 104
3155	006520	060415	EM17	
3156	006522	063631	DH15	
3157	006524	066134	DT13	
3158	006526	066660	DF21	
3159				:ERR 105
3160	006530	060436	EM18	
3161	006532	063631	DH15	
3162	006534	066134	DT13	
3163	006536	066660	DF21	
3164				:ERR 106
3165	006540	060457	EM19	
3166	006542	063631	DH15	
3167	006544	066134	DT13	
3168	006546	066660	DF21	
3169				:ERR 107
3170	006550	060500	EM20	
3171	006552	063631	DH15	
3172	006554	066134	DT13	
3173	006556	066660	DF21	
3174				:ERR 110
3175	006560	060163	EM13	
3176	006562	063660	DH16	
3177	006564	065576	DT1	
3178	006566	066530	DF10	
3179				:ERR 111
3180	006570	061512	EM33	
3181	006572	064216	DH23	
3182	006574	065576	DT1	
3183	006576	066530	DF10	
3184				:ERR 112
3185	006600	060163	EM13	
3186	006602	063026	DH5	

:AFTER TMO

:PORT AVAIL
:BEFORE TMO OR RLS

:AO ERROR
:WHILE PORT UNAVAIL

:NO ATTN
:TO ALLOW PORT TO SEIZE

:ATTN SET
:W/O REQUEST PENDING

:NO ATTN
:AFT TMO

3187	006604	065576	DT1	
3188	006606	066530	DF10	
3189			:ERR 113	
3190	006610	061450	EM31	:PORT NO AVAIL
3191	006612	064164	DH22	:AFTER DRIVE CLEAR CMD
3192	006614	065576	DT1	
3193	006616	066530	DF10	
3194			:ERR 114	
3195	006620	061535	EM34	:ATTN CLEARED
3196	006622	064246	DH24	:BY DR CLR TO OTHER PORT.
3197	006624	065576	DT1	
3198	006626	066530	DF10	
3199			:ERR 115	
3200	006630	061512	EM33	:ATTN SET
3201	006632	064550	DH35	:AFT RELEASE ISSUED
3202	006634	065576	DT1	
3203	006636	066530	DF10	
3204			:ERROR 116	
3205	006640	060125	EM12	:CONT NOT RDY
3206	006642	064060	DH19	:AFTER PACK CMD
3207	006644	065576	DT1	
3208	006646	066530	DF10	
3209			:ERROR 117	
3210	006650	060125	EM12	:CONT NOT RDY
3211	006652	064103	DH20	:AFTER SEL DR CMD
3212	006654	065576	DT1	
3213	006656	066530	DF10	
3214			:ERROR 120	
3215	006660	060125	EM12	
3216	006662	064136	DH21	:AFTER SUBSYS CLEAR
3217	006664	065576	DT1	
3218	006666	066530	DF10	
3219			:ERROR 121	
3220	006670	060125	EM12	
3221	006672	063303	DH9	:AFTER START SPINDLE CMD
3222	006674	065576	DT1	
3223	006676	066530	DF10	
3224			:ERR 122	
3225	006700	060163	EM13	:NO ATTN
3226	006702	064656	DH37	:AFT RLS & REQUEST PENDING
3227	006704	065576	DT1	
3228	006706	066530	DF10	
3229			:ERR 123	
3230	006710	061535	EM34	:ATTN CLEARED
3231	006712	064550	DH35	:AFTER RELEASE
3232	006714	065576	DT1	
3233	006716	066530	DF10	
3234			:ERROR 124	
3235	006720	060125	EM12	
3236	006722	064007	DH17	:AFTER RECAL CMD
3237	006724	065576	DT1	
3238	006726	066530	DF10	
3239			:ERR 125	
3240	006730	062215	EM73	:CTO SET
3241	006732	062452	EM84	:WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
3242	006734	065576	DT1	

3243	006736	066460	DF5	
3244			:ERR 126	
3245	006740	062263	EM79	:NED SET
3246	006742	062452	EM84	
3247	006744	065576	DT1	
3248	006746	066460	DF5	
3249			:ERR 127	
3250	006750	057414	EM5	:MDS SET
3251	006752	062452	EM84	
3252	006754	065576	DT1	
3253	006756	066460	DF5	
3254			:ERROR 130	
3255	006760	061564	EM35	:CERR NOT SET
3256	006762	064575	DH36	:AFT SEL DRV CMD & NO DRA
3257	006764	065576	DT1	
3258	006766	066530	DF10	
3259			:ERROR 131	
3260	006770	060125	EM12	:NO RDY
3261	006772	064313	DH25	:AFTER SEEK CMD
3262	006774	065576	DT1	
3263	006776	066530	DF10	
3264			:ERROR 132	
3265	007000	060163	EM13	:NO ATTN
3266	007002	064313	DH25	
3267	007004	065576	DT1	
3268	007006	066530	DF10	
3269			:ERR 133	
3270	007010	060415	EM17	:AO ERROR
3271	007012	064550	DH35	:AFTER RLS ISSUED
3272	007014	066134	DT13	
3273	007016	066660	DF21	
3274			:ERR 134	
3275	007020	060436	EM18	
3276	007022	064550	DH35	
3277	007024	066134	DT13	
3278	007026	066660	DF21	
3279			:ERR 135	
3280	007030	060457	EM19	
3281	007032	064550	DH35	
3282	007034	066134	DT13	
3283	007036	066660	DF21	
3284			:ERR 136	
3285	007040	060500	EM20	
3286	007042	064550	DH35	
3287	007044	066134	DT13	
3288	007046	066660	DF21	
3289			:ERROR 137	
3290	007050	061761	EM39	:CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3291	007052	064313	DH25	
3292	007054	065576	DT1	
3293	007056	066530	DF10	
3294			:ERR 140	
3295	007060	060415	EM17	:MSG AO ERROR
3296	007062	065517	DH51	:AFTER SEEK TO SELF
3297	007064	066134	DT13	
3298	007066	066660	DF21	

3299			:ERR 141	
3300	007070	060436		EM18
3301	007072	065517		DH51
3302	007074	066134		DT13
3303	007076	066660		DF21
3304			:ERR 142	
3305	007100	060457		EM19
3306	007102	065517		DH51
3307	007104	066134		DT13
3308	007106	066660		DF21
3309			:ERR 143	
3310	007110	060500		EM20
3311	007112	065517		DH51
3312	007114	066134		DT13
3313	007116	066660		DF21
3314			:ERR 144	
3315	007120	061655		EM37
3316	007122	063026		DH5
3317	007124	065576		DT1
3318	007126	066530		DF10
3319			:ERR 145	
3320	007130	060415		EM17
3321	007132	064732		DH38
3322	007134	066134		DT13
3323	007136	066660		DF21
3324			:ERR 146	
3325	007140	060436		EM18
3326	007142	064732		DH38
3327	007144	066134		DT13
3328	007146	066660		DF21
3329			:ERR 147	
3330	007150	060457		EM19
3331	007152	064732		DH38
3332	007154	066134		DT13
3333	007156	066660		DF21
3334			:ERR 150	
3335	007160	060500		EM20
3336	007162	064732		DH38
3337	007164	066134		DT13
3338	007166	066660		DF21
3339			:ERROR 151	
3340	007170	060125		EM12
3341	007172	064164		DH22
3342	007174	065576		DT1
3343	007176	066530		DF10
3344			:ERR 152	
3345	007200	060163		EM13
3346	007202	064732		DH38
3347	007204	065576		DT1
3348	007206	066530		DF10
3349			:ERR 153	
3350	007210	061706		EM38
3351	007212	065240		DH43
3352	007214	065576		DT1
3353	007216	066530		DF10
3354			:ERROR 154	

:MULT ATTN
:AFTER TMO

:AO ERROR
:AFTER SEEK & IMMED RELEASE

:NO RDY
:AFTER CLEAR CMD

:NO ATTN
:AFTER SEEK & IMMED RLS

:RE-TRIGGER NO GOOD
:AFTER RE-SEIZE MIDWAY THRU TMO

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 B 6
ERROR POINTER TABLE PAGE 67

SEQ 0066

3355 007220 062066
3356 007222 064164
3357 007224 065576
3358 007226 066530
3359

EM55
DH22
DT1
DF10

;ATTN NOT CLEARED

;ERR 155

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 C 6
ERROR POINTER TABLE PAGE 68

SEQ 0067

3360 007230 061361
3361 007232 065376
3362 007234 065576

EM29
DH45
DT1

;PORT AVAIL-TIMERS NOT INHIB.
;BEFORE RLS WHEN UNLOADED

3363	007236	066530		
3364			:ERR 156	DF10
3365	007240	060163		EM13
3366	007242	064313		DH25
3367	007244	065576		DT1
3368	007246	066530		DF10
3369			:ERR 157	
3370	007250	061450		EM31
3371	007252	064313		DH25
3372	007254	065576		DT1
3373	007256	066530		DF10
3374			:ERROR 160	
3375	007260	061512		EM33
3376	007262	064732		DH38
3377	007264	065576		DT1
3378	007266	066530		DF10
3379			:ERR 161	
3380	007270	060415		EM17
3381	007272	064313		DH25
3382	007274	066134		DT13
3383	007276	066660		DF21
3384			:ERR 162	
3385	007300	060436		EM18
3386	007302	064313		DH25
3387	007304	066134		DT13
3388	007306	066660		DF21
3389			:ERR 163	
3390	007310	060457		EM19
3391	007312	064313		DH25
3392	007314	066134		DT13
3393	007316	066660		DF21
3394			:ERR 164	
3395	007320	060500		EM20
3396	007322	064313		DH25
3397	007324	066134		DT13
3398	007326	066660		DF21
3399			:ERR 165	
3400	007330	060415		EM17
3401	007332	063026		DH5
3402	007334	066134		DT13
3403	007336	066660		DF21
3404			:ERR 166	
3405	007340	060436		EM18
3406	007342	063026		DH5
3407	007344	066134		DT13
3408	007346	066660		DF21
3409			:ERR 167	
3410	007350	060457		EM19
3411	007352	063026		DH5
3412	007354	066134		DT13
3413	007356	066660		DF21
3414			:ERR 170	
3415	007360	060500		EM20
3416	007362	063026		DH5
3417	007364	066134		DT13
3418	007366	066660		DF21

:NO ATTN
:AFTER SEEK CMD

:PORT NOT AVAIL
:AFTER SEEK CMD

:ATTN SET
:AFTER SEEK & IMMED RLS

:MSG AO ERROR
:AFTER SEEK CMD

:MSG AO ERROR
:AFTER TMO

3419			:ERROR 171	
3420	007370	060125	EM12	:NO RDY
3421	007372	064437	DH30	:AFTER READ HEADER CMD
3422	007374	065576	DT1	
3423	007376	066530	DF10	
3424			:ERROR 172	
3425	007400	061450	EM31	:PORT NOT AVAIL
3426	007402	064550	DH35	:AFT RELEASE
3427	007404	065576	DT1	
3428	007406	066530	DF10	
3429			:ERROR 173	
3430	007410	062121	EM63	:DLT SET
3431	007412	064437	DH30	
3432	007414	065576	DT1	
3433	007416	066570	DF15	
3434			:ERROR 174	
3435	007420	060521	EM21	:CERR SET
3436	007422	064437	DH30	
3437	007424	065576	DT1	
3438	007426	066570	DF15	
3439			:ERROR 175	
3440	007430	061761	EM39	:CYL DIFF NOT CLEARED
3441	007432	063356	DH10	:AT END OF HEAD LOADING
3442	007434	065576	DT1	
3443	007436	066530	DF10	
3444			:ERROR 176	
3445	007440	062030	EM40	:CYL ADDR NOT CLEARED.
3446	007442	063356	DH10	
3447	007444	065576	DT1	
3448	007446	066530	DF10	
3449			:ERROR 177	
3450	007450	061361	EM29	:PORT AVAIL-TIMERS NOT INHIB
3451	007452	065073	DH41	:DURING RECAL CMD
3452	007454	065576	DT1	
3453	007456	066530	DF10	
3454			:ERROR 200	
3455	007460	060125	EM12	:NO RDY
3456	007462	065004	DH39	:AFTER WRITE HEADER CMD
3457	007464	065576	DT1	
3458	007466	066570	DF15	
3459			:ERROR 201	
3460	007470	060521	EM21	:CERR SET
3461	007472	065004	DH39	
3462	007474	065576	DT1	
3463	007476	066570	DF15	
3464			:ERROR 202	
3465	007500	062142	EM65	:READ HEADER ERROR
3466	007502	062624	DH1	
3467	007504	065754	DT7	
3468	007506	066550	DF14	
3469			:ERROR 203	
3470	007510	061450	EM31	:PORT NOT AVAIL
3471	007512	065073	DH41	:DURING RECAL CMD
3472	007514	065576	DT1	
3473	007516	066530	DF10	
3474			:ERROR 204	

3475	007520	000000	0	
3476	007522	000000	0	
3477	007524	000000	0	
3478	007526	000000	0	
3479			:ERROR 205	
3480	007530	000000	0	
3481	007532	000000	0	
3482	007534	000000	0	
3483	007536	000000	0	
3484			:ERROR 206	
3485	007540	061612	EM36	:CYL ADDR IN RKMR3 INCORRECT
3486	007542	064313	DH25	:AFTER SEEK CMD
3487	007544	066022	DT8	
3488	007546	066504	DF6	
3489			:ERROR 207	
3490	007550	061612	EM36	:CYL ADDR IN RKMR3 INCORRECT
3491	007552	064313	DH25	:AFTER SEEK CMD
3492	007554	065642	DT4	
3493	007556	066504	DF6	
3494			:ERROR 210	
3495	007560	060521	EM21	:CERR SET
3496	007562	064313	DH25	
3497	007564	065576	DT1	
3498	007566	066530	DF10	
3499			:ERR 211	
3500	007570	060415	EM17	:MSG A0 ERROR
3501	007572	064550	DH35	:AFTER RLS ISSUED
3502	007574	066134	DT13	
3503	007576	066660	DF21	
3504			:ERR 212	
3505	007600	060436	EM18	
3506	007602	064550	DH35	
3507	007604	066134	DT13	
3508	007606	066660	DF21	
3509			:ERR 213	
3510	007610	060457	EM19	
3511	007612	064550	DH35	
3512	007614	066134	DT13	
3513	007616	066660	DF21	
3514			:ERR 214	
3515	007620	060500	EM20	
3516	007622	064550	DH35	
3517	007624	066134	DT13	
3518	007626	066660	DF21	
3519			:ERROR 215	
3520	007630	000000	0	
3521	007632	000000	0	
3522	007634	000000	0	
3523	007636	000000	0	
3524			:ERROR 216	
3525	007640	000000	0	
3526	007642	000000	0	
3527	007644	000000	0	
3528	007646	000000	0	
3529			:ERROR 217	
3530	007650	000000	0	

3531	007652	000000	0	
3532	007654	000000	0	
3533	007656	000000	0	
3534			:ERROR 220	
3535	007660	000000	0	
3536	007662	000000	0	
3537	007664	000000	0	
3538	007666	000000	0	
3539			:ERROR 221	
3540	007670	060415	EM17	:MSG A0 ERROR
3541	007672	064007	DH17	
3542	007674	066134	DT13	
3543	007676	066660	DF21	
3544			:ERROR 222	
3545	007700	060457	EM19	:MSG A1 ERROR
3546	007702	064007	DH17	
3547	007704	066134	DT13	
3548	007706	066660	DF21	
3549			:ERROR 223	
3550	007710	060521	EM21	:CERR SET
3551	007712	063110	DH7	:AFT SCLR
3552	007714	065576	DT1	
3553	007716	066530	DF10	
3554			:ERR 224	
3555	007720	061612	EM36	:CYL ADDR IN RKMR3 BAD
3556	007722	063422	DH12	:AFT SEEK CMD
3557	007724	065642	DT4	
3558	007726	066504	DF6	
3559			:ERROR 225	
3560	007730	000000	0	
3561	007732	000000	0	
3562	007734	000000	0	
3563	007736	000000	0	
3564			:ERROR 226	
3565	007740	060125	EM12	:NO RDY
3566	007742	064336	DH26	:AFTER READ DATA CMD
3567	007744	065576	DT1	
3568	007746	066530	DF10	
3569			:ERROR 227	
3570	007750	060521	EM21	:CERR SET
3571	007752	064336	DH26	
3572	007754	065576	DT1	
3573	007756	066570	DF15	
3574			:ERROR 230	
3575	007760	060352	EM16	:CANNOT READ BSE INFO
3576	007762	063513	DH13	:ON SEC 10, 12, 14, 16, 18, 20
3577	007764	065576	DT1	
3578	007766	066610	DF17	
3579			:ERROR 231	
3580	007770	000000	0	
3581	007772	000000	0	
3582	007774	000000	0	
3583	007776	000000	0	
3584			:ERROR 232	
3585	010000	000000	0	
3586	010002	000000	0	

```

3587 010004 000000      0
3588 010006 000000      0
3589      :ERROR 233
3590 010010 060352      EM16      :CANNOT READ BSE INFO
3591 010012 065167      DH42      :ON SECT 0,2,4,6,8
3592 010014 065576      DT1
3593 010016 066610      DF17
3594      :ERROR 234
3595 010020 000000      0
3596 010022 000000      0
3597 010024 000000      0
3598 010026 000000      0
3599      :ERROR 235
3600 010030 062164      EM69      :ALIGN CARTRIDGE USED
3601 010032 065317      DH44      :WILL BYPASS FORMAT & ALL R/W TESTS
3602 010034 065576      DT1
3603 010036 066530      DF10
3604
3605      .SBTTL PROGRAM SETUP
3606
3607 010040 012737 000001 001336 PARSRT: MOV      #1,PARAM      :SET FLAG FOR 220 START: INPUT PARAMETERS
3608 010046 000402      BR          PRGSRT      :START PROGRAM
3609
3610 010050 005037 001336      START: CLR      PARAM      :200 START, DEFAULT
3611 010054 000005      PRGSRT: RESET   :CLEAR ALL INT ENABLE & INIT
3612 010056 012706 001100      MOV      #STACK,SP      :SETUP STACK POINTER
3613 010062 012746 000000      MOV      #PRO,-(SP)      :PSW LOADED TO BE
3614 010066 012746 010074      MOV      #1$,-(SP)      :LSI-11 COMPATABLE
3615 010072 000002      RTI          :ENABLE ALL INTERRUPTS
3616
3617 010074 004737 052332      1$:      JSR      PC,$TKINT      :SETUP KB VECTOR ADDR, PRIORITY 4
3618      :& TURN ON KB INTERRUPT
3619
3620
3621      :*** CPU PRIORITY LEVEL NOW AT 0 ***
3622      :*** ANY DEVICE WHICH SETS ITS      ***
3623      :*** INTERRUPT ENABLE BIT WILL      ***
3624      :*** SERVICED.                      ***
3625
3626      :CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
3627      :RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3628      :KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3629
3630      :ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3631
3632      :SYSMAC 'SETUP'
3633      .SBTTL INITIALIZE THE COMMON TAGS
3634      ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3635 010100 012706 001100      MOV      #CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3636 010104 005026      CLR      (R6)+          ;;CLEAR MEMORY LOCATION
3637 010106 022706 001140      CMP      #SWR,R6 ;;DONE?
3638 010112 001374      BNE      -6            ;;LOOP BACK IF NO
3639 010114 012706 001100      MOV      #STACK,SP      ;;SETUP THE STACK POINTER
3640      ;;INITIALIZE A FEW VECTORS
3641 010120 012737 050366 000020      MOV      #SCOPE,#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3642 010126 012737 000340 000022      MOV      #340,#IOTVEC+2 ;;LEVEL 7

```

```

3643 010134 012737 050646 000030      MOV      # $ERROR,@#EMTVEC      ;; EMT VECTOR FOR ERROR ROUTINE
3644 010142 012737 000340 000032      MOV      #340,@#EMTVEC+2      ;; LEVEL 7
3645 010150 012737 054562 000034      MOV      # $TRAP,@#TRAPVEC     ;; TRAP VECTOR FOR TRAP CALLS
3646 010156 012737 000340 000036      MOV      #340,@#TRAPVEC+2     ;; LEVEL 7
3647 010164 012737 050122 000024      MOV      # $PWRDN,@#PWRVEC     ;; POWER FAILURE VECTOR
3648 010172 012737 000340 000026      MOV      #340,@#PWRVEC+2     ;; LEVEL 7
3649 010200 013737 042740 042732      MOV      $SENDCT,$SEOPCT      ;; SETUP END-OF-PROGRAM COUNTER
3650 010206 005037 001174          CLR      $TIMES                ;; INITIALIZE NUMBER OF ITERATIONS
3651 010212 005037 001176          CLR      $ESCAPE              ;; CLEAR THE ESCAPE ON ERROR ADDRESS
3652 010216 112737 000001 001115      MOV      #1,$ERMAX            ;; ALLOW ONE ERROR PER TEST
3653 010224 012737 010224 001106      MOV      #.,$LPADR            ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
3654 010232 012737 010232 001110      MOV      #.,$LPERR           ;; SETUP THE ERROR LOOP ADDRESS
3655                                     ;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3656                                     ;; EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
3657 010240 013746 000004          MOV      @#ERRVEC,-(SP)       ;; SAVE ERROR VECTOR
3658 010244 012737 010300 000004      MOV      #64,$@#ERRVEC       ;; SET UP ERROR VECTOR
3659 010252 012737 177570 001140      MOV      # $DSWR,$SWR         ;; SETUP FOR A HARDWARE SWICH REGISTER
3660 010260 012737 177570 001142      MOV      # $DDISP,$DISPLAY    ;; AND A HARDWARE DISPLAY REGISTER
3661 010266 022777 177777 170644      CMP      #-1,@$SWR           ;; TRY TO REFERENCE HARDWARE SWR
3662 010274 001012          BNE     66$                  ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
3663                                     ;; AND THE HARDWARE SWR IS NOT = -1
3664 010276 000403          BR     65$                  ;; BRANCH IF NO TIMEOUT
3665 010300 012716 010306      64$:   MOV      #65$,(SP)       ;; SET UP FOR TRAP RETURN
3666 010304 000002          RTI
3667 010306 012737 000176 001140      65$:   MOV      # $SWREG,$SWR     ;; POINT TO SOFTWARE SWR
3668 010314 012737 000174 001142      MOV      # $DISPREG,$DISPLAY
3669 010322 012637 000004      66$:   MOV      (SP)+,@#ERRVEC     ;; RESTORE ERROR VECTOR
3670
3671 010326 005037 001216          CLR      $PASS                ;; CLEAR PASS COUNT
3672 010332 132737 000200 001231      BITB    # $APTSIZE,$ENVM     ;; TEST USER SIZE UNDER APT
3673 010340 001403          BEQ     67$                  ;; YES,USE NON-APT SWITCH
3674 010342 012737 001232 001140      MOV      # $SSWREG,$SWR      ;; NO,USE APT SWITCH REGISTER
3675 010350      67$:
3676 010350 012737 010414 000004      MEMPAR: MOV      #1$,ERRVEC     ;; SETUP TIMEOUT VECTOR
3677 010356 012737 000340 000006      MOV      #PR7,ERRVEC+2
3678
3679 010364 012701 172100          MOV      #MEMBAS,R1          ;; ADDR OF MEM CSR
3680 010370 005011      3$:   CLR      (R1)                ;; SEE IF CAN REFERENCE
3681 010372 012711 000001          MOV      #1,(R1)            ;; SET ENABLE BIT IF YES
3682 010376 012737 050024 000114      MOV      #MEMERR,MEMVEC     ;; LOAD VECTOR IF NO TIMEOUT
3683 010404 012737 000340 000116      MOV      #PR7,MEMVEC+2
3684 010412 000401          BR     2$
3685
3686 010414 022626      1$:   CMP      (SP)+,(SP)+       ;; ADJ STACK
3687 010416 062701 000002      2$:   ADD      #2,R1            ;; TRY NEXT CSR
3688 010422 020127 172140          CMP      R1,#MEMBAS+40     ;; SEE IF TRIED ALL
3689 010426 001360          BNE     3$                  ;; BR IF NO
3690 010430 012737 000006 000004      MOV      #ERRVEC+2,ERRVEC   ;; RESTORE TRAP CATCHER
3691 010436 005037 000006          CLR      ERRVEC+2
3692
3693 010442 004737 043026          JSR     PC,CLRFLG           ;; CLEAR DDUMP THRU UNITB
3694 010446 005037 001220          CLR      $DEVCT
3695 010452 005037 001222          CLR      $UNIT
3696
3697
3698                                     ;; FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE

```

```
3699 :
3700 :
3701 010456 005737 000042 START1: TST 42
3702 010462 001014 BNE 1$ ;BR IF AUTO
3703 010464 004737 043046 JSR PC,TITLE ;MANUAL, TYPE PROG ID
3704 010470 123727 000041 000013 CMPB 41,#13 ;13=LOADED BY XXDP
3705 010476 001010 BNE 2$
3706 010500 005237 005424 INC DDUMP ;SET RK06 DUMP MODE FLAG
3707 010504 104401 056011 TYPE ,MSG2 ;REPLACE DRO PACK W/SCRATCH & DO<CR>
3708 010510 000137 010524 JMP ST2
3709 010514 000137 010570 1$: JMP ST3
3710 010520 005237 005432 2$: INC PPTP ;SET ACT/APT/PTP DUMP MODE FLAG
3711 :
3712 :
3713 ;CHECK IF ALL PARAMETERS DEFAULTED. IF NOT, BEGIN INPUT DIALOGUE
3714 ;WITH OPERATOR. THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
3715 ;DRIVE NOS. SEPERATED BY COMMA & TERMINATED BY <CR>
3716 ; EX: DRIVES TO BE TESTED: 0,2,4<CR>
3717 ;FOR DUAL PORT, ONLY EVEN NUMBERED DRIVES MUST BE TESTED
3718 :
3719 010524 005737 001336 ST2: TST PARAM
3720 010530 001002 BNE 1$ ;BR IF 220 START
3721 010532 000137 010622 JMP ST4 ;200 START, DEFAULT & SIZE THE BUSS
3722 010536 104401 056062 1$: TYPE ,MSG3 ;DRIVES TO BE TESTED
3723 010542 004737 043126 JSR PC,GDRVS ;GET DR NOS.
3724 010546 104401 056135 TYPE ,MSG4 ;BUSS ADDR
3725 010552 004737 043274 JSR PC,GBA ;GET BA
3726 010556 104401 056167 TYPE ,MSG5 ;CONT INT VECTOR
3727 010562 004737 043322 JSR PC,GINT ;GET INT VECTOR
3728 010566 000427 BR ST5
3729 :
3730 :
3731 ;AUTO MODE
3732 ;CHECK IF LOADED BY XXDP OR OTHER. SET FLAGS & NO INPUT DIALOGUE.
3733 ;DEFAULT ALL PARAMETERS. TEST ONLY THOSE DRIVES THAT ARE READY
3734 ;ON THE BUSS
3735 :
3736 :
3737 010570 123727 000041 000013 ST3: CMPB 41,#13 ;13=LOADED BY XXDP
3738 010576 001007 BNE 1$
3739 010600 005237 005426 INC DDPCH ;SET RK06 CHAIN MODE FLAG
3740 010604 004737 043046 JSR PC,TITLE
3741 010610 104401 056271 TYPE ,MSG7 ;DRO NOT TSTD
3742 010614 000402 BR ST4
3743 010616 005237 005430 1$: INC ACT11 ;SET ACT AUTO FLAG.
3744 :
3745 010622 012737 177440 001264 ST4: MOV #177440,$BASE ;DEFAULT VALUE
3746 010630 012737 000210 001314 MOV #210,RKVEC ;DEFAULT VALUE
3747 010636 004737 043354 JSR PC,SETINT
3748 010642 005237 005462 INC SIZEFLG ;DO "SIZE THE BUSS" TEST
3749 :
3750 010646 005037 005276 ST5: CLR UNLD ;INITIALIZE FLAGS
3751 010652 005037 005300 CLR BADHDR ;USED IN 'STOP' ROUTINE
3752 010656 005037 005302 CLR HPEND ;FOR VALID PROGRAM HALTS
3753 010662 005037 001176 CLR $ESCAPE
3754 010666 005037 001172 CLR $TMP5 ;CLEAR RK07 FLAG
```

```

3755 010672 012737 005436 001342      MOV      #DRIVO,DRVPTR      :SETUP
3756 010700 005037 001220              CLR      $DEVCT            :NO. OF DRVS DONE
3757 010704 005037 001222              CLR      $UNIT             :CURRENT DRV UNDER TEST
3758 010710 005037 005464              CLR      UNITB             :PORT A TESTING
3759 010714 112737 000101 056644      MOV      #'A,MSG19A        :PORT A TESTING
3760 010722 012737 010770 000004      MOV      #1$,ERRVEC        :SETUP TIMEOUT ERROR VECTOR
3761 010730 005777 170372              TST      @LKS              :SEE IF L-CLOCK THERE
3762 010734 005237 005456              INC      LCLKF             :PRESENT, SET FLAG.
3763 010740 013700 001330              MOV      LCVEC,RO          :VECTOR ADDR
3764 010744 012737 000026 000004      MOV      #2$,ERRVEC
3765 010752 005777 170342              TST      @PKS              :SEE IF P-CLOCK THERE
3766 010756 005237 005460              INC      PCLKF             :PRESENT, SET FLAG
3767 010762 013700 001332              MOV      PCVEC,RO          :VECTOR ADDR
3768 010766 000412              BR       3$
3769
3770 010770 022626              1$:    CMP      (SP)+,(SP)+      :L-CLOCK NOT THERE, CLEAR STACK
3771 010772 012737 011032 000004      MOV      #4$,ERRVEC
3772 011000 005777 170314              TST      @PKS              :SEE IF P-CLOCK THERE
3773 011004 005237 005460              INC      PCLKF             :PRESENT, SET FLAG
3774 011010 013700 001332              MOV      PCVEC,RO          :VECTOR ADDR
3775 011014 012720 047154              3$:    MOV      #CLOCK,(RO)+   :SERVICE ROUTINE FOR CLOCKS
3776 011020 012710 000300              MOV      #PR6,(RO)
3777 011024 000407              BR       TST1              :;GO TO NEXT TEST
3778
3779 011026 022626              2$:    CMP      (SP)+,(SP)+   :P-CLOCK NOT THERE, CLEAR STACK
3780 011030 000771              BR       3$
3781
3782 011032 022626              4$:    CMP      (SP)+,(SP)+   :NEITHER CLOCK THERE, CLEAR STACK
3783 011034 104401 056444              TYPE    ,MSG13             :NO CLOCKS PRESENT TESTS BYPASSED
3784 011040 000137 042644              JMP     $EOP
3785
3786

```

3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842

.SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

*TEST 1 REFERENCE ALL CONTROLLER REGISTERS

* THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
* CAN BE ACCESSED. THE INABILITY TO BE ACCESSED WILL
* RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE. ANY
* ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
* TESTS AND JUMPING TO 'END OF PASS'

```
TST1: SCOPE
MOV #1,$TIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR

MOV #PRO,-(SP) ;RESET PSW TO PRIORITY 0
MOV #5$,-(SP) ;& MAKE IT LSI COMPATABLE
RTI

5$:

MOV #1$,ERRVEC ;SETUP TIMOUT ERROR VECTOR
MOV $BASE,R5 ;SETUP INDEX REG.
TST RKCS1(R5) ;REFERENCE ALL THE
TST RKCS2(R5) ;CONTROLLER REGISTERS
TST RKWC(R5)
TST RKBA(R5)
TST RKDA(R5)
TST RKDS(R5) ;TIMEOUTS IN THIS SECTION
TST RKER(R5) ;INDICATE THAT THE CONTROLLER
TST RKASOF(R5) ;REGISTERS CANNOT BE READ.
TST RKDC(R5) ;TESTING SHOULD NOT PROCEED
TST RKDB(R5) ;UNTIL THIS IS REMEDIED.
TST RKMR1(R5)
TST RKMR2(R5)
TST RKMR3(R5)
TST RKECPS(R5)
TST RKECPT(R5)

MOV #BADTMO,ERRVEC ;SETUP TIMEOUT HANDLER
MOV #PR7,ERRVEC+2
BR TST2 ;GO TO NEXT TEST

1$: CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
ERROR 7 ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
JMP $EOP1
```

*TEST 2 SIZE THE BUSS

* THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
* EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
* MANUAL MODE.
* EVERY EVEN NUMBERED DRIVE (0,2,4,6) IS ADDRESSED.
* CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE

```
3843 :* DRIVE WILL BE TESTED AS AN RK06. IF SET, THE PROGRAM WILL BYPASS
3844 :* TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
3845 :* MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
3846 :* DICATING THE OTHER PORT IS ACCESSED.
3847 :* IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS AN RK07.
3848 :*
3849 :*
*****
3850 011226 000004          TST2: SCOPE
3851 011230 012737 000001 001174  MOV #1,$TIMES      ;;DO 1 ITERATION
3852 011236 012706 001100          MOV #STACK,SP     ;RESTORE STK PTR
3853
3854 011242 005237 001456          INC BYPCERR       ;DO NOT DO 'CKCERR' ROUTINE
3855
3856
3857 011246 132737 000200 001231  BITB #BIT7,$ENVM  ;SEE IF USE APT SELECTED DRIVES
3858 011254 001002          BNE 14$          ;BR IF YES
3859 011256 000137 011420          JMP 12$          ;ELSE DO NORM SIZING OR VERIFY
3860
3861 011262 104401 056357          14$: TYPE ,MSG10   ;WILL TEST DRIVES
3862 011266 005037 005434          CLR DRVS        ;# OF DRIVES PRESENT
3863 011272 005000          CLR R0          ;DRV ADDR
3864 011274 012701 005436          MOV #DRIVO,R1   ;DRV FLAG
3865 011300 013702 001266          MOV $DEVN,R2    ;APT DEVICE MAP
3866
3867 011304 032702 000001          15$: BIT #BIT0,R2 ;SEE IF DRV IN DEVICE MAP
3868 011310 001420          BEQ 16$          ;BR IF NO
3869 011312 022700 000001          CMP #BIT0,R0    ;SEE IF ODD # DRIVE
3870 011316 001405          BEQ 11$          ;BR IF NO
3871
3872 011320 104401 061253          TYPE ,EM28      ;ONLY EVEN # ALLOWED IN $DEVN
3873 011324 000000          HALT            ;RELOAD $DEVN & PRESS 'CONTINUE'
3874 011326 000137 010054          JMP PRGSR1      ;RESTART
3875 011332 005237 005434          11$: INC DRVS     ;ELSE INCR DRIVE COUNT
3876 011336 005211          INC (R1)        ;& SET DRIVE PRESENT FLAG
3877 011340 104401 001205          TYPE ,$CRLF
3878 011344 010046          MOV R0,-(SP)   ;;SAVE R0 FOR TYPEOUT
3879
3880 011346 104403          TYPOS          ;;TYPE DRIVE #
3881 011350 001          .BYTE 1        ;;GO TYPE--OCTAL ASCII
3882 011351 000          .BYTE 0        ;;TYPE 1 DIGIT(S)
3883
3884 011352 022121          16$: CMP (R1)+,(R1)+ ;ADV PTR TO NEXT EVEN #
3885 011354 062700 000002          ADD #2,R0       ;ADD DRV ADDR TO NEXT EVEN #
3886 011360 022700 000010          CMP #8,R0       ;ALL 4 TESTED?
3887 011364 001402          BEQ 17$         ;BR IF YES
3888
3889 011366 006002          ROR R2          ;ELSE GET NEXT BIT OFF DEVICE MAP
3890 011370 000745          BR 15$         ;& TRY AGAIN
3891
3892 011372 005737 005434          17$: TST DRVS     ;SEE IF MORE DRIVES PRESENT
3893 011376 001402          BEQ 18$        ;BR IF NO
3894 011400 000137 012076          JMP VERIFY     ;ELSE EXIT TEST & SETUP FOR RK07'S
3895
3896 011404 104075          18$: ERROR 75   ;NO DRIVES FOUND IN $DEVN
3897 011406 000000          HALT          ;SETUP CORRECTLY & PRESS 'CONTINUE'
3898 011410 000137 010646          JMP ST5       ;TO TRY AGAIN
```

```

3899 011414 000137 012076      20$:  JMP      VERIFY      ;DO NOT SIZE, GO TO NEXT TEST
3900
3901 011420 012765 000040 000010 12$:  MOV      #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3902 011426 013737 001400 005352  MOV      T10,TEMP1      ;SET TIMEOUT
3903 011434 004737 043466      JSR      PC,FRDY        ;FIND RDY
3904 011440 104120      ERROR    120           ;RDY NOT SET BY END OF SCLR
3905 011442 005737 005462      TST      SIZFLG        ;SIZE BUS?
3906 011446 001762      BEQ      20$           ;BR IF NO
3907 011450 104401 056357      TYPE    ,MSG10        ;WILL TEST DRIVES
3908 011454 005037 005434      CLR     DRIVS         ;# OF DRIVES PRESENT
3909 011460 005000      CLR     R0            ;DRV ADDR
3910 011462 012701 005436      MOV     #DRIVO,R1     ;DRV FLAG
3911 011466
3912 011466 104415      1$:   SCOP1
3913 011470 012706 001100      MOV     #STACK,SP     ;RESTORE STK PTR
3914
3915 011474 012765 000040 000010  MOV     #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3916 011502 013737 001400 005352  MOV     T10,TEMP1     ;SET TIMEOUT
3917 011510 004737 043466      JSR     PC,FRDY       ;FIND RDY
3918 011514 104120      ERROR    120         ;RDY NOT SET BY END OF SCLR
3919 011516 010065 000010      MOV     R0,RKCS2(R5)  ;SELECT THE DRIVE ADDR
3920 011522 012737 000001 005314  MOV     #SELDRV,HCS1
3921 011530 053737 001170 005314  BIS     $TMP4,HCS1    ;ADD CDT IF RK07
3922 011536 013765 005314 000000  MOV     HCS1,RKCS1(R5) ;GET STATUS
3923 011544 013737 001412 005352  MOV     T50000,TEMP1
3924 011552 004737 044242      JSR     PC,DLY        ;DO DELAY TO CATCH MDS
3925 011556 013737 001400 005352  MOV     T10,TEMP1
3926 011564 004737 043466      JSR     PC,FRDY       ;FIND RDY
3927 011570 104117      ERROR    117         ;NO RDY AFTER SELECT DRIVE CMD.
3928 011572 032737 100000 005314  BIT     #CERR,HCS1
3929 011600 001053      BNE     2$
3930 011602 013737 005342 005352  MOV     HMR2,TEMP1
3931 011610 042737 177770 005352  BIC     #^C<DRVMSK>,TEMP1
3932 011616 020037 005352      CMP     R0,TEMP1     ;S/B SAME
3933 011622 001020      BNE     3$
3934 011624 005700      TST     R0
3935 011626 001003      BNE     4$
3936 011630 005737 005426      TST     DDPCH        ;SEE IF XXDP CHAIN MODE
3937 011634 001016      BNE     5$
3938 011636 005237 005434      4$:  INC     DRIVS        ;INC DRIVE COUNT.
3939 011642 005211      INC     (R1)         ;SET DRIVE PRESENT FLAG
3940 011644 053711 001170      BIS     $TMP4,(R1)   ;ADD CDT IF RK07.
3941 011650 104401 001205      TYPE    ,SCLRF
3942 011654 010046      MOV     R0,-(SP)     ;;SAVE R0 FOR TYPEOUT
3943
3944 011656 104403      TYPOS
3945 011660 001      .BYTE  1            ;;TYPE DR #
3946 011661 000      .BYTE  0            ;;GO TYPE--OCTAL ASCII
3947 011662 000403      BR     5$           ;;TYPE 1 DIGIT(S)
3948
3949 011664 004737 044260      3$:  JSR     PC,BYP       ;TYPE BYPASS DR #
3950 011670 104001      ERROR    1           ;WRITTEN DR # DOES NOT MATCH RKMP.2 DR #
3951
3952 011672 022121      5$:  CMP     (R1)+,(R1)+ ;ADV PTR TO NEXT EVEN #
3953 011674 062700      ADD     #2,R0        ;ADV DRV ADDR TO NEXT EVEN #
3954 011700 005037 001170      CLR     $TMP4        ;CLEAR FOR NEXT TRY
  
```



```

3955 011704 022700 000010 CMP #8.,R0 ;TESTED ALL 4 DRIVES?
3956 011710 001266 BNE 1$ ;BR IF NO
3957 011712 005737 005434 TST DRIVS
3958 011716 001065 BNE 10$
3959 011720 104076 ERROR 76 ;NO DRIVES FOUND ON BUSS
3960 011722 000000 HALT ;SETUP CORRECTLY
3961 011724 000137 010646 JMP ST5 ;AND PRESS 'CONTINUE'
3962
3963 011730 032737 000040 005330 2$: BIT #DTYE,HER
3964 011736 001405 BEQ 13$
3965 011740 012737 002000 001170 MOV #CDT,$TMP4 ;ADD CDT
3966 011746 000137 011466 JMP 1$ ;TRY AGAIN
3967 011752 032737 001000 005316 13$: BIT #MDS,HCS2
3968 011760 001015 BNE 6$
3969 011762 032737 000400 005316 BIT #UFE,HCS2
3970 011770 001015 BNE 7$
3971 011772 032737 000001 005326 BIT #DRA,HDS
3972 012000 001015 BNE 8$
3973 012002 032737 010000 005316 BIT #NED,HCS2
3974 012010 001424 BEQ 9$
3975 012012 000727 BR 5$
3976
3977 012014 004737 044260 6$: JSR PC,BYP ;TYPE BYP DR #
3978 012020 104002 ERROR 2 ;MDS DETECTED
3979 012022 000723 BR 5$
3980
3981 012024 004737 044260 7$: JSR PC,BYP
3982 012030 104003 ERROR 3 ;UFE DETECTED
3983 012032 000717 BR 5$
3984
3985 012034 032737 010000 005316 8$: BIT #NED,HCS2
3986 012042 001675 BEQ 4$
3987 012044 104401 056546 TYPE ,MSG15 ;DRV#
3988 012050 010046 MOV R0,-(SP) ;:SAVE R0 FOR TYPEOUT
3989 ;:TYPE DR#
3990 012052 104403 TYPOS ;:GO TYPE--OCTAL ASCII
3991 012054 001 .BYTE 1 ;:TYPE 1 DIGIT(S)
3992 012055 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
3993 012056 104010 ERROR 10 ;DRA & NED BOTH SET
3994 012060 000704 BR 5$
3995
3996 012062 004737 044260 9$: JSR PC,BYP
3997 012066 104004 ERROR 4 ;NO DRA & NO NED = OTHER PORT SELECTED
3998 012070 000700 BR 5$
3999 012072 000137 012472 10$: JMP NUDRV
4000
4001 012076 VERIFY:
4002
4003
4004
4005
4006
4007
4008
4009
4010

```

```

:*****
:*TEST 3 VERIFY OPERATOR DRIVE SELECTIONS
:*
:* THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
:* DEFAULTED. EVERY EVEN NUMBERED DRIVE (0,2,4,6) IS ADDRESSED &
:* CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
:* PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06.

```

```

4011      : *      IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
4012      : *      ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
4013      : *      NED & DRA RESET (WRONG PORT).  IF CERR IS A RESULT OF
4014      : *      NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFORMATION TO
4015      : *      VERIFY IT WAS NOT SPECIFIED.
4016      : *      IF CERR DUE TO DTYE, DRIVE WILL BE TESTED AS AN RK07.
4017      : *
4018      : *****
4019 012076 000004      TST3: SCOPE
4020 012100 012737 000001 001174  MOV #1,$TIMES      ;;DO 1 ITERATION
4021 012106 012706 001100      MOV #STACK,SP      ;RESTORE STK PTR
4022 012112 005000      CLR R0              ;DRIVE ADDR
4023 012114 012701 005436      MOV #DRIVO,R1      ;DRIVE FLAG
4024 012120
4025 012120 104415      1$: SCOP1
4026 012122 012706 001100      MOV #STACK,SP      ;RESTORE STK PTR
4027
4028 012126 012765 000040 000010  MOV #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
4029 012134 013737 001400 005352  MOV T10,TEMP1      ;SET TIME OUT
4030 012142 004737 043466      JSR PC,FRDY        ;FIND RDY
4031 012146 104120      ERROR 120         ;NO RDY AFTER SCLR
4032 012150 010065 000010      MOV R0,RKCS2(R5)  ;DRV ADDR
4033 012154 012737 000001 005314  MOV #SELDRV,HCS1
4034 012162 053737 001170 005314  BIS $TMP4,HCS1     ;ADD CDT IF RK07
4035 012170 013765 005314 000000  MOV HCS1,RKCS1(R5) ;GET STATUS
4036 012176 013737 001412 005352  MOV T5000,TEMP1
4037 012204 004737 044242      JSR PC,DLY         ;DO DELAY TO CATCH MDS
4038 012210 013737 001400 005352  MOV T10,TEMP1
4039 012216 004737 043466      JSR PC,FRDY        ;FIND RDY
4040 012222 104117      ERROR 117         ;NO RDY AFTER SELECT DRIVE CMD.
4041 012224 032737 100000 005314  BIT #CERR,HCS1
4042 012232 001037      BNE 2$
4043 012234 013737 005342 005352  MOV HMR2,TEMP1
4044 012242 042737 177770 005352  BIC #^C<DRVMSK>,TEMP1
4045 012250 020037 005352      CMP R0,TEMP1      ;S/B SAME
4046 012254 001015      BNE 3$
4047 012256 005711      11$: TST (R1)
4048 012260 001402      BEQ 4$
4049 012262 053711 001170      BIS $TMP4,(R1)     ;SET RK07 FLAG
4050 012266 022121      4$: CMP (R1)+,(R1)+  ;ADV PTR TO NEXT EVEN #
4051 012270 062700 000002      ADD #2,R0         ;ADV DRV ADDR TO NEXT EVEN #
4052 012274 005037 001170      CLR $TMP4         ;CLEAR FOR NEXT DRIVE
4053 012300 022700 000010      CMP #8.,R0
4054 012304 001305      BNE 1$            ;MORE LEFT
4055 012306 000475      BR TST4          ;;GO TO NEXT TEST
4056
4057 012310 004737 044260      3$: JSR PC,BYP        ;TRY BYPASS DRIVE#
4058 012314 104001      ERROR 1           ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
4059 012316 005711      TST (R1)
4060 012320 001762      BEQ 4$
4061 012322 005337 005434      12$: DEC DRIVS      ;BRANCH IF NOT SPEC BY INPUT
4062 012326 005011      CLR (R1)         ;DECREMENT TOTAL DRIVS
4063 012330 000756      BR 4$            ;CLEAR DRIVE FLAG
4064
4065 012332 032737 000040 005330  2$: BIT #DTYE,HER
4066 012340 001405      BEQ 13$

```

```

4067 012342 012737 002000 001170      MOV    #CDT,$TMP4      ;ADD CDT
4068 012350 000137 012120      JMP    1$              ;TRY AGAIN
4069
4070 012354 032737 001000 005316 13$:   BIT    #MDS,HCS2
4071 012362 001027                BNE    6$
4072 012364 032737 000400 005316   BIT    #UFE,HCS2
4073 012372 001027                BNE    7$
4074 012374 032737 000001 005326   BIT    #DRA,HDS
4075 012402 001005                BNE    8$
4076 012404 032737 010000 005316   BIT    #NED,HCS2
4077 012412 001423                BEQ    9$
4078 012414 000404                BR     10$
4079 012416 032737 010000 005316 8$:   BIT    #NED,HCS2
4080 012424 001714                BEQ    11$
4081 012426 005711                TST   (R1)
4082 012430 001716                BEQ    4$
4083
4084 012432 004737 044260      JSR    PC,BYP          ;TYPE BYPASS DRIVE#
4085 012436 104006                ERROR  6
4086 012440 000730                BR     12$
4087
4088 012442 004737 044260      6$:   JSR    PC,BYP          ;TYPE BYPASS DRIVE#
4089 012446 104002                ERROR  2                ;MDS DETECTED
4090 012450 000724                BR     12$
4091
4092 012452 004737 044260      7$:   JSR    PC,BYP
4093 012456 104003                ERROR  3                ;UFE DETECTED
4094 012460 000720                BR     12$
4095
4096 012462 004737 044260      9$:   JSR    PC,BYP
4097 012466 104004                ERROR  4                ;DRA & NED RESET - OTHER PORT SELECTED
4098 012470 000714                BR     12$
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109 012472 005037 001456      NUDRV: CLR    BYPCERR    ;ENTER HERE FROM LAST TEST
4110
4111 012476 005037 001170      CLR    $TMP4          ;ALLOW CHECKING CERR IN 'FRDY'
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122 012502 000004      TST4: SCOPE
  
```

```

:
: THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
: DRIVE PRESENT
:
: '$UNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
: UNDER TEST
  
```

```

:*****
: *TEST 4      FIND NEXT DRIVE TO BE TESTED
: *
: * THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
: * ADDRESS IN '$UNIT' & $TMP4 IS SET TO CDT IF RK07.
: * THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
: * THE DRIVE WHOSE ADDRESS IS IN '$UNIT'.
:*****
  
```

```

4123 012504 012737 000001 001174      MOV      #1,$TIMES      ;;DO 1 ITERATION
4124 012512 012706 001100          MOV      #STACK,SP     ;RESTORE STK PTR
4125 012516 012737 000004 001214      MOV      #STN-1,$TESTN
4126 012524 012737 000004 001102      MOV      #STN-1,$STNM
4127
4128 012532 005737 005434          TST      DRVS          ;ANY DRIVES PRESENT?
4129 012536 001004          BNE      4$           ;YES BRANCH
4130 012540 104401 056326          TYPE    ,MSG8         ;ALL DRIVES TESTED
4131 012544 000137 042704          JMP      $EOP1        ;NO, GO TO END
4132
4133 012550 005037 005464          CLR      UNITB        ;SETUP PORT A
4134 012554 112737 000101 056644      MOV      #'A,MSG19A
4135 012562 013701 001342          MOV      DRVPTR,R1    ;ADDR OF NEXT DRIVE FLAG
4136 012566 005737 001220          TST      $DEVCT       ;IS FIRST DRIVE BEING CHECKED
4137 012572 001403          BEQ      2$           ;YES, BRANCH
4138 012574 062737 000002 001222      1$:     ADD      #2,$UNIT   ;INCR TO NEXT EVEN DRIVE
4139 012602 005711          2$:     TST      (R1)      ;IS DRIVE PRESENT?
4140 012604 001003          BNE      5$           ;BR IF YES
4141 012606 062701 000004          ADD      #4,R1        ;ELSE INCR PTR TO NEXT EVEN DRIVE
4142 012612 000770          BR       1$           ;& TRY AGAIN
4143
4144 012614 005737 005426          5$:     TST      DDPCH    ;DDF CHAIN MODE?
4145 012620 001406          BEQ      3$           ;BR IF NO
4146 012622 005737 001222          TST      $UNIT        ;ELSE SEE IF DRV 0
4147 012626 001003          BNE      3$           ;BR IF NO
4148 012630 062701 000004          ADD      #4,R1        ;ELSE FIND NEXT EVEN DRIVE PRESENT
4149 012634 000757          BR       1$
4150
4151 012636 032711 002000          3$:     BIT      #CDT,(R1) ;SEE IF DRIVE UNDER TEST IS RK07
4152 012642 001403          BEQ      6$           ;BR IF NO
4153 012644 012737 002000 001170      MOV      #CDT,$TMP4   ;ELSE SET RK07 FLAG
4154 012652 062701 000004          6$:     ADD      #4,R1
4155 012656 010137 001342          MOV      R1,DRVPTR    ;STORE PTR OF NEXT EVEN DRIVE FLAG
4156 012662 104401 056546          TYPE    ,MSG15        ;'DRIVE'
4157 012666 013700 001222          MOV      $UNIT,R0
4158 012672 010046          MOV      R0,-(SP)     ;;SAVE R0 FOR TYPEOUT
4159
4160 012674 104403          TYPOS    ;DRIVE #
4161 012676 001          .BYTE   1            ;;GO TYPE--OCTAL ASCII
4162 012677 000          .BYTE   0            ;;TYPE 1 DIGIT(S)
4163
4164          ; TYPE    , $CRLF ;29-SEP-77
4165
4166
4167 012700 005737 001170          TST      $TMP4        ;SEE IF RK07 UNDER TEST
4168 012704 001014          BNE      7$           ;BR IF YES
4169 012706 012737 000632 012770      MOV      #632,LC      ;ELSE LOAD RK06 PARAMERERS
4170 012714 005037 012776          CLR      E.DDT
4171 012720 012737 000777 012772      MOV      #777,MASK
4172 012726 012737 160017 012774      MOV      #160017,MASK1
4173 012734 000423          BR       TST5        ;;GOTO NEXT TEST
4174
4175 012736 012737 001456 012770      7$:     MOV      #1456,LC   ;LOAD RK07 PARAMETERS
4176 012744 012737 000400 012776      MOV      #D.DDT,E.DDT
4177 012752 012737 001777 012772      MOV      #1777,MASK
4178 012760 012737 140017 012774      MOV      #140017,MASK1
    
```

4179 012766 000406 BR TST5 ;:GOTO NEXT TEST
4180 ;:LAST CYL
4181 012770 000000 LC: 0
4182 012772 000000 MASK: 0
4183 012774 000000 MASK1: 0
4184 012776 000000 E.DDT: 0 ;:EXPECTED DRIVE TYPE TO E.A0
4185 013000 000000 FC: 0 ;:FIRST CYL
4186 013002 000001 FCP1: 1 ;:FC+1
4187 013004 PFSRT: ;:ENTER HERE FOR POWER FAIL RESTART

.SBTTL DUAL PORT TESTS

:TEST 5 TEST PORT 'A' SEIZE & TIMEOUT
:*****
: * VERIFY THAT THE DRIVE CAN BE SEIZED & THAT THE PORT
: * TIMEOUT RELEASES THE DRIVE.
: *
: * A. SET VOLUME VALID FOR BOTH PORTS & DO RECAL COMMAND
: *
: * B. A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'A'.
: * THE PROGRAM VERIFIES THE DRIVE HAS BEEN SEIZED BY 'DRIVE
: * AVAILABLE' SET.
: *
: * C. A SELECT DRIVE COMMAND IS ISSUED THRU PORT 'B' THE
: * PROGRAM VERIFIES THAT 'DRIVE AVAILABLE' IS NOT SET
: * FOR PORT 'B' & THAT CERR IS SET
: *
: * D. VERIFY THAT FOR ALL MESSAGES REQUESTED THRU PORT 'B', (MSG
: * A0-A3, B0-B3) THAT MESSAGE 0 ALWAYS RETURNS FROM PORT B
: * WHILE PORT 'A' IS SEIZED.
: *
: * E. WAIT FOR THE PORT TIMEOUT TO OCCUR ON PORT 'A' BY
: * CONTINUOUSLY CHECKING FOR ATTN ON PORT 'B'.
: * AFTER ATTN-B IS REC'D, A DRIVE SELECT COMMAND IS
: * ISSUED THRU PORT 'B' TO VERIFY THAT DRIVE AVAILABLE IS SET
: * IN MSG A0.
: *
: * MEASURE THE DURATION OF THE TIMEOUT & TYPE THE VALUE
: * FOR THE FIRST PASS ONLY.
: *
: * F. VERIFY THAT ONLY PORT 'B' GETS 'DSC' & 'ATTN'.
: *
: * G. VERIFY THE DRIVE CLEAR COMMAND CLEARS 'DSC' & 'ATTN'
: * ON PORT 'B' BUT DOES NOT RELEASE THE DRIVE FROM PORT 'B'.
:*****

4225
4226 013004 000004 TST5: SCOPE
4227 013006 012737 000001 001174 MOV #1,\$TIMES ;:DO 1 ITERATION
4228 013014 012706 001100 MOV #STACK,SP
4229 013020 012737 000000 005464 MOV #0,UNITB ;:SETUP PORT A
4230 013026 112737 000101 056644 MOVB #'A,MSG19A
4231 013034 013737 005466 001366 MOV TIMER,COUNT
4232 013042 004737 047204 JSR PC,TMO ;:DO TIMEOUT
4233
4234 013046 004737 045534 JSR PC,SUBCLR

G 7

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59 PAGE 85
CZR6GC.P11 04-JAN-82 12:39 T5 TEST PORT 'A' SEIZE & TIMEOUT SEQ 0084

```

4235 013052 104024                    ERROR    24                    ;CERR AFTER SCLR
4236
4237
4238 013054 004737 044172            JSR      PC,DRAV            ;SEE IF DRIVE AVAIL
4239 013060 104045                    ERROR    45                    ;PORT A NOT AVAIL AFTER TMO
4240 013062                    7$:
4241 013062 012765 100000 000000    MOV      #CCLR,RKCS1(R5)
4242 013070 013765 001222 000010    MOV      $UNIT,RKCS2(R5)
4243 013076 063765 005464 000010    ADD      UNITB,RKCS2(R5)
4244 013104 012737 000003 005314    MOV      #PACK,HCS1
4245 013112 004737 043372            JSR      PC,DOCMD           ;DO PACK CMD & GET CONTR RDY
4246 013116 104116                    ERROR    116                  ;CONTR NOT RDY AFT PACK CMD
4247
4248 013120 032737 000100 005342    BIT      #D.VV,HMR2
4249 013126 001001                    BNE      64$
4250 013130 104027                    ERROR    27                    ;VV NOT SET AFTER PACK CMD
4251 013132                    64$:
4252 013132 012737 042644 001176    MOV      #SEOP,$ESCAPE
4253 013140 005737 005464            TST      UNITB
4254 013144 001022                    BNE      2$
4255 013146 012737 000001 005464    MOV      #1,UNITB           ;SETUP PORT B
4256 013154 112737 000102 056644    MOVB    #'B,MSG19A
4257 013162 013737 005466 001366    MOV      TIMER,COUNT
4258 013170 004737 047204            JSR      PC,TMO            ;DO TIMEOUT
4259
4260 013174 004737 045534            JSR      PC,SUBCLR
4261 013200 104223                    ERROR    223                  ;CERR AFTER SCLR
4262
4263 013202 004737 044172            JSR      PC,DRAV            ;SEE IF DROVE AVAIL
4264 013206 104045                    ERROR    45                    ;PORT B NOT AVAIL AFTER TMO
4265 013210 000724                    BR       7$
4266 013212                    2$:
4267
4268 013212 012765 100000 000000    MOV      #CCLR,RKCS1(R5)
4269 013220 013765 001222 000010    MOV      $UNIT,RKCS2(R5)
4270 013226 063765 005464 000010    ADD      UNITB,RKCS2(R5)    ;ADD 1 IF ON PORT B
4271 013234 012737 000013 005314    MOV      #RECAL,HCS1
4272 013242 004737 043372            JSR      PC,DOCMD           ;DO RECAL CMD & GET CONTR RDY
4273 013246 104124                    ERROR    124                  ;RDY NOT SET AFTER RECAL CMD
4274
4275 013250 012765 000001 000026    MOV      #1,RKMR1(R5)       ;SELECT WORD 1
4276 013256 004737 045146            JSR      PC,GSTAT
4277 013262 032737 020000 005342    BIT      #D.RTZ,HMR2
4278 013270 001001                    BNE      65$
4279 013272 104070                    ERROR    70                    ;RTZ NOT SET DURING RECAL CMD
4280 013274 013737 001400 005354    MOV      T10,TEMP2         ;SETUP TIMEOUT
4281 013302 004737 044006            JSR      PC,FATT1          ;FIND ATTN
4282 013306 104055                    ERROR    55                    ;NO ATTN AFTER RECAL CMD
4283
4284 013310 012737 050340 005404    MOV      #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4285 013316 005037 005406            CLR      E.B0              ;EXPECTED MSG B0
4286 013322 012737 001720 005410    MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4287 013330 012737 000001 005412    MOV      #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
4288 013336 005037 005414            CLR      E.A2              ;EXPECTED MSG A2
4289 013342 012737 000002 005416    MOV      #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
4290 013350 012737 000003 005422    MOV      #3,E.B3           ;MSG ID FOR EXPECTED MSG B3

```

4291										
4292	013356	004737	044274			JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1	
4293	013362	000007				.WORD	T.A2!T.B2!T.B3		; & MSGS SPECIFIED HERE	
4294	013364	104221				ERROR	221		;MSG A0 ERROR AFTER RECAL CMD	
4295	013366	104066				ERROR	66		;MSH B0 ERROR	
4296	013370	104222				ERROR	222		;MSG A1 ERROR	
4297	013372	104067				ERROR	67		;MSG B1 ERROR	
4298	013374	012765	000002	000026		MOV	#2,RKMR1(R5)			
4299	013402	004737	045146			JSR	PC,GSTAT			
4300	013406	005737	001356			TST	CYLDIF		;SEE IF MSG A2=0	
4301	013412	001401				BEQ	66\$;BR IF YES	
4302	013414	104047				ERROR	47		;MSG A2 NOT CLEARED AFTER RECAL CMD	
4303	013416	005737	001360		66\$:	TST	CYLADD		;SEE IF MSG B2=0	
4304	013422	001401				BEQ	67\$;BR IF YES	
4305	013424	104050				ERROR	50		;MSG B2 NOT CLEARED AFTER RECAL CMD	
4306	013426				67\$:					
4307										
4308	013426	012765	100000	000000		MOV	#CCLR,RKCS1(R5)			
4309	013434	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)		;DRIVE#	
4310	013442	063765	005464	000010		ADD	UNITB,RKCS2(R5)		;ADD 1 IF ON PORT B	
4311	013450	012737	000005	005314		MOV	#CLEAR,HCS1			
4312	013456	004737	043372			JSR	PC,DOCMD		;DO DRIVE CLEAR CMD & GET CONTR RDY	
4313	013462	104151				ERROR	151		;NO RDY AFTER DRIVE CLEAR CMD	
4314	013464	004737	043750			JSR	PC,TSTATN		;TEST FOR ATTN	
4315	013470	000401				BR	68\$			
4316	013472	104154				ERROR	154		;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD	
4317	013474				68\$:					
4318										
4319	013474	012737	010340	005404		MOV	#<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0		;EXPECTED MSG A0	
4320	013502	005037	005406			CLR	E.B0		;EXPECTED MSG B0	
4321	013506	012737	001720	005410		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1	
4322	013514	012737	000001	005412		MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1	
4323	013522	005037	005414			CLR	E.A2		;EXPECTED MSG A2	
4324	013526	012737	000002	005416		MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2	
4325	013534	012737	000003	005422		MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3	
4326										
4327	013542	004737	044274			JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1	
4328	013546	000003				.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE	
4329	013550	104033				ERROR	33		;MSG A0 ERROR AFTER DRV CLEAR CMD	
4330	013552	104034				ERROR	34		;MSH B0 ERROR	
4331	013554	104035				ERROR	35		;MSG A1 ERROR	
4332	013556	104036				ERROR	36		;MSG B1 ERROR	
4333										
4334										
4335	013560	012737	000000	005464		MOV	#0,UNITB		;SETUP PORT A	
4336	013566	112737	000101	056644		MOVB	#'A,MSG19A			
4337	013574	013737	005466	001366		MOV	TIMER,COUNT			
4338	013602	004737	047204			JSR	PC,TMO		;DO TIMEOUT	
4339										
4340	013606	004737	045534			JSR	PC,SUBCLR			
4341	013612	104024				ERROR	24		;CERR AFTER SCLR	
4342										
4343										
4344	013614	004737	044172			JSR	PC,DRAV		;SEE IF DRIVE AVAIL	
4345	013620	104045				ERROR	45		;PORT A NOT AVAIL AFTER TMO	
4346	013622	012737	014544	001176		MOV	#5\$, \$ESCAPE			

```
4347
4348 013630 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4349 013636 005037 005406 CLR E.B0 ;EXPECTED MSG B0
4350 013642 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4351 013650 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4352 013656 005037 005414 CLR E.A2 ;EXPECTED MSG A2
4353 013662 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4354 013670 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4355
4356 013676 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4357 013702 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4358 013704 104077 ERROR 77 ;MSG A0 ERROR AFTER TIMEOUT
4359 013706 104100 ERROR 100 ;MSH B0 ERROR
4360 013710 104101 ERROR 101 ;MSG A1 ERROR
4361 013712 104102 ERROR 102 ;MSG B1 ERROR
4362 013714 012737 000001 005464 MOV #1,UNITB ;SELECT PORT B BEFORE TIMEOUT OR RELEASE
4363 013722 112737 000102 056644 MOV #B,MSG19A ;SETUP ERROR MSG FOR PORT B
4364 013730 005037 001176 CLR $ESCAPE
4365 013734 004737 044172 JSR PC,DRAV ;SEE IF DRIVE AVAIL
4366 013740 000401 BR 1$ ;BR IF NOT AVAIL
4367 013742 104103 ERROR 103 ;PORT B AVAIL BEFORE TMO OR RELEASE
4368
4369 013744 032737 100000 005314 1$: BIT #CERR,HCS1
4370 013752 001001 BNE 6$
4371 013754 104130 ERROR 130 ;CERR NOT SET AFTER SEL DRIVE & DRIVE NOT AVAIL
4372 013756 012737 010100 005404 6$: MOV #<D.SPIN!D.VV>,E.A0
4373 013764 013737 005404 005410 MOV E.A0,E.A1 ;MSG 0 & 1 SHOULD ALWAYS RETURN SAME
4374 013772 053737 012776 005410 BIS E.DDT,E.A1
4375 014000 005037 005406 CLR E.B0 ;WORD 0 FOR PORT B
4376 014004 005037 005412 CLR E.B1
4377 014010 012737 014544 001176 MOV #5$, $ESCAPE
4378
4379 014016 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4380 014022 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4381 014024 104104 ERROR 104 ;MSG A0 ERROR WHILE PORT B UNAVAILABLE
4382 014026 104105 ERROR 105 ;MSH B0 ERROR
4383 014030 104106 ERROR 106 ;MSG A1 ERROR
4384 014032 104107 ERROR 107 ;MSG B1 ERROR
4385
4386 014034 005737 001216 TST $PASS
4387 014040 001402 BEQ 8$ ;BR IF FIRST PASS
4388 014042 000137 014544 JMP 5$ ;ELSE EXIT TEST
4389 014046 005037 001176 8$: CLR $ESCAPE
4390 014052 012765 100000 000000 MOV #CCLR,RKCS1(R5)
4391 014060 012737 000000 005464 MOV #0,UNITB ;SETUP FOR PORT A AGAIN
4392 014066 112737 000101 056644 MOV #A,MSG19A
4393 014074 012737 000360 001366 MOV #360,COUNT ;SETUP 4 SEC TIMEOUT
4394 014102 004737 047114 JSR PC,CLKON ;TURN ON CLOCK
4395
4396 014106 004737 044172 JSR PC,DRAV
4397 014112 104045 ERROR 45 ;PORT A NOT AVAIL AFTER TIMEOUT
4398
4399 014114 012737 000001 005464 MOV #1,UNITB ;SELECT PORT B BEFORE TIMEOUT OR RELEASE
4400 014122 112737 000102 056644 MOV #B,MSG19A
4401 014130 004737 044172 JSR PC,DRAV ;SEE IF PORT B DRIVE AVAIL
4402 014134 000401 BR 3$ ;BR IF NOT AVAIL
```



```

4403 014136 104103          ERROR 103          ;PORT B AVAIL BEFORE TMO OR RELEASE
4404
4405 014140 012765 100000 000000 3$: MOV #CCLR,RKCS1(R5)
4406 014146 013704 001222          MOV $UNIT,R4
4407 014152 063704 005464          ADD UNITB,R4
4408 014156 004737 0442          JSR PC,FATT3
4409 014162 104110          ERROR 110          ;NO ATTN ON PORT B TO ALLOW SEIZE
4410
4411 014164 004737 047162          JSR PC,CLKOF
4412 014170 004737 044172          JSR PC,DRAV          ;SEE IF PORT B DRIVE AVAIL
4413 014174 104045          ERROR 45          ;PORT B NOT AVAIL
4414
4415 014176 012737 014544 001176          MOV #5$,$ESCAPE
4416
4417 014204 012737 050340 005404          MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4418 014212 005037 005406          CLR E.B0          ;EXPECTED MSG B0
4419 014216 012737 001720 005410          MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4420 014224 012737 000001 005412          MOV #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
4421 014232 005037 005414          CLR E.A2          ;EXPECTED MSG A2
4422 014236 012737 000002 005416          MOV #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
4423 014244 012737 000003 005422          MOV #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
4424
4425 014252 004737 044274          JSR PC,CHKMSG          ;CHECK MSGS A0, B0, A1, B1
4426 014256 000000          .WORD 0!0!0          ;8 MSGS SPECIFIED HERE
4427 014260 104077          ERROR 77          ;MSG A0 ERROR AFTER TIMEOUT
4428 014262 104100          ERROR 100          ;MSH B0 ERROR
4429 014264 104101          ERROR 101          ;MSG A1 ERROR
4430 014266 104102          ERROR 102          ;MSG B1 ERROR
4431 014270 005037 001176          CLR $ESCAPE
4432 014274 012737 000000 005464          MOV #0,UNITB          ;SETUP FOR PORT A
4433 014302 112737 000101 056644          MOVB #'A,MSG19A
4434 014310 004737 043750          JSR PC,TSTATN          ;TEST FOR ATTN ON PORT A
4435 014314 000401          BR 4$
4436 014316 104111          ERROR 111          ;PORT A ATTN SET W/O REQUEST PENDING
4437
4438 014320 012737 01 544 001176 4$: MOV #5$,$ESCAPE
4439 014326 012737 000001 005464          MOV #1,UNITB          ;SETUP FOR PORT B
4440 014334 112737 000102 056644          MOVB #'B,MSG19A
4441
4442 014342 012765 100000 000000          MOV #CCLR,RKCS1(R5)
4443 014350 013765 001222 000010          MOV $UNIT,RKCS2(R5) ;DRIVE#
4444 014356 063765 005464 000010          ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
4445 014364 012737 000005 005314          MOV #CLEAR,HCS1
4446 014372 004737 043372          JSR PC,DOCMD          ;DO DRIVE CLEAR CMD & GET CONTR RDY
4447 014376 104151          ERROR 151          ;NO RDY AFTER DRIVE CLEAR CMD
4448 014400 004737 043750          JSR PC,TSTATN          ;TEST FOR ATTN
4449 014404 000401          BR 69$
4450 014406 104154          ERROR 154          ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4451 014410          69$:
4452
4453 014410 012737 010340 005404          MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4454 014416 005037 005406          CLR E.B0          ;EXPECTED MSG B0
4455 014422 012737 001720 005410          MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4456 014430 012737 000001 005412          MOV #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
4457 014436 005037 005414          CLR E.A2          ;EXPECTED MSG A2
4458 014442 012737 000002 005416          MOV #2,E.B2          ;MSG ID FOR EXPECTED MSG B2

```

```

4459 014450 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4460
4461 014456 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4462 014462 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4463 014464 104033 ERROR 33 ;MSG A0 ERROR AFTER DRV CLEAR CMD
4464 014466 104034 ERROR 34 ;MSH B0 ERROR
4465 014470 104035 ERROR 35 ;MSG A1 ERROR
4466 014472 104036 ERROR 36 ;MSG B1 ERROR
4467
4468
4469 014474 012701 000360 MOV #360,R1
4470 014500 163701 001366 SUB COUNT,R1 ;R1-COUNT=R1
4471 014504 012746 000021 MOV #17,-(SP) ;:PUT THE MULTIPLIER ON THE STACK
4472 014510 010146 MOV R1,-(SP) ;:PUT THE MULTIPLICAND ON THE STACK
4473 014512 004737 054354 JSR PC,@#SMULT ;:CALL THE MULTIPLY ROUTINE
4474 014516 012616 MOV (SP)+,(SP) ;:DISREGARD THE MSB'S
4475 014520 012601 MOV (SP)+,R1 ;:GET THE LSB'S OF THE PRODUCT
4476 014522 104401 056650 TYPE ,MSG20 ;PORT TIMEOUT
4477 014526 010146 MOV R1,-(SP) ;PUSH BINARY ONTO STACK
4478 014530 004737 054260 JSR PC,$SB2D ;CONVERT TO ASCII
4479 014534 004737 054314 JSR PC,$SUPRS ;TYPE IT
4480 014540 104401 056731 TYPE ,MSG22 ;MS
4481
4482 014544 005037 001176 5$: CLR $ESCAPE
4483 014550 004737 047162 JSR PC,CLKOF
4484
4485 ;:*****
4486 ;*TEST 6 TEST PORT 'B' SEIZE & TIMEOUT*
4487 ;*
4488 ;* THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
4489 ;*
4490 ;:*****
4491 014554 000004 TST6: SCOPE
4492 014556 012737 000001 001174 MOV #1,$TIMES ;:DO 1 ITERATION
4493 014564 012706 001100 MOV #STACK,SP
4494 014570 012737 000001 005464 MOV #1,UNITB ;:SETUP PORT B
4495 014576 112737 000102 056644 MOV #B,MSG19A
4496 014604 013737 005466 001366 MOV TIMER,COUNT
4497 014612 004737 047204 JSR PC,TMO ;:DO TIMEOUT
4498
4499 014616 004737 045534 JSR PC,SUBCLR
4500 014622 104024 ERROR 24 ;CERR AFTER SCLR
4501
4502
4503 014624 004737 044172 JSR PC,DRAV ;:SEE IF DRIVE AVAIL
4504 014630 104045 ERROR 45 ;:PORT B NOT AVAIL AFTER TMO
4505 014632 012737 015554 001176 MOV #5,$ESCAPE
4506
4507 014640 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4508 014646 005037 005406 CLR E.B0 ;EXPECTED MSG B0
4509 014652 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4510 014660 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4511 014666 005037 005414 CLR E.A2 ;EXPECTED MSG A2
4512 014672 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4513 014700 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4514

```

L 7
PAGE 90

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59
CZR6GC.P11 04-JAN-82 12:39 T6 TEST PORT 'B' SEIZE & TIMEOUT SEQ 0089

```

4515 014706 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4516 014712 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4517 014714 104077 ERROR 77 ;MSG A0 ERROR AFTER TIMEOUT
4518 014716 104100 ERROR 100 ;MSH B0 ERROR
4519 014720 104101 ERROR 101 ;MSG A1 ERROR
4520 014722 104102 ERROR 102 ;MSG B1 ERROR
4521 014724 012737 000000 005464 MOV #0,UNITB ;SELECT PORT A BEFORE TIMEOUT OR RELEASE
4522 014732 112737 000101 056644 MOVB #'A,MSG19A ;SETUP ERROR MSG FOR PORT A
4523 014740 005037 001176 CLR $ESCAPE
4524 014744 004737 044172 JSR PC,DRAV ;SEE IF DRIVE AVAIL
4525 014750 000401 BR 1$ ;BR IF NOT AVAIL
4526 014752 104103 ERROR 103 ;PORT A AVAIL BEFORE TMO OR RELEASE
4527
4528 014754 032737 100000 005314 1$: BIT #CERR,HCS1
4529 014762 001001 BNE 6$
4530 014764 104130 ERROR 130 ;CERR NOT SET AFTER SEL DRIVE & DRIVE NOT AVAIL
4531 014766 012737 010100 005404 6$: MOV #<D.SPIN!D.VV>,E.A0
4532 014774 013737 005404 005410 MOV E.A0,E.A1 ;MSG 1 & 0 SHOULD ALWAYS RETURN SAME
4533 015002 053737 012776 005410 BIS E.DDT,E.A1
4534 015010 005037 005406 CLR E.B0 ;WORD 0 FOR PORT A
4535 015014 005037 005412 CLR E.B1
4536 015020 012737 015554 001176 MOV #5$, $ESCAPE
4537
4538 015026 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4539 015032 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4540 015034 104104 ERROR 104 ;MSG A0 ERROR WHILE PORT A UNAVAILABLE
4541 015036 104105 ERROR 105 ;MSH B0 ERROR
4542 015040 104106 ERROR 106 ;MSG A1 ERROR
4543 015042 104107 ERROR 107 ;MSG B1 ERROR
4544
4545 015044 005737 001216 TST $PASS
4546 015050 001402 BEQ 8$ ;BR IF FIRST PASS
4547 015052 000137 015554 JMP 5$ ;ELSE EXIT TEST
4548 015056 005037 001176 CLR $ESCAPE
4549 015062 012765 100000 000000 8$: MOV #CCLR,RKCS1(R5)
4550 015070 012737 000001 005464 MOV #1,UNITB ;SETUP FOR PORT B AGAIN
4551 015076 112737 000102 056644 MOVB #'B,MSG19A
4552 015104 012737 000360 001366 MOV #360,COUNT ;SETUP 4 SEC TIMEOUT
4553 015112 004737 047114 JSR PC,CLKON ;TURN ON CLOCK
4554
4555 015116 004737 044172 JSR PC,DRAV
4556 015122 104045 ERROR 45 ;PORT B NOT AVAIL AFTER TIMEOUT
4557
4558 015124 012737 000000 005464 MOV #0,UNITB ;SELECT PORT A BEFORE TIMEOUT OR RELEASE
4559 015132 112737 000101 056644 MOVB #'A,MSG19A
4560 015140 004737 044172 JSR PC,DRAV ;SEE IF PORT A DRIVE AVAIL
4561 015144 000401 BR 3$ ;BR IF NOT AVAIL
4562 015146 104103 ERROR 103 ;PORT A AVAIL BEFORE TMO OR RELEASE
4563
4564 015150 012765 100000 000000 3$: MOV #CCLR,RKCS1(R5)
4565 015156 013704 001222 MOV $UNIT,R4
4566 015162 063704 005464 ADD UNITB,R4
4567 015166 004737 044214 JSR PC,FAIT3
4568 015172 104110 ERROR 110 ;NO ATTN ON PORT A TO ALLOW SEIZE
4569
4570 015174 004737 047162 JSR PC,CLKOF

```

```

4571 015200 004737 044172 JSR PC,DRAV ;SEE IF PORT A DRIVE AVAIL
4572 015204 104045 ERROR 45 ;PORT A NOT AVAIL
4573
4574 015206 012737 015554 001176 MOV #5$,SESCAPE
4575
4576 015214 012737 050340 005404 MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4577 015222 005037 005406 CLR E.B0 ;EXPECTED MSG B0
4578 015226 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4579 015234 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4580 015242 005037 005414 CLR E.A2 ;EXPECTED MSG A2
4581 015246 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4582 015254 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4583
4584 015262 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4585 015266 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4586 015270 104077 ERROR 77 ;MSG A0 ERROR AFTER TIMEOUT
4587 015272 104100 ERROR 100 ;MSH B0 ERROR
4588 015274 104101 ERROR 101 ;MSG A1 ERROR
4589 015276 104102 ERROR 102 ;MSG B1 ERROR
4590 015300 005037 001176 CLR $ESCAPE
4591 015304 012737 000001 005464 MOV #1,UNITB ;SETUP FOR PORT B
4592 015312 112737 000102 056644 MOVB #'B,MSG19A
4593 015320 004737 043750 JSR PC,TSTATN ;TEST FOR ATTN ON PORT B
4594 015324 000401 BR 4$
4595 015326 104111 ERROR 111 ;PORT B ATTN SET W/O REQUEST PENDING
4596
4597 015330 012737 015554 001176 4$: MOV #5$,SESCAPE
4598 015336 012737 000000 005464 MOV #0,UNITB ;SETUP FOR PORT A
4599 015344 112737 000101 056644 MOVB #'A,MSG19A
4600
4601 015352 012765 100000 000000 MOV #CLR,RKCS1(R5)
4602 015360 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
4603 015366 063765 005464 000010 ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
4604 015374 012737 000005 005314 MOV #CLEAR,HCS1
4605 015402 004737 043372 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
4606 015406 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
4607 015410 004737 043750 JSR PC,TSTATN ;TEST FOR ATTN
4608 015414 000401 BR 64$
4609 015416 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4610 015420 64$:
4611
4612 015420 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4613 015426 005037 005406 CLR E.B0 ;EXPECTED MSG B0
4614 015432 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4615 015440 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4616 015446 005037 005414 CLR E.A2 ;EXPECTED MSG A2
4617 015452 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4618 015460 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4619
4620 015466 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4621 015472 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4622 015474 104033 ERROR 33 ;MSG A0 ERROR AFTER DRV CLEAR CMD
4623 015476 104034 ERROR 34 ;MSH B0 ERROR
4624 015500 104035 ERROR 35 ;MSG A1 ERROR
4625 015502 104036 ERROR 36 ;MSG B1 ERROR
4626

```


4683
4684 015706 004737 054314

JSR PC,\$SUPRS ;TYPE
TYPE ,SCLRF ;29-SEP-77
TYPE ,SCLRF ;29-SEP-77

4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703

:TEST 10 TEST PORT 'A' COMMAND SEIZE & ATTENTION
:*****
:VERIFY THE OPERATION OF 'DSC' & 'ATTN' BITS AFTER A COMMAND.
:*****
:A. ISSUE A SEEK COMMAND TO CYLINDER 10 THRU PORT 'A'.
:*****
:B. VERIFY SEIZURE & THAT 'DSC' & 'ATTN' SETS FOR PORT 'A'
: ONLY AFTER SEEK COMPLETION
:*****
:C. VERIFY 'ATTN' REMAINS SET BEYOND TIMECUT
:*****
:D. VERIFY A DRIVE CLEAR COMMAND RESETS 'DSC' & 'ATTN'
: & DOES NOT RELEASE THE DRIVE FROM PORT 'A'.
:*****

4704 015712 000004
4705 015714 012737 000001 001174
4706 015722 012706 001100
4707 015726 012737 000000 005464
4708 015734 112737 000101 056644
4709 015742 013737 005466 001366
4710 015750 004737 047204
4711
4712 015754 004737 045534
4713 015760 104024
4714
4715
4716 015762 004737 044172
4717 015766 104045
4718 015770 012765 000012 000020
4719
4720 015776 012737 000017 005314
4721 016004 004737 043372
4722 016010 104131
4723
4724 016012 013737 001412 005352
4725 016020 004737 044106
4726 016024 104132
4727
4728 016026 032737 100000 005314
4729 016034 001401
4730 016036 104210
4731
4732 016040
4733
4734 016040 012737 050340 005404
4735 016046 005037 005406
4736 016052 012737 001720 005410
4737 016060 012737 000001 005412
4738 016066 005037 005414

TST10: SCOPE
MOV #1,\$TIMES ;;DO 1 ITERATION
MOV #STACK,SP
MOV #0,UNITB ;SETUP PORT A
MOVB #'A,MSG19A
MOV TIMER,COUNT
JSR PC,TMO ;DO TIMEOUT
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
JSR PC,DRAV ;SEE IF DRIVE AVAIL
ERROR 45 ;PORT A NOT AVAIL AFTER TMO
MOV #10.,RKDC(R5) ;SEEK TO CYL 10.
MOV #SEEK,HCS1
JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
ERROR 131 ;NO RDY AFTER SEEK CMD
MOV T5000,TEMP1 ;SETUP TIMEOUT
JSR PC,FATT2 ;FIND ATTN
ERROR 132 ;NO ATTN AFTER SEEK CMD
BIT #CERR,HCS1
BEQ 65\$
ERROR 210 ;CERR AFTER SEEK CMD
65\$:
MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
CLR E.B0 ;EXPECTED MSG B0
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
CLR E.A2 ;EXPECTED MSG A2

```

CZR6GCO RK611 DU PORT LGC          MACY11 30(1046) 04-JAN-82 12:59 C 8
CZR6GC.P11 04-JAN-82 12:39         T10      TEST PORT 'A' COMMAND SEIZE & ATTENTION PAGE 94
                                                    SEQ 0093
4739 016072 012737 000002 005416  MOV #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
4740 016100 012737 000003 005422  MOV #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
4741                                     ;
4742 016106 004737 044274          JSR PC,CHKMSG    ;CHECK MSGS A0, B0, A1, B1
4743 016112 000000                    .WORD 0!0!0      ;& MSGS SPECIFIED HERE
4744 016114 104161                    ERROR 161        ;MSG A0 ERROR AFTER SEEK CMD
4745 016116 104162                    ERROR 162        ;MSH B0 ERROR
4746 016120 104163                    ERROR 163        ;MSG A1 ERROR
4747 016122 104164                    ERROR 164        ;MSG B1 ERROR
4748                                     ;
4749 016124 004737 045756          JSR PC,RDCYLA    ;READ CYL ADDR
4750 016130 023727 001360 000012  CMP CYLADD,#10. ;SEE IF CYL 10.
4751 016136 001415                    BEQ 64$          ;BR IF YES
4752 016140 012737 000000 001344  MOV #0,FRCYL
4753 016146 012737 000012 001346  MOV #10.,TOCYL
4754 016154 012737 000012 001354  MOV #10.,CALDIF
4755 016162 012765 000012 000020  MOV #10.,RKDC(R5) ;REFRESH RKDC
4756 016170 104224                    ERROR 224        ;DID NOT SEEK TO CYL 10.
4757                                     ;
4758 016172 012737 000226 001366 64$: MOV #150.,COUNT
4759 016200 004737 047204          JSR PC,TMO      ;DO 2.5 SEC TIMEOUT
4760 016204 004737 044106          JSR PC,FATT2
4761 016210 104112                    ERROR 112        ;ATTN CLEARED AFTER TMO
4762                                     ;
4763 016212 012765 100000 000000  MOV #CCLR,RKCS1(R5)
4764 016220 013765 001222 000010  MOV $UNIT,RKCS2(R5) ;DRIVE#
4765 016226 063765 005464 000010  ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
4766 016234 012737 000005 005314  MOV #CLEAR,HCS1
4767 016242 004737 043372          JSR PC,DOCMD    ;DO DRIVE CLEAR CMD & GET CONTR RDY
4768 016246 104151                    ERROR 151        ;NO RDY AFTER DRIVE CLEAR CMD
4769 016250 004737 043750          JSR PC,TSTATN  ;TEST FOR ATTN
4770 016254 000401                    BR 66$
4771 016256 104154                    ERROR 154        ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4772 016260          66$:
4773                                     ;
4774 016260 012737 010340 005404  MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4775 016266 005037 005406          CLR E.B0        ;EXPECTED MSG B0
4776 016272 012737 001720 005410  MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4777 016300 012737 000001 005412  MOV #1,E.B1     ;MSG ID FOR EXPECTED MSG B1
4778 016306 005037 005414          CLR E.A2        ;EXPECTED MSG A2
4779 016312 012737 000002 005416  MOV #2,E.B2     ;MSG ID FOR EXPECTED MSG B2
4780 016320 012737 000003 005422  MOV #3,E.B3     ;MSG ID FOR EXPECTED MSG B3
4781                                     ;
4782 016326 004737 044274          JSR PC,CHKMSG    ;CHECK MSGS A0, B0, A1, B1
4783 016332 000003                    .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4784 016334 104033                    ERROR 33        ;MSG A0 ERROR AFTER DRV CLEAR CMD
4785 016336 104034                    ERROR 34        ;MSH B0 ERROR
4786 016340 104035                    ERROR 35        ;MSG A1 ERROR
4787 016342 104036                    ERROR 36        ;MSG B1 ERROR
4788                                     ;
4789 016344 004737 044172          JSR PC,DRAV
4790 016350 104113                    ERROR 113        ;PORT A NOT AVAIL AFTER DRIVE CLEAR CMD
4791                                     ;
4792                                     ;*****
4793                                     ;*TEST 11      TEST PORT 'B' COMMAND SEIZE & ATTENTION
4794                                     ;*

```

4795
4796
4797
4798
4799 016352 000004
4800 016354 012737 000001 001174
4801 016362 012706 001100
4802 016366 012737 000001 005464
4803 016374 112737 000102 056644
4804 016402 013737 005466 001366
4805 016410 004737 047204
4806
4807 016414 004737 045534
4808 016420 104024
4809
4810
4811 016422 004737 044172
4812 016426 104045
4813 016430 012765 000000 000020
4814
4815 016436 012737 000017 005314
4816 016444 004737 043372
4817 016450 104131
4818
4819 016452 013737 001412 005352
4820 016460 004737 044106
4821 016464 104132
4822
4823 016466 032737 100000 005314
4824 016474 001401
4825 016476 104210
4826
4827 016500
4828
4829 016500 012737 050340 005404
4830 016506 005037 005406
4831 016512 012737 001720 005410
4832 016520 012737 000001 005412
4833 016526 005037 005414
4834 016532 012737 000002 005416
4835 016540 012737 000003 005422
4836
4837 016546 004737 044274
4838 016552 000000
4839 016554 104161
4840 016556 104162
4841 016560 104163
4842 016562 104164
4843
4844 016564 004737 045756
4845 016570 023727 001360 000000
4846 016576 001415
4847 016600 012737 000012 001344
4848 016606 012737 000000 001346
4849 016614 012737 000000 001354
4850 016622 012765 000000 000020

*** THE PREVIOUS TEST IS REPEATED FOR PORT 'B',
*** BUT THE SEEK IS TO CYLINDER 0

TST11: SCOPE
MOV #1,\$TIMES ;:DO 1 ITERATION
MOV #STACK,SP
MOV #1,UNITB ;:SETUP PORT B
MOVB #'B,MSG19A
MOV TIMER,COUNT
JSR PC,TMO ;:DO TIMEOUT
JSR PC,SUBCLR
ERROR 24 ;:CERR AFTER SCLR
JSR PC,DRAV ;:SEE IF DRIVE AVAIL
ERROR 45 ;:PORT B NOT AVAIL AFTER TMO
MOV #0,RKDC(R5) ;:SEEK TO CYL 0
MOV #SEEK,HCS1
JSR PC,DOCMD ;:DO SEEK CMD & GET CONTR READY
ERROR 131 ;:NO RDY AFTER SEEK CMD
MOV T50000,TEMP1 ;:SETUP TIMEOUT
JSR PC,FATT2 ;:FIND ATTN
ERROR 132 ;:NO ATTN AFTER SEEK CMD
BIT #CERR,HCS1
BEQ 65\$
ERROR 210 ;:CERR AFTER SEEK CMD
65\$:
MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;:EXPECTED MSG A0
CLR E.B0 ;:EXPECTED MSG B0
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;:EXPECTED A1
MOV #1,E.B1 ;:MSG ID FOR EXPECTED MSG B1
CLR E.A2 ;:EXPECTED MSG A2
MOV #2,E.B2 ;:MSG ID FOR EXPECTED MSG B2
MOV #3,E.B3 ;:MSG ID FOR EXPECTED MSG B3
JSR PC,CHKMSG ;:CHECK MSGS A0, B0, A1, B1
.WORD 0!0!0 ;:& MSGS SPECIFIED HERE
ERROR 161 ;:MSG A0 ERROR AFTER SEEK CMD
ERROR 162 ;:MSH B0 ERROR
ERROR 163 ;:MSG A1 ERROR
ERROR 164 ;:MSG B1 ERROR
JSR PC,RDCYLA ;:READ CYL ADDR
CMP CYLADD,#0 ;:SEE IF CYL 0
BEQ 64\$;:BR IF YES
MOV #10,,FRCYL
MOV #0,TOCYL
MOV #0,CALDIF
MOV #0,RKDC(R5) ;:REFRESH RKDC


```

4851 016630 104224 ERROR 224 ;DID NOT SEEK TO CYL 0
4852
4853 016632 012737 000226 001366 64$: MOV #150,COUNT
4854 016640 004737 047204 JSR PC,TMO ;DO 2.5 SEC TIMEOUT
4855 016644 004737 044106 JSR PC,FATT2
4856 016650 104112 ERROR 112 ;ATTN CLEARED AFTER TMO
4857
4858 016652 012765 100000 000000 MOV #CCLR,RKCS1(R5)
4859 016660 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
4860 016666 063765 005464 000010 ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
4861 016674 012737 000005 005314 MOV #CLEAR,HCS1
4862 016702 004737 043372 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
4863 016706 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
4864 016710 004737 043750 JSR PC,TSTATN ;TEST FOR ATTN
4865 016714 000401 BR 66$
4866 016716 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4867 016720 66$:
4868
4869 016720 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4870 016726 005037 005406 CLR E.B0 ;EXPECTED MSG B0
4871 016732 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4872 016740 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4873 016746 005037 005414 CLR E.A2 ;EXPECTED MSG A2
4874 016752 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4875 016760 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4876
4877 016766 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4878 016772 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4879 016774 104033 ERROR 33 ;MSG A0 ERROR AFTER DRV CLEAR CMD
4880 016776 104034 ERROR 34 ;MSH B0 ERROR
4881 017000 104035 ERROR 35 ;MSG A1 ERROR
4882 017002 104036 ERROR 36 ;MSG B1 ERROR
4883
4884 017004 004737 044172 JSR PC,DRAV
4885 017010 104113 ERROR 113 ;PORT B NOT AVAIL AFTER DRIVE CLEAR CMD
4886
4887

```

```

*****
*TEST 12 TEST RESET PORT 'A' ATTENTION BY DRIVE CLEAR COMMAND
*
* VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF
* THE SEIZING PORT
*
* A. SET EACH PORT'S ATTENTION BIT BY PERFORMING SEEK
* COMMANDS TO CYLINDER 0 & ALLOWING TIMEOUTS.
*
* B. SEIZE THE DRIVE THRU PORT 'A' & ISSUE A DRIVE CLEAR COMMAND
* VERIFY THAT 'DSC' & 'ATTN' FOR PORT 'A' HAVE BEEN CLEARED
*
* C. SEIZE THE DRIVE THRU PORT 'B' & VERIFY 'DSC' & 'ATTN'
* HAVE NOT CLEARED
*****

```

```

4901
4902 017012 000004 TST12: SCOPE
4903 017014 012737 000001 001174 MOV #1,$TIMES ;;DO 1 ITERATION
4904 017022 012706 001100 MOV #STACK,SP
4905 017026 012737 000000 005464 MOV #0,UNITB ;SETUP PORT A
4906 017034 112737 000101 056644 MOVB #'A,MSG19A

```

F 8
PAGE 97

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59
 CZR6GC.P11 04-JAN-82 12:39 T12 TEST RESET PORT 'A' ATTENTION BY DRIVE CLEAR COMMAND SEQ 0096

4907	017042	013737	005466	001366	MOV	TIMER,COUNT	
4908	017050	004737	047204		JSR	PC,TMO	;DO TIMEOUT
4909							
4910	017054	004737	045534		JSR	PC,SUBCLR	
4911	017060	104024			ERROR	24	;CERR AFTER SCLR
4912							
4913							
4914	017062	004737	044172		JSR	PC,DRAV	;SEE IF DRIVE AVAIL
4915	017066	104045			ERROR	45	;PORT A NOT AVAIL AFTER TMO
4916							
4917	017070	012737	000017	005314	MOV	#SEEK,HCS1	
4918	017076	004737	043372		JSR	PC,DOCMD	;DO SEEK CMD & GET CONTR READY
4919	017102	104131			ERROR	131	;NO RDY AFTER SEEK CMD
4920							
4921	017104	013737	001412	005352	MOV	T50000,TEMP1	;SETUP TIMEOUT
4922	017112	004737	044106		JSR	PC,FATT2	;FIND ATTN
4923	017116	104132			ERROR	132	;NO ATTN AFTER SEEK CMD
4924							
4925	017120	032737	100000	005314	BIT	#CERR,HCS1	
4926	017126	001401			BEQ	64\$	
4927	017130	104210			ERROR	210	;CERR AFTER SEEK CMD
4928							
4929	017132			64\$:			
4930							
4931	017132	012737	050340	005404	MOV	#<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	;EXPECTED MSG A0
4932	017140	005037	005406		CLR	E.B0	;EXPECTED MSG B0
4933	017144	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4934	017152	012737	000001	005412	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4935	017160	005037	005414		CLR	E.A2	;EXPECTED MSG A2
4936	017164	012737	000002	005416	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4937	017172	012737	000003	005422	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4938							
4939	017200	004737	044274		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4940	017204	000000			.WORD	0!0!0	;8 MSGS SPECIFIED HERE
4941	017206	104161			ERROR	161	;MSG A0 ERROR AFTER SEEK CMD
4942	017210	104162			ERROR	162	;MSG B0 ERROR
4943	017212	104163			ERROR	163	;MSG A1 ERROR
4944	017214	104164			ERROR	164	;MSG B1 ERROR
4945							
4946	017216	013737	005466	001366	MOV	TIMER,COUNT	
4947	017224	004737	047204		JSR	PC,TMO	;DO 1.5 SEC TIMEOUT
4948	017230	012737	000001	005464	MOV	#1,UNITB	;SETUP PORT B
4949	017236	112737	000102	056644	MOVB	#'B,MSG19A	
4950	017244	004737	044172		JSR	PC,DRAV	
4951	017250	104045			ERROR	45	;PORT B NOT AVAIL AFTER TMO
4952							
4953							
4954	017252	012737	010340	005404	MOV	#<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0	;EXPECTED MSG A0
4955	017260	005037	005406		CLR	E.B0	;EXPECTED MSG B0
4956	017264	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4957	017272	012737	000001	005412	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4958	017300	005037	005414		CLR	E.A2	;EXPECTED MSG A2
4959	017304	012737	000002	005416	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4960	017312	012737	000003	005422	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4961							
4962	017320	004737	044274		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046)
T12

04-JAN-82 12:59 PAGE 98
TEST RESET PORT 'A' ATTENTION BY DRIVE CLEAR COMMAND

SEQ 0097

```
4963 017324 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4964 017326 104165 ERROR 165 ;MSG A0 ERROR AFTER TIMEOUT
4965 017330 104166 ERROR 166 ;MSH B0 ERROR
4966 017332 104167 ERROR 167 ;MSG A1 ERROR
4967 017334 104170 ERROR 170 ;MSG B1 ERROR
4968
4969 017336 012737 000017 005314 MOV #SEEK,HCS1
4970 017344 004737 043372 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
4971 017350 104131 ERROR 131 ;NO RDY AFTER SEEK CMD
4972
4973 017352 013737 001412 005352 MOV T50000,TEMP1 ;SETUP TIMEOUT
4974 017360 004737 044106 JSR PC,FATT2 ;FIND ATTN
4975 017364 104132 ERROR 132 ;NO ATTN AFTER SEEK CMD
4976
4977 017366 032737 100000 005314 BIT #CERR,HCS1
4978 017374 001401 BEQ 65$
4979 017376 104210 ERROR 210 ;CERR AFTER SEEK CMD
4980
4981 017400 65$:
4982
4983 017400 012737 050340 005404 MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
4984 017406 005037 005406 CLR E.B0 ;EXPECTED MSG B0
4985 017412 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4986 017420 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4987 017426 005037 005414 CLR E.A2 ;EXPECTED MSG A2
4988 017432 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4989 017440 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4990
4991 017446 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4992 017452 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4993 017454 104161 ERROR 161 ;MSG A0 ERROR AFTER SEEK CMD
4994 017456 104162 ERROR 162 ;MSH B0 ERROR
4995 017460 104163 ERROR 163 ;MSG A1 ERROR
4996 017462 104164 ERROR 164 ;MSG B1 ERROR
4997
4998 017464 013737 005466 001366 MOV TIMER,COUNT
4999 017472 004737 047204 JSR PC,TMO
5000 017476 012737 000000 005464 MOV #0,UNITB ;SETUP PORTC
5001 017504 112737 000101 056644 MOVB #'A,MSG19A
5002 017512 004737 044172 JSR PC,DRAV
5003 017516 104045 ERROR 45 ;PORT A NOT AVAIL AFTER TMO
5004
5005
5006 017520 012737 050340 005404 MOV #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
5007 017526 005037 005406 CLR E.B0 ;EXPECTED MSG B0
5008 017532 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5009 017540 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5010 017546 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5011 017552 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5012 017560 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5013
5014 017566 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5015 017572 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
5016 017574 104165 ERROR 165 ;MSG A0 ERROR AFTER TIMEOUT
5017 017576 104166 ERROR 166 ;MSH B0 ERROR
5018 017600 104167 ERROR 167 ;MSG A1 ERROR
```

```

5019 017602 104170 ERROR 170 ;MSG B1 ERROR
5020
5021 017604 012765 100000 000000 MOV #CCLR,RKCS1(R5)
5022 017612 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
5023 017620 063765 005464 000010 ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
5024 017626 012737 000005 005314 MOV #CLEAR,HCS1
5025 017634 004737 043372 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
5026 017640 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
5027 017642 004737 043750 JSR PC,TSTATN ;TEST FOR ATTN
5028 017646 000401 BR 66$
5029 017650 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5030 017652
5031 66$:
5032 017652 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
5033 017660 005037 005406 CLR E.B0 ;EXPECTED MSG B0
5034 017664 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5035 017672 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5036 017700 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5037 017704 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5038 017712 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5039
5040 017720 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5041 017724 000003 .WORD T.A2:T.B2!0 ;& MSGS SPECIFIED HERE
5042 017726 104033 ERROR 33 ;MSG A0 ERROR AFTER DRV CLEAR CMD
5043 017730 104034 ERROR 34 ;MSH B0 ERROR
5044 017732 104035 ERROR 35 ;MSG A1 ERROR
5045 017734 104036 ERROR 36 ;MSG B1 ERROR
5046
5047 017736 013737 005466 001366 MOV TIMER,COUNT
5048 017744 004737 047204 JSR PC,TMO ;DO 1.5 SEC TIMEOUT
5049 017750 012737 000001 005464 MOV #1,UNITB ;SETUP PORT B
5050 017756 112737 000102 056644 MOV #B,MSG19A
5051 017764 004737 043750 JSR PC,TSTATN
5052 017770 104114 ERROR 114 ;ATTN RESET ON PORT B AFTER DR CLR CMD
5053 ;ON PORT A
5054
5055
5056 017772 012737 050340 005404 MOV #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
5057 020000 005037 005406 CLR E.B0 ;EXPECTED MSG B0
5058 020004 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5059 020012 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5060 020020 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5061 020024 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5062 020032 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5063
5064 020040 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5065 020044 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
5066 020046 104165 ERROR 165 ;MSG A0 ERROR AFTER TIMEOUT
5067 020050 104166 ERROR 166 ;MSH B0 ERROR
5068 020052 104167 ERROR 167 ;MSG A1 ERROR
5069 020054 104170 ERROR 170 ;MSG B1 ERROR
5070
5071
5072
5073
5074

```

```

*****
*TEST 13 TEST RESET PORT 'B' ATTENTION BY DRIVE CLEAR COMMAND
*
* THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.

```

```
5075
5076
5077 020056 000004
5078 020060 012737 000001 001174
5079 020066 012706 001100
5080 020072 012737 000001 005464
5081 020100 112737 000102 056644
5082 020106 013737 005466 001366
5083 020114 004737 047204
5084
5085 020120 004737 045534
5086 020124 104024
5087
5088
5089 020126 004737 044172
5090 020132 104045
5091
5092 020134 012737 000017 005314
5093 020142 004737 043372
5094 020146 104131
5095
5096 020150 013737 001412 005352
5097 020156 004737 044106
5098 020162 104132
5099
5100 020164 032737 100000 005314
5101 020172 001401
5102 020174 104210
5103
5104 020176
5105
5106 020176 012737 050340 005404
5107 020204 005037 005406
5108 020210 012737 001720 005410
5109 020216 012737 000001 005412
5110 020224 005037 005414
5111 020230 012737 000002 005416
5112 020236 012737 000003 005422
5113
5114 020244 004737 044274
5115 020250 000000
5116 020252 104161
5117 020254 104162
5118 020256 104163
5119 020260 104164
5120
5121 020262 013737 005466 001366
5122 020270 004737 047204
5123 020274 012737 000000 005464
5124 020302 112737 000101 056644
5125 020310 004737 044172
5126 020314 104045
5127
5128
5129 020316 012737 010340 005404
5130 020324 005037 005406
```

TST13: SCOPE
MOV #1,\$TIMES ;DO 1 ITERATION
MOV #STACK,SP
MOV #1,UNITB ;SETUP PORT B
MOVB #'B,MSG19A
MOV TIMER,COUNT
JSR PC,TMO ;DO TIMEOUT
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
JSR PC,DRAV ;SEE IF DRIVE AVAIL
ERROR 45 ;PORT B NOT AVAIL AFTER TMO
MOV #SEEK,HCS1
JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
ERROR 131 ;NO RDY AFTER SEEK CMD
MOV T50000,TEMP1 ;SETUP TIMEOUT
JSR PC,FATT2 ;FIND ATTN
ERROR 132 ;NO ATTN AFTER SEEK CMD
BIT #CERR,HCS1
BEQ 64\$
ERROR 210 ;CERR AFTER SEEK CMD
64\$:
MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
CLR E.B0 ;EXPECTED MSG B0
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
CLR E.A2 ;EXPECTED MSG A2
MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
.WORD 0!0!0 ;& MSGS SPECIFIED HERE
ERROR 161 ;MSG A0 ERROR AFTER SEEK CMD
ERROR 162 ;MSH B0 ERROR
ERROR 163 ;MSG A1 ERROR
ERROR 164 ;MSG B1 ERROR
MOV TIMER,COUNT
JSR PC,TMO ;DO 1.5 SEC TIMEOUT
MOV #0,UNITB ;SETUP PORT A
MOVB #'A,MSG19A
JSR PC,DRAV
ERROR 45 ;PORT A NOT AVAIL AFTER TMO
MOV #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
CLR E.B0 ;EXPECTED MSG B0

```

5131 020330 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5132 020336 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5133 020344 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5134 020350 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5135 020356 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5136
5137 020364 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5138 020370 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
5139 020372 104165 ERROR 165 ;MSG A0 ERROR AFTER TIMEOUT
5140 020374 104166 ERROR 166 ;MSH B0 ERROR
5141 020376 104167 ERROR 167 ;MSG A1 ERROR
5142 020400 104170 ERROR 170 ;MSG B1 ERROR
5143
5144 020402 012737 000017 005314 MOV #SEEK,HCS1
5145 020410 004737 043372 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
5146 020414 104131 ERROR 131 ;NO RDY AFTER SEEK CMD
5147
5148 020416 013737 001412 005352 MOV T50000,TEMP1 ;SETUP TIMEOUT
5149 020424 004737 044106 JSR PC,FATT2 ;FIND ATTN
5150 020430 104132 ERROR 132 ;NO ATTN AFTER SEEK CMD
5151
5152 020432 032737 100000 005314 BIT #CERR,HCS1
5153 020440 001401 BEQ 65$
5154 020442 104210 ERROR 210 ;CERR AFTER SEEK CMD
5155
5156 020444 65$:
5157
5158 020444 012737 050340 005404 MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
5159 020452 005037 005406 CLR E.B0 ;EXPECTED MSG B0
5160 020456 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5161 020464 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5162 020472 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5163 020476 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5164 020504 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5165
5166 020512 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5167 020516 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
5168 020520 104161 ERROR 161 ;MSG A0 ERROR AFTER SEEK CMD
5169 020522 104162 ERROR 162 ;MSH B0 ERROR
5170 020524 104163 ERROR 163 ;MSG A1 ERROR
5171 020526 104164 ERROR 164 ;MSG B1 ERROR
5172
5173 020530 013737 005466 001366 MOV TIMER,COUNT
5174 020536 004737 047204 JSR PC,TMO
5175 020542 012737 000001 005464 MOV #1,UNITB ;SETUP PORTC
5176 020550 112737 000102 056644 MOVB #'B,MSG19A
5177 020556 004737 044172 JSR PC,DRAV
5178 020562 104045 ERROR 45 ;PORT B NOT AVAIL AFTER TMO
5179
5180
5181 020564 012737 050340 005404 MOV #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
5182 020572 005037 005406 CLR E.B0 ;EXPECTED MSG B0
5183 020576 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5184 020604 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5185 020612 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5186 020616 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2

```

K 8
PAGE 102

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59
CZR6GC.P11 04-JAN-82 12:39 T13 TEST RESET PORT 'B' ATTENTION BY DRIVE CLEAR COMMAND SEQ 0101

```

5187 020624 012737 000003 005422      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
5188
5189 020632 004737 044274      JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
5190 020636 000000      .WORD      0!0!0      ;& MSGS SPECIFIED HERE
5191 020640 104165      ERROR      165      ;MSG A0 ERROR AFTER TIMEOUT
5192 020642 104166      ERROR      166      ;MSH B0 ERROR
5193 020644 104167      ERROR      167      ;MSG A1 ERROR
5194 020646 104170      ERROR      170      ;MSG B1 ERROR
5195
5196 020650 012765 100000 000000      MOV      #CLR,RKCS1(R5)
5197 020656 013765 001222 000010      MOV      SUNIT,RKCS2(R5)      ;DRIVE#
5198 020664 063765 005464 000010      ADD      UNITB,RKCS2(R5)      ;ADD 1 IF ON PORT B
5199 020672 012737 000005 005314      MOV      #CLEAR,HCS1
5200 020700 004737 043372      JSR      PC,DOCMD      ;DO DRIVE CLEAR CMD & GET CONTR RDY
5201 020704 104151      ERROR      151      ;NO RDY AFTER DRIVE CLEAR CMD
5202 020706 004737 043750      JSR      PC,TSTATN      ;TEST FOR ATTN
5203 020712 000401      BR      66$
5204 020714 104154      ERROR      154      ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5205 020716      66$:
5206
5207 020716 012737 010340 005404      MOV      #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0      ;EXPECTED MSG A0
5208 020724 005037 005406      CLR      E.B0      ;EXPECTED MSG B0
5209 020730 012737 001720 005410      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1      ;EXPECTED A1
5210 020736 012737 000001 005412      MOV      #1,E.B1      ;MSG ID FOR EXPECTED MSG B1
5211 020744 005037 005414      CLR      E.A2      ;EXPECTED MSG A2
5212 020750 012737 000002 005416      MOV      #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
5213 020756 012737 000003 005422      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
5214
5215 020764 004737 044274      JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
5216 020770 000003      .WORD      T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
5217 020772 104033      ERROR      33      ;MSG A0 ERROR AFTER DRV CLEAR CMD
5218 020774 104034      ERROR      34      ;MSH B0 ERROR
5219 020776 104035      ERROR      35      ;MSG A1 ERROR
5220 021000 104036      ERROR      36      ;MSG B1 ERROR
5221
5222 021002 013737 005466 001366      MOV      TIMER,COUNT
5223 021010 004737 047204      JSR      PC,TMO      ;DO 1.5 SEC TIMEOUT
5224 021014 012737 000000 005464      MOV      #0,UNITB      ;SETUP PORT A
5225 021022 112737 000101 056644      MOV      #A,MSG19A
5226 021030 004737 043750      JSR      PC,TSTATN
5227 021034 104114      ERROR      114      ;ATTN RESET ON PORT A AFTER DR CLR CMD
5228      ;ON PORT B
5229
5230
5231 021036 012737 050340 005404      MOV      #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0      ;EXPECTED MSG A0
5232 021044 005037 005406      CLR      E.B0      ;EXPECTED MSG B0
5233 021050 012737 001720 005410      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1      ;EXPECTED A1
5234 021056 012737 000001 005412      MOV      #1,E.B1      ;MSG ID FOR EXPECTED MSG B1
5235 021064 005037 005414      CLR      E.A2      ;EXPECTED MSG A2
5236 021070 012737 000002 005416      MOV      #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
5237 021076 012737 000003 005422      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
5238
5239 021104 004737 044274      JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
5240 021110 000000      .WORD      0!0!0      ;& MSGS SPECIFIED HERE
5241 021112 104165      ERROR      165      ;MSG A0 ERROR AFTER TIMEOUT
5242 021114 104166      ERROR      166      ;MSH B0 ERROR

```

5243 021116 104167
5244 021120 104170

ERROR 167 ;MSG A1 ERROR
ERROR 170 ;MSG B1 ERROR

5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256

```
*****  
:TEST 14 TEST RELEASE, DRIVE SEIZED BY PORT 'A'  
:  
: A. SEIZE THE DRIVE THRU PORT 'A'  
:  
: B. ISSUE A RELEASE USING RKCS2 THRU PORT 'A'  
:  
: C. VERIFY PORT 'B' CAN ACCESS THE DRIVE IMMEDIATELY &  
: THAT NEITHER PORT SEES 'DSC' OR 'ATTN'  
:  
*****
```

5257 021122 000004
5258 021124 012737 000001 001174
5259 021132 012706 001100
5260 021136 012737 000000 005464
5261 021144 112737 000101 056644
5262 021152 013737 005466 001366
5263 021160 004737 047204
5264
5265 021164 004737 045534
5266 021170 104024
5267
5268
5269 021172 004737 044172
5270 021176 104045
5271 021200 013765 001222 000010
5272 021206 012737 000000 005464
5273 021214 063765 005464 000010
5274 021222 112737 000101 056644
5275 021230 062765 000010 000010
5276 021236 012737 000001 005314
5277 021244 004737 043372
5278 021250 104117
5279
5280 021252 013765 001222 000010
5281 021260 012737 000001 005464
5282 021266 063765 005464 000010
5283 021274 112737 000102 056644
5284 021302 012737 000001 005314
5285 021310 004737 043372
5286 021314 104117
5287
5288 021316 032737 000040 005342
5289 021324 001001
5290 021326 104071
5291 021330
5292 021330 004737 043750
5293 021334 000401
5294 021336 104115
5295 021340
5296
5297 021340 012737 010340 005404
5298 021346 005037 005406

```
TST14: SCOPE  
MOV #1,$TIMES ;:DO 1 ITERATION  
MOV #STACK,SP  
MOV #0,UNITB ;:SETUP PORT A  
MOVB #'A,MSG19A  
MOV TIMER,COUNT  
JSR PC,TMO ;:DO TIMEOUT  
JSR PC,SUBCLR  
ERROR 24 ;:CERR AFTER SCLR  
JSR PC,DRAV ;:SEE IF DRIVE AVAIL  
ERROR 45 ;:PORT A NOT AVAIL AFTER TMO  
MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT A  
MOV #0,UNITB  
ADD UNITB,RKCS2(R5)  
MOVB #'A,MSG19A  
ADD #RLS,RKCS2(R5) ;:RELEASE PORT A  
MOV #SELDRV,HCS1  
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY  
ERROR 117 ;:NO RDY AFTER SEL DRV CMD  
MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT B  
MOV #1,UNITB  
ADD UNITB,RKCS2(R5)  
MOVB #'B,MSG19A  
MOV #SELDRV,HCS1  
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY  
ERROR 117 ;:NO RKY AFTER SEL DRV CMD  
BIT #D.DRA,HMR2 ;:SEE IF DRIVE AVAIL ON PORT B  
BNE 64$ ;:BR IF YES  
ERROR 71 ;:PORT B NOT AVAIL AFTER PORT A RLS  
64$: JSR PC,TSTATN  
BR 1$  
ERROR 115 ;:ATTN SET IN PORT B AFTER RLS OF PORT A  
1$:  
MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;:EXPECTED MSG A0  
CLR E.B0 ;:EXPECTED MSG B0
```



```

5299 021352 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5300 021360 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5301 021366 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5302 021372 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5303 021400 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5304
5305 021406 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5306 021412 000000 .WORD 0!0!0 ;8 MSGS SPECIFIED HERE
5307 021414 104133 ERROR 133 ;MSG A0 ERROR AFTER RELEASED TO PORT B
5308 021416 104134 ERROR 134 ;MSH B0 ERROR
5309 021420 104135 ERROR 135 ;MSG A1 ERROR
5310 021422 104136 ERROR 136 ;MSG B1 ERROR
5311
5312 021424 012737 000000 005464 MOV #0,UNITB ;ADDRESS PORT A
5313 021432 112737 000101 056644 MOVB #'A,MSG19A
5314 021440 004737 043750 JSR PC,TSTATN
5315 021444 000433 BR TST15 ;:GOTO NEXT TST
5316 021446 104115 ERROR 115 ;ATTN SET AFTER RLS ISSUED
5317
5318 021450 012737 010340 005404 MOV #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
5319 021456 005037 005406 CLR E.B0 ;EXPECTED MSG B0
5320 021462 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5321 021470 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5322 021476 005037 005414 CLR E.A2 ;EXPECTED MSG A2
5323 021502 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5324 021510 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5325
5326 021516 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5327 021522 000000 .WORD 0!0!0 ;8 MSGS SPECIFIED HERE
5328 021524 104211 ERROR 211 ;MSG A0 ERROR AFTER RELEASE ISSUED
5329 021526 104212 ERROR 212 ;MSH B0 ERROR
5330 021530 104213 ERROR 213 ;MSG A1 ERROR
5331 021532 104214 ERROR 214 ;MSG B1 ERROR
5332
5333
5334
5335
5336
5337
5338
5339 021534 000004 TST15: SCOPE
5340 021536 012737 000001 001174 MOV #1,$TIMES ;:DO 1 ITERATION
5341 021544 012706 001100 MOV #STACK,SP
5342 021550 012737 000001 005464 MOV #1,UNITB ;SETUP PORT B
5343 021556 112737 000102 056644 MOVB #'B,MSG19A
5344 021564 013737 005466 001366 MOV TIMER,COUNT
5345 021572 004737 047204 JSR PC,TMO ;DO TIMEOUT
5346
5347 021576 004737 045534 JSR PC,SUBCLR
5348 021602 104024 ERROR 24 ;CERR AFTER SCLR
5349
5350
5351 021604 004737 044172 JSR PC,DRAV ;SEE IF DRIVE AVAIL
5352 021610 104045 ERROR 45 ;PORT B NOT AVAIL AFTER TMO
5353 021612 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5354 021620 012737 000001 005464 MOV #1,UNITB
  
```

```

*****
*TEST 15 TEST RELEASE, DRIVE SEIZED BY PORT 'B'
*
* THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
*****
  
```

5355	021626	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
5356	021634	112737	000102	056644	MOVB	#'B,MSG19A	
5357	021642	062765	000010	000010	ADD	#RLS,RKCS2(R5)	;RELEASE PORT B
5358	021650	012737	000001	005314	MOV	#SELDRV,HCS1	
5359	021656	004737	043372		JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
5360	021662	104117			ERROR	117	;NO RDY AFTER SEL DRV CMD
5361							
5362	021664	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT A
5363	021672	012737	000000	005464	MOV	#0,UNITB	
5364	021700	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
5365	021706	112737	000101	056644	MOVB	#'A,MSG19A	
5366	021714	012737	000001	005314	MOV	#SELDRV,HCS1	
5367	021722	004737	043372		JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
5368	021726	104117			ERROR	117	;NO RKY AFTER SEL DRV CMD
5369							
5370	021730	032737	000040	005342	BIT	#D.DRA,HMR2	;SEE IF DRIVE AVAIL ON PORT A
5371	021736	001001			BNE	64\$;BR IF YES
5372	021740	104071			ERROR	71	;PORT A NOT AVAIL AFTER PORT B RLS
5373	021742						
5374	021742	004737	043750		JSR	PC,TSTATN	
5375	021746	000401			BR	1\$	
5376	021750	104115			ERROR	115	;ATTN SET IN PORT A AFTER RLS OF PORT B
5377	021752						
5378							
5379	021752	012737	010340	005404	MOV	#<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	;EXPECTED MSG A0
5380	021760	005037	005406		CLR	E.B0	;EXPECTED MSG B0
5381	021764	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5382	021772	012737	000001	005412	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5383	022000	005037	005414		CLR	E.A2	;EXPECTED MSG A2
5384	022004	012737	000002	005416	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5385	022012	012737	000003	005422	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5386							
5387	022020	004737	044274		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5388	022024	000000			.WORD	0!0!0	;# MSGS SPECIFIED HERE
5389	022026	104133			ERROR	133	;MSG A0 ERROR AFTER RELEASED TO PORT A
5390	022030	104134			ERROR	134	;MSH B0 ERROR
5391	022032	104135			ERROR	135	;MSG A1 ERROR
5392	022034	104136			ERROR	136	;MSG B1 ERROR
5393							
5394	022036	012737	000001	005464	MOV	#1,UNITB	;ADDRESS PORT B
5395	022044	112737	000102	056644	MOVB	#'B,MSG19A	
5396	022052	004737	043750		JSR	PC,TSTATN	
5397	022056	000433			BR	TST16	::GOTO NEXT TST
5398	022060	104115			ERROR	115	;ATTN SET AFTER RLS ISSUED
5399							
5400	022062	012737	010340	005404	MOV	#<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	;EXPECTED MSG A0
5401	022070	005037	005406		CLR	E.B0	;EXPECTED MSG B0
5402	022074	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5403	022102	012737	000001	005412	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5404	022110	005037	005414		CLR	E.A2	;EXPECTED MSG A2
5405	022114	012737	000002	005416	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5406	022122	012737	000003	005422	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5407							
5408	022130	004737	044274		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5409	022134	000000			.WORD	0!0!0	;# MSGS SPECIFIED HERE
5410	022136	104211			ERROR	211	;MSG A0 ERROR AFTER RELEASE ISSUED

5411 022140 104212
5412 022142 104213
5413 022144 104214

ERROR 212 ;MSH B0 ERROR
ERROR 213 ;MSG A1 ERROR
ERROR 214 ;MSG B1 ERROR

*TEST 16 TEST RELEASE FROM PORT 'A' WITH PORT 'B' REQUESTING
*
* A. PORT 'A' SEIZES THE DRIVE & DOES A SEEK TO SELF COMMAND.
* THE PROGRAM VERIFIES 'DSC' & 'ATTN' ON PORT 'A' ONLY
* ON COMPLETION
*
* B. PORT 'B' TRIES TO ACCESS THE DRIVE. THE PROGRAM VERIFIES
* DRIVE NOT AVAILABLE
*
* C. A RELEASE BY PORT 'A' IS ISSUED. VERIFY PORT 'B' CAN
* ACCESS THE DRIVE IMMEDIATELY & THAT 'DSC' & 'ATTN'
* ARE SEEN ON PORT 'B'
*
* D. VERIFY PORT 'A' 'DSC' & 'ATTN' REMAINS SET AFTER RELEASE
*
* E. THE PROGRAM ISSUES A DRIVE CLEAR COMMAND TO PORT 'B'
* & VERIFIES 'DSC' & 'ATTN' RESETS.
*
* F. THE PROGRAM THEN VERIFIES THAT PORT 'B' DOES NOT SEE
* FURTHER (MULTIPLE) ATTENTIONS FROM WHAT WOULD HAVE BEEN
* NORMAL TIMEOUT FROM PORT 'A'.

5439 022146 000004
5440 022150 012737 000001 001174
5441 022156 012706 001100
5442 022162 013765 001222 000010
5443 022170 012737 000001 005464
5444 022176 063765 005464 000010
5445 022204 112737 000102 056644
5446 022212 062765 000010 000010
5447 022220 012737 000001 005314
5448 022226 004737 043372
5449 022232 104117
5450
5451 022234 013765 001222 000010
5452 022242 012737 000000 005464
5453 022250 063765 005464 000010
5454 022256 112737 000101 056644
5455 022264 012737 000001 005314
5456 022272 004737 043372
5457 022276 104117
5458
5459 022300 032737 000040 005342
5460 022306 001001
5461 022310 104071
5462 022312
5463
5464 022312 012737 010340 005404
5465 022320 005037 005406
5466 022324 012737 001720 005410

TST16: SCOPE
MOV #1,\$TIMES ;:DO 1 ITERATION
MOV #STACK,SP
MOV \$UNIT,RKCS2(R5) ;SETUP FOR PORT B
MOV #1,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'B,MSG19A
ADD #RLS,RKCS2(R5) ;RELEASE PORT B
MOV #SELDRV,HCS1
JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;NO RDY AFTER SEL DRV CMD

MOV \$UNIT,RKCS2(R5) ;SETUP FOR PORT A
MOV #0,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'A,MSG19A
MOV #SELDRV,HCS1
JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;NO RKY AFTER SEL DRV CMD

BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT A
BNE 64\$;BR IF YES
ERROR 71 ;PORT A NOT AVAIL AFTER PORT B RLS

64\$:
MOV #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
CLR E.B0 ;EXPECTED MSG B0
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1

C 9
PAGE 107

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59
 CZR6GC.P11 04-JAN-82 12:39 T16 TEST RELEASE FROM PORT 'A' WITH PORT 'B' REQUESTING SEQ 0106

5467	022332	012737	000001	005412	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5468	022340	005037	005414		CLR	E.A2	;EXPECTED MSG A2
5469	022344	012737	000002	005416	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5470	022352	012737	000003	005422	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5471							
5472	022360	004737	044274		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5473	022364	000000			.WORD	0!0!0	; & MSGS SPECIFIED HERE
5474	022366	104211			ERROR	211	;MSG A0 ERROR AFTER RELEASE ISSUED
5475	022370	104212			ERROR	212	;MSH B0 ERROR
5476	022372	104213			ERROR	213	;MSG A1 ERROR
5477	022374	104214			ERROR	214	;MSG B1 ERROR
5478	022376	004737	045534		JSR	PC,SUBCLR	
5479	022402	104024			ERROR	24	;CERR AFTER SCLR
5480							
5481	022404	012737	000017	005314	MOV	#SEEK,HCS1	
5482	022412	004737	043372		JSR	PC,DOCMD	;DO SEEK CMD & GET CONTR READY
5483	022416	104131			ERROR	131	;NO RDY AFTER SEEK CMD
5484							
5485	022420	013737	001412	005352	MOV	T50000,TEMP1	;SETUP TIMEOUT
5486	022426	004737	044106		JSR	PC,FATT2	;FIND ATTN
5487	022432	104132			ERROR	132	;NO ATTN AFTER SEEK CMD
5488							
5489	022434	032737	100000	005314	BIT	#CERR,HCS1	
5490	022442	001401			BEQ	65\$	
5491	022444	104210			ERROR	210	;CERR AFTER SEEK CMD
5492							
5493	022446					65\$:	
5494							
5495	022446	012737	050340	005404	MOV	#<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	;EXPECTED MSG A0
5496	022454	005037	005406		CLR	E.B0	;EXPECTED MSG B0
5497	022460	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5498	022466	012737	000001	005412	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5499	022474	005037	005414		CLR	E.A2	;EXPECTED MSG A2
5500	022500	012737	000002	005416	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5501	022506	012737	000003	005422	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5502							
5503	022514	004737	044274		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5504	022520	000000			.WORD	0!0!0	; & MSGS SPECIFIED HERE
5505	022522	104161			ERROR	161	;MSG A0 ERROR AFTER SEEK CMD
5506	022524	104162			ERROR	162	;MSH B0 ERROR
5507	022526	104163			ERROR	163	;MSG A1 ERROR
5508	022530	104164			ERROR	164	;MSG B1 ERROR
5509							
5510							
5511	022532	012737	000001	005464	MOV	#1,UNITB	;SETUP PORT B
5512	022540	112737	000102	056644	MOV	#'B,MSG19A	
5513	022546	004737	044172		JSR	PC,DRAV	;SEE IF DRIVE AVAIL
5514	022552	000401			BR	1\$	
5515	022554	104103			ERROR	103	;PORT B AVAIL BEFORE TMO OR RELEASE
5516							
5517	022556	032737	100000	005314	BIT	#CERR,HCS1	
5518	022564	001001			BNE	2\$	
5519	022566	104130			ERROR	130	;CERR NOT SET AFTER SEL DRIVE CMD
5520							; & NO DRA
5521	022570	004737	043750		JSR	PC,TSTATN	
5522	022574	000401			BR	3\$	

5523	022576	104115			ERROR	115		:ATTN SET IN PORT B AFTER RLS FROM PORT A
5524								
5525	022600	012765	100000	000000	3\$:	MOV	#CCLR,RKCS1(R5)	
5526	022606	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT A
5527	022614	012737	000000	005464		MOV	#0,UNITB	
5528	022622	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5529	022630	112737	000101	056644		MOVB	#'A,MSG19A	
5530	022636	062765	000010	000010		ADD	#RLS,RKCS2(R5)	:RELEASE PORT A
5531	022644	012737	000001	005314		MOV	#SELDRV,HCS1	
5532	022652	004737	043372			JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5533	022656	104117				ERROR	117	:NO RDY AFTER SEL DRV CMD
5534								
5535	022660	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT B
5536	022666	012737	000001	005464		MOV	#1,UNITB	
5537	022674	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5538	022702	112737	000102	056644		MOVB	#'B,MSG19A	
5539	022710	012737	000001	005314		MOV	#SELDRV,HCS1	
5540	022716	004737	043372			JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5541	022722	104117				ERROR	117	:NO RKY AFTER SEL DRV CMD
5542								
5543	022724	032737	000040	005342		BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT B
5544	022732	001001				BNE	66\$:BR IF YES
5545	022734	104071				ERROR	71	:PORT B NOT AVAIL AFTER PORT A RLS
5546	022736				66\$:			
5547	022736	004737	043750			JSR	PC,TSTATN	
5548	022742	104122				ERROR	122	:NO ATTN IN PORT B AFTER RLS FROM PORT A
5549								
5550								
5551	022744	012737	050340	005404		MOV	#<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	:EXPECTED MSG A0
5552	022752	005037	005406			CLR	E.B0	:EXPECTED MSG B0
5553	022756	012737	001720	005410		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
5554	022764	012737	000001	005412		MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
5555	022772	005037	005414			CLR	E.A2	:EXPECTED MSG A2
5556	022776	012737	000002	005416		MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
5557	023004	012737	000003	005422		MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
5558								
5559	023012	004737	044274			JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
5560	023016	000000				.WORD	0!0!0	:& MSGS SPECIFIED HERE
5561	023020	104211				ERROR	211	:MSG A0 ERROR AFTER RELEASE ISSUED
5562	023022	104212				ERROR	212	:MSH B0 ERROR
5563	023024	104213				ERROR	213	:MSG A1 ERROR
5564	023026	104214				ERROR	214	:MSG B1 ERROR
5565	023030	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT B
5566	023036	012737	000001	005464		MOV	#1,UNITB	
5567	023044	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5568	023052	112737	000102	056644		MOVB	#'B,MSG19A	
5569	023060	062765	000010	000010		ADD	#RLS,RKCS2(R5)	:RELEASE PORT B
5570	023066	012737	000001	005314		MOV	#SELDRV,HCS1	
5571	023074	004737	043372			JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5572	023100	104117				ERROR	117	:NO RDY AFTER SEL DRV CMD
5573								
5574	023102	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT A
5575	023110	012737	000000	005464		MOV	#0,UNITB	
5576	023116	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5577	023124	112737	000101	056644		MOVB	#'A,MSG19A	
5578	023132	012737	000001	005314		MOV	#SELDRV,HCS1	

E 9
PAGE 109

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59
CZR6GC.P11 04-JAN-82 12:39 T16 TEST RELEASE FROM PORT 'A' WITH PORT 'B' REQUESTING SEQ 0108

5579	023140	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5580	023144	104117			ERROR	117	:NO RKY AFTER SEL DRV CMD
5581							
5582	023146	032737	000040	005342	BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT A
5583	023154	001001			BNE	67\$:BR IF YES
5584	023156	104071			ERROR	71	:PORT A NOT AVAIL AFTER PORT B RLS
5585	023160						
5586	023160	004737	043750		JSR	PC,TSTATN	
5587	023164	104123			ERROR	123	:ATTN CLEARED IN PORT A AFT RLS FROM PORT B
5588							
5589							
5590	023166	012737	050340	005404	MOV	#<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	:EXPECTED MSG A0
5591	023174	005037	005406		CLR	E.B0	:EXPECTED MSG B0
5592	023200	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
5593	023206	012737	000001	005412	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
5594	023214	005037	005414		CLR	E.A2	:EXPECTED MSG A2
5595	023220	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
5596	023226	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
5597							
5598	023234	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
5599	023240	000000			.WORD	0!0!0	:& MSGS SPECIFIED HERE
5600	023242	104211			ERROR	211	:MSG A0 ERROR AFTER RELEASE ISSUED
5601	023244	104212			ERROR	212	:MSH B0 ERROR
5602	023246	104213			ERROR	213	:MSG A1 ERROR
5603	023250	104214			ERROR	214	:MSG B1 ERROR
5604	023252	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT A
5605	023260	012737	000000	005464	MOV	#0,UNITB	
5606	023266	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
5607	023274	112737	000101	056644	MOVB	#'A,MSG19A	
5608	023302	062765	000010	000010	ADD	#RLS,RKCS2(R5)	:RELEASE PORT A
5609	023310	012737	000001	005314	MOV	#SELDRV,HCS1	
5610	023316	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5611	023322	104117			ERROR	117	:NO RDY AFTER SEL DRV CMD
5612							
5613	023324	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT B
5614	023332	012737	000001	005464	MOV	#1,UNITB	
5615	023340	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
5616	023346	112737	000102	056644	MOVB	#'B,MSG19A	
5617	023354	012737	000001	005314	MOV	#SELDRV,HCS1	
5618	023362	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5619	023366	104117			ERROR	117	:NO RKY AFTER SEL DRV CMD
5620							
5621	023370	032737	000040	005342	BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT B
5622	023376	001001			BNE	68\$:BR IF YES
5623	023400	104071			ERROR	71	:PORT B NOT AVAIL AFTER PORT A RLS
5624	023402						
5625							
5626	023402	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	
5627	023410	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:DRIVE#
5628	023416	063765	005464	000010	ADD	UNITB,RKCS2(R5)	:ADD 1 IF ON PORT B
5629	023424	012737	000005	005314	MOV	#CLEAR,HCS1	
5630	023432	004737	043372		JSR	PC,DOCMD	:DO DRIVE CLEAR CMD & GET CONTR RDY
5631	023436	104151			ERROR	151	:NO RDY AFTER DRIVE CLEAR CMD
5632	023440	004737	043750		JSR	PC,TSTATN	:TEST FOR ATTN
5633	023444	000401			BR	69\$	
5634	023446	104154			ERROR	154	:ATTN NOT CLEARED AFTER DRIVE CLEAR CMD

5635 023450
 5636
 5637 023450 012737 010340 005404
 5638 023456 005037 005406
 5639 023462 012737 001720 005410
 5640 023470 012737 000001 005412
 5641 023476 005037 005414
 5642 023502 012737 000002 005416
 5643 023510 012737 000003 005422
 5644
 5645 023516 004737 044274
 5646 023522 000003
 5647 023524 104033
 5648 023526 104034
 5649 023530 104035
 5650 023532 104036
 5651
 5652 023534 013737 005466 001366
 5653 023542 004737 047204
 5654 023546 004737 043750
 5655 023552 000401
 5656 023554 104144
 5657
 5658
 5659
 5660
 5661
 5662
 5663
 5664 023556 000004
 5665 023560 012737 000001 001174
 5666 023566 012706 001100
 5667 023572 013765 001222 000010
 5668 023600 012737 000000 005464
 5669 023606 063765 005464 000010
 5670 023614 112737 000101 056644
 5671 023622 062765 000010 000010
 5672 023630 012737 000001 005314
 5673 023636 004737 043372
 5674 023642 104117
 5675
 5676 023644 013765 001222 000010
 5677 023652 012737 000001 005464
 5678 023660 063765 005464 000010
 5679 023666 112737 000102 056644
 5680 023674 012737 000001 005314
 5681 023702 004737 043372
 5682 023706 104117
 5683
 5684 023710 032737 000040 005342
 5685 023716 001001
 5686 023720 104071
 5687 023722
 5688
 5689 023722 012737 010340 005404
 5690 023730 005037 005406

69\$:
 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
 CLR E.B0 ;EXPECTED MSG B0
 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
 CLR E.A2 ;EXPECTED MSG A2
 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
 ERROR 33 ;MSG A0 ERROR AFTER DRV CLEAR CMD
 ERROR 34 ;MSH B0 ERROR
 ERROR 35 ;MSG A1 ERROR
 ERROR 36 ;MSG B1 ERROR
 MOV TIMER,COUNT
 JSR PC,TMO ;DO 1.5 SEC TIMEOUT ON PORT B
 JSR PC,TSTAT
 BR TST17 ;;GOTO NEXT TST
 ERROR 144 ;MULT ATTN ON PORT B

 *TEST 17 TEST RELEASE FROM PORT 'B' WITH PORT 'A' REQUESTING
 *
 * THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
 *

 TST17: SCOPE
 MOV #1,\$TIMES ;;DO 1 ITERATION
 MOV #STACK,SP
 MOV \$UNIT,RKCS2(R5) ;SETUP FOR PORT A
 MOV #0,UNITB
 ADD UNITB,RKCS2(R5)
 MOVB #'A,MSG19A
 ADD #RLS,RKCS2(R5) ;RELEASE PORT A
 MOV #SELDRV,HCS1
 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 ERROR 117 ;NO RDY AFTER SEL DRV CMD
 MOV \$UNIT,RKCS2(R5) ;SETUP FOR PORT B
 MOV #1,UNITB
 ADD UNITB,RKCS2(R5)
 MOVB #'B,MSG19A
 MOV #SELDRV,HCS1
 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
 ERROR 117 ;NO RKY AFTER SEL DRV CMD
 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT B
 BNE 64\$;BR IF YES
 ERROR 71 ;PORT B NOT AVAIL AFTER PORT A RLS
 64\$:
 MOV #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
 CLR E.B0 ;EXPECTED MSG B0

G 9

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59 PAGE 111
 CZR6GC.P11 04-JAN-82 12:39 T17 TEST RELEASE FROM PORT 'B' WITH PORT 'A' REQUESTING SEQ 0110

5691	023734	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
5692	023742	012737	000001	005412	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
5693	023750	005037	005414		CLR	E.A2	:EXPECTED MSG A2
5694	023754	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
5695	023762	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
5696							
5697	023770	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
5698	023774	000000			.WORD	0!0!0	:& MSGS SPECIFIED HERE
5699	023776	104211			ERROR	211	:MSG A0 ERROR AFTER RELEASE ISSUED
5700	024000	104212			ERROR	212	:MSH B0 ERROR
5701	024002	104213			ERROR	213	:MSG A1 ERROR
5702	024004	104214			ERROR	214	:MSG B1 ERROR
5703	024006	004737	045534		JSR	PC,SUBCLR	
5704	024012	104024			ERROR	24	:CERR AFTER SCLR
5705							
5706	024014	012737	000017	005314	MOV	#SEEK,HCS1	
5707	024022	004737	043372		JSR	PC,DOCMD	:DO SEEK CMD & GET CONTR READY
5708	024026	104131			ERROR	131	:NO RDY AFTER SEEK CMD
5709							
5710	024030	013737	001412	005352	MOV	T50000,TEMP1	:SETUP TIMEOUT
5711	024036	004737	044106		JSR	PC,FATT2	:FIND ATTN
5712	024042	104132			ERROR	132	:NO ATTN AFTER SEEK CMD
5713							
5714	024044	032737	100000	005314	BIT	#CERR,HCS1	
5715	024052	001401			BEQ	65\$	
5716	024054	104210			ERROR	210	:CERR AFTER SEEK CMD
5717							
5718	024056						
5719							
5720	024056	012737	050340	005404	MOV	#<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	:EXPECTED MSG A0
5721	024064	005037	005406		CLR	E.B0	:EXPECTED MSG B0
5722	024070	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
5723	024076	012737	000001	005412	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
5724	024104	005037	005414		CLR	E.A2	:EXPECTED MSG A2
5725	024110	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
5726	024116	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
5727							
5728	024124	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
5729	024130	000000			.WORD	0!0!0	:& MSGS SPECIFIED HERE
5730	024132	104161			ERROR	161	:MSG A0 ERROR AFTER SEEK CMD
5731	024134	104162			ERROR	162	:MSH B0 ERROR
5732	024136	104163			ERROR	163	:MSG A1 ERROR
5733	024140	104164			ERROR	164	:MSG B1 ERROR
5734							
5735							
5736	024142	012737	000000	005464	MOV	#0,UNITB	:SETUP PORT A
5737	024150	112737	000101	056644	MOVB	#'A,MSG19A	
5738	024156	004737	044172		JSR	PC,DRAV	:SEE IF DRIVE AVAIL
5739	024162	000401			BR	1\$	
5740	024164	104103			ERROR	103	:PORT A AVAIL BEFORE TMO OR RELEASE
5741							
5742	024166	032737	100000	005314	1\$: BIT	#CERR,HCS1	
5743	024174	001001			BNE	2\$	
5744	024176	104130			ERROR	130	:CERR NOT SET AFTER SEL DRIVE CMD
5745							:& NO DRA
5746	024200	004737	043750		2\$: JSR	PC,TSTATN	

5747	024204	000401			BR	3\$		
5748	024206	104115			ERROR	115		;ATTN SET IN PORT A AFTER RLS FROM PORT B
5749								
5750	024210	012765	100000	000000	3\$:	MOV	#CCLR,RKCS1(R5)	
5751	024216	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT B
5752	024224	012737	000001	005464		MOV	#1,UNITB	
5753	024232	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5754	024240	112737	000102	056644		MOVB	#'B,MSG19A	
5755	024246	062765	000010	000010		ADD	#RLS,RKCS2(R5)	;RELEASE PORT B
5756	024254	012737	000001	005314		MOV	#SELDRV,HCS1	
5757	024262	004737	043372			JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
5758	024266	104117				ERROR	117	;NO RDY AFTER SEL DRV CMD
5759								
5760	024270	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT A
5761	024276	012737	000000	005464		MOV	#0,UNITB	
5762	024304	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5763	024312	112737	000101	056644		MOVB	#'A,MSG19A	
5764	024320	012737	000001	005314		MOV	#SELDRV,HCS1	
5765	024326	004737	043372			JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
5766	024332	104117				ERROR	117	;NO RKY AFTER SEL DRV CMD
5767								
5768	024334	032737	000040	005342		BIT	#D.DRA,HMR2	;SEE IF DRIVE AVAIL ON PORT A
5769	024342	001001				BNE	66\$;BR IF YES
5770	024344	104071				ERROR	71	;PORT A NOT AVAIL AFTER PORT B RLS
5771	024346				66\$:			
5772	024346	004737	043750			JSR	PC,TSTATN	
5773	024352	104122				ERROR	122	;NO ATTN IN PORT A AFTER RLS FROM PORT B
5774								
5775								
5776	024354	012737	050340	005404		MOV	#<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	;EXPECTED MSG A0
5777	024362	005037	005406			CLR	E.B0	;EXPECTED MSG B0
5778	024366	012737	001720	005410		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5779	024374	012737	000001	005412		MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5780	024402	005037	005414			CLR	E.A2	;EXPECTED MSG A2
5781	024406	012737	000002	005416		MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5782	024414	012737	000003	005422		MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5783								
5784	024422	004737	044274			JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5785	024426	000000				.WORD	0!0!0	;# MSGS SPECIFIED HERE
5786	024430	104211				ERROR	211	;MSG A0 ERROR AFTER RELEASE ISSUED
5787	024432	104212				ERROR	212	;MSH B0 ERROR
5788	024434	104213				ERROR	213	;MSG A1 ERROR
5789	024436	104214				ERROR	214	;MSG B1 ERROR
5790	024440	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT A
5791	024446	012737	000000	005464		MOV	#0,UNITB	
5792	024454	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5793	024462	112737	000101	056644		MOVB	#'A,MSG19A	
5794	024470	062765	000010	000010		ADD	#RLS,RKCS2(R5)	;RELEASE PORT A
5795	024476	012737	000001	005314		MOV	#SELDRV,HCS1	
5796	024504	004737	043372			JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
5797	024510	104117				ERROR	117	;NO RDY AFTER SEL DRV CMD
5798								
5799	024512	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT B
5800	024520	012737	000001	005464		MOV	#1,UNITB	
5801	024526	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
5802	024534	112737	000102	056644		MOVB	#'B,MSG19A	

I 9

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59 PAGE 113
 CZR6GC.P11 04-JAN-82 12:39 T17 TEST RELEASE FROM PORT 'B' WITH PORT 'A' REQUESTING SEQ 0112

5803	024542	012737	000001	005314	MOV	#SELDRV,HCS1	
5804	024550	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5805	024554	104117			ERROR	117	:NO RKY AFTER SEL DRV CMD
5806							
5807	024556	032737	000040	005342	BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT B
5808	024564	001001			BNE	67\$:BR IF YES
5809	024566	104071			ERROR	71	:PORT B NOT AVAIL AFTER PORT A RLS
5810	024570						
5811	024570	004737	043750		JSR	PC,TSTATN	
5812	024574	104123			ERROR	123	:ATTN CLEARED IN PORT B AFT RLS FROM PORT A
5813							
5814							
5815	024576	012737	050340	005404	MOV	#<D.DRA!D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	:EXPECTED MSG A0
5816	024604	005037	005406		CLR	E.B0	:EXPECTED MSG B0
5817	024610	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
5818	024616	012737	000001	005412	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
5819	024624	005037	005414		CLR	E.A2	:EXPECTED MSG A2
5820	024630	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
5821	024636	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
5822							
5823	024644	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
5824	024650	000000			.WORD	0!0!0	:& MSGS SPECIFIED HERE
5825	024652	104211			ERROR	211	:MSG A0 ERROR AFTER RELEASE ISSUED
5826	024654	104212			ERROR	212	:MSH B0 ERROR
5827	024656	104213			ERROR	213	:MSG A1 ERROR
5828	024660	104214			ERROR	214	:MSG B1 ERROR
5829	024662	013765	001222	000010	MCV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT B
5830	024670	012737	000001	005464	MOV	#1,UNITB	
5831	024676	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
5832	024704	112737	000102	056644	MOVB	#'B,MSG19A	
5833	024712	062765	000010	000010	ADD	#RLS,RKCS2(R5)	:RELEASE PORT B
5834	024720	012737	000001	005314	MOV	#SELDRV,HCS1	
5835	024726	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5836	024732	104117			ERROR	117	:NO RDY AFTER SEL DRV CMD
5837							
5838	024734	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT A
5839	024742	012737	000000	005464	MOV	#0,UNITB	
5840	024750	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
5841	024756	112737	000101	056644	MOVB	#'A,MSG19A	
5842	024764	012737	000001	005314	MOV	#SELDRV,HCS1	
5843	024772	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
5844	024776	104117			ERROR	117	:NO RKY AFTER SEL DRV CMD
5845							
5846	025000	032737	000040	005342	BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT A
5847	025006	001001			BNE	68\$:BR IF YES
5848	025010	104071			ERROR	71	:PORT A NOT AVAIL AFTER PORT B RLS
5849	025012						
5850							
5851	025012	012765	100000	000000	MOV	#CLR,RKCS1(R5)	
5852	025020	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:DRIVE#
5853	025026	063765	005464	000010	ADD	UNITB,RKCS2(R5)	:ADD 1 IF ON PORT B
5854	025034	012737	000005	005314	MOV	#CLEAR,HCS1	
5855	025042	004737	043372		JSR	PC,DOCMD	:DO DRIVE CLEAR CMD & GET CONTR RDY
5856	025046	104151			ERROR	151	:NO RDY AFTER DRIVE CLEAR CMD
5857	025050	004737	043750		JSR	PC,TSTATN	:TEST FOR ATTN
5858	025054	000401			BR	69\$	

```

5859 025056 104154          ERROR 154          ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5860 025060          69$:
5861
5862 025060 012737 010340 005404  MOV    #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
5863 025066 005037 005406          CLR    E.B0          ;EXPECTED MSG B0
5864 025072 012737 001720 005410  MOV    #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5865 025100 012737 000001 005412  MOV    #1,E.B1        ;MSG ID FOR EXPECTED MSG B1
5866 025106 005037 005414          CLR    E.A2          ;EXPECTED MSG A2
5867 025112 012737 000002 005416  MOV    #2,E.B2        ;MSG ID FOR EXPECTED MSG B2
5868 025120 012737 000003 005422  MOV    #3,E.B3        ;MSG ID FOR EXPECTED MSG B3
5869
5870 025126 004737 044274          JSR    PC,CHKMSG     ;CHECK MSGS A0, B0, A1, B1
5871 025132 000003          .WORD  T.A2!T.B2!0  ;& MSGS SPECIFIED HERE
5872 025134 104033          ERROR  33          ;MSG A0 ERROR AFTER DRV CLEAR CMD
5873 025136 104034          ERROR  34          ;MSH B0 ERROR
5874 025140 104035          ERROR  35          ;MSG A1 ERROR
5875 025142 104036          ERROR  36          ;MSG B1 ERROR
5876
5877 025144 013737 005466 001366  MOV    TIMER,COUNT
5878 025152 004737 047204          JSR    PC,TMO       ;DO 1.5 SEC TIMEOUT ON PORT A
5879 025156 004737 043750          JSR    PC,TSTATN
5880 025162 000401          BR     TST20
5881 025164 104144          ERROR  144        ;;GOTO NEXT TST
                    ;MULT ATTN ON PORT A
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893

```

```

*****
*TEST 20          TEST RELEASE FROM REQUESTING PORT 'B' INHIBITS 'ATTN'
*
*
*   A. PORT 'A' SEIZES THE DRIVE
*   B. PORT 'B' ATTEMPTS TO SEIZE THE DRIVE
*   C. PORT 'B' & PORT 'A' RELEASE THE DRIVE, IN THAT ORDER
*   D. THE PROGRAM VERIFIES THAT NEITHER PORT 'A' OR 'B' ATTENTION
*       BITS SET
*****

```

```

5894 025166 000004          TST20: SCOPE
5895 025170 012737 000001 001174  MOV    #1,$TIMES     ;;DO 1 ITERATION
5896 025176 012706 001100          MOV    #STACK,SP
5897 025202 013765 001222 000010  MOV    $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5898 025210 012737 000001 005464  MOV    #1,UNITB
5899 025216 063765 005464 000010  ADD    UNITB,RKCS2(R5)
5900 025224 112737 000102 056644  MOVB   #'B,MSG19A
5901 025232 062765 000010 000010  ADD    #RLS,RKCS2(R5) ;RELEASE PORT B
5902 025240 012737 000001 005314  MOV    #SELDRV,HCS1
5903 025246 004737 043372          JSR    PC,DOCMD     ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5904 025252 104117          ERROR  117        ;NO RDY AFTER SEL DRV CMD
5905
5906 025254 013765 001222 000010  MOV    $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5907 025262 012737 000000 005464  MOV    #0,UNITB
5908 025270 063765 005464 000010  ADD    UNITB,RKCS2(R5)
5909 025276 112737 000101 056644  MOVB   #'A,MSG19A
5910 025304 012737 000001 005314  MOV    #SELDRV,HCS1
5911 025312 004737 043372          JSR    PC,DOCMD     ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5912 025316 104117          ERROR  117        ;NO RKY AFTER SEL DRV CMD
5913
5914 025320 032737 000040 005342  BIT    #D.DRA,HMR2  ;SEE IF DRIVE AVAIL ON PORT A

```

5915	025326	001001				BNE	64\$:BR IF YES
5916	025330	104071				ERROR	71		:PORT A NOT AVAIL AFTER PORT B RLS
5917	025332				64\$:				
5918									
5919	025332	012737	010340	005404		MOV	#<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0		:EXPECTED MSG A0
5920	025340	005037	005406			CLR	E.B0		:EXPECTED MSG B0
5921	025344	012737	001720	005410		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		:EXPECTED A1
5922	025352	012737	000001	005412		MOV	#1,E.B1		:MSG ID FOR EXPECTED MSG B1
5923	025360	005037	005414			CLR	E.A2		:EXPECTED MSG A2
5924	025364	012737	000002	005416		MOV	#2,E.B2		:MSG ID FOR EXPECTED MSG B2
5925	025372	012737	000003	005422		MOV	#3,E.B3		:MSG ID FOR EXPECTED MSG B3
5926									
5927	025400	004737	044274			JSR	PC,CHKMSG		:CHECK MSGS A0, B0, A1, B1
5928	025404	000000				.WORD	0!0!0		:& MSGS SPECIFIED HERE
5929	025406	104211				ERROR	211		:MSG A0 ERROR AFTER RELEASE ISSUED
5930	025410	104212				ERROR	212		:MSH B0 ERROR
5931	025412	104213				ERROR	213		:MSG A1 ERROR
5932	025414	104214				ERROR	214		:MSG B1 ERROR
5933	025416	004737	045534			JSR	PC,SUBCLR		
5934	025422	104024				ERROR	24		:CERR AFTER SCLR
5935									
5936	025424	012737	000001	005464		MOV	#1,UNITB		:SETUP PORT B
5937	025432	112737	000102	056644		MOVB	#'B,MSG19A		
5938	025440	004737	044172			JSR	PC,DRAV		:SEE IF DRIVE AVAIL
5939	025444	000401				BR	1\$		
5940	025446	104103				ERROR	103		:PORT B AVAIL BEFORE TMO OR RELEASE
5941									
5942	025450	032737	100000	005314	1\$:	BIT	#CERR,HCS1		
5943	025456	001001				BNE	2\$		
5944	025460	104130				ERROR	130		:CERR NOT SET BY NO DRA
5945									
5946	025462	004737	043750		2\$:	JSR	PC,TSTATN		
5947	025466	000401				BR	3\$		
5948	025470	104115				ERROR	115		:ATTN SET IN PORT B AFTER RLS FROM PORT A
5949									
5950	025472	012765	100000	000000	3\$:	MOV	#CCLR,RKCS1(R5)		
5951	025500	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)		:SETUP FOR PORT B
5952	025506	012737	000001	005464		MOV	#1,UNITB		
5953	025514	063765	005464	000010		ADD	UNITB,RKCS2(R5)		
5954	025522	112737	000102	056644		MOVB	#'B,MSG19A		
5955	025530	062765	000010	000010		ADD	#RLS,RKCS2(R5)		:RELEASE PORT B
5956	025536	012737	000001	005314		MOV	#SELDRV,HCS1		
5957	025544	004737	043372			JSR	PC,DOCMD		:DO SELDRV (STATUS) CMD & GET CONTR RDY
5958	025550	104117				ERROR	117		:NO RDY AFTER SEL DRV CMD
5959									
5960	025552	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)		:SETUP FOR PORT A
5961	025560	012737	000000	005464		MOV	#0,UNITB		
5962	025566	063765	005464	000010		ADD	UNITB,RKCS2(R5)		
5963	025574	112737	000101	056644		MOVB	#'A,MSG19A		
5964	025602	012737	000001	005314		MOV	#SELDRV,HCS1		
5965	025610	004737	043372			JSR	PC,DOCMD		:DO SELDRV (STATUS) CMD & GET CONTR RDY
5966	025614	104117				ERROR	117		:NO RKY AFTER SEL DRV CMD
5967									
5968	025616	032737	000040	005342		BIT	#D.DRA,HMR2		:SEE IF DRIVE AVAIL ON PORT A
5969	025624	001001				BNE	65\$:BR IF YES
5970	025626	104071				ERROR	71		:PORT A NOT AVAIL AFTER PORT B RLS

```

5971 025630
5972 025630 013765 001222 000010 65$: MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
5973 025636 012737 000000 005464 MOV #0,UNITB
5974 025644 063765 005464 000010 ADD UNITB,RKCS2(R5)
5975 025652 112737 000101 056644 MOVB #'A,MSG19A
5976 025660 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT A
5977 025666 012737 000001 005314 MOV #SELDRV,HCS1
5978 025674 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5979 025700 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
5980
5981 025702 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
5982 025710 012737 000001 005464 MOV #1,UNITB
5983 025716 063765 005464 000010 ADD UNITB,RKCS2(R5)
5984 025724 112737 000102 056644 MOVB #'B,MSG19A
5985 025732 012737 000001 005314 MOV #SELDRV,HCS1
5986 025740 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
5987 025744 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD
5988
5989 025746 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT B
5990 025754 001001 BNE 66$ ;BR IF YES
5991 025756 104071 ERROR 71 ;PORT B NOT AVAIL AFTER PORT A RLS
5992 025760
5993 025760 013737 005466 001366 66$: MOV TIMER,COUNT
5994 025766 004737 047204 JSR PC,TMO ;DO 1.5 SEC TIMEOUT ON PORT B
5995 025772 004737 043750 JSR PC,TSTATN
5996 025776 000401 BR 4$
5997 026000 104115 ERROR 115 ;ATTN SET ON PORT B AFTER RLS FROM PORT B
5998
5999 026002 4$:
6000
6001 026002 012737 010340 005404 MOV #<D.DRA!0!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6002 026010 005037 005406 CLR E.B0 ;EXPECTED MSG B0
6003 026014 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6004 026022 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
6005 026030 005037 005414 CLR E.A2 ;EXPECTED MSG A2
6006 026034 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
6007 026042 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
6008
6009 026050 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
6010 026054 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
6011 026056 104211 ERROR 211 ;MSG A0 ERROR AFTER RELEASE ISSUED
6012 026060 104212 ERROR 212 ;MSH B0 ERROR
6013 026062 104213 ERROR 213 ;MSG A1 ERROR
6014 026064 104214 ERROR 214 ;MSG B1 ERROR
6015 026066 012737 000000 005464 MOV #0,UNITB ;SETUP PORT A
6016 026074 112737 000101 056644 MOVB #'A,MSG19A
6017 026102 004737 043750 JSR PC,TSTATN
6018 026106 000401 BR TST21 ;;GOTO NEXT TST
6019 026110 104115 ERROR 115 ;ATTN SET ON PORT A, AFTER RLS FROM PORT A
6020
6021
6022 *****
6023 *TEST 21 TEST RELEASE FROM REQUESTING PORT 'A' INHIBITS 'ATTN'
6024 *
6025 * THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
6026 *****
    
```

```

6027 026112 000004          TST21: SCOPE
6028 026114 012737 000001 001174  MOV      #1,STIMES      ;;DO 1 ITERATION
6029 026122 012706 001100          MOV      #STACK,SP
6030 026126 013765 001222 000010  MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6031 026134 012737 000000 005464  MOV      #0,UNITB
6032 026142 063765 005464 000010  ADD      UNITB,RKCS2(R5)
6033 026150 112737 000101 056644  MOV      #A,MSG19A
6034 026156 062765 000010 000010  ADD      #RLS,RKCS2(R5) ;RELEASE PORT A
6035 026164 012737 000001 005314  MOV      #SELDRV,HCS1
6036 026172 004737 043372          JSR      PC,DOCMD      ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6037 026176 104117          ERROR    117          ;NO RDY AFTER SEL DRV CMD
6038
6039 026200 013765 001222 000010  MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6040 026206 012737 000001 005464  MOV      #1,UNITB
6041 026214 063765 005464 000010  ADD      UNITB,RKCS2(R5)
6042 026222 112737 000102 056644  MOV      #B,MSG19A
6043 026230 012737 000001 005314  MOV      #SELDRV,HCS1
6044 026236 004737 043372          JSR      PC,DOCMD      ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6045 026242 104117          ERROR    117          ;NO RKY AFTER SEL DRV CMD
6046
6047 026244 032737 000040 005342  BIT      #D.DRA,HMR2      ;SEE IF DRIVE AVAIL ON PORT B
6048 026252 001001          BNE     64$             ;BR IF YES
6049 026254 104071          ERROR    71             ;PORT B NOT AVAIL AFTER PORT A RLS
6050 026256          64$:
6051
6052 026256 012737 010340 005404  MOV      #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6053 026264 005037 005406          CLR      E.B0          ;EXPECTED MSG B0
6054 026270 012737 001720 005410  MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6055 026276 012737 000001 005412  MOV      #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
6056 026304 005037 005414          CLR      E.A2          ;EXPECTED MSG A2
6057 026310 012737 000002 005416  MOV      #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
6058 026316 012737 000003 005422  MOV      #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
6059
6060 026324 004737 044274          JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
6061 026330 000000          .WORD   0!0!0          ;& MSGS SPECIFIED HERE
6062 026332 104211          ERROR    211          ;MSG A0 ERROR AFTER RELEASE ISSUED
6063 026334 104212          ERROR    212          ;MSH B0 ERROR
6064 026336 104213          ERROR    213          ;MSG A1 ERROR
6065 026340 104214          ERROR    214          ;MSG B1 ERROR
6066 026342 004737 045534          JSR      PC,SUBCLR      ;CERR AFTER SCLR
6067 026346 104024          ERROR    24
6068
6069 026350 012737 000000 005464  MOV      #0,UNITB      ;SETUP PORT A
6070 026356 112737 000101 056644  MOV      #A,MSG19A
6071 026364 004737 044172          JSR      PC,DRAV        ;SEE IF DRIVE AVAIL
6072 026370 000401          BR      1$             ;PORT A AVAIL BEFORE TMO OR RELEASE
6073 026372 104103          ERROR    103
6074
6075 026374 032737 100000 005314 1$: BIT      #CERR,HCS1
6076 026402 001001          BNE     2$
6077 026404 104130          ERROR    130          ;CERR NOT SET BY NO DRA
6078
6079 026406 004737 043750          2$: JSR      PC,TSTATN
6080 026412 000401          BR      3$
6081 026414 104115          ERROR    115          ;ATTN SET IN PORT A AFTER RLS FROM PORT B
6082
    
```

```

6083 026416 012765 100000 000000 3$: MOV #CCLR,RKCS1(R5)
6084 026424 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6085 026432 012737 000000 005464 MOV #0,UNITB
6086 026440 063765 005464 000010 ADD UNITB,RKCS2(R5)
6087 026446 112737 000101 056644 MOVB #'A,MSG19A
6088 026454 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT A
6089 026462 012737 000001 005314 MOV #SELDRV,HCS1
6090 026470 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6091 026474 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6092
6093 026476 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6094 026504 012737 000001 005464 MOV #1,UNITB
6095 026512 063765 005464 000010 ADD UNITB,RKCS2(R5)
6096 026520 112737 000102 056644 MOVB #'B,MSG19A
6097 026526 012737 000001 005314 MOV #SELDRV,HCS1
6098 026534 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6099 026540 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD
6100
6101 026542 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT B
6102 026550 001001 BNE 65$ ;BR IF YES
6103 026552 104071 ERROR 71 ;PORT B NOT AVAIL AFTER PORT A RLS
6104 026554
6105 026554 013765 001222 000010 65$: MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6106 026562 012737 000001 005464 MOV #1,UNITB
6107 026570 063765 005464 000010 ADD UNITB,RKCS2(R5)
6108 026576 112737 000102 056644 MOVB #'B,MSG19A
6109 026604 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT B
6110 026612 012737 000001 005314 MOV #SELDRV,HCS1
6111 026620 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6112 026624 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6113
6114 026626 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6115 026634 012737 000000 005464 MOV #0,UNITB
6116 026642 063765 005464 000010 ADD UNITB,RKCS2(R5)
6117 026650 112737 000101 056644 MOVB #'A,MSG19A
6118 026656 012737 000001 005314 MOV #SELDRV,HCS1
6119 026664 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6120 026670 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD
6121
6122 026672 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT A
6123 026700 001001 BNE 66$ ;BR IF YES
6124 026702 104071 ERROR 71 ;PORT A NOT AVAIL AFTER PORT B RLS
6125 026704
6126 026704 013737 005466 001366 66$: MOV TIMER,COUNT
6127 026712 004737 047204 JSR PC,TMO ;DO 1.5 SEC TIMEOUT ON PORT A
6128 026716 004737 043750 JSR PC,TSTATN
6129 026722 000401 BR 4$
6130 026724 104115 ERROR 115 ;ATTN SET ON PORT A AFTER RLS FROM PORT A
6131
6132 026726 4$:
6133
6134 026726 012737 010340 005404 MOV #<D.DRA!O!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6135 026734 005037 005406 CLR E.B0 ;EXPECTED MSG B0
6136 026740 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6137 026746 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
6138 026754 005037 005414 CLR E.A2 ;EXPECTED MSG A2
    
```

```

6139 026760 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
6140 026766 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
6141
6142 026774 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
6143 027000 000000 .WORD 0!0!0 ;# MSGS SPECIFIED HERE
6144 027002 104211 ERROR 211 ;MSG A0 ERROR AFTER RELEASE ISSUED
6145 027004 104212 ERROR 212 ;MSH B0 ERROR
6146 027006 104213 ERROR 213 ;MSG A1 ERROR
6147 027010 104214 ERROR 214 ;MSG B1 ERROR
6148 027012 012737 000001 005464 MOV #1,UNITB ;SETUP PORT B
6149 027020 112737 000102 056644 MOVB #'B,MSG19A
6150 027026 004737 043750 JSR PC,TSTATN
6151 027032 000401 BR TST22 ;:GOTO NEXT TST
6152 027034 104115 ERROR 115 ;ATTN SET ON PORT B, AFTER RLS FROM PORT B
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165

```

```

:*****
:*TEST 22 TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A'
:*
:* VERIFY THAT A RELEASE ISSUED BY ONE PORT IS NOT RECOGNIZED IF
:* THE DRIVE IS SEIZED BY THE OTHER PORT
:*
:* A. SEIZE THE DRIVE THRU PORT 'A'.
:*
:* B. ISSUE A RELEASE THRU PORT 'B' & VERIFY DRIVE STILL SEIZED
:* BY PORT 'A'.
:*****

```

```

6166 027036 000004 TST22: SCOPE
6167 027040 012737 000001 001174 MOV #1,$TIMES ;:DO 1 ITERATION
6168 027046 012706 001100 MOV #STACK,SP
6169 027052 012737 000000 005464 MOV #0,UNITB ;SETUP PORT A
6170 027060 112737 000101 056644 MOVB #'A,MSG19A
6171 027066 013737 005466 001366 MOV TIMER,COUNT
6172 027074 004737 047204 JSR PC,TMO ;DO TIMEOUT
6173
6174 027100 004737 045534 JSR PC,SUBCLR
6175 027104 104024 ERROR 24 ;CERR AFTER SCLR
6176
6177
6178 027106 004737 044172 JSR PC,DRAV ;SEE IF DRIVE AVAIL
6179 027112 104045 ERROR 45 ;PORT A NOT AVAIL AFTER TMO
6180 027114 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6181 027122 012737 000001 005464 MOV #1,UNITB
6182 027130 063765 005464 000010 ADD UNITB,RKCS2(R5)
6183 027136 112737 000102 056644 MOVB #'B,MSG19A
6184 027144 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT B
6185 027152 012737 000001 005314 MOV #SELDRV,HCS1
6186 027160 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6187 027164 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6188
6189 027166 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6190 027174 012737 000000 005464 MOV #0,UNITB
6191 027202 063765 005464 000010 ADD UNITB,RKCS2(R5)
6192 027210 112737 000101 056644 MOVB #'A,MSG19A
6193 027216 012737 000001 005314 MOV #SELDRV,HCS1
6194 027224 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY

```



```

CZR6GCO RK611 DU PORT LGC          MACY11 30(1046) 04-JAN-82 12:59 C 10
CZR6GC.P11 04-JAN-82 12:39        T22      TEST RELEASE BY PORT 'B' WHEN SEIZED BY PORT 'A' PAGE 120
                                                    SEQ 0119

6195 027230 104117          ERROR 117          ;NO RKY AFTER SEL DRV CMD
6196
6197 027232 032737 000040 005342  BIT #D.DRA,HMR2    ;SEE IF DRIVE AVAIL ON PORT A
6198 027240 001001          BNE 64$           ;BR IF YES
6199 027242 104071          ERROR 71          ;PORT A NOT AVAIL AFTER PORT B RLS
6200 027244          64$:
6201
6202 027244 012737 010340 005404  MOV #<D.DRA!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6203 027252 005037 005406          CLR E.B0         ;EXPECTED MSG B0
6204 027256 012737 001720 005410  MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6205 027264 012737 000001 005412  MOV #1,E.B1      ;MSG ID FOR EXPECTED MSG B1
6206 027272 005037 005414          CLR E.A2         ;EXPECTED MSG A2
6207 027276 012737 000002 005416  MOV #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
6208 027304 012737 000003 005422  MOV #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
6209
6210 027312 004737 044274          JSR PC,CHKMSG    ;CHECK MSGS A0, BC, A1, B1
6211 027316 000000          .WORD 0!0!0     ;8 MSGS SPECIFIED HERE
6212 027320 104211          ERROR 211       ;MSG A0 ERROR AFTER RELEASE ISSUED
6213 027322 104212          ERROR 212       ;MSH B0 ERROR
6214 027324 104213          ERROR 213       ;MSG A1 ERROR
6215 027326 104214          ERROR 214       ;MSG B1 ERROR
6216
6217
6218
6219
6220
6221
6222
6223 027330 000004          *****
6224 027332 012737 000001 001174  *TEST 23      TEST RELEASE BY PORT 'A' WHEN SEIZED BY PORT 'B'
6225 027340 012706 001100          *
6226 027344 012737 000001 005464          *      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
6227 027352 112737 000102 056644          *
6228 027360 013737 005466 001366  *****
6229 027366 004737 047204          TST23: SCOPE
6230
6231 027372 004737 045534          MOV #1,$TIMES   ;;DO 1 ITERATION
6232 027376 104024          MOV #STACK,SP  ;SETUP PORT B
6233
6234
6235 027400 004737 044172          MOV #1,UNITB   ;SETUP PORT B
6236 027404 104045          MOV #B,MSG19A ;SETUP PORT B
6237 027406 013765 001222 000010  MOV #B,MSG19A  ;SETUP PORT B
6238 027414 012737 000000 005464          MOV #0,UNITB   ;SETUP PORT B
6239 027422 063765 005464 000010  ADD UNITB,RKCS2(R5) ;SETUP PORT B
6240 027430 112737 000101 056644          MOV #A,MSG19A  ;SETUP PORT B
6241 027436 062765 000010 000010  ADD #RLS,RKCS2(R5) ;RELEASE PORT A
6242 027444 012737 000001 005314  MOV #SELDRV,HCS1 ;RELEASE PORT A
6243 027452 004737 043372          JSR PC,DOCMD    ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6244 027456 104117          ERROR 117       ;NO RDY AFTER SEL DRV CMD
6245
6246 027460 013765 001222 000010  MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6247 027466 012737 000001 005464          MOV #1,UNITB   ;SETUP FOR PORT B
6248 027474 063765 005464 000010  ADD UNITB,RKCS2(R5) ;SETUP FOR PORT B
6249 027502 112737 000102 056644          MOV #B,MSG19A  ;SETUP FOR PORT B
6250 027510 012737 000001 005314          MOV #SELDRV,HCS1 ;SETUP FOR PORT B

```


6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286 027622 000004
6287 027624 012737 000001 001174
6288 027632 012706 001100
6289 027636 013765 001222 000010
6290 027644 012737 000001 005464
6291 027652 063765 005464 000010
6292 027660 112737 000102 056644
6293 027666 062765 000010 000010
6294 027674 012737 000001 005314
6295 027702 004737 043372
6296 027706 104117
6297
6298 027710 013765 001222 000010
6299 027716 012737 000000 005464
6300 027724 063765 005464 000010
6301 027732 112737 000101 056644
6302 027740 012737 000001 005314
6303 027746 004737 043372
6304 027752 104117
6305
6306 027754 032737 000040 005342
6307 027762 001001
6308 027764 104071
6309 027766
6310 027766 004737 045534
6311 027772 104024
6312
6313 027774 012765 000012 000020
6314 030002 012737 000017 005314
6315 030010 004737 043372
6316 030014 104131
6317 030016 013765 001222 000010
6318 030024 012737 000000 005464
6319 030032 063765 005464 000010
6320 030040 112737 000101 056644
6321 030046 062765 000010 000010
6322 030054 012737 000001 005314
6323 030062 004737 043372
6324 030066 104117
6325
6326 030070 013765 001222 000010
6327 030076 012737 000001 005464
6328 030104 063765 005464 000010
6329 030112 112737 000102 056644

```
*****
:TEST 24 TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE: PORT A
:
: A. ISSUE A SEEK COMMAND TO CYL 10 FROM PORT 'A' & AN
: IMMEDIATE RELEASE TO PORT 'A'
:
: B. VERIFY THE DRIVE IS AVAILABLE TO PORT 'B' & PORT B SEES ATTN
:
: C. VERIFY PORT A DOES NOT RAISE ATTN WHEN SEEK COMPLETED.
:
:*****
TST24: SCOPE
MOV #1,$TIMES ;:DO 1 ITERATION
MOV #STACK,SP
MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT B
MOV #1,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'B,MSG19A
ADD #RLS,RKCS2(R5) ;:RELEASE PORT B
MOV #SELDRV,HCS1
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;:NO RDY AFTER SEL DRV CMD

MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT A
MOV #0,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'A,MSG19A
MOV #SELDRV,HCS1
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;:NO RKY AFTER SEL DRV CMD

BIT #D.DRA,HMR2 ;:SEE IF DRIVE AVAIL ON PORT A
BNE 64$ ;:BR IF YES
ERROR 71 ;:PORT A NOT AVAIL AFTER PORT B RLS

64$: JSR PC,SUBCLR ;:CERR AFTER SCLR
ERROR 24

MOV #10.,RKDC(R5)
MOV #SEEK,HCS1
JSR PC,DOCMD ;:DO SEEK CMD & GET CONTR READY
ERROR 131 ;:NO RDY AFTER SEEK CMD

MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT A
MOV #0,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'A,MSG19A
ADD #RLS,RKCS2(R5) ;:RELEASE PORT A
MOV #SELDRV,HCS1
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;:NO RDY AFTER SEL DRV CMD

MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT B
MOV #1,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'B,MSG19A
```

6330	030120	012737	000001	005314	MOV	#SELDRV,HCS1	
6331	030126	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
6332	030132	104117			ERROR	117	:NO RKY AFTER SEL DRV CMD
6333							
6334	030134	032737	000040	005342	BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT B
6335	030142	001001			BNE	65\$:BR IF YES
6336	030144	104071			ERROR	71	:PORT B NOT AVAIL AFTER PORT A RLS
6337	030146						
6338	030146	013737	001412	005352	MOV	T50000,TEMP1	
6339	030154	004737	044106		JSR	PC,FATT2	
6340	030160	104152			ERROR	152	:NO ATTN ON PORT B AFTER SEEK & RLS FROM PORT A
6341							
6342							
6343	030162	012737	050340	005404	MOV	#<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0	:EXPECTED MSG A0
6344	030170	005037	005406		CLR	E.B0	:EXPECTED MSG B0
6345	030174	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
6346	030202	012737	000001	005412	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
6347	030210	005037	005414		CLR	E.A2	:EXPECTED MSG A2
6348	030214	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
6349	030222	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
6350							
6351	030230	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
6352	030234	000000			.WORD	0!0!0	:& MSGS SPECIFIED HERE
6353	030236	104145			ERROR	145	:MSG A0 ERROR AFTER SEEK & RLS FROM PORT A
6354	030240	104146			ERROR	146	:MSH B0 ERROR
6355	030242	104147			ERROR	147	:MSG A1 ERROR
6356	030244	104150			ERROR	150	:MSG B1 ERROR
6357	030246	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT B
6358	030254	012737	000001	005464	MOV	#1,UNITB	
6359	030262	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
6360	030270	112737	000102	056644	MOVB	#'B,MSG19A	
6361	030276	062765	000010	000010	ADD	#RLS,RKCS2(R5)	:RELEASE PORT B
6362	030304	012737	000001	005314	MOV	#SELDRV,HCS1	
6363	030312	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
6364	030316	104117			ERROR	117	:NO RDY AFTER SEL DRV CMD
6365							
6366	030320	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT A
6367	030326	012737	000000	005464	MOV	#0,UNITB	
6368	030334	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
6369	030342	112737	000101	056644	MOVB	#'A,MSG19A	
6370	030350	012737	000001	005314	MOV	#SELDRV,HCS1	
6371	030356	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
6372	030362	104117			ERROR	117	:NO RKY AFTER SEL DRV CMD
6373							
6374	030364	032737	000040	005342	BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT A
6375	030372	001001			BNE	66\$:BR IF YES
6376	030374	104071			ERROR	71	:PORT A NOT AVAIL AFTER PORT B RLS
6377	030376						
6378	030376	004737	043750		JSR	PC,TSTATN	
6379	030402	000401			BR	1\$	
6380	030404	104160			ERROR	160	:ATTN ON PORT A AFTER SEEK & RLS FROM PORT A
6381							
6382	030406						
6383							
6384	030406	012737	010340	005404	MOV	#<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	:EXPECTED MSG A0
6385	030414	005037	005406		CLR	E.B0	:EXPECTED MSG B0

CZR6GC0 RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046)
T24

04-JAN-82 12:59 PAGE 124
TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE: PORT A

SEQ 0123

6386	030420	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
6387	030426	012737	000001	005412	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
6388	030434	005037	005414		CLR	E.A2	:EXPECTED MSG A2
6389	030440	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
6390	030446	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
6391							
6392	030454	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
6393	030460	000000			.WORD	0!0!0	:# MSGS SPECIFIED HERE
6394	030462	104145			ERROR	145	:MSG A0 ERROR AFTER SEEK & RLS FROM PORT A
6395	030464	104146			ERROR	146	:MSH B0 ERROR
6396	030466	104147			ERROR	147	:MSG A1 ERROR
6397	030470	104150			ERROR	150	:MSG B1 ERROR
6398							
6399							

 :*TEST 25 TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE: PORT B
 :*

THE PREVIOUS TEST IS REPEATED FOR PORT 'B',
 BUT THE SEEK IS TO CYLINDER 0

6400							
6401							
6402							
6403							
6404							
6405							
6406	030472	000004			TST25: SCOPE		
6407	030474	012737	000001	001174	MOV	#1,\$TIMES	::DO 1 ITERATION
6408	030502	012706	001100		MOV	#STACK,SP	
6409	030506	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT A
6410	030514	012737	000000	005464	MOV	#0,UNITB	
6411	030522	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
6412	030530	112737	000101	056644	MOVB	#'A,MSG19A	
6413	030536	062765	000010	000010	ADD	#RLS,RKCS2(R5)	:RELEASE PORT A
6414	030544	012737	000001	005314	MOV	#SELDRV,HCS1	
6415	030552	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
6416	030556	104117			ERROR	117	:NO RDY AFTER SEL DRV CMD
6417							
6418	030560	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT B
6419	030566	012737	000001	005464	MOV	#1,UNITB	
6420	030574	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
6421	030602	112737	000102	056644	MOVB	#'B,MSG19A	
6422	030610	012737	000001	005314	MOV	#SELDRV,HCS1	
6423	030616	004737	043372		JSR	PC,DOCMD	:DO SELDRV (STATUS) CMD & GET CONTR RDY
6424	030622	104117			ERROR	117	:NO RKY AFTER SEL DRV CMD
6425							
6426	030624	032737	000040	005342	BIT	#D.DRA,HMR2	:SEE IF DRIVE AVAIL ON PORT B
6427	030632	001001			BNE	64\$:BR IF YES
6428	030634	104071			ERROR	71	:PORT B NOT AVAIL AFTER PORT A RLS
6429	030636						
6430	030636	004737	045534		JSR	PC,SUBCLR	
6431	030642	104024			ERROR	24	:CERR AFTER SCLR
6432							
6433	030644	012765	000000	000020	MOV	#0,RKDC(R5)	
6434	030652	012737	000017	005314	MOV	#SEEK,HCS1	
6435	030660	004737	043372		JSR	PC,DOCMD	:DO SEEK CMD & GET CONTR READY
6436	030664	104131			ERROR	131	:NO RDY AFTER SEEK CMD
6437	030666	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:SETUP FOR PORT B
6438	030674	012737	000001	005464	MOV	#1,UNITB	
6439	030702	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
6440	030710	112737	000102	056644	MOVB	#'B,MSG19A	
6441	030716	062765	000010	000010	ADD	#RLS,RKCS2(R5)	:RELEASE PORT B

64\$:

H 10
PAGE 125

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59
CZR6GC.P11 04-JAN-82 12:39 T25 TEST COMMAND FOLLOWED BY IMMEDIATE RELEASE: PORT B SEQ 0124

```

6442 030724 012737 000001 005314  MOV      #SELDRV,HCS1
6443 030732 004737 043372  JSR      PC,DOCMD      ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6444 030736 104117  ERROR   117           ;NO RDY AFTER SEL DRV CMD
6445
6446 030740 013765 001222 000010  MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6447 030746 012737 000000 005464  MOV      #0,UNITB
6448 030754 063765 005464 000010  ADD      UNITB,RKCS2(R5)
6449 030762 112737 000101 056644  MOVB    #'A,MSG19A
6450 030770 012737 000001 005314  MOV      #SELDRV,HCS1
6451 030776 004737 043372  JSR      PC,DOCMD      ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6452 031002 104117  ERROR   117           ;NO RKY AFTER SEL DRV CMD
6453
6454 031004 032737 000040 005342  BIT      #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT A
6455 031012 001001  BNE     65$           ;BR IF YES
6456 031014 104071  ERROR   71           ;PORT A NOT AVAIL AFTER PORT B RLS
6457 031016
6458 031016 013737 001412 005352  MOV      T50000,TEMP1   65$:
6459 031024 004737 044106  JSR      PC,FATT2
6460 031030 104152  ERROR   152           ;NO ATTN ON PORT A AFTER SEEK & RLS FROM PORT B
6461
6462
6463 031032 012737 050340 005404  MOV      #<D.DRA!D.DSC!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6464 031040 005037 005406  CLR     E.B0           ;EXPECTED MSG B0
6465 031044 012737 001720 005410  MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6466 031052 012737 000001 005412  MOV      #1,E.B1        ;MSG ID FOR EXPECTED MSG B1
6467 031060 005037 005414  CLR     E.A2           ;EXPECTED MSG A2
6468 031064 012737 000002 005416  MOV      #2,E.B2        ;MSG ID FOR EXPECTED MSG B2
6469 031072 012737 000003 005422  MOV      #3,E.B3        ;MSG ID FOR EXPECTED MSG B3
6470
6471 031100 004737 044274  JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
6472 031104 000000  .WORD   0!0!0         ;& MSGS SPECIFIED HERE
6473 031106 104145  ERROR   145           ;MSG A0 ERROR AFTER SEEK & RLS FROM PORT B
6474 031110 104146  ERROR   146           ;MSH B0 ERROR
6475 031112 104147  ERROR   147           ;MSG A1 ERROR
6476 031114 104150  ERROR   150           ;MSG B1 ERROR
6477 031116 013765 001222 000010  MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6478 031124 012737 000000 005464  MOV      #0,UNITB
6479 031132 063765 005464 000010  ADD      UNITB,RKCS2(R5)
6480 031140 112737 000101 056644  MOVB    #'A,MSG19A
6481 031146 062765 000010 000010  ADD      #RLS,RKCS2(R5) ;RELEASE PORT A
6482 031154 012737 000001 005314  MOV      #SELDRV,HCS1
6483 031162 004737 043372  JSR      PC,DOCMD      ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6484 031166 104117  ERROR   117           ;NO RDY AFTER SEL DRV CMD
6485
6486 031170 013765 001222 000010  MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6487 031176 012737 000001 005464  MOV      #1,UNITB
6488 031204 063765 005464 000010  ADD      UNITB,RKCS2(R5)
6489 031212 112737 000102 056644  MOVB    #'B,MSG19A
6490 031220 012737 000001 005314  MOV      #SELDRV,HCS1
6491 031226 004737 043372  JSR      PC,DOCMD      ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6492 031232 104117  ERROR   117           ;NO RKY AFTER SEL DRV CMD
6493
6494 031234 032737 000040 005342  BIT      #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6495 031242 001001  BNE     66$           ;BR IF YES
6496 031244 104071  ERROR   71           ;PORT B NOT AVAIL AFTER PORT A RLS
6497 031246

```

66\$:

```

6498 031246 004737 043750 JSR PC,TSTATN
6499 031252 000401 BR 1$
6500 031254 104160 ERROR 160 ;ATTN ON PORT B AFTER SEEK & RLS FROM PORT B
6501
6502 031256 1$:
6503
6504 031256 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6505 031264 005037 005406 CLR E.B0 ;EXPECTED MSG B0
6506 031270 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6507 031276 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
6508 031304 005037 005414 CLR E.A2 ;EXPECTED MSG A2
6509 031310 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
6510 031316 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
6511
6512 031324 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
6513 031330 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
6514 031332 104145 ERROR 145 ;MSG A0 ERROR AFTER SEEK & RLS FROM PORT B
6515 031334 104146 ERROR 146 ;MSH B0 ERROR
6516 031336 104147 ERROR 147 ;MSG A1 ERROR
6517 031340 104150 ERROR 150 ;MSG B1 ERROR
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532

```

```

*****
: *TEST 26 TEST TIMEOUT RETRIGGER THRU PORT 'A'
: *
: * VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED.
: *
: * A. PORT 'A' SEIZES THE DRIVE
: *
: * B. THE PROGRAM WAITS 500MS & RE-SEIZES THE DRIVE THRU PORT 'A'
: *
: * C. PORT 'B' ATTEMPTS TO SEIZE THE DRIVE & THE PROGRAM
: * VERIFIES THAT FULL TIMEOUT TOOK PLACE FROM STEP B ABOVE.
: *
*****

```

```

6533 031342 000004 TST26: SCOPE
6534 031344 012737 000001 001174 MOV #1,$TIMES ;;DO 1 ITERATION
6535 031352 012706 001100 MOV #STACK,SP
6536 031356 004737 045534 JSR PC,SUBCLR
6537 031362 104024 ERROR 24 ;CERR AFTER SCLR
6538 031364 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6539 031372 012737 000001 005464 MOV #1,UNITB
6540 031400 063765 005464 000010 ADD UNITB,RKCS2(R5)
6541 031406 112737 000102 056644 MOVB #'B,MSG19A
6542 031414 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT B
6543 031422 012737 000001 005314 MOV #SELDRV,HCS1
6544 031430 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6545 031434 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6546
6547 031436 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6548 031444 012737 000000 005464 MOV #0,UNITB
6549 031452 063765 005464 000010 ADD UNITB,RKCS2(R5)
6550 031460 112737 000101 056644 MOVB #'A,MSG19A
6551 031466 012737 000001 005314 MOV #SELDRV,HCS1
6552 031474 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6553 031500 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD

```

```

6554
6555 031502 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT A
6556 031510 001001 BNE 64$ ;BR IF YES
6557 031512 104071 ERROR 71 ;PORT A NOT AVAIL AFTER PORT B RLS
6558 031514 64$: JSR PC,SUBCLR
6559 031514 004737 045534 ERROR 24 ;CERR AFTER SCLR
6560 031520 104024
6561
6562 031522 012737 000036 001366 MOV #30,COUNT
6563 031530 004737 047204 JSR PC,TMO ;DO 500MS TIMEOUT
6564 031534 004737 044172 JSR PC,DRAV ;RE-SEIZE DRIVE THRU PORT A
6565 031540 104045 ERROR 45 ;PORT A NOT AVAIL AFTER TIMEOUT
6566
6567 031542 012737 010340 005404 MOV #<D.DRA!O!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6568 031550 005037 005406 CLR E.B0 ;EXPECTED MSG B0
6569 031554 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6570 031562 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
6571 031570 005037 005414 CLR E.A2 ;EXPECTED MSG A2
6572 031574 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
6573 031602 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
6574
6575 031610 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
6576 031614 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
6577 031616 104165 ERROR 165 ;MSG A0 ERROR AFTER TIMEOUT
6578 031620 104166 ERROR 166 ;MSH B0 ERROR
6579 031622 104167 ERROR 167 ;MSG A1 ERROR
6580 031624 104170 ERROR 170 ;MSG B1 ERROR
6581 031626 012737 177777 001366 MOV #-1,COUNT
6582 031634 004737 047114 JSR PC,CLKON ;TURN ON CLOCK
6583 031640 012737 000001 005464 MOV #1,UNITB ;SETUP PORT B
6584 031646 004737 044172 JSR PC,DRAV ;SEE IF DRV AVAIL
6585 031652 000401 BR 1$ ;BR IF NO
6586 031654 104103 ERROR 103 ;PORT A AVAIL BEFORE TMO OR RLS
6587 031656 112737 000102 056644 1$: MOV #B,MSG19A
6588 031664 013704 001222 MOV $UNIT,R4
6589 031670 063704 005464 ADD UNITB,R4
6590 031674 004737 044214 JSR PC,FATT3
6591 031700 104110 ERROR 110 ;NO ATTN ON PORT B TO ALLOW SEIZE
6592
6593 031702 004737 047162 JSR PC,CLKOF ;TURN CLOCK OFF
6594 031706 005137 001366 COM COUNT ;GET ACTUAL COUNT OF TIMEOUT
6595 031712 023727 001366 000043 CMP COUNT,#35. ;COMPARE COUNT AGAINST APPROX 1 SEC
6596 031720 002001 BGE TST27 ;GO TO NEXT TEST
6597 031722 104153 ERROR 153 ;TIMEOUT DID NOT RE-TRIGGER
6598
6599
6600 *****
6601 *TEST 27 TEST TIMEOUT RETRIGGER THRU PORT 'B'
6602 *
6603 * THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
6604 *
6605 *****
6606 TST27: SCOPE
6607 031724 000004 MOV #1,$TIMES ;DO 1 ITERATION
6608 031726 012737 000001 001174 MOV #STACK,SP
6609 031734 012706 001100 JSR PC,SUBCLR
6608 031740 004737 045534 ERROR 24 ;CERR AFTER SCLR
6609 031744 104024

```



```

6610 031746 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6611 031754 012737 000000 005464 MOV #0,UNITB
6612 031762 063765 005464 000010 ADD UNITB,RKCS2(R5)
6613 031770 112737 000101 056644 MOVB #'A,MSG19A
6614 031776 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT A
6615 032004 012737 000001 005314 MOV #SELDRV,HCS1
6616 032012 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6617 032016 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6618
6619 032020 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6620 032026 012737 000001 005464 MOV #1,UNITB
6621 032034 063765 005464 000010 ADD UNITB,RKCS2(R5)
6622 032042 112737 000102 056644 MOVB #'B,MSG19A
6623 032050 012737 000001 005314 MOV #SELDRV,HCS1
6624 032056 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6625 032062 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD
6626
6627 032064 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT B
6628 032072 001001 BNE 64$ ;BR IF YES
6629 032074 104071 ERROR 71 ;PORT B NOT AVAIL AFTER PORT A RLS
6630 032076 64$:
6631 032076 004737 045534 JSR PC,SUBCLR
6632 032102 104024 ERROR 24 ;CERR AFTER SCLR
6633
6634 032104 012737 000036 001366 MOV #30,COUNT
6635 032112 004737 047204 JSR PC,TMO ;DO 500MS TIMEOUT
6636 032116 004737 044172 JSR PC,DRAV ;RE-SEIZE DRIVE THRU PORT B
6637 032122 104045 ERROR 45 ;PORT B NOT AVAIL AFTER TIMEOUT
6638
6639 032124 012737 010340 005404 MOV #<D.DRA!O!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
6640 032132 005037 005406 CLR E.B0 ;EXPECTED MSG B0
6641 032136 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6642 032144 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
6643 032152 005037 005414 CLR E.A2 ;EXPECTED MSG A2
6644 032156 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
6645 032164 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
6646
6647 032172 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
6648 032176 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
6649 032200 104165 ERROR 165 ;MSG A0 ERROR AFTER TIMEOUT
6650 032202 104166 ERROR 166 ;MSH B0 ERROR
6651 032204 104167 ERROR 167 ;MSG A1 ERROR
6652 032206 104170 ERROR 170 ;MSG B1 ERROR
6653 032210 012737 177777 001366 MOV #-1,COUNT
6654 032216 004737 047114 JSR PC,CLKON ;TURN ON CLOCK
6655 032222 012737 000000 005464 MOV #0,UNITB ;SETUP PORT A
6656 032230 004737 044172 JSR PC,DRAV ;SEE IF DRV AVAIL
6657 032234 000401 BR 1$ ;BR IF NO
6658 032236 104103 ERROR 103 ;PORT B AVAIL BEFORE TMO OR RLS
6659 032240 112737 000101 056644 1$: MOVB #'A,MSG19A
6660 032246 013704 001222 MOV $UNIT,R4
6661 032252 063704 005464 ADD UNITB,R4
6662 032256 004737 044214 JSR PC,FAIT3
6663 032262 104110 ERROR 110 ;NO ATTN ON PORT A TO ALLOW SEIZE
6664
6665 032264 004737 047162 JSR PC,CLKOF ;TURN CLOCK OFF
  
```

6666 032270 005137 001366
6667 032274 023727 001366 000043
6668 032302 002001
6669 032304 104153
6670

COM	COUNT	:GET ACTUAL COUNT OF TIMEOUT
CMP	COUNT,#35.	:COMPARE COUNT AGAINST APPROX 1 SEC
BGE	TST30	::GO TO NEXT TEST
ERROR	153	:TIMEOUT DID NOT RE-TRIGGER

6671
6672
6673
6674
6675
6676
6677
6678
6679
6680
6681
6682
6683
6684
6685
6686
6687
6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726

```
*****  
:TEST 30 TEST PORT 'A' TIMER INHIBIT  
:  
:A. PORT 'A' SEIZES THE DRIVE  
:B. PORT 'B' ATTEMPTS TO SEIZE THE DRIVE  
:C. PORT 'A' RELEASES THE DRIVE  
:D. PORT 'A' ATTEMPTS TO GET THE DRIVE BACK.  
:  
:THE PROGRAM VERIFIES THAT PORT 'A' CANNOT ACCESS  
:THE DRIVE FOR APPROX 1 SEC  
:  
*****
```

```
TST30: SCOPE  
MOV #1,$TIMES ;:DO 1 ITERATION  
MOV #STACK,SP  
JSR PC,SUBCLR  
ERROR 24 ;:CERR AFTER SCLR  
MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT B  
MOV #1,UNITB  
ADD UNITB,RKCS2(R5)  
MOVB #'B,MSG19A  
ADD #RLS,RKCS2(R5) ;:RELEASE PORT B  
MOV #SELDRV,HCS1  
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY  
ERROR 117 ;:NO RDY AFTER SEL DRV CMD  
  
MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT A  
MOV #0,UNITB  
ADD UNITB,RKCS2(R5)  
MOVB #'A,MSG19A  
MOV #SELDRV,HCS1  
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY  
ERROR 117 ;:NO RKY AFTER SEL DRV CMD  
  
BIT #D.DRA,HMR2 ;:SEE IF DRIVE AVAIL ON PORT A  
BNE 64$ ;:BR IF YES  
ERROR 71 ;:PORT A NOT AVAIL AFTER PORT B RLS  
  
64$: JSR PC,SUBCLR  
ERROR 24 ;:CERR AFTER SCLR  
  
MOV #1,UNITB ;:SETUP FOR PORT B  
MOVB #'B,MSG19A  
JSR PC,DRAV ;:PORT B TRIES TO SEIZE THE DRIVE  
BR 1$ ;:BR IF NOT AVAIL  
ERROR 103 ;:PORT B AVAIL BEFORE TMO OR RELEASE  
  
1$: MOV #CCLR,RKCS1(R5)  
MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT A  
MOV #0,UNITB  
ADD UNITB,RKCS2(R5)  
MOVB #'A,MSG19A  
ADD #RLS,RKCS2(R5) ;:RELEASE PORT A  
MOV #SELDRV,HCS1  
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY
```

```

6727 032570 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6728
6729 032572 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6730 032600 012737 000001 005464 MOV #1,UNITB
6731 032606 063765 005464 000010 ADD UNITB,RKCS2(R5)
6732 032614 112737 000102 056644 MOVB #'B,MSG19A
6733 032622 012737 000001 005314 MOV #SELDRV,HCS1
6734 032630 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6735 032634 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD
6736
6737 032636 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT B
6738 032644 001001 BNE 65$ ;BR IF YES
6739 032646 104071 ERROR 71 ;PORT B NOT AVAIL AFTER PORT A RLS
6740 032650
6741 032650 012737 000000 005464 65$: MOV #0,UNITB ;SETUP FOR PORT A
6742 032656 112737 000101 056644 MOVB #'A,MSG19A
6743 032664 004737 044172 JSR PC,DRAV
6744 032670 000401 BR 2$
6745 032672 104103 ERROR 103 ;PORT A AVAIL BEFORE TMO OR RELEASE
6746
6747 032674 012765 100000 000000 2$: MOV #CCLR,RKCS1(R5)
6748 032702 012737 177777 001366 MOV #-1,COUNT
6749 032710 004737 047114 JSR PC,CLKON
6750 032714 013704 001222 MOV $UNIT,R4
6751 032720 063704 005464 ADD UNITB,R4
6752 032724 004737 044214 JSR PC,FATT3
6753 032730 104110 ERROR 110 ;NO ATTN ON PORT A TO ALLOW SEIZE
6754
6755 032732 004737 047162 JSR PC,CLKOF ;TURN CLOCK OFF
6756 032736 005137 001366 COM COUNT ;GET ACTUAL COUNT OF TIMEOUT
6757 032742 023727 001366 000043 CMP COUNT,#35. ;COMPARE AGAINST APPROX 1 SEC
6758 032750 002001 BGE TST31 ;GO TO NEXT TST
6759 032752 104153 ERROR 153 ;TIMEOUT DID NOT RE-TRIGGER
6760
6761 *****
6762 :*TEST 31 TEST PORT 'B' TIMER INHIBIT
6763 :*
6764 :* THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
6765 :*
6766 :* *****
6767 TST31: SCOPE
6768 032754 000004 MOV #1,$TIMES ;:DO 1 ITERATION
6769 032756 012737 000001 001174 MOV #STACK,SP
6770 032764 012706 001100 JSR PC,SUBCLR
6771 032770 004737 045534 ERROR 24 ;CERR AFTER SCLR
6772 032776 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6773 033004 012737 000000 005464 MOV #0,UNITB
6774 033012 063765 005464 000010 ADD UNITB,RKCS2(R5)
6775 033020 112737 000101 056644 MOVB #'A,MSG19A
6776 033026 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT A
6777 033034 012737 000001 005314 MOV #SELDRV,HCS1
6778 033042 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6779 033046 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6780
6781 033050 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6782 033056 012737 000001 005464 MOV #1,UNITB
  
```

6783	033064	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
6784	033072	112737	000102	056644		MOVB	#'B,MSG19A	
6785	033100	012737	000001	005314		MOV	#SELDRV,HCS1	
6786	033106	004737	043372			JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
6787	033112	104117				ERROR	117	;NO RKY AFTER SEL DRV CMD
6788								
6789	033114	032737	000040	005342		BIT	#D.DRA,HMR2	;SEE IF DRIVE AVAIL ON PORT B
6790	033122	001001				BNE	64\$;BR IF YES
6791	033124	104071				ERROR	71	;PORT B NOT AVAIL AFTER PORT A RLS
6792	033126				64\$:			
6793	033126	004737	045534			JSR	PC,SUBCLR	
6794	033132	104024				ERROR	24	;CERR AFTER SCLR
6795								
6796	033134	012737	000000	005464		MOV	#0,UNITB	;SETUP FOR PORT A
6797	033142	112737	000101	056644		MOVB	#'A,MSG19A	
6798	033150	004737	044172			JSR	PC,DRAV	;PORT A TRIES TO SEIZE THE DRIVE
6799	033154	000401				BR	1\$;BR IF NOT AVAIL
6800	033156	104103				ERROR	103	;PORT A AVAIL BEFORE TMO OR RELEASE
6801								
6802	033160	012765	100000	000000	1\$:	MOV	#CCLR,RKCS1(R5)	
6803	033166	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT B
6804	033174	012737	000001	005464		MOV	#1,UNITB	
6805	033202	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
6806	033210	112737	000102	056644		MOVB	#'B,MSG19A	
6807	033216	062765	000010	000010		ADD	#RLS,RKCS2(R5)	;RELEASE PORT B
6808	033224	012737	000001	005314		MOV	#SELDRV,HCS1	
6809	033232	004737	043372			JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
6810	033236	104117				ERROR	117	;NO RDY AFTER SEL DRV CMD
6811								
6812	033240	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT A
6813	033246	012737	000000	005464		MOV	#0,UNITB	
6814	033254	063765	005464	000010		ADD	UNITB,RKCS2(R5)	
6815	033262	112737	000101	056644		MOVB	#'A,MSG19A	
6816	033270	012737	000001	005314		MOV	#SELDRV,HCS1	
6817	033276	004737	043372			JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
6818	033302	104117				ERROR	117	;NO RKY AFTER SEL DRV CMD
6819								
6820	033304	032737	000040	005342		BIT	#D.DRA,HMR2	;SEE IF DRIVE AVAIL ON PORT A
6821	033312	001001				BNE	65\$;BR IF YES
6822	033314	104071				ERROR	71	;PORT A NOT AVAIL AFTER PORT B RLS
6823	033316				65\$:			
6824	033316	012737	000001	005464		MOV	#1,UNITB	;SETUP FOR PORT B
6825	033324	112737	000102	056644		MOVB	#'B,MSG19A	
6826	033332	004737	044172			JSR	PC,DRAV	
6827	033336	000401				BR	2\$	
6828	033340	104103				ERROR	103	;PORT B AVAIL BEFORE TMO OR RELEASE
6829								
6830	033342	012765	100000	000000	2\$:	MOV	#CCLR,RKCS1(R5)	
6831	033350	012737	177777	001366		MOV	#-1,COUNT	
6832	033356	004737	047114			JSR	PC,CLKON	
6833	033362	013704	001222			MOV	\$UNIT,R4	
6834	033366	063704	005464			ADD	UNITB,R4	
6835	033372	004737	044214			JSR	PC,FATT3	
6836	033376	104110				ERROR	110	;NO ATTN ON PORT B TO ALLOW SEIZE
6837								
6838	033400	004737	047162			JSR	PC,CLKOF	;TURN CLOCK OFF

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59 C 11
CZR6GC.P11 04-JAN-82 12:39 T31 TEST PORT 'B' TIMER INHIBIT PAGE 133

SEQ 0132

6839 033404 005137 001366
6840 033410 023727 001366 000043
6841 033416 002001
6842 033420 104153
6843

COM COUNT ;GET ACTUAL COUNT OF TIMEOUT
CMP COUNT,#35. ;COMPARE AGAINST APPROX 1 SEC
BGE TST32 ;GO TO NEXT TST
ERROR 153 ;TIMEOUT DID NOT RE-TRIGGER

6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862
6863
6864
6865 033422 000004
6866 033424 012737 000001 001174
6867 033432 004737 045534
6868 033436 104024
6869 033440 012706 001100
6870 033444 013765 001222 000010
6871 033452 012737 000001 005464
6872 033460 063765 005464 000010
6873 033466 112737 000102 056644
6874 033474 062765 000010 000010
6875 033502 012737 000001 005314
6876 033510 004737 043372
6877 033514 104117
6878
6879 033516 013765 001222 000010
6880 033524 012737 000000 005464
6881 033532 063765 005464 000010
6882 033540 112737 000101 056644
6883 033546 012737 000001 005314
6884 033554 004737 043372
6885 033560 104117
6886
6887 033562 032737 000040 005342
6888 033570 001001
6889 033572 104071
6890 033574
6891 033574 004737 045534
6892 033600 104024
6893
6894 033602 005237 005276
6895 033606 012737 034046 001176
6896 033614 012737 000007 005314
6897 033622 053737 001170 005314
6898 033630 013765 005314 000000
6899 033636 013737 001400 005352

```

*****
*TEST 32      TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'A'
*
*      VERIFY THAT THE UNLOAD COMMAND THRU A PORT, SEIZES THAT
*      PORT FOR AS LONG AS HEADS ARE UNLOADED & RELEASE IS NOT
*
*      ISSUED.
*
*      A.  ISSUE AN UNLOAD COMMAND THRU PORT 'A'.
*          VERIFY DRIVE UNLOADS & ATTENTION IS SET.
*
*      B.  DELAY FOR 5 SECONDS & VERIFY DRIVE NOT AVAILABLE
*          TO PORT 'B' TO INSURE TIMERS INHIBITED
*
*
*      C.  ISSUE A RELEASE FROM PORT 'A'.  VERIFY DRIVE BECOMES
*          AVAILABLE TO PORT 'B'
*
*      D.  LOAD HEADS FROM PORT 'B' & VERIFY 'ATTN-B' AT COMPLETION
*****

```

```

*****
TST32:  SCOPE
        MOV      #1,$TIMES      ;;DO 1 ITERATION
        JSR      PC,SUBCLR      ;SUB SYSTEM CLEAR 29-SEP-77
        ERROR    24
        MOV      #STACK,SP
        MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT B
        MOV      #1,UNITB
        ADD      UNITB,RKCS2(R5)
        MOVB     #'B,MSG19A
        ADD      #RLS,RKCS2(R5) ;RELEASE PORT B
        MOV      #SELDRV,HCS1
        JSR      PC,DOCMD       ;DO SELDRV (STATUS) CMD & GET CONTR RDY
        ERROR    117           ;NO RDY AFTER SEL DRV CMD
        MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT A
        MOV      #0,UNITB
        ADD      UNITB,RKCS2(R5)
        MOVB     #'A,MSG19A
        MCV      #SELDRV,HCS1
        JSR      PC,DOCMD       ;DO SELDRV (STATUS) CMD & GET CONTR RDY
        ERROR    117           ;NO RKY AFTER SEL DRV CMD
        BIT      #D.DRA,HMR2    ;SEE IF DRIVE AVAIL ON PORT A
        BNE     64$            ;BR IF YES
        ERROR    71            ;PORT A NOT AVAIL AFTER PORT B RLS
64$:    JSR      PC,SUBCLR
        ERROR    24            ;CERR AFTER SCLR
        INC      UNLD           ;USED FOR VALID HALT
        MOV      #2$,$ESCAPE
        MOV      #UNLOAD,HCS1  ;UNLOAD CMD
        BIS      $TMP4,HCS1
        MOV      HCS1,RKCS1(R5)
        MOV      T10,TEMP1     ;SETUP TIMEOUT
*****

```

```

CZR6GCO RK611 DU PORT LGC          MACY11 30(1046) 04-JAN-82 12:59 E 11
CZR6GC.P11 04-JAN-82 12:39          T32      TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'A' PAGE 135
                                                                                               SEQ 0134

6900 033644 004737 044106          JSR    PC,FATT2          ;FIND ATTN
6901 033650 104073                   ERROR  73                ;NO ATTN AFTER UNLD CMD
6902
6903 033652 012737 000454 001366  MOV    #300,COUNT
6904 033660 004737 047204          JSR    PC,TMO           ;DO 5 SEC DELAY
6905 033664 012737 000001 005464  MOV    #1,UNITB        ;SETUP FOR PORT B
6906 033672 112737 000102 056644  MOVB   #'B,MSG19A
6907 033700 004737 044172          JSR    PC,DRAV
6908 033704 000401                   BR     1$
6909 033706 104155                   ERROR  155              ;PORT B AVAIL BEFORE RLS WHEN UNLOADED
6910                                     ;UNLOAD DID NOT INHIBIT TIMERS
6911
6912 033710 012765 100000 000000 1$:   MOV    #CCLR,RKCS1(R5)
6913 033716 013765 001222 000010  MOV    $UNIT,RKCS2(R5) ;SETUP FOR PORT A
6914 033724 012737 000000 005464  MOV    #0,UNITB
6915 033732 063765 005464 000010  ADD    UNITB,RKCS2(R5)
6916 033740 112737 000101 056644  MOVB   #'A,MSG19A
6917 033746 062765 000010 000010  ADD    #RLS,RKCS2(R5) ;RELEASE PORT A
6918 033754 012737 000001 005314  MOV    #SELDRV,HCS1
6919 033762 004737 043372          JSR    PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6920 033766 104117                   ERROR  117              ;NO RDY AFTER SEL DRV CMD
6921
6922 033770 013765 001222 000010  MOV    $UNIT,RKCS2(R5) ;SETUP FOR PORT B
6923 033776 012737 000001 005464  MOV    #1,UNITB
6924 034004 063765 005464 000010  ADD    UNITB,RKCS2(R5)
6925 034012 112737 000102 056644  MOVB   #'B,MSG19A
6926 034020 012737 000001 005314  MOV    #SELDRV,HCS1
6927 034026 004737 043372          JSR    PC,DOCMD        ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6928 034032 104117                   ERROR  117              ;NO RKY AFTER SEL DRV CMD
6929
6930 034034 032737 000040 005342  BIT    #D.DRA,HMR2     ;SEE IF DRIVE AVAIL ON PORT B
6931 034042 001001                   BNE   65$              ;BR IF YES
6932 034044 104071                   ERROR  71                ;PORT B NOT AVAIL AFTER PORT A RLS
6933 034046
6934 034046 005037 001176          65$: CLR    $ESCAPE
6935                                     2$:
6936 034052 004737 045534          JSR    PC,SUBCLR
6937 034056 104024                   ERROR  24                ;CERR AFTER SCLR
6938
6939 034060 012737 000011 005314  MOV    #SRTSPL,HCS1
6940 034066 004737 043372          JSR    PC,DOCMD        ;DO START SPINDLE CMD & GET CONTR RDY
6941 034072 104121                   ERROR  121              ;RDY NOT SET AFTER ST SPIN CMD.
6942
6943 034074 013737 001406 005354  MOV    T100,TEMP2     ;SETUP TIMEOUT
6944 034102 004737 044006          JSR    PC,FATT1        ;FIND ATTN
6945 034106 104074                   ERROR  74                ;NO ATTN AFTER ST SPIN CMD.
6946
6947 034110 005037 005276          CLR    UNLD
6948
6949 034114 005037 005276          CLR    UNLD
6950
6951                                     ;*****
6952                                     ;*TEST 33      TEST UNLOAD COMMAND TIMER INHIBIT THRU PORT 'B'*
6953                                     ;*
6954                                     ;*      THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.
6955                                     ;*

```



```
6956  
6957 034120 000004  
6958 034122 012737 000001 001174  
6959 034130 004737 045534  
6960 034134 104024  
6961 034136 012706 001100  
6962 034142 013765 001222 000010  
6963 034150 012737 000000 005464  
6964 034156 063765 005464 000010  
6965 034164 112737 000101 056644  
6966 034172 062765 000010 000010  
6967 034200 012737 000001 005314  
6968 034206 004737 043372  
6969 034212 104117  
6970  
6971 034214 013765 001222 000010  
6972 034222 012737 000001 005464  
6973 034230 063765 005464 000010  
6974 034236 112737 000102 056644  
6975 034244 012737 000001 005314  
6976 034252 004737 043372  
6977 034256 104117  
6978  
6979 034260 032737 000040 005342  
6980 034266 001001  
6981 034270 104071  
6982 034272  
6983 034272 004737 045534  
6984 034276 104024  
6985  
6986 034300 005237 005276  
6987 034304 012737 034544 001176  
6988 034312 012737 000007 005314  
6989 034320 053737 001170 005314  
6990 034326 013765 005314 000000  
6991 034334 013737 001400 005352  
6992 034342 004737 044106  
6993 034346 104073  
6994  
6995 034350 012737 000454 001366  
6996 034356 004737 047204  
6997 034362 012737 000000 005464  
6998 034370 112737 000101 056644  
6999 034376 004737 044172  
7000 034402 000401  
7001 034404 104155  
7002  
7003  
7004 034406 012765 100000 000000 1$:  
7005 034414 013765 001222 000010  
7006 034422 012737 000001 005464  
7007 034430 063765 005464 000010  
7008 034436 112737 000102 056644  
7009 034444 062765 000010 000010  
7010 034452 012737 000001 005314  
7011 034460 004737 043372
```

TST33: SCOPE
MOV #1,\$TIMES ;:DO 1 ITERATION
JSR PC,SUBCLR ;:SUB SYSTEM CLEAR 29-SEP-77
ERROR 24
MOV #STACK,SP
MOV \$UNIT,RKCS2(R5) ;:SETUP FOR PORT A
MOV #0,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'A,MSG19A
ADD #RLS,RKCS2(R5) ;:RELEASE PORT A
MOV #SELDRV,HCS1
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;:NO RDY AFTER SEL DRV CMD
MOV \$UNIT,RKCS2(R5) ;:SETUP FOR PORT B
MOV #1,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'B,MSG19A
MOV #SELDRV,HCS1
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;:NO RKY AFTER SEL DRV CMD
BIT #D.DRA,HMR2 ;:SEE IF DRIVE AVAIL ON PORT B
BNE 64\$;:BR IF YES
ERROR 71 ;:PORT B NOT AVAIL AFTER PORT A RLS
64\$: JSR PC,SUBCLR
ERROR 24 ;:CERR AFTER SCLR
INC UNLD ;:USED FOR VALID HALT
MOV #2\$,SESCAPE
MOV #UNLOAD,HCS1 ;:UNLOAD CMD
BIS \$TMP4,HCS1
MOV HCS1,RKCS1(R5)
MOV T10,TEMP1 ;:SETUP TIMEOUT
JSR PC,FATT2 ;:FIND ATTN
ERROR 73 ;:NO ATTN AFTER UNLD CMD
MOV #300,COUNT
JSR PC,TMO ;:DO 5 SEC DELAY
MOV #0,UNITB ;:SETUP FOR PORT A
MOVB #'A,MSG19A
JSR PC,DRAV
BR 1\$
ERROR 155 ;:PORT A AVAIL BEFORE RLS WHEN UNLOADED
;:UNLOAD DID NOT INHIBIT TIMERS
1\$: MOV #CCLR,RKCS1(R5)
MOV \$UNIT,RKCS2(R5) ;:SETUP FOR PORT B
MOV #1,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'B,MSG19A
ADD #RLS,RKCS2(R5) ;:RELEASE PORT B
MOV #SELDRV,HCS1
JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY

```

7012 034464 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
7013
7014 034466 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7015 034474 012737 000000 005464 MOV #0,UNITB
7016 034502 063765 005464 000010 ADD UNITB,RKCS2(R5)
7017 034510 112737 000101 056644 MOVB #'A,MSG19A
7018 034516 012737 000001 005314 MOV #SELDRV,HCS1
7019 034524 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7020 034530 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD
7021
7022 034532 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT A
7023 034540 001001 BNE 65$ ;BR IF YES
7024 034542 104071 ERROR 71 ;PORT A NOT AVAIL AFTER PORT B RLS
7025 034544 65$:
7026 034544 005037 001176 2$: CLR $ESCAPE
7027
7028 034550 004737 045534 JSR PC,SUBCLR
7029 034554 104024 ERROR 24 ;CERR AFTER SCLR
7030
7031 034556 012737 000011 005314 MOV #SRTSPL,HCS1
7032 034564 004737 043372 JSR PC,DOCMD ;DO START SPINDLE CMD & GET CONTR RDY
7033 034570 104121 ERROR 121 ;RDY NOT SET AFTER ST SPIN CMD.
7034
7035 034572 013737 001406 005354 MOV T100,TEMP2 ;SETUP TIMEOUT
7036 034600 004737 044006 JSR PC,FATT1 ;FIND ATTN
7037 034604 104074 ERROR 74 ;NO ATTN AFTER ST SPIN CMD.
7038
7039 034606 005037 005276 CLR UNLD
7040
7041 034612 005037 005276 CLR UNLD
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
    
```

```

*****
:*TEST 34 TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'A'
:*
:* VERIFY THAT THE RECAL COMMAND THRU A PORT SEIZES THAT PORT
:* FOR AS LONG AS THE RECAL IS IN PROGRESS & RELEASE IS NOT ISSUED.
:*
:* A. ISSUE A RECAL COMMAND FROM CYL 410 THRU PORT 'A'
:*
:* B. VERIFY PORT 'B' CANNOT SEIZE THE DRIVE UNTIL PORT 'A'
:* RECEIVES ATTN AND TIMES OUT.THIS INSURES THAT
:* THE TIMERS ARE INHIBITED.
*****
    
```

```

7057 034616 000004 TST34: SCOPE
7058 034620 012737 000001 001174 MOV #1,$TIMES ;:DO 1 ITERATION
7059 034626 004737 045534 JSR PC,SUBCLR ;SUB SYSTEM CLEAR 29-SEP-77
7060 034632 104024 ERROR 24 ;29-SEP-77
7061 034634 012706 001100 MOV #STACK,SP
7062 034640 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7063 034646 012737 000001 005464 MOV #1,UNITB
7064 034654 063765 005464 000010 ADD UNITB,RKCS2(R5)
7065 034662 112737 000102 056644 MOVB #'B,MSG19A
7066 034670 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT B
7067 034676 012737 000001 005314 MOV #SELDRV,HCS1
    
```

```

CZR6GCO RK611 DU PORT LGC          MACY11 30(1046) 04-JAN-82 12:59 H 11
CZR6GC.P11 04-JAN-82 12:39          T34      TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'A' PAGE 138
                                                    SEQ 0137

7068 034704 004737 043372          JSR      PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7069 034710 104117                  ERROR    117              ;NO RDY AFTER SEL DRV CMD
7070
7071 034712 013765 001222 000010    MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7072 034720 012737 000000 005464    MOV      #0,UNITB
7073 034726 063765 005464 000010    ADD      UNITB,RKCS2(R5)
7074 034734 112737 000101 056644    MOV      #'A,MSG19A
7075 034742 012737 000001 005314    MOV      #SELDV,HCS1
7076 034750 004737 043372          JSR      PC,DOCMD          ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7077 034754 104117                  ERROR    117              ;NO RKY AFTER SEL DRV CMD
7078
7079 034756 032737 000040 005342    BIT      #D.DRA,HMR2      ;SEE IF DRIVE AVAIL ON PORT A
7080 034764 001001                  BNE     64$              ;BR IF YES
7081 034766 104071                  ERROR    71              ;PORT A NOT AVAIL AFTER PORT B RLS
7082 034770
7083 034770 004737 045534          JSR      PC,SUBCLR
7084 034774 104024                  ERROR    24              ;CERR AFTER SCLR
7085
7086 034776 012737 035464 001176    MOV      #5$, $ESCAPE
7087 035004 013765 012770 000020    MOV      LC,RKDC(R5)      ;SEEK TO LAST CYL
7088
7089 035012 012737 000017 005314    MOV      #SEEK,HCS1
7090 035020 004737 043372          JSR      PC,DOCMD          ;DO SEEK CMD & GET CONTR READY
7091 035024 104131                  ERROR    131             ;NO RDY AFTER SEEK CMD
7092
7093 035026 013737 001412 005352    MOV      T50000,TEMP1    ;SETUP TIMEOUT
7094 035034 004737 044106          JSR      PC,FATT2         ;FIND ATTN
7095 035040 104132                  ERROR    132             ;NO ATTN AFTER SEEK CMD
7096
7097 035042 032737 100000 005314    BIT      #CERR,HCS1
7098 035050 001401                  BEQ     65$
7099 035052 104210                  ERROR    210             ;CERR AFTER SEEK CMD
7100
7101 035054
7102
7103 035054 012737 050340 005404    MOV      #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7104 035062 005037 005406          CLR     E.B0              ;EXPECTED MSG B0
7105 035066 012737 001720 005410    MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7106 035074 012737 000001 005412    MOV      #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
7107 035102 005037 005414          CLR     E.A2              ;EXPECTED MSG A2
7108 035106 012737 000002 005416    MOV      #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
7109 035114 012737 000003 005422    MOV      #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
7110
7111 035122 004737 044274          JSR      PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
7112 035126 000000                  .WORD  0!0!0             ;& MSGS SPECIFIED HERE
7113 035130 104161                  ERROR    161             ;MSG A0 ERROR AFTER SEEK CMD
7114 035132 104162                  ERROR    162             ;MSG B0 ERROR
7115 035134 104163                  ERROR    163             ;MSG A1 ERROR
7116 035136 104164                  ERROR    164             ;MSG B1 ERROR
7117
7118
7119 035140 012765 100000 000000    MOV      #CCLR,RKCS1(R5) ;DRIVE#
7120 035146 013765 001222 000010    MOV      $UNIT,RKCS2(R5) ;ADD 1 IF ON PORT B
7121 035154 063765 005464 000010    ADD      UNITB,RKCS2(R5)
7122 035162 012737 000005 005314    MOV      #CLEAR,HCS1
7123 035170 004737 043372          JSR      PC,DOCMD          ;DO DRIVE CLEAR CMD & GET CONTR RDY

```

```

CZR6GCO RK611 DU PORT LGC          MACY11 30(1046) 04-JAN-82 12:59 I 11
CZR6GC.P11 04-JAN-82 12:39          T34      TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'A' PAGE 139
                                                    SEQ 0138

7124 035174 104151                ERROR 151                ;NO RDY AFTER DRIVE CLEAR CMD
7125 035176 004737 043750          JSR   PC,TSTATN         ;TEST FOR ATTN
7126 035202 000401                BR    66$
7127 035204 104154                ERROR 154                ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7128 035206                66$:
7129
7130 035206 012737 010340 005404  MOV   #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7131 035214 005037 005406          CLR   E.B0              ;EXPECTED MSG B0
7132 035220 012737 001720 005410  MOV   #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7133 035226 012737 000001 005412  MOV   #1,E.B1           ;MSG ID FOR EXPECTED MSG B1
7134 035234 005037 005414          CLR   E.A2              ;EXPECTED MSG A2
7135 035240 012737 000002 005416  MOV   #2,E.B2           ;MSG ID FOR EXPECTED MSG B2
7136 035246 012737 000003 005422  MOV   #3,E.B3           ;MSG ID FOR EXPECTED MSG B3
7137
7138 035254 004737 044274          JSR   PC,CHKMSG         ;CHECK MSGS A0, B0, A1, B1
7139 035260 000003                .WORD T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
7140 035262 104033                ERROR 33                ;MSG A0 ERROR AFTER DRV CLEAR CMD
7141 035264 104034                ERROR 34                ;MSH B0 ERROR
7142 035266 104035                ERROR 35                ;MSG A1 ERROR
7143 035270 104036                ERROR 36                ;MSG B1 ERROR
7144
7145 035272 012737 000013 005314  MOV   #RECAL,HCS1      ;RECAL COMMAND
7146 035300 053737 001170 005314  BIS   $TMP4,HCS1
7147 035306 013765 005314 000000  MOV   HCS1,RKCS1(R5)
7148 035314 013704 001222          MOV   $UNIT,R4
7149 035320 063704 005464          ADD   UNITB,R4
7150 035324 136465 005304 000017 1$:  BITB  ATTN(R4),RKASOF+1(R5) ;SEE IF ATTN SET
7151 035332 001027                BNE   3$                ;BR IF YES
7152
7153 035334 012737 000001 005464  MOV   #1,UNITB         ;SETUP FOR PORT B
7154 035342 112737 000102 056644  MOVB  #'B,MSG19A
7155 035350 004737 044172          JSR   PC,DRAV          ;SEE IF DRV AVAIL
7156 035354 000401                BR    2$                ;RETURN HERE IF NO
7157 035356 104177                ERROR 177              ;PORT B AVAIL
7158                                ;RECAL DID NOT INHIBIT TIMERS
7159 035360 012765 100000 000000 2$:  MOV   #CCLR,RKCS1(R5)
7160 035366 012737 000000 005464  MOV   #0,UNITB
7161 035374 112737 000101 056644  MOVB  #'A,MSG19A      ;SETUP PORT A
7162 035402 004737 044172          JSR   PC,DRAV          ;SEE IF DRV AVAIL
7163 035406 104203                ERROR 203              ;PORT A NOT REMAIN AVAIL DURING RECAL
7164 035410 000745                BR    1$
7165
7166 035412 004737 045146                3$:  JSR   PC,GSTAT
7167 035416 032737 020000 005342  BIT   #D.PIP,HMR2     ;SEE IF ANY MOTION
7168 035424 001401                BEQ   4$                ;BR IF NO
7169 035426 104072                ERROR 72                ;PIP SET AFTER ATTN RECB FROM RECAL
7170 035430 012737 000170 001366  4$:  MOV   #120.,COUNT
7171 035436 004737 047204          JSR   PC,TMO           ;DO 2 SEC DLY
7172 035442 012737 000001 005464  MOV   #1,UNITB        ;SETUP FOR PORT B
7173 035450 112737 000102 056644  MOVB  #'B,MSG19A
7174 035456 004737 044172          JSR   PC,DRAV          ;SEE IF DRV NOW AVAIL
7175 035462 104045                ERROR 45                ;PORT B NOT AVAIL AFTER TMO
7176
7177 035464 005037 001176                5$:  CLR   $ESCAPE
7178
7179                                ;:*****

```

```
7180 ;*TEST 35 TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'B'  
7181 ;*  
7182 ;* THE PREVIOUS TEST IS REPEATED FOR PORT 'B'.  
7183 ;*  
7184 ;*  
7185 ;*****  
7185 035470 000004 TST35: SCOPE  
7186 035472 012737 000001 001174 MOV #1,$TIMES ;:DO 1 ITERATION  
7187 035500 004737 045534 JSR PC,SUBCLR ;:SUBSYSTEM CLEAR 29-SEP-77  
7188 035504 104024 ERROR 24 ;:29-SEP-77  
7189 035506 012706 001100 MOV #STACK,SP  
7190 035512 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT A  
7191 035520 012737 000000 005464 MOV #0,UNITB  
7192 035526 063765 005464 000010 ADD UNITB,RKCS2(R5)  
7193 035534 112737 000101 056644 MOVB #'A,MSG19A  
7194 035542 062765 000010 000010 ADD #RLS,RKCS2(R5) ;:RELEASE PORT A  
7195 035550 012737 000001 005314 MOV #SELDRV,HCS1  
7196 035556 004737 043372 JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY  
7197 035562 104117 ERROR 117 ;:NO RDY AFTER SEL DRV CMD  
7198  
7199 035564 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;:SETUP FOR PORT B  
7200 035572 012737 000001 005464 MOV #1,UNITB  
7201 035600 063765 005464 000010 ADD UNITB,RKCS2(R5)  
7202 035606 112737 000102 056644 MOVB #'B,MSG19A  
7203 035614 012737 000001 005314 MOV #SELDRV,HCS1  
7204 035622 004737 043372 JSR PC,DOCMD ;:DO SELDRV (STATUS) CMD & GET CONTR RDY  
7205 035626 104117 ERROR 117 ;:NO RKY AFTER SEL DRV CMD  
7206  
7207 035630 032737 000040 005342 BIT #D.DRA,HMR2 ;:SEE IF DRIVE AVAIL ON PORT B  
7208 035636 001001 BNE 64$ ;:BR IF YES  
7209 035640 104071 ERROR 71 ;:PORT B NOT AVAIL AFTER PORT A RLS  
7210 035642 64$:  
7211 035642 004737 045534 JSR PC,SUBCLR  
7212 035646 104024 ERROR 24 ;:CERR AFTER SCLR  
7213  
7214 035650 012737 036336 001176 MOV #5$,$ESCAPE  
7215 035656 013765 012770 000020 MOV LC,RKDC(R5) ;:SEEK TO LAST CYL  
7216  
7217 035664 012737 000017 005314 MOV #SEEK,HCS1  
7218 035672 004737 043372 JSR PC,DOCMD ;:DO SEEK CMD & GET CONTR READY  
7219 035676 104131 ERROR 131 ;:NO RDY AFTER SEEK CMD  
7220  
7221 035700 013737 001412 005352 MOV T50000,TEMP1 ;:SETUP TIMEOUT  
7222 035706 004737 044106 JSR PC,FATT2 ;:FIND ATTN  
7223 035712 104132 ERROR 132 ;:NO ATTN AFTER SEEK CMD  
7224  
7225 035714 032737 100000 005314 BIT #CERR,HCS1  
7226 035722 001401 BEQ 65$  
7227 035724 104210 ERROR 210 ;:CERR AFTER SEEK CMD  
7228  
7229 035726 65$:  
7230  
7231 035726 012737 050340 005404 MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;:EXPECTED MSG A0  
7232 035734 005037 005406 CLR E.B0 ;:EXPECTED MSG B0  
7233 035740 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;:EXPECTED A1  
7234 035746 012737 000001 005412 MOV #1,E.B1 ;:MSG ID FOR EXPECTED MSG B1  
7235 035754 005037 005414 CLR E.A2 ;:EXPECTED MSG A2
```

K 11
PAGE 141

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59 TEST RECAL COMMAND TIMER INHIBIT THRU PORT 'B'

CZR6GC.P11 04-JAN-82 12:39 T35 SEQ 0140

7236	035760	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
7237	035766	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
7238							
7239	035774	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
7240	036000	000000			.WORD	0!0!0	:& MSGS SPECIFIED HERE
7241	036002	104161			ERROR	161	:MSG A0 ERROR AFTER SEEK CMD
7242	036004	104162			ERROR	162	:MSH B0 ERROR
7243	036006	104163			ERROR	163	:MSG A1 ERROR
7244	036010	104164			ERROR	164	:MSG B1 ERROR
7245							
7246							
7247	036012	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	
7248	036020	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	:DRIVE#
7249	036026	063765	005464	000010	ADD	UNITB,RKCS2(R5)	:ADD 1 IF ON PORT B
7250	036034	012737	000005	005314	MOV	#CLEAR,HCS1	
7251	036042	004737	043372		JSR	PC,DOCMD	:DO DRIVE CLEAR CMD & GET CONTR RDY
7252	036046	104151			ERROR	151	:NO RDY AFTER DRIVE CLEAR CMD
7253	036050	004737	043750		JSR	PC,TSTATN	:TEST FOR ATTN
7254	036054	000401			BR	66\$	
7255	036056	104154			ERROR	154	:ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7256	036060						
7257							
7258	036060	012737	010340	005404	MOV	#<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0	:EXPECTED MSG A0
7259	036066	005037	005406		CLR	E.B0	:EXPECTED MSG B0
7260	036072	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	:EXPECTED A1
7261	036100	012737	000001	005412	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
7262	036106	005037	005414		CLR	E.A2	:EXPECTED MSG A2
7263	036112	012737	000002	005416	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
7264	036120	012737	000003	005422	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
7265							
7266	036126	004737	044274		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
7267	036132	000003			.WORD	T.A2!T.B2!0	:& MSGS SPECIFIED HERE
7268	036134	104033			ERROR	33	:MSG A0 ERROR AFTER DRV CLEAR CMD
7269	036136	104034			ERROR	34	:MSH B0 ERROR
7270	036140	104035			ERROR	35	:MSG A1 ERROR
7271	036142	104036			ERROR	36	:MSG B1 ERROR
7272							
7273	036144	012737	000013	005314	MOV	#RECAL,HCS1	:RECAL COMMAND
7274	036152	053737	001170	005314	BIS	\$TMP4,HCS1	
7275	036160	013765	005314	000000	MOV	HCS1,RKCS1(R5)	
7276	036166	013704	001222		MOV	\$UNIT,R4	
7277	036172	063704	005464		ADD	UNITB,R4	
7278	036176	136465	005304	000017	1\$: BITB	ATTN(R4),RKASOF+1(R5)	:SEE IF ATTN SET
7279	036204	001027			BNE	3\$:BR IF YES
7280							
7281	036206	012737	000000	005464	MOV	#0,UNITB	:SETUP FOR PORT A
7282	036214	112737	000101	056644	MOVB	#'A,MSG19A	
7283	036222	004737	044172		JSR	PC,DRAV	:SEE IF DRV AVAIL
7284	036226	000401			BR	2\$:RETURN HERE IF NO
7285	036230	104177			ERROR	177	:PORT A AVAIL
7286							:RECAL DID NOT INHIBIT TIMERS
7287	036232	012765	100000	000000	2\$: MOV	#CCLR,RKCS1(R5)	
7288	036240	012737	000001	005464	MOV	#1,UNITB	
7289	036246	112737	000102	056644	MOVB	#'B,MSG19A	:SETUP PORT B
7290	036254	004737	044172		JSR	PC,DRAV	:SEE IF DRV AVAIL
7291	036260	104203			ERROR	203	:PORT B NOT REMAIN AVAIL DURING RECAL

```

7292 036262 000745 BR 1$
7293
7294 036264 004737 045146 3$: JSR PC,GSTAT
7295 036270 032737 020000 005342 BIT #D,PIP,HMR2 ;SEE IF ANY MOTION
7296 036276 001401 BEQ 4$ ;BR IF NO
7297 036300 104072 ERROR 72 ;PIP SET AFTER ATTN RECA FROM RECAL
7298 036302 012737 000170 001366 4$: MOV #120.,COUNT
7299 036310 004737 047204 JSR PC,TMO ;DO 2 SEC DLY
7300 036314 012737 000000 005464 MOV #0,UNITB ;SETUP FOR PORT A
7301 036322 112737 000101 056644 MOVB #'A,MSG19A
7302 036330 004737 044172 JSR PC,DRAV ;SEE IF DRV NOW AVAIL
7303 036334 104045 ERROR 45 ;PORT A NOT AVAIL AFTER TMO
7304
7305 036336 005037 001176 5$: CLR $ESCAPE
7306
7307
7308
7309
7310
7311
7312
7313
7314
7315
7316
7317
7318
7319
7320
7321
7322
7323
7324
7325
7326
7327
7328 036342 000004
7329 036344 012737 000001 001174
7330 036352 012706 001100
7331
7332 036356 004737 045534
7333 036362 104024
7334
7335 036364 013765 001222 000010
7336 036372 012737 000001 005464
7337 036400 063765 005464 000010
7338 036406 112737 000102 056644
7339 036414 062765 000010 000010
7340 036422 012737 000001 005314
7341 036430 004737 043372
7342 036434 104117
7343
7344 036436 013765 001222 000010
7345 036444 012737 000000 005464
7346 036452 063765 005464 000010
7347 036460 112737 000101 056644
    
```

```

*****
*TEST 36 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
*
* THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
* THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
* FACTORY DURING MANF. ALL BAD SECTOR INFO (BSE) WILL BE STORED
* AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
*
* SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
* SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
*
* IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
* IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
* A MESSAGE WILL BE TYPED INDICATING THAT ALL
* FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
* THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
*
* THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
    
```

```

*****
TST36: SCOPE
MOV #1,$TIMES ;:DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
7332 036356 004737 045534 JSR PC,SUBCLR
7333 036362 104024 ERROR 24 ;CERR AFTER SCLR
7334
7335 036364 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7336 036372 012737 000001 005464 MOV #1,UNITB
7337 036400 063765 005464 000010 ADD UNITB,RKCS2(R5)
7338 036406 112737 000102 056644 MOVB #'B,MSG19A
7339 036414 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT B
7340 036422 012737 000001 005314 MOV #SELDRV,HCS1
7341 036430 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7342 036434 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
7343
7344 036436 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7345 036444 012737 000000 005464 MOV #0,UNITB
7346 036452 063765 005464 000010 ADD UNITB,RKCS2(R5)
7347 036460 112737 000101 056644 MOVB #'A,MSG19A
    
```

```

CZR6GCO RK611 DU PORT LGC          MACY11 30(1046) 04-JAN-82 12:59 M 11
CZR6GC.P11 04-JAN-82 12:39          T36        READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #
                                                    PAGE 143
                                                    SEQ 0142

7348 036466 012737 000001 005314      MOV      #SELDRV,HCS1
7349 036474 004737 043372              JSR      PC,DOCMD      ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7350 036500 104117              ERROR    117          ;NO RKY AFTER SEL DRV CMD
7351
7352 036502 032737 000040 005342      BIT      #D.DRA,HMR2   ;SEE IF DRIVE AVAIL ON PORT A
7353 036510 001001              BNE     64$          ;BR IF YES
7354 036512 104071              ERROR    71          ;PORT A NOT AVAIL AFTER PORT B RLS
7355 036514          64$:
7356
7357 036514 004737 045534      JSR      PC,SUBCLR
7358 036520 104024              ERROR    24          ;CERR AFTER SCLR
7359
7360 036522 005037 005354      CLR      TEMP2        ;SECTOR CTR
7361 036526 005037 005356      CLR      TEMP3        ;0=22 SECTOR HARDWARE DETECTED TABLE
7362
7363
7364 036532 012737 002276 005360      MOV      #BSE22H,TEMP4 ;STORE 22 SECTOR HARDWARE BSE ADDR.
7365 036540 013765 005360 000004      MOV      TEMP4,RKBA(R5)
7366 036546 012737 001000 005362      MOV      #1000,TEMP5   ;TRACK 2, SECTOR 0
7367 036554 013765 005362 000006      MOV      TEMP5,RKDA(R5)
7368
7369 036562 013765 012770 000020 1$:      MOV      LC,RKDC(R5)   ;LAST CYL
7370 036570 012765 177400 000002      MOV      #-256.,RKWC(R5) ;LOAD WORD CT
7371 036576 013765 001222 000010      MOV      $UNIT,RKCS2(R5)
7372 036604 063765 005464 000010      ADD     UNITB,RKCS2(R5)
7373 036612 012737 000021 005314      MOV      #RDDATA,HCS1
7374 036620 004737 043430      JSR      PC,DATCMD    ;DO READ DATA CMD & GET CONTR READY
7375 036624 104226              ERROR    226          ;NO RDY AFTER READ DATA CMD
7376 036626 004737 045146      JSR      PC,GSTAT     ;GET FRESH DATA
7377 036632 032737 100000 005314      BIT      #CERR,HCS1
7378 036640 001470              BEQ     8$
7379 036642 104227              ERROR    227          ;CERR AFTER READ DATA CMD
7380
7381 036644 012737 010340 005404      MOV      #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7382 036652 005037 005406      CLR      E.B0         ;EXPECTED MSG B0
7383 036656 012737 001720 005410      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7384 036664 012737 000001 005412      MOV      #1,E.B1      ;MSG ID FOR EXPECTED MSG B1
7385 036672 005037 005414      CLR      E.A2         ;EXPECTED MSG A2
7386 036676 012737 000002 005416      MOV      #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
7387 036704 012737 000003 005422      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
7388
7389 036712 004737 044274      JSR      PC,CHKMSG    ;CHECK MSGS A0, B0, A1, B1
7390 036716 000000              .WORD   0!0!0        ;& MSGS SPECIFIED HERE
7391 036720 104054              ERROR    54          ;MSG A0 ERROR AFTER READ DATA CMD
7392 036722 104026              ERROR    26          ;MSG B0 ERROR
7393 036724 104056              ERROR    56          ;MSG A1 ERROR
7394 036726 104030              ERROR    30          ;MSG B1 ERROR
7395
7396 036730 004737 045534      JSR      PC,SUBCLR
7397 036734 104024              ERROR    24          ;CERR AFTER SUBCLR
7398
7399 036736 005237 005354          INC     TEMP2
7400 036742 023727 005354 000005      CMP     TEMP2,#5      ;READ ALL 5 SECTORS?
7401 036750 001007              BNE     5$
7402 036752 005737 005356      TST     TEMP3
7403 036756 001002              BNE     2$

```



```

7404 036760 104233          ERROR 233          ;CANT READ SECTORS 0,2,4,6,8
7405 036762 000414          BR      3$
7406 036764 104230          2$:  ERROR 230          ;CANT READ SECTORS 10,12,14,16,18,20
7407 036766 000412          BR      3$
7408
7409 036770 013765 005360 000004 5$:  MOV    TEMP4,RKBA(R5) ;RESTORE TABLE ADDR
7410 036776 062737 000002 005362  ADD    #2,TEMP5      ;SETUP TO READ 2 SECTORS FROM LAST
7411 037004 013765 005362 000006  MOV    TEMP5,RKDA(R5)
7412 037012 000663          BR      1$
7413
7414 037014 005237 001452          3$:  INC    BSERR          ;SET BSE FLAG
7415 037020 000454          BR      TST37         ;GO TO NEXT TEST
7416 037022 005737 002304          8$:  TST    BSE22H+6     ;TEST CARTRIDGE TYPE
7417 037026 001404          BEQ    9$             ;BRANCH IF DATA CARTRIDGE
7418 037030 104235          ERROR 235          ;ALIGNMENT CARTRIDGE USED
7419 037032 005237 001452          INC    BSERR          ;SET BSE ERROR FLAG
7420 037036 000426          BR      10$
7421
7422 037040 005237 005356          9$:  INC    TEMP3
7423 037044 023727 005356 000001  CMP    TEMP3,#1
7424 037052 001020          BNE    10$
7425 037054 005037 005354          CLR    TEMP2
7426 037060 012737 003276 005360  MOV    #BSE22S,TEMP4 ;STORE 22 SECTOR SOFTWARE BSE ADDR
7427 037066 013765 005360 000004  MOV    TEMP4,RKBA(R5)
7428 037074 012737 001012 005362  MOV    #1012,TEMP5   ;TRACK 2, SECTOR 12
7429 037102 013765 005362 000006  MOV    TEMP5,RKDA(R5)
7430 037110 000137 036562          JMP    1$             ;REPEAT
7431
7432 037114 005737 001216          10$: TST    $PASS
7433 037120 001014          BNE    TST37         ;GO TO NEXT TST IF NOT 1'ST PASS
7434 037122 104401 056605          TYPE  ,MSG17         ;CART SERIAL #
7435 037126 012746 002276          MOV    #BSE22H,-(SP)
7436 037132 004737 053744          JSR    PC,$DB20     ;CONVERT DBL BINARY WORD TO OCTAL
7437 037136 004737 054314          JSR    PC,$SUPRS    ;TYPE SERIAL #
7438 037142 104401 001205          TYPE  ,$CRLF
7439 037146 104401 001205          TYPE  ,$CRLF
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458 037152 000004          TST37: SCOPE
7459 037154 012737 000001 001174  MOV    #1,$TIMES    ;DO 1 ITERATION
  
```

```

*****
: *TEST 37      DATA TESTS
: *
: *  VERIFY UNIQUE DATA CAN BE WRITTEN THRU EITHER PORT & READ
: *  BACK CORRECTLY THRU BOTH PORTS.
: *
: *  A.  ALL 0'S ARE WRITTEN THRU PORT 'A' ON CYL 0, SECTOR 0,
: *      TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.
: *
: *  B.  ALL 1'S ARE WRITTEN THRU PORT 'B' ON CYL 10, SECTOR 0,
: *      TRACK 0 & VERIFIED BY READING BACK THRU BOTH PORTS.
: *
: *  C.  THE PROGRAM CHECKS THAT CYL 0 WAS NOT OVERWRITTEN
: *      BY READING & VERIFYING ALL 0'S THRU PORT 'B'.
: *
*****
TST37: SCOPE
MOV    #1,$TIMES    ;DO 1 ITERATION
  
```

7460	037162	012706	001100		MOV	#STACK,SP	
7461							
7462	037166	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT B
7463	037174	012737	000001	005464	MOV	#1,UNITB	
7464	037202	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
7465	037210	112737	000102	056644	MOVB	#'B,MSG19A	
7466	037216	062765	000010	000010	ADD	#RLS,RKCS2(R5)	;RELEASE PORT B
7467	037224	012737	000001	005314	MOV	#SELDRV,HCS1	
7468	037232	004737	043372		JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
7469	037236	104117			ERROR	117	;NO RDY AFTER SEL DRV CMD
7470							
7471	037240	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	;SETUP FOR PORT A
7472	037246	012737	000000	005464	MOV	#0,UNITB	
7473	037254	063765	005464	000010	ADD	UNITB,RKCS2(R5)	
7474	037262	112737	000101	056644	MOVB	#'A,MSG19A	
7475	037270	012737	000001	005314	MOV	#SELDRV,HCS1	
7476	037276	004737	043372		JSR	PC,DOCMD	;DO SELDRV (STATUS) CMD & GET CONTR RDY
7477	037302	104117			ERROR	117	;NO RKY AFTER SEL DRV CMD
7478							
7479	037304	032737	000040	005342	BIT	#D.DRA,HMR2	;SEE IF DRIVE AVAIL ON PORT A
7480	037312	001001			BNE	64\$;BR IF YES
7481	037314	104071			ERROR	71	;PORT A NOT AVAIL AFTER PORT B RLS
7482	037316						
7483	037316	004737	045534		JSR	PC,SUBCLR	
7484	037322	104024			ERROR	24	;CERR AFTER SCLR
7485							
7486	037324	005037	001374		CLR	SECTOR	
7487	037330	005037	001360		CLR	CYLADD	
7488	037334	012737	001434	001436	MOV	#DATA0,DATA01	;HOLD DATA
7489	037342	013765	001374	000006	MOV	SECTOR,RKDA(R5)	;SETUP TO WRITE
7490	037350	013765	001436	000004	MOV	DATA01,RKBA(R5)	;SETUP DATA
7491	037356	013765	001360	000020	MOV	CYLADD,RKDC(R5)	;SETUP CYLINDER
7492	037364	052765	000020	000010	BIS	#BAI,RKCS2(R5)	
7493	037372	012765	177400	000002	MOV	#-256.,RKWC(R5)	
7494							
7495	037400	012737	000023	005314	MOV	#<WRDATA>,HCS1	
7496	037406	004737	043430		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
7497	037412	104011			ERROR	11	;NO RDY AFTER WRITE DATA CMD
7498	037414	004737	045146		JSR	PC,GSTAT	;GET FRESH STATUS
7499	037420	032737	100000	005314	BIT	#CERR,HCS1	
7500	037426	001465			BEQ	68\$;BR IF NO ERRORS
7501							
7502	037430	032737	000200	005330	BIT	#BSE,HER	;SEE IF BAD SECTOR FLAG
7503	037436	001421			BEQ	66\$;BR IF NO
7504	037440	004737	047002		JSR	PC,TRUERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE
7505	037444	000455			BR	67\$;RETURN HERE IF NO
7506							
7507	037446	005237	001374		INC	SECTOR	;RETURN HERE IF YES
7508	037452	023727	001374	000012	CMP	SECTOR,#10.	;ARE 10 CONSEC. SECTORS BAD
7509	037460	001003			BNE	65\$;BR IF NO
7510	037462	104046			ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS
7511	037464	000137	040246		JMP	7\$;BYPASS TEST
7512	037470	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	;TRY ANOTHER SECTOR
7513	037476	000137	037342		JMP	1\$	
7514	037502	104012			ERROR	12	;CERR WITH WRITE DATA CMD
7515							

```

CZR6GCO RK611 DU PORT LGC          MACY11 30(1046) 04-JAN-82 12:59 C 12
CZR6GC.P11 04-JAN-82 12:39          T37      DATA TESTS          PAGE 146
                                                                    SEQ 0145

7516 037504 012737 010340 005404  MOV      #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7517 037512 005037 005406          CLR      E.B0 ;EXPECTED MSG B0
7518 037516 012737 001720 005410  MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7519 037524 012737 000001 005412  MOV      #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7520 037532 005037 005414          CLR      E.A2 ;EXPECTED MSG A2
7521 037536 012737 000002 005416  MOV      #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7522 037544 012737 000003 005422  MOV      #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7523
7524 037552 004737 044274          JSR      PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7525 037556 000003          .WORD   T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7526 037560 104052          ERROR   52 ;MSG A0 ERROR AFTER WRITE DATA CMD
7527 037562 104023          ERROR   23 ;MSH B0 ERROR
7528 037564 104053          ERROR   53 ;MSG A1 ERROR
7529 037566 104025          ERROR   25 ;MSG B1 ERROR
7530 037570 104401 056673          TYPE   ,MSG21 ;ABORTING BALANCE OF TESTS
7531 037574 000137 042644          JMP     $EOP
7532 037600 104043          67$:   ERROR   43 ;BAD SECTOR NOT LISTED IN TABLE
7533 037602          68$:
7534
7535 037602 012737 010340 005404  MOV      #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7536 037610 005037 005406          CLR      E.B0 ;EXPECTED MSG B0
7537 037614 012737 001720 005410  MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7538 037622 012737 000001 005412  MOV      #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7539 037630 005037 005414          CLR      E.A2 ;EXPECTED MSG A2
7540 037634 012737 000002 005416  MOV      #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7541 037642 012737 000003 005422  MOV      #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7542
7543 037650 004737 044274          JSR      PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7544 037654 000003          .WORD   T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7545 037656 104052          ERROR   52 ;MSG A0 ERROR AFTER WRITE DATA CMD
7546 037660 104023          ERROR   23 ;MSH B0 ERROR
7547 037662 104053          ERROR   53 ;MSG A1 ERROR
7548 037664 104025          ERROR   25 ;MSG B1 ERROR
7549 037666 005000          CLR      R0 ;CLEAR TO DO PORT A FIRST
7550
7551 037670 005700          2$:   TST     R0 ;SEE IF DOING PORT 'A'
7552 037672 001056          BNE    3$ ;BR IF NO
7553 037674 013765 001222 000010  MOV      $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7554 037702 012737 000001 005464  MOV      #1,UNITB
7555 037710 063765 005464 000010  ADD     UNITB,RKCS2(R5)
7556 037716 112737 000102 056644  MOVB   #'B,MSG19A
7557 037724 062765 000010 000010  ADD     #RLS,RKCS2(R5) ;RELEASE PORT B
7558 037732 012737 000001 005314  MOV     #SELDRV,HCS1
7559 037740 004737 043372          JSR     PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7560 037744 104117          ERROR   117 ;NO RDY AFTER SEL DRV CMD
7561
7562 037746 013765 001222 000010  MOV     $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7563 037754 012737 000000 005464  MOV     #0,UNITB
7564 037762 063765 005464 000010  ADD     UNITB,RKCS2(R5)
7565 037770 112737 000101 056644  MOVB   #'A,MSG19A
7566 037776 012737 000001 005314  MOV     #SELDRV,HCS1
7567 040004 004737 043372          JSR     PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7568 040010 104117          ERROR   117 ;NO RKY AFTER SEL DRV CMD
7569
7570 040012 032737 000040 005342  BIT     #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT A
7571 040020 001001          BNE    69$ ;BR IF YES

```

```

7572 040022 104071          ERROR 71          ;PORT A NOT AVAIL AFTER PORT B RLS
7573 040024          69$:          JMP 7$          ;GO & WRITE CHECK THRU PORT 'A'
7574 040024 000137 040246          CMP R0,#1          ;SEE IF DOING PORT 'B'
7575          3$:          BNE 4$          ;BR IF NO
7576 040030 020027 000001          MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7577 040034 001056          MOV #0,UNITB
7578 040036 013765 001222 000010          ADD UNITB,RKCS2(R5)
7579 040044 012737 000000 005464          MOVB #'A,MSG19A
7580 040052 063765 005464 000010          ADD #RLS,RKCS2(R5) ;RELEASE PORT A
7581 040060 112737 000101 056644          MOV #SELDRV,HCS1
7582 040066 062765 000010 000010          JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7583 040074 012737 000001 005314          ERROR 117 ;NO RDY AFTER SEL DRV CMD
7584 040102 004737 043372
7585 040106 104117
7586
7587 040110 013765 001222 000010          MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7588 040116 012737 000001 005464          MOV #1,UNITB
7589 040124 063765 005464 000010          ADD UNITB,RKCS2(R5)
7590 040132 112737 000102 056644          MOVB #'B,MSG19A
7591 040140 012737 000001 005314          MOV #SELDRV,HCS1
7592 040146 004737 043372          JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7593 040152 104117          ERROR 117 ;NO RKY AFTER SEL DRV CMD
7594
7595 040154 032737 000040 005342          BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT B
7596 040162 001001          BNE 70$
7597 040164 104071          ERROR 71 ;BR IF YES
7598 040166          70$:          ;PORT B NOT AVAIL AFTER PORT A RLS
7599 040166 000137 040246          JMP 7$          ;GO WRITE CHECK THRU PORT 'B'
7600
7601 040172 020027 000002          4$:          CMP R0,#2          ;SEE IF DID WRT CHK THRU PORT A & B
7602 040176 001402          BEQ 5$          ;BR IF YES
7603 040200 000137 040562          JMP 8$          ;ELSE R0=3, EXIT TEST
7604
7605 040204 023727 001436 001434 5$:          CMP DATA01,#DATA0 ;SEE IF JUST WROTE 0'S ON CYL 0
7606 040212 001010          BNE 6$          ;BR IF NO
7607 040214 012737 000012 001360          MOV #10,CYLADD ;ELSE WRITE 1'S ON CYL 10
7608 040222 012737 001440 001436          MOV #DATA1,DATA01
7609 040230 000137 037342          JMP 1$          ;GO DO IT
7610
7611 040234 005037 001360          6$:          CLR CYLADD ;RECHK CYL 0 FOR 0'S
7612 040240 012737 001434 001436          MOV #DATA0,DATA01
7613
7614 040246 052765 000020 000010 7$:          BIS #BAI,RKCS2(R5) ;SETUP TO WRITE CHECK
7615 040254 013765 001436 000004          MOV DATA01,RKBA(R5)
7616 040262 013765 001360 000020          MOV CYLADD,RKDC(R5)
7617 040270 012765 177400 000002          MOV #-256,RKWC(R5)
7618 040276 013765 001374 000006          MOV SECTOR,RKDA(R5)
7619 040304 013737 001374 001370          MOV SECTOR,SEC ;COPY FOR FUTURE USE
7620
7621 040312 012737 000031 005314          MOV #<WRTCHK>,HCS1
7622 040320 004737 043430          JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
7623 040324 104015          ERROR 15 ;NO RDY AFTER WRITE CHECK CMD
7624 040326 004737 045146          JSR PC,GSTAT ;GET FRESH STATUS
7625 040332 032737 100000 005314          BIT #CERR,HCS1
7626 040340 001453          BEQ 72$
7627 040342 032737 040000 005316          BIT #WCE,HCS2 ;SEE IF WRITE CHECK ERROR
  
```

```

7628 040350 001410 BEQ 71$
7629 040352 016537 000024 001414 MOV RKDB(R5),WD1 ;ACTUAL WORD FOR PRINTOUT
7630 040360 013737 001434 001416 MOV DATA0,WD2 ;EXPECTED WORD FOR TYPEOUT
7631 040366 104016 ERROR 16 ;WCE AFTER WRITE CMD
7632 040370 000437 BR 72$
7633
7634 040372 104022 71$: ERROR 22 ;CERR AFTER WRITE CHECK CMD
7635
7636 040374 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7637 040402 005037 005406 CLR E.B0 ;EXPECTED MSG B0
7638 040406 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7639 040414 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7640 040422 005037 005414 CLR E.A2 ;EXPECTED MSG A2
7641 040426 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7642 040434 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7643
7644 040442 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7645 040446 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7646 040450 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
7647 040452 104031 ERROR 31 ;MSH B0 ERROR
7648 040454 104060 ERROR 60 ;MSG A1 ERROR
7649 040456 104032 ERROR 32 ;MSG B1 ERROR
7650 040460 104401 056673 TYPE ,MSG21 ;ABORTING BALANCE OF TESTS
7651 040464 000137 042644 JMP $EOP
7652
7653 040470 72$:
7654
7655 040470 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7656 040476 005037 005406 CLR E.B0 ;EXPECTED MSG B0
7657 040502 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7658 040510 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7659 040516 005037 005414 CLR E.A2 ;EXPECTED MSG A2
7660 040522 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7661 040530 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7662
7663 040536 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7664 040542 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7665 040544 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
7666 040546 104031 ERROR 31 ;MSH B0 ERROR
7667 040550 104060 ERROR 60 ;MSG A1 ERROR
7668 040552 104032 ERROR 32 ;MSG B1 ERROR
7669
7670 040554 005200 INC R0
7671 040556 000137 037670 JMP 2$
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
  
```

```

:*****
:*TEST 40 ALTERNATING SEEK INTERACTION TEST
:*
:* THIS TEST VERIFIES THAT THERE ARE NO TIMING INTERACTION PROBLEMS
:* BETWEEN SEEKS FROM BOTH PORTS.
:*
:* A. PORT 'A' SEIZES THE DRIVE & SEEKS TO CYLINDER 0 & RELEASES
:* THE DRIVE AFTER 'ATTN' IS RECEIVED.
:*
:* THE PROGRAM VERIFIES THAT UNTIL ATTN IS RECEIVED,
  
```

7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699 040562 000004
7700 040564 012737 000001 001174
7701 040572 012706 001100
7702
7703 040576 004737 045534
7704 040602 104024
7705
7706 040604 013765 001222 000010
7707 040612 012737 000001 005464
7708 040620 063765 005464 000010
7709 040626 112737 000102 056644
7710 040634 062765 000010 000010
7711 040642 012737 000001 005314
7712 040650 004737 043372
7713 040654 104117
7714
7715 040656 013765 001222 000010
7716 040664 012737 000000 005464
7717 040672 063765 005464 000010
7718 040700 112737 000101 056644
7719 040706 012737 000001 005314
7720 040714 004737 043372
7721 040720 104117
7722
7723 040722 032737 000040 005342
7724 040730 001001
7725 040732 104071
7726 040734
7727
7728 040734 005037 001414
7729 040740 013737 012770 001416
7730
7731 040746 012765 100000 000000
7732 040754 013765 001222 000010
7733 040762 063765 005464 000010
7734 040770 013765 001414 000020
7735 040776 012737 000017 005314
7736 041004 004737 043372
7737 041010 104131
7738
7739 041012 013737 001412 005352

:* PORT 'B' SEES CONTROLLER ERROR & DRIVE NOT AVAILABLE.
:*
:* B. PORT 'B' SEIZES THE DRIVE & SEEKS TO THE LAST CYL
:* & RELEASES THE DRIVE AFTER 'ATTN' IS RECEIVED
:*
:* THE PROGRAM VERIFIES THAT UNTIL ATTN IS RECEIVED,
:* PORT 'A' SEES CONTROLLER ERROR & DRIVE NOT AVAILABLE.
:*
:* C. THE ABOVE IS REPEATED FOR A PATTERN OF CONVERGING SEEKS
:* TOWARD THE CENTER OF THE CARTRIDGE.
:*
:* D. THE PROGRAM VERIFIES MULTIPLE ATTENTIONS OR ERRORS
:* DO NOT OCCUR AS A RESULT OF TIMING PROBLEMS.
:*
:*****
TST40: SCOPE
MOV #1,\$TIMES ;:DO 1 ITERATION
MOV #STACK,SP
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
MOV \$UNIT,RKCS2(R5) ;SETUP FOR PORT B
MOV #1,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'B,MSG19A
ADD #RLS,RKCS2(R5) ;RELEASE PORT B
MOV #SELDRV,HCS1
JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;NO RDY AFTER SEL DRV CMD
MOV \$UNIT,RKCS2(R5) ;SETUP FOR PORT A
MOV #0,UNITB
ADD UNITB,RKCS2(R5)
MOVB #'A,MSG19A
MOV #SELDRV,HCS1
JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
ERROR 117 ;NO RKY AFTER SEL DRV CMD
BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT A
BNE 64\$;BR IF YES
ERROR 71 ;PORT A NOT AVAIL AFTER PORT B RLS
64\$:
CLR WD1 ;SETUP CONVERGING LIMITS
MOV LC,WD2
10\$:
MOV #CCLR,RKCS1(R5)
MOV \$UNIT,RKCS2(R5)
ADD UNITB,RKCS2(R5)
MOV WD1,RKDC(R5)
MOV #SEEK,HCS1
JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
ERROR 131 ;NO RDY AFTER SEEK CMD-PORT A
MOV T50000,TEMP1

```

7740 041020 013704 001222      1$:  MOV  $UNIT,R4
7741 041024 136465 005304 000017  BITB  ATTN(R4),RKASOF+1(R5) ;TEST FOR ATTN ON PORT A
7742 041032 001032          BNE   3$ ;BR IF THERE
7743 041034 012737 000001 005464  MOV  #1,UNITB ;ELSE VERIFY PORT B NOT AVAIL
7744 041042 112737 000102 056644  MOVB #'B,MSG19A
7745 041050 004737 044172  JSR  PC,DRAV
7746 041054 000403          BR   2$
7747 041056 104103          ERROR 103 ;PORT B AVAIL BEFORE TMO OR RELEASE
7748 041060 000137 042644  JMP  11$
7749
7750 041064 012765 100000 000000 2$:  MOV  #CCLR,RKCS1(R5)
7751 041072 005337 005352          DEC  TEMP1
7752 041076 001350          BNE  1$ ;SEE IF PORT A HAS ATTN
7753 041100 005037 005464          CLR  UNITB
7754 041104 112737 000101 056644  MOVB #'A,MSG19A
7755 041112 104156          ERROR 156 ;NO ATTN ON PORT A AFTER SEEK
7756 041114 000137 042644  JMP  11$ ;EXIT TEST
7757
7758 041120 005037 005464          CLR  UNITB ;SETUP FOR PORT A
7759 041124 112737 000101 056644  MOVB #'A,MSG19A
7760 041132 004737 044172  JSR  PC,DRAV
7761 041136 104157          ERROR 157 ;PORT A NOT AVAIL AFTER SEEK
7762
7763 041140 032737 100000 005314  BIT  #CERR,HCS1
7764 041146 001401          BEQ  4$
7765 041150 104210          ERROR 210 ;CERR AFTER SEEK CMD ON PORT A
7766
7767 041152          4$:
7768
7769 041152 012737 050340 005404  MOV  #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7770 041160 005037 005406          CLR  E.B0 ;EXPECTED MSG B0
7771 041164 012737 001720 005410  MOV  #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7772 041172 012737 000001 005412  MOV  #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7773 041200 005037 005414          CLR  E.A2 ;EXPECTED MSG A2
7774 041204 012737 000002 005416  MOV  #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7775 041212 012737 000003 005422  MOV  #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7776
7777 041220 004737 044274          JSR  PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7778 041224 000000          .WORD 0!0!0 ;8 MSGS SPECIFIED HERE
7779 041226 104161          ERROR 161 ;MSG A0 ERROR AFTER SEEK CMD
7780 041230 104162          ERROR 162 ;MSG B0 ERROR
7781 041232 104163          ERROR 163 ;MSG A1 ERROR
7782 041234 104164          ERROR 164 ;MSG B1 ERROR
7783 041236 013737 001414 001346  MOV  WD1,TOCYL ;SETUP TO SEE IF ON CORRECT CYL
7784 041244 013765 001414 000020  MOV  WD1,RKDC(R5)
7785 041252 013765 001222 000010  MOV  $UNIT,RKCS2(R5)
7786 041260 063765 005464 000010  ADD  UNITB,RKCS2(R5)
7787
7788
7789 041266 012700 001666          MOV  #RHTAB,R0
7790 041272 012737 000025 005314  MOV  #<RDHEAD>,HCS1
7791 041300 004737 043430          JSR  PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
7792 041304 104171          ERROR 171 ;NO RDY AFTER READ HEADER CMD
7793 041306 032737 100000 005314  BIT  #CERR,HCS1
7794 041314 001401          BEQ  66$
7795 041316 104174          ERROR 174 ;CERR AFTER READ HEADER CMD
  
```

```

7796
7797 041320 016520 000024      66$:  MOV   RKDB(R5),(R0)+ ;1'ST WORD FROM SILO TO RHTAB
7798 041324 016520 000024      MOV   RKDB(R5),(R0)+ ;2'ND WORD
7799 041330 016520 000024      MOV   RKDB(R5),(R0)+ ;3'RD WORD
7800
7801
7802 041334 032765 100000 000010      BIT   #DLT,RKCS2(R5)
7803 041342 001403                BEQ   67$
7804 041344 004737 045146                JSR   PC,GSTAT
7805 041350 104173                ERROR 173 ;DLT AFTER READ HEADER CMD
7806 041352
7807      67$:
7808 041352 012737 010340 005404      MOV   #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7809 041360 005037 005406      CLR   E.B0 ;EXPECTED MSG B0
7810 041364 012737 001720 005410      MOV   #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7811 041372 012737 000001 005412      MOV   #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7812 041400 005037 005414      CLR   E.A2 ;EXPECTED MSG A2
7813 041404 012737 000002 005416      MOV   #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7814 041412 012737 000003 005422      MOV   #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7815
7816 041420 004737 044274                JSR   PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7817 041424 000003                .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7818 041426 104301                ERROR 301 ;MSG A0 ERROR AFTER READ HEADER CMD
7819 041430 104271                ERROR 271 ;MSH B0 ERROR
7820 041432 104302                ERROR 302 ;MSG A1 ERROR
7821 041434 104272                ERROR 272 ;MSG B1 ERROR
7822
7823 041436 023737 001666 001346      CMP   RHTAB,TOCYL ;CHECK WORD 0 ONLY, CYL#
7824 041444 001401                BEQ   65$ ;BR IF SAME
7825 041446 104051                ERROR 51 ;WRONG CYL# ON HEADER
7826 041450
7827      65$:
7828 041450 004737 045534                JSR   PC,SUBCLR
7829 041454 104024                ERROR 24 ;CERR AFTER SCLR
7830
7831 041456 013765 001222 000010      MOV   $UNIT,RKCS2(R5) ;SETUP FOR PORT A
7832 041464 012737 000000 005464      MOV   #0,UNITB
7833 041472 063765 005464 000010      ADD   UNITB,RKCS2(R5)
7834 041500 112737 000101 056644      MOV   #A,MSG19A
7835 041506 062765 000010 000010      ADD   #RLS,RKCS2(R5) ;RELEASE PORT A
7836 041514 012737 000001 005314      MOV   #SELDRV,HCS1
7837 041522 004737 043372                JSR   PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7838 041526 104117                ERROR 117 ;NO RDY AFTER SEL DRV CMD
7839
7840 041530 013765 001222 000010      MOV   $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7841 041536 012737 000001 005464      MOV   #1,UNITB
7842 041544 063765 005464 000010      ADD   UNITB,RKCS2(R5)
7843 041552 112737 000102 056644      MOV   #B,MSG19A
7844 041560 012737 000001 005314      MOV   #SELDRV,HCS1
7845 041566 004737 043372                JSR   PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7846 041572 104117                ERROR 117 ;NO RKY AFTER SEL DRV CMD
7847
7848 041574 032737 000040 005342      BIT   #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT B
7849 041602 001001                BNE  68$ ;BR IF YES
7850 041604 104071                ERROR 71 ;PORT B NOT AVAIL AFTER PORT A RLS
7851 041606
      68$:

```



```

7852 041606 012765 100000 000000 MOV #CCLR,RKCS1(R5)
7853 041614 013765 001222 000010 MOV $UNIT,RKCS2(R5)
7854 041622 063765 005464 000010 ADD UNITB,RKCS2(R5)
7855 041630 013765 001416 000020 MOV WD2,RKDC(R5)
7856 041636 012737 000017 005314 MOV #SEEK,HCS1
7857 041644 0C4737 043372 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
7858 041650 104131 ERROR 131 ;NO READY AFTER SEEK CMD-PORT B
7859
7860 041652 013737 001412 005352 MOV T50000,TEMP1
7861 041660 013704 001222 5$: MOV $UNIT,R4
7862 041664 005204 INC R4
7863 041666 136465 005304 000017 BITB ATTN(R4),RKASOF+1(R5) ;TEST FOR ATTN ON PORT B
7864 041674 001032 BNE 7$ ;BR IF THERE
7865 041676 005037 005464 CLR UNITB ;ELSE VERIFY PORT A NOT AVAIL
7866 041702 112737 000101 056644 MOVB #'A,MSG19A
7867 041710 004737 044172 JSR PC,DRAV
7868 041714 000403 BR 6$
7869 041716 104103 ERROR 103 ;PORT A AVAIL BEFORE TMO OR RELEASE
7870 041720 000137 042644 JMP 11$
7871
7872 041724 012765 100000 000000 6$: MOV #CCLR,RKCS1(R5)
7873 041732 005337 005352 DEC TEMP1
7874 041736 001350 BNE 5$ ;SEE IF PORT B HAS ATTN
7875 041740 012737 000001 005464 MOV #1,UNITB
7876 041746 112737 000102 056644 MOVB #'B,MSG19A
7877 041754 104156 ERROR 156 ;NO ATTN ON PORT B AFTER SEEK
7878 041756 000137 042644 JMP 11$ ;EXIT
7879
7880 041762 012737 000001 005464 7$: MOV #1,UNITB ;SETUP FOR PORT B
7881 041770 112737 000102 056644 MOVB #'B,MSG19A
7882 041776 004737 044172 JSR PC,DRAV
7883 042002 104157 ERROR 157 ;PORT B NOT AVAIL AFTER SEEK
7884 042004 032737 100000 005314 BIT #CERR,HCS1
7885 042012 001401 BEQ 8$
7886 042014 104210 ERROR 210 ;CERR AFTER SEEK CMD ON PORT B
7887
7888 042016 8$:
7889
7890 042016 012737 050340 005404 MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7891 042024 005037 005406 CLR E.B0 ;EXPECTED MSG B0
7892 042030 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7893 042036 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7894 042044 005037 005414 CLR E.A2 ;EXPECTED MSG A2
7895 042050 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7896 042056 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7897
7898 042064 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7899 042070 000000 .WORD 0!0!0 ;8 MSGS SPECIFIED HERE
7900 042072 104161 ERROR 161 ;MSG A0 ERROR AFTER SEEK CMD
7901 042074 104162 ERROR 162 ;MSH B0 ERROR
7902 042076 104163 ERROR 163 ;MSG A1 ERROR
7903 042100 104164 ERROR 164 ;MSG B1 ERROR
7904 042102 013737 001416 001346 MOV WD2,TOCYL ;SETUP TO SEE IF ON CORRECT CYL
7905 042110 013765 001416 000020 MOV WD2,RKDC(R5)
7906 042116 013765 001222 000010 MOV $UNIT,RKCS2(R5)
7907 042124 063765 005464 000010 ADD UNITB,RKCS2(R5)

```

7908									
7909									
7910	042132	012700	001666		MOV	#RHTAB,RO			
7911	042136	012737	000025	005314	MOV	#<RDHEAD>,HCS1			
7912	042144	004737	043430		JSR	PC,DATCMD			:DO DATA X FOR CMD & GET CONTR RDY
7913	042150	104171			ERROR	171			:NO RDY AFTER READ HEADER CMD
7914	042152	032737	100000	005314	BIT	#CERR,HCS1			
7915	042160	001401			BEQ	70\$			
7916	042162	104174			ERROR	174			:CERR AFTER READ HEADER CMD
7917									
7918	042164	016520	000024		70\$:	MOV	RKDB(R5),(R0)+		:1'ST WORD FROM SILO TO RHTAB
7919	042170	016520	000024			MOV	RKDB(R5),(R0)+		:2'ND WORD
7920	042174	016520	000024			MOV	RKDB(R5),(R0)+		:3'RD WORD
7921									
7922									
7923	042200	032765	100000	000010	BIT	#DLT,RKCS2(R5)			
7924	042206	001403			BEQ	71\$			
7925	042210	004737	045146		JSR	PC,GSTAT			
7926	042214	104173			ERROR	173			:DLT AFTER READ HEADER CMD
7927	042216				71\$:				
7928									
7929	042216	012737	010340	005404	MOV	#<D.DRA!D.CPIN!D.DRDY!D.VV>,E.A0			:EXPECTED MSG A0
7930	042224	005037	005406		CLR	E.B0			:EXPECTED MSG B0
7931	042230	012737	001720	005410	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1			:EXPECTED A1
7932	042236	012737	000001	005412	MOV	#1,E.B1			:MSG ID FOR EXPECTED MSG B1
7933	042244	005037	005414		CLR	E.A2			:EXPECTED MSG A2
7934	042250	012737	000002	005416	MOV	#2,E.B2			:MSG ID FOR EXPECTED MSG B2
7935	042256	012737	000003	005422	MOV	#3,E.B3			:MSG ID FOR EXPECTED MSG B3
7936									
7937	042264	004737	044274		JSR	PC,CHKMSG			:CHECK MSGS A0, B0, A1, B1
7938	042270	000003			.WORD	T.A2!T.B2!0			:& MSGS SPECIFIED HERE
7939	042272	104301			ERROR	301			:MSG A0 ERROR AFTER READ HEADER CMD
7940	042274	104271			ERROR	271			:MSH B0 ERROR
7941	042276	104302			ERROR	302			:MSG A1 ERROR
7942	042300	104272			ERROR	272			:MSG B1 ERROR
7943									
7944	042302	023737	001666	001346	CMP	RHTAB,TOCYL			:CHECK WORD 0 ONLY, CYL#
7945	042310	001401			BEQ	69\$:BR IF SAME
7946	042312	104051			ERROR	51			:WRONG CYL# ON HEADER
7947	042314				69\$:				
7948									
7949	042314	004737	045534		JSR	PC,SUBCLR			
7950	042320	104024			ERROR	24			:CERR AFTER SCLR
7951									
7952	042322	005237	001414		INC	WD1			:CONVERGE THE CYLINDERS
7953	042326	005337	001416		DEC	WD2			
7954	042332	023737	001414	001416	CMP	WD1,WD2			:SEE IF CONVERGED TOGETHER
7955	042340	001063			BNE	9\$:BR IF NO & REPEAT
7956	042342	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)			
7957	042350	063765	005464	000010	ADD	UNITB,RKCS2(R5)			
7958									
7959	042356	012737	000017	005314	MOV	#SEEK,HCS1			
7960	042364	004737	043372		JSR	PC,DOCMD			:DO SEEK CMD & GET CONTR READY
7961	042370	104131			ERROR	131			:NO RDY AFTER SEEK CMD
7962									
7963	042372	013737	001412	005352	MOV	T5000,TEMP1			:SETUP TIMEOUT

K 12
PAGE 154

CZR6GCO RK611 DU PORT LGC MACY11 30(1046) 04-JAN-82 12:59
 CZR6GC.P11 04-JAN-82 12:39 T40 ALTERNATING SEEK INTERACTION TEST SEQ 0153

```

7964 042400 004737 044106 JSR PC,FATT2 ;FIND ATTN
7965 042404 104132 ERROR 132 ;NO ATTN AFTER SEEK CMD
7966
7967 042406 032737 100000 005314 BIT #CERR,HCS1
7968 042414 001401 BEQ 72$
7969 042416 104210 ERROR 210 ;CERR AFTER SEEK CMD
7970
7971 042420 72$:
7972
7973 042420 012737 050340 005404 MOV #<D.DSC!D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
7974 042426 005037 005406 CLR E.B0 ;EXPECTED MSG B0
7975 042432 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7976 042440 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7977 042446 005037 005414 CLR E.A2 ;EXPECTED MSG A2
7978 042452 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7979 042460 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7980
7981 042466 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
7982 042472 000000 .WORD 0!0!0 ;8 MSGS SPECIFIED HERE
7983 042474 104161 ERROR 161 ;MSG A0 ERROR AFTER SEEK CMD
7984 042476 104162 ERROR 162 ;MSH B0 ERROR
7985 042500 104163 ERROR 163 ;MSG A1 ERROR
7986 042502 104164 ERROR 164 ;MSG B1 ERROR
7987
7988 042504 000137 042644 JMP 11$ ;EXIT TEST
7989 042510 9$:
7990 042510 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT B
7991 042516 012737 000001 005464 MOV #1,UNITB
7992 042524 063765 005464 000010 ADD UNITB,RKCS2(R5)
7993 042532 112737 000102 056644 MOVB #'B,MSG19A
7994 042540 062765 000010 000010 ADD #RLS,RKCS2(R5) ;RELEASE PORT B
7995 042546 012737 000001 005314 MOV #SELDRV,HCS1
7996 042554 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
7997 042560 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
7998
7999 042562 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;SETUP FOR PORT A
8000 042570 012737 000000 005464 MOV #0,UNITB
8001 042576 063765 005464 000010 ADD UNITB,RKCS2(R5)
8002 042604 112737 000101 056644 MOVB #'A,MSG19A
8003 042612 012737 000001 005314 MOV #SELDRV,HCS1
8004 042620 004737 043372 JSR PC,DOCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
8005 042624 104117 ERROR 117 ;NO RKY AFTER SEL DRV CMD
8006
8007 042626 032737 000040 005342 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAIL ON PORT A
8008 042634 001001 BNE 73$ ;BR IF YES
8009 042636 104071 ERROR 71 ;PORT A NOT AVAIL AFTER PORT B RLS
8010 042640 73$:
8011 042640 000137 040746 JMP 10$
8012
8013 042644 11$:
8014
8015
8016
8017
8018

```

```
8019 .SBTTL END OF PASS ROUTINE
8020
8021 ::*****
8022 ::*INCREMENT THE PASS NUMBER ($PASS)
8023 ::*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
8024 ::*IF THERES A MONITOR GO TO IT
8025 ::*IF THERE ISN'T JUMP TO ST5
8026
8027 042644 $EOP:
8028
8029 042644 000004 SCOPE
8030 042646 005037 001176 CLR $ESCAPE
8031 042652 012737 000001 001174 MOV #1,$TIMES
8032 042660 012706 001100 MOV #STACK,SP
8033 042664 005237 001220 INC $DEVCT ;INCR COUNT FOR # DRIVES CHECKED
8034 042670 023737 005434 001220 CMP DRIVS,$DEVCT ;ARE ALL DRIVES PRESENT TESTED?
8035 042676 001403 BEQ $EOP1+2 ;BR IF YES
8036 042700 000137 012472 JMP NUDRV ;ELSE TEST NEXT DRIVE PRESENT
8037 042704 000004 $EOP1: SCOPE
8038 042706 005037 001102 CLR $STSTM ;:ZERO THE TEST NUMBER
8039 042712 005037 001174 CLR $TIMES ;:ZERO THE NUMBER OF ITERATIONS
8040 042716 005237 001216 INC $PASS ;:INCREMENT THE PASS NUMBER
8041 042722 042737 100000 001216 BIC #100000,$PASS ;:DON'T ALLOW A NEG. NUMBER
8042 042730 005327 DEC (PC)+ ;:LOOP?
8043 042732 000001 $EOPCT: .WORD 1
8044 042734 003022 BGT $DOAGN ;:YES
8045 042736 012737 MOV (PC)+,@(PC)+ ;:RESTORE COUNTER
8046 042740 000001 $ENDCT: .WORD 1
8047 042742 042732 $EOPCT
8048 042744 104401 043011 TYPE $SENDMG ;:TYPE 'END PASS #'
8049 042750 013746 001216 MOV $PASS,-(SP) ;:SAVE $PASS FOR TYPEOUT
8050 042754 104405 TYPDS ;:GO TYPE--DECIMAL ASCII WITH SIGN
8051 042756 104401 043006 TYPE $ENULL ;:TYPE A NULL CHARACTER
8052 042762 013700 000042 $GET42: MOV @#42,R0 ;:GET MONITOR ADDRESS
8053 042766 001405 BEQ $DOAGN ;:BRANCH IF NO MONITOR
8054 042770 000005 RESET ;:CLEAR THE WORLD
8055 042772 004710 $ENDAD: JSR PC,(R0) ;:GO TO MONITOR
8056 042774 000240 NOP ;:SAVE ROOM
8057 042776 000240 NOP ;:FOR
8058 043000 000240 NOP ;:ACT11
8059 043002 $DOAGN:
8060 043002 000137 JMP @(PC)+ ;:RETURN
8061 043004 010646 $RTNAD: .WORD ST5
8062 043006 377 377 000 $ENULL: .BYTE -1,-1,0 ;:NULL CHARACTER STRING
8063 043011 015 042412 042116 $ENDMG: .ASCIZ <15><12>/END PASS #/
8064 043016 050040 051501 020123
8065 043024 000043
```

```
8066      .SBTTL SUBROUTINES
8067
8068      ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU UNITB
8069      ;
8070
8071      043026 012700 005424 CLRFLG: MOV #DDUMP,R0
8072      043032 012701 177757      MOV #-17.,R1
8073      043036 005020 1$: CLR (R0)+
8074      043040 005201      INC R1
8075      043042 001375      BNE 1$
8076      043044 000207      RTS PC
8077
8078
8079      ;
8080      ;TYPE PROGRAM ID IF FTITLE=0
8081      ;
8082      043046 005737 001340 TITLE: TST FTITLE
8083      043052 001024      BNE 1$
8084      043054 005237 001340      INC FTITLE
8085      043060 104401 054652      TYPE ,MSG1 ;PROGRAM ID
8086      .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
8087      043064 005737 000042 TST @#42 ;:ARE WE RUNNING UNDER XXDP/ACT?
8088      043070 001012      BNE 64$ ;:BRANCH IF YES
8089      043072 123727 001230 000001 CMPB $ENV,#1 ;:ARE WE RUNNING UNDER APT?
8090      043100 001406      BEQ 64$ ;:BRANCH IF YES
8091      043102 023727 001140 000176 CMP SWR,#SWREG ;:SOFTWARE SWITCH REG SELECTED?
8092      043110 001005      BNE 65$ ;:BRANCH IF NO
8093      043112 104406      GTSWR ;:GET SOFT-SWR SETTINGS
8094      043114 000403      BR 65$
8095      043116 112737 000001 001134 64$: MOVB #1,$AUTOB ;:SET AUTO-MODE INDICATOR
8096      043124      65$:
8097      043124 000207 1$: RTS PC
8098
8099
8100      ;
8101      ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
8102      ;DRIVS, DRIVO-DRIV7 REGISTERS APPROPRIATELY
8103      ;ONLY EVEN NUMBERS ALLOWED (0,2,4,6)
8104      ;
8105      043126 104411 GDRVS: RDLIN
8106      043130 012600      MOV (SP)+,R0 ;GET STARTING ADDR OF ASCII STRING
8107      043132 012701 177774      MOV #-4,R1 ;SET UP COUNT
8108      043136 112002 1$: MOVB (R0)+,R2 ;GET ASCII CHAR
8109      043140 042702 177400      BIC #177400,R2 ;MASK HI BYTE
8110      043144 012703 005436      MOV #DRIVO,R3 ;DRIVE FLAG ADDR
8111      043150 012704 000060      MOV #60,R4
8112
8113      2$: CMP R4,R2 ;WAS TYPED CHAR 0 THRU 6?
8114      043156 001415      BEQ 3$ ;BRANCH IF YES
8115      043160 005723      TST (R3)+ ;NO, INCREMENT DR FLAG ADDR
8116      043162 005204      INC R4
8117      043164 020427 000067      CMP R4,#67
8118      043170 001371      BNE 2$ ;S/B 0-6 OR TERMINATOR
8119      043172 005702      TST R2 ;TERMINATOR=0
8120      043174 001025      BNE 4$
8121      043176 020127 177774      CMP R1,#-4
```

```
8122 043202 001431          BEQ      6$          ;DEFAULT ALL DRIVES
8123 043204 005037 005462   7$:     CLR      SIZFLG ;BYPASS TEST 1 (SIZING)
8124 043210 000207          RTS      PC          ;FOUND TERMINATOR, EXIT
8125
8126 043212 032704 000001   3$:     BIT      #BIT0,R4 ;SEE IF ODD #
8127 043216 001014          BNE      4$
8128 043220 005213          INC      @R3        ;SET UP FLAG FOR THE DRIVE
8129 043222 005237 005434   INC      DRIVS      ;INCREMENT TOTAL # DRIVES TO BE TESTED
8130 043226 112002          MOV      (R0)+,R2   ;GET NEXT ASCII CHAR.
8131 043230 042702 177400   BIC      #177400,R2 ;MASK
8132 043234 022702 000054   CMP      #54,R2    ;IS IT A COMMA?
8133 043240 001407          BEQ      5$        ;YES, GO TO NEXT WORD.
8134 043242 005702          TST      R2        ;NO, IS IT A TERMINATOR?
8135 043244 001001          BNE      4$        ;IF NOT, SOMETHING WRONG.
8136 043246 000756          BR       7$        ;FOUND TERMINATOR, EXIT
8137
8138 043250 104401 057122   4$:     TYPE     ,EM1  ;ONLY 0,2,4,6 ALLOWED.
8139 043254 000137 010054   JMP      PRGSRT    ;START ALL OVER
8140
8141 043260 005201          INC      R1        ;S/B NO MORE THAN 4 DIFF
8142 043262 001325          BNE      1$        ;DRIVES TYPED IN.
8143 043264 000771          BR       4$        ;IF NORE, HAVE ERROR.
8144
8145 043266 005237 005462   6$:     INC      SIZFLG ;DO TEST 1 (SIZING)
8146 043272 000207          RTS      PC          ;EXIT.
8147
8148
8149
8150
8151
8152
8153
8154
8155
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169
8170
8171
8172
8173
8174
8175
8176
8177
```

ROUTINE TO INPUT RKBAS OR DEFAULT.

```
GBA:  RDOCT
      MOV      (SP)+,R0 ;GET LOW ORDER FROM STACK
      TST      R0
      BEQ      1$      ;BRANCH IF DEFAULT.
      MOV      R0,$BASE
      RTS      PC
1$:   MOV      #177440,$BASE ;DEFAULT VALUE
      RTS      PC
```

ROUTINE TO INPUT RKVEC OR DEFAULT

```
GINT: RDOCT
      MOV      (SP)+,R0 ;GET LOW ORDER FROM STACK
      TST      R0
      BEQ      1$      ;BRANCH IF DEFAULT
      MOV      R0,RKVEC
      JSR      PC,SETINT
2$:   JSR      PC,SETINT
      RTS      PC
1$:   MOV      #210,RKVEC ;DEFAULT VALUE
      BR       2$
```

ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY

```
8178
8179 043354 013700 001314
8180 043360 012720 050066
8181 043364 013710 001316
8182 043370 000207
8183
8184
8185
8186
8187
8188 043372 053737 001170 005314
8189 043400 013765 005314 000000
8190 043406 013737 001400 005352
8191 043414 004737 043466
8192 043420 000207
8193 043422 062716 000002
8194 043426 000207
8195
8196
8197
8198
8199 043430 053737 001170 005314
8200 043436 013765 005314 000000
8201 043444 013737 001412 005352
8202 043452 004737 043466
8203 043456 000207
8204 043460 062716 000002
8205 043464 000207
8206
8207
8208
8209
8210
8211
8212
8213
8214 043466 032765 000200 000000
8215 043474 001010
8216 043476 005337 005352
8217 043502 001371
8218 043504 004737 043622
8219 043510 004737 045064
8220 043514 000207
8221 043516 062716 000002
8222 043522 004737 043622
8223 043526 004737 045064
8224 043532 000207
8225
8226
8227
8228 043534 032765 000200 000000
8229 043542 001014
8230 043544 005337 005352
8231 043550 001371
8232 043552 016537 000034 005342
8233 043560 016537 000036 005344
```

SETINT: MOV RKVEC,R0
MOV #INTER,(R0)+ ;INTER ADDR TO RKVEC
MOV RKPRI,(R0) ;PR5 TO RKVEC+2
RTS PC

;
;THIS ROUTINE SETS CDT IN RKCS1 IF DRIVE UNDER TEST IS AN RK07.
;ENTER WITH COMMAND IN HCS1
;
DOCMD: BIS \$TMP4,HCS1 ;ADD CDT IF RK07
MOV HCS1,RKCS1(R5) ;DO COMMAND
MOV T10,TEMP1
JSR PC,FRDY ;FIND CONTR READY
RTS PC ;SET HERE IF NOT RDY
ADD #2,(SP) ;ELSE SKIP OVER ERROR
RTS PC

;
;THIS ROUTINE IS SIMILAR TO THE ABOVE BUT IS USED FOR DATA TRANSFERS
;& REQUIRES A LONGER TIMEOUT
;
DATCMD: BIS \$TMP4,HCS1 ;ADD CDT IF RK07
MOV HCS1,RKCS1(R5) ;DO CMD
MOV T50000,TEMP1
JSR PC,FRDY ;FIND CONTR RDY
RTS PC
ADD #2,(SP)
RTS PC

;
;ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
;ENTER WITH A COUNT IN TEMP1
;RETURN IF RDY NOT PRESENT (ERROR CONDITION)
;RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
;
FRDY: BIT #RDY,RKCS1(R5)
BNE 1\$
DEC TEMP1
BNE FRDY
JSR PC,HOLD ;STORE ALL RK611 REGS IN HOLDING REGS.
JSR PC,CKCERR ;CHECK FOR SPECIAL CERR
RTS PC ;NO RDY, EXIT
1\$: ADD #2,(SP) ;SKIP OVER ERROR
JSR PC,HOLD
JSR PC,CKCERR ;CHECK FOR SPECIAL CERR
RTS PC

;
;ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
;
FRDY1: BIT #RDY,RKCS1(R5)
BNE 1\$
DEC TEMP1
BNE FRDY1
MOV RKMR2(R5),HMR2
MOV RKMR3(R5),HMR3

```
8234 043566 004737 045064 JSR PC,CKCERR ;CHECK FOR SPECIAL CERR CONDITIONS
8235 043572 000207 RTS PC ;NO RDY, EXIT
8236 043574 062716 000002 1$: ADD #2,(SP) ;SKIP OVER ERROR
8237 043600 016537 000034 005342 MOV RKMR2(R5),HMR2
8238 043606 016537 000036 005344 MOV RKMR3(R5),HMR3
8239 043614 004737 045064 JSR PC,CKCERR ;CHECK FOR SPECIAL CERR CONDITIONS
8240 043620 000207 RTS PC
8241
8242
8243 ;STORE ALL RK611 REGISTERS IN HOLDING REGS
8244 ;
8245
8246 043622 016537 000000 005314 HOLD: MOV RKCS1(R5),HCS1
8247 043630 016537 000010 005316 MOV RKCS2(R5),HCS2
8248 043636 016537 000002 005320 MOV RKWC(R5),HWC
8249 043644 016537 000004 005322 MOV RKBA(R5),HBA
8250 043652 016537 000006 005324 MOV RKDA(R5),HDA
8251 043660 016537 000012 005326 MOV RKDS(R5),HDS
8252 043666 016537 000014 005330 MOV RKER(R5),HER
8253 043674 016537 000016 005332 MOV RKASOF(R5),HASOF
8254 043702 016537 000020 005334 MOV RKDC(R5),HDC
8255 043710 016537 000026 005340 MOV RKMR1(R5),HMR1
8256 043716 016537 000034 005342 MOV RKMR2(R5),HMR2
8257 043724 016537 000036 005344 MOV RKMR3(R5),HMR3
8258 043732 016537 000030 005346 MOV RKECPS(R5),HPOS
8259 043740 016537 000032 005350 MOV RKECPT(R5),HPAT
8260 043746 000207 RTS PC
8261
8262
8263 ;ROUTINE TO CHECK FOR CORRECT ATTN
8264 ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
8265 ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
8266 ;
8267 043750 010446 ;STATN: MOV R4,-(SP) ;SAV R4
8268 043752 013704 001222 MOV $UNIT,R4
8269 043756 063704 005464 ADD UNITB,R4 ;ADD 1 IF ON PORT B
8270 043762 136437 005304 005333 BITB ATTN(R4),HASOF+1
8271 043770 001404 BEQ 1$ ;BRANCH IF ATTN NOT PRESENT
8272 043772 012604 MOV (SP)+,R4 ;RESTOR R4
8273 043774 062716 000002 ADD #2,(SP) ;INCR RET ADDR TO JUMP OVER ERROR.
8274 044000 000207 RTS PC
8275 044002 012604 1$: MOV (SP)+,R4 ;RESTOR R4
8276 044004 000207 RTS PC
8277
8278
8279 ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
8280 ;ENTER WITH TIME IN SECONDS IN TEMP2
8281 ;RETURN IF NO ATTN (ERROR CONDITION)
8282 ;RETURN +2 IF ATTN FOUND
8283 ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8284 ;
8285
8286 044006 010446 FATT1: MOV R4,-(SP) ;SAV R4
8287 044010 012737 177777 005352 3$: MOV #-1,TEMP1
8288 044016 013704 001222 MOV $UNIT,R4
8289 044022 063704 005464 ADD UNITB,R4 ;ADD 1 IF ON PORT B
```



```

8290 044026 136465 005304 000017 1$: BITB ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
8291 044034 001014 BNE 2$
8292 044036 005337 005352 DEC TEMP1
8293 044042 001371 BNE 1$
8294 044044 005337 005354 DEC TEMP2
8295 044050 001357 BNE 3$
8296 044052 005065 000026 CLR RKMR1(R5) ;SELECT WORD 0
8297 044056 004737 045146 JSR PC,GSTAT ;GET LATEST STATUS
8298 044062 012604 MOV (SP)+,R4 ;RESTOR R4
8299 044064 000207 RTS PC
8300 044066 005065 000026 2$: CLR RKMR1(R5)
8301 044072 004737 045146 JSR PC,GSTAT ;GET STATUS AFTER ATTN SEEN
8302 044076 012604 MOV (SP)+,R4 ;RESTOR R4
8303 044100 062716 000002 ADD #2,(SP) ;SKIP OVER ERROR
8304 044104 000207 RTS PC

```

```

8305
8306
8307 ;ROUTINE TO FIND ATTN WITHIN 1 SEC
8308 ;ENTER WITH COUNT IN TEMP1
8309 ;RETURN IF NO ATTN (ERROR)
8310 ;RETURN +2 IF ATTN FOUND
8311 ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
8312
8313 ;

```

```

8314 044106 010446 FATT2: MOV R4,-(SP) ;SAV R4
8315 044110 013704 001222 2$: MOV $UNIT,R4
8316 044114 063704 005464 ADD UNITB,R4 ;ADD 1 IF ON PORT B
8317 044120 136465 005304 000017 BITB ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
8318 044126 001011 BNE 1$
8319 044130 005337 005352 DEC TEMP1
8320 044134 001365 BNE 2$
8321 044136 005065 000026 CLR RKMR1(R5) ;SELECT WORD 0
8322 044142 004737 045146 JSR PC,GSTAT ;GET LATEST STATUS.
8323 044146 012604 MOV (SP)+,R4 ;RESTOR R4
8324 044150 000207 RTS PC
8325 044152 005065 000026 1$: CLR RKMR1(R5)
8326 044156 004737 045146 JSR PC,GSTAT
8327 044162 012604 MOV (SP)+,R4 ;RESTOR R4
8328 044164 062716 000002 ADD #2,(SP) ;SKIP OVER ERROR
8329 044170 000207 RTS PC

```

```

8330
8331 ;THIS ROUTINE CHECKS 'DRIVE AVAILABLE' IN MSG A0 AFTER A SELECT COMMAND
8332 ; IF NOT SET, IT DOES A NORMAL RETURN
8333 ; IF SET, IT DOES A RETURN+2
8334 ;

```

```

8335 044172 004737 045146 005342 DRAW: JSR PC,GSTAT
8336 044176 032737 000040 BIT #D.DRA,HMR2 ;SEE IF DRIVE AVAILABLE SET
8337 044204 001402 BEQ 1$ ;BR IF NO & RETURN
8338 044206 062716 000002 ADD #2,(SP) ;ELSE RETURN+2
8339 044212 000207 1$: RTS PC

```

```

8340
8341 ;
8342 ;THIS ROUTINE LOOKS FOR ATTN
8343 ; RETURN+2 IF FOUND BEFORE COUNT=0
8344 ; RETURN IF COUNT=0 (ERROR CONDITION)
8345 ;

```

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 PAGE 161
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0160

```

8346 044214 136465 005304 000017 FATT3: BITB   ATTN(R4),RKASOF+1(R5) ;TEST FOR ATTN
8347 044222 001004          BNE     1$
8348 044224 005737 001366          TST     COUNT           ;SEE IF TIME UP
8349 044230 001371          BNE     FATT3           ;BR IF NO
8350 044232 000207          RTS     PC
8351 044234 062716 000002          1$:    ADD     #2,(SP)       ;JUMP OVER ERROR
8352 044240 000207          RTS     PC
8353
8354
8355          ;ENTER WITH A COUNT IN TEMP1
8356          ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
8357          ;WHEN COUNT IS 0. BASED ON AN 11/05
8358
8359 044242 005737 005352          DLY:    TST     TEMP1       ;5.6 US
8360 044246 001403          BEQ     1$               ;2.5 US
8361 044250 005337 005352          DEC     TEMP1           ;6.8 US
8362 044254 000772          BR     DLY               ;2.5 US
8363 044256 000207          1$:    RTS     PC         ;3.8 US
8364
8365          ;THIS ROUTINE TYPES BYPASSED DRIVE#. ENTER WITH DRIVE# IN R0
8366
8367
8368 044260 104401 056523          BYP:    TYPE    ,MSG14     ;BYPASS DRIVE
8369 044264 010046          MOV     R0,-(SP)        ;:SAVE R0 FOR TYPEOUT
8370
8371          TYPOS
8372          .BYTE 1           ;:TYPE DR#
8373          .BYTE 0           ;:GO TYPE--OCTAL ASCII
8374          .BYTE 0           ;:TYPE 1 DIGIT(S)
8375          RTS     PC         ;:SUPPRESS LEADING ZEROS
8376
8377          ;THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
8378 044274 017637 000000 001460          CHKMSG: MOV     @ (SP),CHKFLG ;PASS MSGS TO BE TESTED
8379 044302 062716 000002          ADD     #2,(SP)         ;BUMP RETURN ADDR TO 1ST ERROR
8380 044306 004737 045210          JSR     PC,GSTAT1       ;GET ALL ACTUAL DRIVE & CONTR STATUS
8381
8382 044312 053737 001222 005404          BIS     $UNIT,E.A0     ;SET UNIT #
8383 044320 063737 005464 005404          ADD     UNITB,E.A0     ;ADD 1 IF ON PORT B
8384 044326 053737 001222 005410          BIS     $UNIT,E.A1
8385 044334 063737 005464 005410          ADD     UNITB,E.A1
8386 044342 053737 001222 005414          BIS     $UNIT,E.A2
8387 044350 063737 005464 005414          ADD     UNITB,E.A2
8388 044356 053737 001222 005420          BIS     $UNIT,E.A3
8389 044364 063737 005464 005420          ADD     UNITB,E.A3
8390 044372 053737 012776 005404          BIS     E.DDT,E.A0     ;SET CDT IF RK07
8391
8392 044400 013746 005352          MOV     TEMP1,-(SP)    ;SAVE TEMP1
8393
8394 044404 013737 005404 005352          MOV     E.A0,TEMP1
8395 044412 004737 047224          JSR     PC,SBPAR       ;GET PARITY FOR MSG A0
8396 044416 013737 005352 005404          MOV     TEMP1,E.A0
8397
8398 044424 013737 005410 005352          MOV     E.A1,TEMP1
8399 044432 004737 047224          JSR     PC,SBPAR       ;GET PARITY FOR MSG A1
8400 044436 013737 005352 005410          MOV     TEMP1,E.A1
8401

```

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 PAGE 162
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0161

```

8402 044444 013737 005414 005352      MOV      E.A2,TEMP1
8403 044452 004737 047224              JSR      PC,SBPAR      ;GET PARITY FOR MSG A2
8404 044456 013737 005352 005414      MOV      TEMP1,E.A2
8405
8406 044464 013737 005406 005352      MOV      E.B0,TEMP1
8407 044472 004737 047224              JSR      PC,SBPAR      ;GET PARITY FOR MSG B0
8408 044476 013737 005352 005406      MOV      TEMP1,E.B0
8409
8410 044504 013737 005412 005352      MOV      E.B1,TEMP1
8411 044512 004737 047224              JSR      PC,SBPAR      ;GET PARITY FOR MSG B1
8412 044516 013737 005352 005412      MOV      TEMP1,E.B1
8413
8414 044524 013737 005416 005352      MOV      E.B2,TEMP1
8415 044532 004737 047224              JSR      PC,SBPAR      ;GET PARITY FOR MSG B2
8416 044536 013737 005352 005416      MOV      TEMP1,E.B2
8417
8418 044544 013737 005422 005352      MOV      E.B3,TEMP1
8419 044552 004737 047224              JSR      PC,SBPAR      ;GET PARITY FOR MSG B3
8420 044556 013737 005352 005422      MOV      TEMP1,E.B3
8421
8422 044564 012637 005352              MOV      (SP)+,TEMP1    ;RESTORE TEMP1
8423 044570 013737 001176 001172      MOV      $ESCAPE,$TMP5 ;SAVE ESCAPE
8424
8425 044576 023737 005364 005404      CMP      H.A0,E.A0      ;TEST MSG A0
8426 044604 001411                    BEQ      2$              ;BR IF OK
8427 044606 012737 044620 001176      MOV      #1,$ESCAPE     ;ELSE SETUP ESCAPE
8428 044614 011646                    MOV      (SP),-(SP)     ;COPY RET ADDR.
8429 044616 000207                    RTS      PC              ;& RETURN TO MAINLINE ERROR
8430
8431 044620 032777 001000 134312 1$:      BIT      #SW9,@SWR      ;RET HERE FROM MAINLINE ERROR
8432 044626 001107                    BNE      20$            ;& BR IF LOOP ON ERROR
8433 044630 062716 000002              ADD      #2,(SP)        ;BUMP RET ADDR TO NEXT ERROR
8434
8435 044634 023737 005366 005406      CMP      H.B0,E.B0      ;TEST MSG B0
8436 044642 001411                    BEQ      5$              ;BR IF OK
8437 044644 012737 044656 001176      MOV      #4,$ESCAPE     ;ELSE SETUP ESCAPE
8438 044652 011646                    MOV      (SP),-(SP)     ;COPY RET ADDR
8439 044654 000207                    RTS      PC              ;& RETURN TO MAINLINE ERROR
8440
8441 044656 032777 001000 134254 4$:      BIT      #SW9,@SWR      ;RETURN HERE FROM MAINLINE ERROR
8442 044664 001070                    BNE      20$            ;& BR IF LOOP ON ERROR
8443 044666 062716 000002              ADD      #2,(SP)        ;BUMP RET ADDR TO NEXT ERROR
8444
8445 044672 023737 005370 005410      CMP      H.A1,E.A1      ;TEST MSG A1
8446 044700 001411                    BEQ      8$              ;BR IF OK
8447 044702 012737 044714 001176      MOV      #7,$ESCAPE     ;ELSE SETUP ESCAPE
8448 044710 011646                    MOV      (SP),-(SP)     ;COPY RET ADDR
8449 044712 000207                    RTS      PC              ;& RETURN TO MAINLINE ERROR
8450
8451 044714 032777 001000 134216 7$:      BIT      #SW9,@SWR      ;RETURN HERE FROM MAINLINE ERROR
8452 044722 001051                    BNE      20$            ;& BR IF LOOP ON ERROR
8453 044724 062716 000002              ADD      #2,(SP)        ;BUMP RET ADDR TO NEXT ERROR
8454
8455 044730 023737 005372 005412      CMP      H.B1,E.B1      ;TEST MSG B1
8456 044736 001411                    BEQ      11$            ;BR IF OK
8457 044740 012737 044752 001176      MOV      #10,$ESCAPE

```

```

8458 044746 011646          MOV    (SP),-(SP)
8459 044750 000207          RTS    PC
8460
8461 044752 032777 001000 134160 10$:  BIT    #SW9,@SWR
8462 044760 001032          BNE    20$
8463 044762 062716 000002          ADD    #2,(SP)
8464
8465 044766 032737 000001 001460 12$:  BIT    #T.A2,CHKFLG      ;TEST MSG A2?
8466 044774 001402          BEQ    13$              ;BR IF NO
8467 044776 004737 046026          JSR    PC,RCYLD        ;PUT INFO CYLDIF, DO NOT CHECK
8468 045002 032737 000002 001460 13$:  BIT    #T.B2,CHKFLG      ;TEST MSG B2?
8469 045010 001402          BEQ    14$              ;BR IF NO
8470 045012 004737 046100          JSR    PC,RCYLA        ;PUT INFO IN CYLADD, DO NOT CHECK
8471
8472 045016 032737 000004 001460 14$:  BIT    #T.B3,CHKFLG      ;TEST MSG B3?
8473 045024 001404          BEQ    15$
8474 045026 004737 046136          JSR    PC,RSEC         ;PUT INFO IN SECTOR, DO NOT CHECK
8475 045032 004737 046174          JSR    PC,RHEAD        ;PUT INFO IN HEADA, DO NOT CHECK
8476
8477 045036 013737 001172 001176 15$:  MOV    $TMP5,$ESCAPE    ;RESTORE
8478 045044 000207          RTS    PC
8479
8480 045046 012706 001100          MOV    #STACK,SP       ;RESET STACK PTR
8481 045052 013737 001172 001176 20$:  MOV    $TMP5,$ESCAPE    ;RESTORE
8482 045060 000177 134024          JMP    @SLPERR
8483
8484          ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
8485          ; I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
8486
8487          CKCERR: TST    BYPCERR
8488 045064 005737 001456          BNE    4$
8489 045070 001025          BIT    #CERR,HCS1
8490 045072 032737 100000 005314 1$:  BNE    1$              ;BR IF CERR
8491 045100 001001          RTS    PC
8492
8493 045104 032737 004000 005314 1$:  BIT    #CTO,HCS1
8494 045112 001402          BEQ    2$              ;BR IF NOT CTO
8495 045114 104125          ERROR 125             ;CTO ERROR, MSG A & B INVALID
8496 045116 000207          RTS    PC
8497
8498 045120 032737 010000 005316 2$:  BIT    #NED,HCS2
8499 045126 001401          BEQ    3$              ;BR IF NOT NED
8500 045130 104126          ERROR 126             ;NED ERROR, MSG A & B INVALID
8501
8502 045132 032737 001000 005316 3$:  BIT    #MDS,HCS2
8503 045140 001401          BEQ    4$              ;MDS ERROR, MSG A & B INVALID
8504 045142 104127          ERROR 127
8505
8506 045144 000207          4$:  RTS    PC
8507
8508          ; THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
8509          ; IT THEN WAITS FOR CONTROLLER READY
8510          ; IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
8511
8512
8513

```

```
8514 045146 013746 005352          GSTAT: MOV    TEMP1,-(SP)      ;SAVE TEMP1
8515 045152 013765 001222 000010    MOV    $UNIT,RKCS2(R5)      ;CURRENT DRIVE #
8516 045160 063765 005464 000010    ADD    UNITB,RKCS2(R5)     ;ADD 1 IF ON PORT B
8517 045166 012737 000001 005314    MOV    #SELDRV,HCS1
8518 045174 004737 043372          JSR    PC,DOCMD            ;DO SELDRV (STATUS) CMD & GET CONTR RDY
8519 045200 104117          ERROR  117                ;RDY NOT SET BY END OF SELECT DRIVE CMD
8520 045202 012637 005352          MOV    (SP)+,TEMP1        ;RESTOR TEMP1.
8521 045206 000207          RTS    PC
8522
8523          ; THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
8524          ; & ALL CONTROLLER REGISTERS.
8525          ;
8526 045210 013746 005352          GSTAT1: MOV   TEMP1,-(SP)    ;SAVE TEMP1
8527 045214 004737 043622          JSR    PC,HOLD            ;GET ALL CONTR REG
8528 045220 012765 100000 000000    MOV    #CCLR,RKCS1(R5)    ;CLEAR CONTR
8529 045226 013765 001222 000010    MOV    $UNIT,RKCS2(R5)    ;CURRENT DRIVE #
8530 045234 063765 005464 000010    ADD    UNITB,RKCS2(R5)     ;ADD 1 IF ON PORT B
8531 045242 012765 000003 000026    MOV    #3,RKMR1(R5)      ;SELECT WORD 3
8532 045250 004737 045470          JSR    PC,GSTAT2
8533 045254 104117          ERROR  117                ;RDY NOT SET BY END OF SELECT DRV CMD
8534 045256 013737 005342 005400    MOV    HMR2,H.A3         ;STORE MSG A3
8535 045264 013737 005344 005402    MOV    HMR3,H.B3         ;STORE MSG B3
8536
8537 045272 012765 100000 000000    MOV    #CCLR,RKCS1(R5)
8538 045300 013765 001222 000010    MOV    $UNIT,RKCS2(R5)
8539 045306 063765 005464 000010    ADD    UNITB,RKCS2(R5)     ;ADD 1 IF ON PORT B
8540 045314 012765 000002 000026    MOV    #2,RKMR1(R5)      ;SELECT WORD 2
8541 045322 004737 045470          JSR    PC,GSTAT2
8542 045326 104117          ERROR  117                ;RDY NOT SET BY END OF SELECT DRV CMD
8543 045330 013737 005342 005374    MOV    HMR2,H.A2         ;STORE MSG A2
8544 045336 013737 005344 005376    MOV    HMR3,H.B2         ;STORE MSG B2
8545
8546 045344 012765 100000 000000    MOV    #CCLR,RKCS1(R5)
8547 045352 013765 001222 000010    MOV    $UNIT,RKCS2(R5)
8548 045360 063765 005464 000010    ADD    UNITB,RKCS2(R5)     ;ADD 1 IF ON PORT B
8549 045366 012765 000001 000026    MOV    #1,RKMR1(R5)      ;SELECT WORD 1
8550 045374 004737 045470          JSR    PC,GSTAT2
8551 045400 104117          ERROR  117                ;RDY NOT SET BY END OF SELECT DRV CMD
8552 045402 013737 005342 005370    MOV    HMR2,H.A1         ;STORE MSG A1
8553 045410 013737 005344 005372    MOV    HMR3,H.B1         ;STORE MSG B1
8554
8555 045416 012765 100000 000000    MOV    #CCLR,RKCS1(R5)
8556 045424 013765 001222 000010    MOV    $UNIT,RKCS2(R5)
8557 045432 063765 005464 000010    ADD    UNITB,RKCS2(R5)     ;ADD 1 IF ON PORT B
8558 045440 004737 045470          JSR    PC,GSTAT2
8559 045444 104117          ERROR  117                ;RDY NOT SET BY END OF SEL DRV CMD
8560 045446 013737 005342 005364    MOV    HMR2,H.A0         ;STORE MSG A0
8561 045454 013737 005344 005366    MOV    HMR3,H.B0         ;STORE MSG B0
8562
8563 045462 012637 005352          MOV    (SP)+,TEMP1        ;RESTORE TEMP1
8564 045466 000207          RTS    PC
8565
8566 045470 012737 000001 005314    GSTAT2: MOV   #SELDRV,HCS1
8567 045476 053737 001170 005314    BIS    $TMP4,HCS1         ;ADD CDT IF RK07
8568 045504 013765 005314 000000    MOV    HCS1,RKCS1(R5)    ;GET STATUS
8569 045512 013737 001400 005352    MOV    TIO,TEMP1
```

```

8570 045520 004737 043534      JSR    PC,FRDY1      :FIND CONTR RDY & STORE DRIVE REGS ONLY
8571 045524 000207              RTS    PC            :RET HERE IF NOT RDY
8572 045526 062716 000002      ADD    #2,(SP)       :RET HERE IF OK
8573 045532 000207              RTS    PC
8574
8575
8576      ; THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
8577      ; IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
8578      ; THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
8579      ; RETURN IF CERR SET
8580      ; RETURN +2 IF CERR CLEAR
8581
8582 045534 012765 000040 000010  SUBCLR: MOV    #SCLR,RKCS2(R5) ;SUBSYS CLEAR
8583 045542 013737 001400 005352      MOV    T10,TEMP1
8584 045550 004737 043466      JSR    PC,FRDY      :FIND RDY
8585 045554 104120              ERROR  120          :RDY NOT SET BY END OF SCLR
8586 045556 013765 001222 000010  MOV    $UNIT,RKCS2(R5) ;CURRENT DRIVE #
8587 045564 063765 005464 000010  ADD    UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
8588 045572 005065 000026      CLR    RKMR1(R5)    ;SELECT WORD 0
8589 045576 004737 045146      JSR    PC,GSTAT     ;GET STATUS
8590 045602 032737 100000 005314  BIT    #CERR,HCS1   ;CHECK FOR CONT ERROR
8591 045610 001401              BEQ    1$
8592 045612 000207              RTS    PC
8593 045614 062716 000002      1$:  ADD    #2,(SP)    ;SKIP OVER ERROR
8594 045620 000207              RTS    PC
8595
8596
8597      ; READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
8598
8599 045622 012765 000003 000026  RDSEC: MOV    #3,RKMR1(R5) ;WORD 3
8600 045630 004737 045146      JSR    PC,GSTAT
8601 045634 013737 005344 001374  MOV    HMR3,SECTOR
8602 045642 042737 177017 001374  BIC    #^C<M.SECT>,SECTOR
8603 045650 006237 001374      ASR    SECTOR       ;RIGHT JUSTIFY
8604 045654 006237 001374      ASR    SECTOR       ;SECTOR
8605 045660 006237 001374      ASR    SECTOR       ;INFO
8606 045664 006237 001374      ASR    SECTOR
8607 045670 000207              RTS    PC
8608
8609      ; READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
8610
8611 045672 012765 000002 000026  RDCYLD: MOV    #2,RKMR1(R5) ;WORD 2
8612 045700 004737 045146      JSR    PC,GSTAT
8613 045704 013737 005342 001356  MOV    HMR2,CYLDIF
8614 045712 043737 012774 001356  BIC    MASK1,CYLDIF
8615 045720 006237 001356      ASR    CYLDIF       ;RIGHT JUSTIFY
8616 045724 006237 001356      ASR    CYLDIF       ;CYL DIFF/OFFSET
8617 045730 006237 001356      ASR    CYLDIF       ;INFO
8618 045734 006237 001356      ASR    CYLDIF
8619 045740 023737 001356 012772  CMP    CYLDIF,MASK  ;CHK TO SEE IF RET IN COMPL. FORM
8620 045746 001002              BNE    1$
8621 045750 005037 001356      CLR    CYLDIF       ;CLR IF YES
8622 045754 000207      1$:  RTS    PC
8623
8624
8625      ; READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'

```

```

8626
8627 045756 012765 000002 000026 RDCYLA: MOV #2,RKMR1(R5) ;WORD 2
8628 045764 004737 045146 JSR PC,GSTAT
8629 045770 013737 005344 001360 MOV HMR3,CYLADD
8630 045776 043737 012774 001360 BIC MASK1,CYLADD
8631 046004 006237 001360 ASR CYLADD ;RIGHT JUSTIFY
8632 046010 006237 001360 ASR CYLADD ;CYL ADDR
8633 046014 006237 001360 ASR CYLADD ;INFO
8634 046020 006237 001360 ASR CYLADD
8635 046024 000207 RTS PC
8636
8637 ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
8638
8639 046026 013737 005374 001356 RCYLD: MOV H.A2,CYLDIF
8640 046034 043737 012774 001356 BIC MASK1,CYLDIF ;CLEAR UNWANTED INFO
8641 046042 006237 001356 ASR CYLDIF ;RIGHT JUSTIFY
8642 046046 006237 001356 ASR CYLDIF
8643 046052 006237 001356 ASR CYLDIF
8644 046056 006237 001356 ASR CYLDIF
8645 046062 023737 001356 012772 CMP CYLDIF,MASK ;CHK TO SEE IF RET IN COMPL. FORM
8646 046070 001002 BNE 1$ ;BR IF NO
8647 046072 005037 001356 CLR CYLDIF ;ELSE CLEAR
8648 046076 000207 1$: RTS PC
8649
8650 ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
8651
8652 046100 013737 005376 001360 RCYLA: MOV H.B2,CYLADD
8653 046106 043737 012774 001360 BIC MASK1,CYLADD ;CLEAR UNWANTED INFO
8654 046114 006237 001360 ASR CYLADD ;RIGHT JUSTIFY
8655 046120 006237 001360 ASR CYLADD
8656 046124 006237 001360 ASR CYLADD
8657 046130 006237 001360 ASR CYLADD
8658 046134 000207 RTS PC
8659
8660 ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
8661
8662 046136 013737 005402 001374 RSEC: MOV H.B3,SECTOR
8663 046144 042737 177017 001374 BIC #*C<M.SECT>,SECTOR ;CLEAR UNWANTED INFO
8664 046152 006237 001374 ASR SECTOR ;RIGHT JUSTIFY
8665 046156 006237 001374 ASR SECTOR
8666 046162 006237 001374 ASR SECTOR
8667 046166 006237 001374 ASR SECTOR
8668 046172 000207 RTS PC
8669
8670 ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEAD'
8671
8672 046174 013737 005402 001422 RHEAD: MOV H.B3,HEAD
8673 046202 042737 170777 001422 BIC #*C<M.HEAD>,HEAD ;CLEAR UNWANTED INFO
8674 046210 006237 001422 ASR HEAD ;RIGHT JUSTIFY IT
8675 046214 000337 001422 SWAB HEAD
8676 046220 000207 RTS PC
8677
8678 ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
8679 ;ENTER WITH TIME IN SECONDS IN TEMP2
8680 ;RETURN IF NOT FOUND
8681 ;RETURN+2 IF FOUND - SKIP OVER ERROR

```

```

8682
8683 046222 012737 177777 005352 FHDHM: MOV # -1,TEMP1 ;ALL 1'S
8684 046230 012765 000001 000026 MOV #1,RKMR1(R5) ;WORD 1
8685 046236 004737 045146 JSR PC,GSTAT
8686 046242 032737 000040 005342 1$: BIT #D.HDHM,HMR2
8687 046250 001007 BNE 2$
8688 046252 005337 005352 DEC TEMP1
8689 046256 001367 BNE 1$
8690 046260 005337 005354 DEC TEMP2
8691 046264 001356 BNE FHDHM
8692 046266 000207 RTS PC
8693 046270 062716 C00002 2$: ADD #2,(SP) ;SKIP OVER ERROR
8694 046274 000207 RTS PC
8695
8696 ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE THE TIMEOUT
8697 ;RETURN IF NOT FOUND
8698 ;RETURN+2 IF FOUND: SKIP OVER ERROR
8699
8700 046276 012737 000372 005352 FLOAD: MOV #250,TEMP1
8701 046304 012765 000001 000026 MOV #1,RKMR1(R5) ;WORD 1
8702 046312 004737 045146 JSR PC,GSTAT
8703 046316 032737 010000 005342 1$: BIT #D.LOAD,HMR2
8704 046324 001004 BNE 2$
8705 046326 005337 005352 DEC TEMP1
8706 046332 001367 BNE 1$
8707 046334 000207 RTS PC
8708 046336 062716 000002 2$: ADD #2,(SP) ;SKIP OVER ERROR
8709 046342 000207 RTS PC
8710
8711 ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
8712 ;ENTER WITH CYL # IN 'CALADD'
8713 ;ENTER WITH HEAD # IN 'HEAD'
8714 ;ENTER WITH FORMAT IN 'FORMAT'
8715
8716 046344 010046 FHDTAB: MOV R0,-(SP) ;SAV R0
8717 046346 010146 MOV R1,-(SP) ;SAV R1
8718 046350 012700 001462 MOV #HDTAB,R0 ;HEADER WORD TABLE ADDR
8719 046354 005001 CLR R1 ;SECTOR COUNTER
8720 046356 013737 001420 001424 MOV HEAD,HD1
8721 046364 006337 001424 ASL HD1
8722 046370 006337 001424 ASL HD1
8723 046374 006337 001424 ASL HD1
8724 046400 006337 001424 ASL HD1
8725 046404 006337 001424 ASL HD1 ;SETUP HEAD # FOR WORD 2 OF HEADER
8726 046410 013737 001426 001430 MOV FORMAT,FMT1
8727 046416 000337 001430 SWAB FMT1
8728 046422 006337 001430 ASL FMT1 ;SETUP FORMAT FOR WORD 2 OF HEADER
8729
8730 046426 013720 001362 1$: MOV CALADD,(R0)+ ;HEADER WORD 1-CYL ADDR
8731 046432 010110 MOV R1,(R0) ;HEADER WORD 2-SECTOR NO
8732 046434 053710 001424 BIS HD1,(R0) ;
8733 046440 053710 001430 BIS FMT1,(R0) ; -HEAD NO
; -FORMAT

```


CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 L 13
GET VALUE FOR SOFTWARE SWITCH REGISTER PAGE 168

SEQ 0167

8734 046444 004737 046524

JSR PC,SECFLG

```
8735
8736 046450 013737 001362 005352      MOV      CALADD,TEMP1
8737 046456 011037 005354      MOV      (R0),TEMP2
8738 046462 043737 001362 005354      BIC      CALADD,TEMP2
8739 046470 042037 005352      BIC      (R0)+,TEMP1
8740 046474 053737 005352 005354      BIS      TEMP1,TEMP2
8741 046502 013720 005354      MOV      TEMP2,(R0)+      ;HEADER WORD 3-HEADER CHECK
8742
8743 046506 005201      INC      R1                ;SECTOR CTR
8744 046510 020127 000026      CMP      R1,#22.          ;ALL 22 SECTORS DONE? (66 WORDS)
8745 046514 001344      BNE      1$                ;BR IF NO
8746
8747 046516 012601      MOV      (SP)+,R1          ;RESTOR R1
8748 046520 012600      MOV      (SP)+,R0          ;RESTOR R0
8749 046522 000207      RTS      PC
8750
8751      ; THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS
8752      ; TEST & SETS BITS 14 & 15 APPROPRIATLY.
8753      ;
```

```
8754 046524 010246          SECFLG: MOV    R2,-(SP)          ;SAVE R2
8755 046526 005737 001426    TST    FORMAT
8756 046532 001016          BNE    1$                    ;BR IF 20 SECTOR FORMAT
8757          ;NOTE: ONLY 22 SECTOR PERFORMED
8758 046534 012702 002306    MOV    #BSE22H+8.,R2
8759 046540 004737 046574    JSR    PC,FLGTST             ;GET HARDWARE DETECTED FLAG
8760 046544 052710 100000    BIS    #BIT15,(R0)          ;RETURN HERE IF GOOD SECTOR
8761
8762 046550 012702 003306    MOV    #BSE22S+8.,R2       ;ELSE RETURN HERE
8763 046554 004737 046574    JSR    PC,FLGTST             ;GET SOFTWARE DETECTED FLAG
8764 046560 052710 040000    BIS    #BIT14,(R0)          ;RETURN HERE IF GOOD SECTOR
8765
8766 046564 012602          MOV    (SP)+,R2             ;ELSE RETURN HERE
8767 046566 000207          RTS    PC
8768
8769 046570 012602          1$:  MOV    (SP)+,R2          ;RESTORE R2
8770 046572 000207          RTS    PC
8771
8772          ;
8773          ;THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
8774          ;ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
8775          ;RETURN IF NO COMPARE
8776          ;RETURN+4 IF COMPARE
8777          ;
8778 046574 010346          FLGTST: MOV   R3,-(SP)        ;SAVE R3
8779
8780 046576 021227 177777    1$:  CMP    (R2),#-1           ;SEE IF ALL 1'S
8781 046602 001002          BNE    2$                    ;BR IF NO
8782 046604 012603          MOV    (SP)+,R3             ;RESTORE R3
8783 046606 000207          RTS    PC
8784
8785 046610 022237 001362    2$:  CMP    (R2)+,CALADD       ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
8786 046614 001403          BEQ    3$                    ;
8787 046616 062702 000002    ADD    #2,R2                 ;GO TO NEXT CYL WORD IN TABLE
8788 046622 000765          BR     1$
8789
8790 046624 013703 001420    3$:  MOV    HEAD,R3              ;GET HEAD # FROM FHDTAB ROUTINE
8791 046630 000303          SWAB   R3
8792 046632 050103          BIS    R1,R3                 ;ADD SECTOR # FROM FHDTAB ROUTINE
8793 046634 022203          CMP    (R2)+,R3             ;SEE IF SECTOR/HEAD COMPARE
8794          ;& INCR PTR TO NEXT CYL WORD
8795 046636 001401          BEQ    4$                    ;BR IF COMPARE
8796 046640 000756          BR     1$                    ;ELSE TRY NEXT CYL
8797
8798 046642 012603          4$:  MOV    (SP)+,R3             ;RESTORE R3
8799 046644 062716 000004    ADD    #4,(SP)              ;INCREMENT RET ADDR
8800 046650 000207          RTS    PC
8801
8802          ;
8803          ;THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
8804          ;WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
8805          ;
8806 046652 010046          SORT: MOV   R0,-(SP)          ;SAVE R0
8807 046654 010146          MOV   R1,-(SP)              ;SAVE R1
8808 046656 004737 045622    JSR   PC,RDSEC
8809 046662 062737 000001 001374  ADD   #1,SECTOR
```

```

8810 046670 004737 046760      JSR      PC,MULT6      ;MULT SECTOR BY 6
8811
8812 046674 012700 000204      MOV      #132.,R0
8813 046700 163700 001374      SUB      SECTOR,R0      ;R0-SECTOR TO R0 = INDEX
8814 046704 010037 001374      MOV      R0,SECTOR
8815 046710 062737 001666 001374  ADD      #RHTAB,SECTOR ;SAVE INDEX
8816
8817 046716 062700 001666      ADD      #RHTAB,R0      ;INDEX TO BOT HALF OF RHTAB
8818 046722 012701 002072      MOV      #SRTTAB,R1     ;INDEX TO TOP HALF OF SRTTAB
8819
8820 046726 012021      1$:     MOV      (R0)+,(R1)+     ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
8821 046730 020027 002072      CMP      R0,#RHTAB+132.
8822 046734 001374      BNE      1$
8823
8824 046736 012700 001666      MOV      #RHTAB,R0      ;PUT TOP OF RHTAB TO BOT OF SRTTAB
8825 046742 012021      2$:     MOV      (R0)+,(R1)+
8826 046744 020037 001374      CMP      R0,SECTOR
8827 046750 001374      BNE      2$
8828
8829 046752 012601      MOV      (SP)+,R1      ;RESTOR R1
8830 046754 012600      MOV      (SP)+,R0      ;RESTOR R0
8831 046756 000207      RTS      PC
8832
8833      ;
8834      ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
8835      ;
8836 046760 006337 001374  MULT6:  ASL      SECTOR      ;2 X SECTOR
8837 046764 013746 001374      MOV      SECTOR,-(SP)
8838 046770 006337 001374      ASL      SECTOR      ;4 X SECTOR
8839 046774 062637 001374      ADD      (SP)+,SECTOR ;:(4 X S)+(2 X S) = 6 X SECTOR
8840 047000 000207      RTS      PC
8841
8842      ;
8843      ;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
8844      ;CMD. IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
8845      ;CYLINDER AT CYL 410, TRACK 2.
8846      ;
8847      ;RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
8848      ;RETURN+2 IF LISTED, SKIP OVER ERROR
8849      ;
8850 047002 010446  TRUERR: MOV      R4,-(SP)      ;SAVE R4
8851
8852 047004 032737 010000 005314  BIT      #CFMT,HCS1     ;CHECK FORMAT
8853 047012 001014      BNE      3$            ;BR FOR 20 SECTOR FORMAT
8854      ;ONLY 22 SECTOR FMT IN THIS PGM
8855
8856 047014 012704 002306      MOV      #BSE22H+8.,R4
8857 047020 004737 047054      JSR      PC,TERR1      ;SEE IF ON HARDWARE DETECTED TABLE
8858 047024 000407      BR       3$            ;RETURN HERE IF YES
8859
8860 047026 012704 003306      MOV      #BSE22S+8.,R4
8861 047032 004737 047054      JSR      PC,TERR1      ;& SEE IF ON SOFTWARE DETECTED TABLE
8862 047036 000402      BR       3$            ;RETURN HERE IF YES
8863
8864 047040 012604  1$:     MOV      (SP)+,R4      ;RESTORE R4
8865 047042 000207      RTS      PC            ;RETURN WITHOUT JUMPING OVER ERROR

```

```

8866
8867 047044 012604          3$:  MOV    (SP)+,R4      ;RESTORE R4
8868 047046 062716 000002  ADD    #2,(SP)        ;SKIP OVER ERROR ON RETURN
8869 047052 000207          RTS    PC
8870
8871
8872 ;THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
8873 ;THE BSE TABLE FOR THE ABOVE SUBROUTINE.
8874 ;RETURN IF FOUND ON TABLE
8875 ;RETURN+2 IF NOT FOUND
8876
8877 047054 021427 177777  TERR1: CMP    (R4),#-1      ;SEE IF ALL 1'S
8878 047060 001405          BEQ    1$              ;BR IF YES, NOT ON TABLE
8879 047062 022437 005334  CMP    (R4)+,HDC      ;SEE IF CYL MATCH
8880 047066 001405          BEQ    2$              ;BR IF YES
8881 047070 005724          TST    (R4)+          ;ELSE ADV TO NEXT CYL WORD
8882 047072 000770          BR     TERR1          ;& TRY AGAIN.
8883
8884 047074 062716 000002  1$:  ADD    #2,(SP)
8885 047100 000207          RTS    PC
8886
8887 047102 022437 005324  2$:  CMP    (R4)+,HDA      ;SEE IF SECTOR & TRACK MATCH
8888 047106 001401          BEQ    3$              ;BR IF YES
8889 047110 000761          BR     TERR1          ;OR TRY AGAIN
8890
8891 047112 000207          3$:  RTS    PC
8892
8893
8894 ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
8895
8896
8897 047114 005037 001372  CLKON: CLR    TIMUP
8898 047120 005737 005460  TST    PCLKF
8899 047124 001004          BNE    1$              ;BRANCH IF P-CLOCK PRESENT
8900 047126 012777 000100 132172 MOV    #100,@LKS      ;L-CLOCK, ENABLE INT
8901 047134 000207          RTS    PC
8902 047136 012777 177777 132156 1$:  MOV    #-1,@PKSB      ;P-CLOCK, ALL 1'S
8903 047144 012777 000135 132146  MOV    #135,@PKS      ;ENABLE INT, CT UP, REP INT
8904 047152 000207          RTS    PC              ;LINE FREQ & RUN
8905
8906 ;KW11-L & KW11-P INTERRUPT HANDLER
8907
8908 047154 005337 001366  CLOCK: DEC    COUNT
8909 047160 000002          RTI
8910
8911 ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
8912
8913 047162 005737 005460  CLKOF: TST    PCLKF
8914 047166 001003          BNE    1$              ;BRACH IF P-CLOCK PRESENT
8915 047170 005077 132132  CLR    @LKS           ;L-CLOCK, CLEAR INTERRUPT
8916 047174 000207          RTS    PC
8917 047176 005077 132116  1$:  CLR    @PKS           ;P-CLOCK, CLEAR INTERRUPT
8918 047202 000207          RTS    PC
8919
8920 ;THIS ROUTINE DOES A TIMEOUT DEPENDING ON THE VALUE IN 'COUNT' WHICH THE
8921

```

```
8922      ;'CLOCK' HANDLER DECREMENTS EVERY 1/60 SEC.
8923      ;
8924 047204 004737 047114      TMO:   JSR   PC,CLKON      ;TURN CLOCK ON
8925 047210 005737 001366      1$:   TST   COUNT        ;TIME UP?
8926      ;                   BNE   TMO          ;BR IF NO S9-SEP-77
8927 047214 001375              ;                   BNE   1$          ;29-SEP-77
8928 047216 004737 047162      ;                   JSR   PC,CLKOF     ;ELSE, TURN CLOCK OFF
8929 047222 000207              ;                   RTS    PC
8930
8931      ;
8932      ;THIS ROUTINE GENERATES PARITY FOR EXPECTED MESSAGES
8933      ;ENTER WITH THE EXPECTED WORD IN TEMP1
8934      ; TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
8935      ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES ( TEMP1 BACK TO ORIG),
8936      ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
8937      ; THE PARITY BIT IS NOT SET IN B .
8938      ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
8939      ;SET IN TEMP1
8940      ;
8941 047224 010046              SBPAR:  MOV   R0,-(SP)      ;SAVE R0
8942 047226 010146              ;                   MOV   R1,-(SP)      ;SAVE R1
8943 047230 012700 000021      ;                   MOV   #17.,R0     ;SHIFT COUNTER
8944 047234 005001              ;                   CLR   R1          ;COUNT # OF 1'S IN TEMP1
8945 047236 000241              ;                   CLC           ;CLEAR CARRY
8946
8947 047240 006137 005352      1$:   ROL   TEMP1
8948 047244 103001              ;                   BCC   2$          ;BR IF CARRY CLEAR
8949 047246 005201              ;                   INC   R1          ;COUNT # OF 1'S
8950 047250 005300      2$:   DEC   R0          ;SHIFT COUNTER
8951 047252 001372              ;                   BNE   1$
8952
8953 047254 032701 000001              ;                   BIT   #BIT0,R1
8954 047260 001003              ;                   BNE   3$          ;BR IF ODD # IN R0
8955 047262 052737 100000 005352      3$:   BIS   #M.PAR,TEMP1    ;SET PARITY BIT
8956 047270 012601              ;                   MOV   (SP)+,R1    ;RESTORE R1
8957 047272 012600              ;                   MOV   (SP)+,R0    ;RESTORE R0
8958 047274 000207              ;                   RTS    PC
8959
8960      ;
8961      ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
8962      ;WHEN $LPERR SET BY OTHER THAN SCOPE ROUTINE
8963      ; IE: MY LOOP MACRO
8964      ;
8965 047276 032777 001000 131634      SCOP1$: BIT   #SW9,@SWR     ;LOOP ON ERROR?
8966 047304 001406              ;                   BEQ   1$          ;BR IF NO
8967 047306 105737 001103      ;                   TSTB  $ERFLG     ;HAD ERROR?
8968 047312 001403              ;                   BEQ   1$          ;BR IF NO
8969 047314 013716 001110      ;                   MOV   $LPERR,(SP)
8970 047320 000002              ;                   RTI
8971
8972 047322 011637 001110      1$:   MOV   (SP),$LPERR     ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
8973 047326 000002              ;                   RTI
8974
8975
8976
8977
```

```

8978
8979
8980
8981
8982
8983
8984 047330 022626
8985
8986 047332 004737 045534
8987 047336 104024
8988
8989 047340 005737 005276
8990 047344 001431
8991 047346 005737 000042
8992 047352 001403
8993 047354 104401 056760
8994 047360 000402
8995 047362 104401 057026
8996 047366
8997
8998 047366 004737 045534
8999 047372 104024
9000
9001 047374 012737 000011 005314
9002 047402 004737 043372
9003 047406 104121
9004
9005 047410 013737 001406 005354
9006 047416 004737 044006
9007 047422 104074
9008
9009 047424 005037 005276
9010
9011
9012 047430 005737 005300
9013 047434 001520
9014 047436 005237 005302
9015
9016 047442 012765 100000 000000
9017 047450 013765 001222 000010
9018 047456 063765 005464 000010
9019 047464 012737 000013 005314
9020 047472 004737 043372
9021 047476 104124
9022
9023 047500 012765 000001 000026
9024 047506 004737 045146
9025 047512 032737 020000 005342
9026 047520 001001
9027 047522 104070
9028 047524 013737 001400 005354
9029 047532 004737 044006
9030 047536 104055
9031
9032 047540 012765 100000 000000
9033 047546 013765 001222 000010

```

: THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
: IT IS USED TO ALLOW THE OPERATOR TO HLT THE CPU WHILE INSURING
: THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
: THE CPU.
:
STOP: CMP (SP)+,(SP)+ ;RESTORE STACK FROM INTERRUPT
JSR PC,SUBCLR ;CERR AFTER
ERROR 24
TST UNLD ;SEE IF HEADS UNLOADED
BEQ 3\$;BR IF NO
TST 42 ;SEE IF MANUAL OR AUTO MODE
BEQ 1\$;BR IF MANUAL MODE
TYPE ,MSG74 ;PGM ABORT PENDING
BR 2\$
1\$: TYPE ,MSG75 ;HALT PENDING
2\$:
JSR PC,SUBCLR ;CERR AFTER SCLR
ERROR 24
MOV #SRTSPL,HCS1
JSR PC,DOCMD ;DO START SPINDLE CMD & GET CONTR RDY
ERROR 121 ;RDY NOT SET AFTER ST SPIN CMD.
MOV T100,TEMP2 ;SETUP TIMEOUT
JSR PC,FATT1 ;FIND ATTN
ERROR 74 ;NO ATTN AFTER ST SPIN CMD.
CLR UNLD
3\$: TST BADHDR ;SEE IF HEADERS VALID
BEQ 4\$;BR IF YES
INC HPEND
MOV #CCLR,RKCS1(R5)
MOV \$UNIT,RKCS2(R5)
ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
MOV #RECAL,HCS1
JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
ERROR 124 ;RDY NOT SET AFTER RECAL CMD
MOV #1,RKMR1(R5) ;SELECT WORD 1
JSR PC,GSTAT
BIT #D.RTZ,HMR2
BNE 64\$
ERROR 70 ;RTZ NOT SET DURING RECAL CMD
MOV T10,TEMP2 ;SETUP TIMEOUT
JSR PC,FATT1 ;FIND ATTN
ERROR 55 ;NO ATTN AFTER RECAL CMD
MOV #CCLR,RKCS1(R5)
MOV \$UNIT,RKCS2(R5) ;DRIVE#

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 PAGE 175
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0174

```

9034 047554 063765 005464 000010 ADD UNITB,RKCS2(R5) ;ADD 1 IF ON PORT B
9035 047562 012737 000005 005314 MOV #CLEAR,HCS1
9036 047570 004737 043372 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
9037 047574 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
9038 047576 004737 043750 JSR PC,TSTATN ;TEST FOR ATTN
9039 047602 000401 BR 65$
9040 047604 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
9041 047606
9042
65$:
9043 047606 012737 010340 005404 MOV #<D.DRA!D.SPIN!D.DRDY!D.VV>,E.A0 ;EXPECTED MSG A0
9044 047614 005037 005406 CLR E.B0 ;EXPECTED MSG B0
9045 047620 012737 001720 005410 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
9046 047626 012737 000001 005412 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
9047 047634 005037 005414 CLR E.A2 ;EXPECTED MSG A2
9048 047640 012737 000002 005416 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
9049 047646 012737 000003 005422 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
9050
9051 047654 004737 044274 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
9052 047660 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
9053 047662 104033 ERROR 33 ;MSG A0 ERROR AFTER DRV CLEAR CMD
9054 047664 104034 ERROR 34 ;MSH B0 ERROR
9055 047666 104035 ERROR 35 ;MSG A1 ERROR
9056 047670 104036 ERROR 36 ;MSG B1 ERROR
9057
9058
9059 047672 000137 047736 JMP FORM ;WRITE VALID FORMATS
9060
9061 047676 005737 000042 4$: TST 42 ;SEE IF MANUAL OR AUTO MODE
9062 047702 001410 BEQ 5$ ;BR IF MANUAL MODE
9063 047704 104401 057063 TYPE ,MSG76 ;PGM ABORTED
9064 047710 005037 042732 CLR $EOPCT ;SET UP EOP TO EXIT TO MONITOR
9065 047714 005037 001176 CLR $ESCAPE
9066 047720 000137 042704 JMP $EOP1 ;ABORT PGM
9067
9068 047724 104401 057105 5$: TYPE ,MSG77 ;CPU HALTED
9069 047730 000000 HALT
9070 047732 000137 010646 JMP ST5 ;START OVER IF CONTINUE PRESSED
9071 047736
FORM:
9072 .SBTTL UNEXPECTED TIMEOUT HANDLER
9073
9074
9075 ;THIS ROUTINE IS ENTERED IF THERE IS
9076 ; A. NON EXISTANT MEMORY (NO SSYN)
9077 ; B. BOUNDRY ERROR
9078 ; C. STACK OVERFLOW
9079 ;
9080
9081 047736 011600 BADTMO: MOV (SP),R0 ;SAVE PC WHERE TIMEOUT OCCURRED.
9082 047740 005740 TST -(R0) ;GET PC BEFORE UPDATE
9083 047742 032777 020000 131170 BIT #SW13,@SWR ;INHIBIT ERR TYPOUT?
9084 047750 001005 BNE 1$ ;YES, DON'T TYPE
9085 047752 104401 057265 TYPE ,EM3 ;ABORT TESTS,UNEXP T.O. @ PC=
9086 047756 010046 MOV R0,-(SP) ;SAVE R0 FOR TYPEOUT
9087 ;TYPE PC
9088 047760 104403 TYPOS ;GO TYPE--OCTAL ASCII
9089 047762 006 .BYTE 6 ;TYPE 6 DIGIT(S)

```



```

9090 047763 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
9091 047764 032777 001000 131146 1$: BIT #SW9,@SWR ;:LOOP ON ERROR?
9092 047772 001403 BEQ 2$ ;:NO BRANCH
9093 047774 022626 CMP (SP)+,(SP)+ ;:YES, RESTORE STACK
9094 047776 000177 131104 JMP @SLPADR ;:GO TO STARTING ADDR OF TEST
9095 ;:THAT GAVE BAD TIMEOUT
9096 050002 032777 040000 131130 2$: BIT #SW14,@SWR ;:LOOP ON TEST?
9097 050010 001401 BEQ 3$ ;:NO BRANCH
9098 050012 000002 RTI ;:YES
9099
9100 050014 000000 3$: HALT ;:UNEXPECTED TIME OUT OCCURRED
9101 ;:AS INDICATED. YOU CAN LOOP ON
9102 ;:ERROR, LOOP ON TEST OR INHIBIT
9103 ;:ERROR TYPEOUT BY SETTING THOSE
9104 ;:SWITCHES.
9105
9106 050016 022626 CMP (SP)+,(SP)+ ;:RESTORE STACK
9107 050020 000137 042704 JMP $EOP1 ;:ABORT TESTS
9108
9109 .SBTTL MEMORY CHECK ENABLE TRAP
9110
9111 050024 012737 050040 001176 MEMERR: MOV #1$, $ESCAPE
9112 050032 011637 001334 MOV (SP), TRAPPC ;:STORE PC
9113 050036 104017 ERROR 17 ;:UNEXP MEM PARITY TRAP
9114 050040 005037 001176 1$: CLR $ESCAPE
9115 050044 032777 001000 131066 BIT #SW9,@SWR ;:CHECK IF LOOP ON ERROR
9116 050052 001001 BNE 2$ ;:YES, FORCE STACK AND TRY AGAIN
9117 050054 000002 RTI ;:ELSE RETURN
9118
9119 050056 012706 001100 2$: MOV #STACK, SP ;:INIT STACK
9120 050062 000177 131022 JMP @SLPERR ;:LOOP ON ERROR
9121
9122
9123 .SBTTL RK06 INTERRUPT HANDLER
9124
9125 050066 000240 INTER: NOP
9126 050070 000240 NOP
9127 050072 000240 NOP
9128 050074 011600 MOV (SP), R0 ;:SAVE PC WHERE INT OCCURRED.
9129 050076 005740 TST -(R0) ;:GET PC BEFORE UPDATE.
9130 050100 104401 056235 TYPE ,MSG6 ;:INT AT PC=
9131 050104 010046 MOV R0,-(SP) ;:SAVE R0 FOR TYPEOUT
9132 ;:TYPE PC
9133 050106 104403 TYPOS ;:GO TYPE--OCTAL ASCII
9134 050110 006 .BYTE 6 ;:TYPE 6 DIGIT(S)
9135 050111 000 .BYTE 0 ;:SUPPRESS LEADING ZEROS
9136 050112 000000 HALT
9137 050114 000240 NOP
9138 050116 000240 NOP
9139 050120 000002 RTI
9140
9141 .SBTTL POWER DOWN AND UP ROUTINES
9142
9143 ;:POWER DOWN ROUTINE
9144
9145 050122 012737 050134 000024 $PWRDN: MOV # $PWRUP, PWRVEC ;:SET UP VECTOR

```

```

9146 050130 000000          HALT
9147 050132 000776          BR      .-2          ;HANG UP.
9148
9149          ;POWER UP ROUTINE
9150
9151 050134 005037 050206    $PWRUP: CLR      $PWRCT          ;WAIT LOOP FOR TTY
9152 050140 005237 050206    1$:      INC      $PWRCT          ;WAIT FOR THE INCR
9153 050144 001375          BNE      1$              ;OF WORD
9154 050146 012737 050122 000024    MOV      #SPWRDN,PWRVEC ;SET POWER DOWN VECTOR
9155 050154 012737 000340 000026    MOV      #PR7,PWRVEC+2 ;PRIORITY 7
9156 050162 012737 000340 000036    MOV      #PR7,TRAPVEC+2 ;LOCKOUT ALL INTERRUPTS FOR TRAPS
9157 050170 012706 001100    MOV      #STACK,SP      ;INITIALIZE STACK
9158 050174 104401 056404    TYPE     ,MSG11         ;REPORT POWER FAIL
9159 050200 000005
9160 050202 000137 013004    RESET
9161          JMP      PFSRT
9162 050206 000000          $PWRCT: 0              ;WAIT COUNT FOR TTY
9163
9164          ;
9165          ;DIVISION UTILITY ROUTINE
9166          ;
9167          ;R0-R1-R2-R3=DIVIDEND
9168          ;R4-R5=DIVISOR
9169          ;R0-R1=REMAINDER AFTER DIVISION
9170          ;R2-R3=QUOTIENT AFTER DIVISION
9171          ;ENTER WITH JSR PC,M.DPID
9172          ;
9173 050210 012746 000040    M.DPID: MOV      #40,-(SP)      ;COUNTER FOR DIVISION CYCLES
9174 050214 010446          MOV      R4,-(SP)        ;HI ORDER
9175 050216 010546          MOV      R5,-(SP)        ;LO ORDER TO THE STACK
9176 050220 005466 000002    NEG      2(SP)          ;FORM NEGATIVE
9177 050224 005416          NEG      @SP             ;VERSION OF DIVISOR
9178 050226 005666 000002    SBC      2(SP)
9179 050232 061601          ADD      @SP,R1
9180 050234 005500          ADC      R0              ;PERFORM INIT SUBT.
9181 050236 066600 000002    ADD      2(SP),R0
9182 050242 103445          BCS      M.DP50          ;IF CARRY THEN OVERFLOW HAS OCCURRED
9183 050244 005046          CLR      -(SP)          ;THIS IS A LONGER LASTING CARRY BIT
9184 050246 006103    M.DP40: ROL      R3
9185 050250 006102          ROL      R2
9186 050252 006101          ROL      R1
9187 050254 006100          ROL      R0
9188 050256 005716          TST      @SP             ;TEST CARRY INDICATOR
9189 050260 001410          BEQ      M.DP41          ;IF TO CARRY THEN ADD, ELSE SUBT.
9190 050262 005016          CLR      @SP             ;CLEAR UP FOR NEXT TIME
9191 050264 066601 000002    ADD      2(SP),R1
9192 050270 005500          ADC      R0              ;ADD -(DIVISOR)
9193 050272 005516          ADC      @SP             ;SET CARRY
9194 050274 066600 000004    ADD      4(SP),R0
9195 050300 000404          BR      M.DP42
9196
9197 050302 060501    M.DP41: ADD      R5,R1
9198 050304 005500          ADC      R0              ;ADD +(DIVISOR)
9199 050306 005516          ADC      @SP             ;SET CARRY
9200 050310 060400          ADD      R4,R0
9201 050312 005516    M.DP42: ADC      @SP             ;SET CARRY

```

9202	050314	005716		TST	@SP		;TEST THE UPDATE INDICATOR
9203	050316	001401		BEQ	.+4		;IF 0,FORGET IT
9204	050320	005203		INC	R3		;NO CARRY POSSIBLE HERE
9205	050322	005366	000006	DEC	6(SP)		;DECREMENT CTR
9206	050326	003347		BGT	M.DP40		;BR IF MORE TO DO
9207	050330	006003		ROR	R3		
9208	050332	103404		BCS	M.DP44		
9209	050334	060501		ADD	R5,R1		
9210	050336	005500		ADC	R0		
9211	050340	060400		ADD	R4,R0		
9212	050342	000241		CLC			
9213							
9214	050344	006103		M.DP44: ROL	R3		
9215	050346	062706	000010	ADD	#10,SP		;ADJUST STACK BY 4 WORDS
9216	050352	000242		CLV			
9217	050354	000207		RTS	PC		
9218							
9219	050356	062706	000006	M.DP50: ADD	#6,SP		
9220	050362	000262		SEV			
9221	050364	000207		RTS	PC		
9222							

```
9223 .SBTTL SCOPE HANDLER ROUTINE
9224
9225 ::*****
9226 ::*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9227 ::*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
9228 ::*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
9229 ::*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9230 ::*SW14=1 LOOP ON TEST
9231 ::*SW11=1 INHIBIT ITERATIONS
9232 ::*SW09=1 LOOP ON ERROR
9233 ::*SW08=1 LOOP ON TEST IN SWR<7:0>
9234 ::*CALL
9235 ::* SCOPE ::SCOPE=IOT
9236
9237 $SCOPE:
9238 050366 104407 CKSWR ::TEST FOR CHANGE IN SOFT-SWR
9239 050370 032777 040000 130542 1$: BIT #BIT14,@SWR ::LOOP ON PRESENT TEST?
9240 050376 001114 BNE $OVER ::YES IF SW14=1
9241 :####START OF CODE FOR THE XOR TESTER####
9242 050400 000416 $XTSTR: BR 6$ ::IF RUNNING ON THE 'XOR' TESTER CHANGE
9243 ::THIS INSTRUCTION TO A 'NOP' (NOP=240)
9244 050402 013746 000004 MOV @#ERRVEC,-(SP) ::SAVE THE CONTENTS OF THE ERROR VECTOR
9245 050406 012737 050426 000004 MOV #5,@#ERRVEC ::SET FOR TIMEOUT
9246 050414 005737 177060 TST @#177060 ::TIME OUT ON XOR?
9247 050420 012637 000004 MOV (SP)+,@#ERRVEC ::RESTORE THE ERROR VECTOR
9248 050424 000463 BR $SVLAD ::GO TO THE NEXT TEST
9249 050426 022626 5$: CMP (SP)+,(SP)+ ::CLEAR THE STACK AFTER A TIME OUT
9250 050430 012637 000004 MOV (SP)+,@#ERRVEC ::RESTORE THE ERROR VECTOR
9251 050434 000423 BR 7$ ::LOOP ON THE PRESENT TEST
9252 050436 6$:####END OF CODE FOR THE XOR TESTER####
9253 050436 032777 000400 130474 BIT #BIT08,@SWR ::LOOP ON SPEC. TEST?
9254 050444 C01404 BEQ 2$ ::BR IF NO
9255 050446 127737 130466 001102 CMPB @SWR,$TSTNM ::ON THE RIGHT TEST? SWR<7:0>
9256 050454 001465 BEQ $OVER ::BR IF YES
9257 050456 105737 001103 2$: TSTB $ERFLG ::HAS AN ERROR OCCURRED?
9258 050462 001421 BEQ 3$ ::BR IF NO
9259 050464 123737 001115 001103 CMPB $ERMAX,$ERFLG ::MAX. ERRORS FOR THIS TEST OCCURRED?
9260 050472 101015 BHI 3$ ::BR IF NO
9261 050474 032777 001000 130436 BIT #BIT09,@SWR ::LOOP ON ERROR?
9262 050502 001404 BEQ 4$ ::BR IF NO
9263 050504 013737 001110 001106 7$: MOV $LPERR,$LPADR ::SET LOOP ADDRESS TO LAST SCOPE
9264 050512 000446 BR $OVER
9265 050514 105037 001103 4$: CLRB $ERFLG ::ZERO THE ERROR FLAG
9266 050520 005037 001174 CLR $TIMES ::CLEAR THE NUMBER OF ITERATIONS TO MAKE
9267 050524 000415 BR 1$ ::ESCAPE TO THE NEXT TEST
9268 050526 032777 004000 130404 3$: BIT #BIT11,@SWR ::INHIBIT ITERATIONS?
9269 050534 001011 BNE 1$ ::BR IF YES
9270 050536 005737 001216 TST $PASS ::IF FIRST PASS OF PROGRAM
9271 050542 001406 BEQ 1$ :: INHIBIT ITERATIONS
9272 050544 005237 001104 INC $ICNT ::INCREMENT ITERATION COUNT
9273 050550 023737 001174 001104 CMP $TIMES,$ICNT ::CHECK THE NUMBER OF ITERATIONS MADE
9274 050556 002024 BGE $OVER ::BR IF MORE ITERATION REQUIRED
9275 050560 012737 000001 001104 1$: MOV #1,$ICNT ::REINITIALIZE THE ITERATION COUNTER
9276 050566 013737 050644 001174 MOV $MXCNT,$TIMES ::SET NUMBER OF ITERATIONS TO DO
9277 050574 105237 001102 $SVLAD: INCB $TSTNM ::COUNT TEST NUMBERS
9278 050600 113737 001102 001214 MOVB $TSTNM,$TESTN ::SET TEST NUMBER IN APT MAILBOX
```

```
9279 050606 011637 001106      MOV      (SP), $LPADR      ;;SAVE SCOPE LOOP ADDRESS
9280 050612 011637 001110      MOV      (SP), $LPERR     ;;SAVE ERROR LOOP ADDRESS
9281 050616 005037 001176      CLR      $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
9282 050622 112737 000001 001115  MOVVB   #1, $ERMAX        ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
9283 050630 013777 001102 130304 $OVER:  MOV      $STNM, @DISPLAY ;;DISPLAY TEST NUMBER
9284 050636 013716 001106      MOV      $LPADR, (SP)    ;;FUDGE RETURN ADDRESS
9285 050642 000002          RTI                       ;;FIXES PS
9286 050644 003720          $MXCNT: 2000.           ;;MAX. NUMBER OF ITERATIONS
9287          .SBTTL  ERROR HANDLER ROUTINE
9288
9289          ;;*****
9290          ;;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
9291          ;;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9292          ;;AND GO TO TYPERR ON ERROR
9293          ;;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9294          ;;*SW15=1      HALT ON ERROR
9295          ;;*SW13=1      INHIBIT ERROR TYPEOUTS
9296          ;;*SW10=1     BELL ON ERROR
9297          ;;*SW09=1     LOOP ON ERROR
9298          ;;*CALL
9299          ;;*      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
9300
9301          $ERROR:
9302 050646 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
9303 050650 105237 001103 7$:      INCB          $ERFLG      ;;SET THE ERROR FLAG
9304 050654 001775          BEQ          7$          ;;DON'T LET THE FLAG GO TO ZERO
9305 050656 013777 001102 130256      MOV      $STNM, @DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
9306 050664 032777 002000 130246      BIT      #BIT10, @SWR    ;;BELL ON ERROR?
9307 050672 001402          BEQ          1$          ;;NO - SKIP
9308 050674 104401 001200          TYPE      , $BELL      ;;RING BELL
9309 050700 005237 001112 1$:      INC          $ERTTL     ;;COUNT THE NUMBER OF ERRORS
9310 050704 011637 001116          MOV      (SP), $ERRPC   ;;GET ADDRESS OF ERROR INSTRUCTION
9311 050710 162737 000002 001116      SUB      #2, $ERRPC
9312 050716 117737 130174 001114      MOVVB   @ $ERRPC, $ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
9313 050724 032777 020000 130206      BIT      #BIT13, @SWR   ;;SKIP TYPEOUT IF SET
9314 050732 001004          BNE          20$        ;;SKIP TYPEOUTS
9315 050734 004737 067004          JSR      PC, TYPERR    ;;GO TO USER ERROR ROUTINE
9316 050740 104401 001205          TYPE      , $CRLF
9317 050744          20$:
9318 050744 122737 000001 001230      CMPB    #APTENV, $ENV   ;;RUNNING IN APT MODE
9319 050752 001007          BNE          2$        ;;NO, SKIP APT ERROR REPORT
9320 050754 113737 001114 050766      MOVVB   $ITEMB, 21$    ;;SET ITEM NUMBER AS ERROR NUMBER
9321 050762 004737 051644          JSR      PC, $ATY4     ;;REPORT FATAL ERROR TO APT
9322 050766 000          21$:      .BYTE    0
9323 050767 000          .BYTE    0
9324 050770 000777          BR       22$          ;;APT ERROR LOOP
9325 050772 005777 130142 22$:      BR       @SWR         ;;HALT ON ERROR
9326 050776 100002          BPL      3$          ;;SKIP IF CONTINUE
9327 051000 000000          HALT          ;;HALT ON ERROR!
9328 051002 104407          CKSWR
9329 051004 032777 001000 130126 3$:      BIT      #BIT09, @SWR  ;;TEST FOR CHANGE IN SOFT-SWR
9330 051012 001402          BEQ      4$          ;;LOOP ON ERROR SWITCH SET?
9331 051014 013716 001110          MOV      $LPERR, (SP)  ;;BR IF NO
9332 051020 005737 001176          TST     $ESCAPE       ;;FUDGE RETURN FOR LOOPING
9333 051024 001402          BEQ      5$          ;;CHECK FOR AN ESCAPE ADDRESS
9334 051026 013716 001176          MOV     $ESCAPE, (SP) ;;BR IF NONE
9334          ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```

9335 051032
9336 051032 022737 042772 000042 5$:      CMP      #SENDAD,@#42      ;;ACT-11 AUTO-ACCEPT?
9337 051040 001001                      BNE      6$              ;;BRANCH IF NO
9338 051042 000000                      HALT                      ;;YES
9339 051044
9340 051044 000002 6$:      RTI              ;;RETURN
9341
9342 .SBTTL  TYPE ROUTINE
9343
9344 *****
9345 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
9346 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
9347 *NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
9348 *NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
9349 *NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
9350
9351 *CALL:
9352 *1) USING A TRAP INSTRUCTION
9353 *      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
9354 *OR
9355 *      TYPE
9356 *      MESADR
9357 *
9358 051046 105737 001157 $TYPE:  TSTB      $TPFLG      ;;IS THERE A TERMINAL?
9359 051052 10C002          BPL      1$              ;;BR IF YES
9360 051054 000000          HALT                      ;;HALT HERE IF NO TERMINAL
9361 051056 000430          BR              ;;LEAVE
9362 051060 010046 1$:      MOV      RO,-(SP)      ;;SAVE RO
9363 051062 017600 000002  MOV      @2(SP),RO      ;;GET ADDRESS OF ASCIZ STRING
9364 051066 122737 000001 001230  CMPB     #APTENV,$ENV    ;;RUNNING IN APT MODE
9365 051074 001011          BNE      62$           ;;NO,GO CHECK FOR APT CONSOLE
9366 051076 132737 000100 001231  BITB     #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
9367 051104 001405          BEQ      62$           ;;NO,GO CHECK FOR CONSOLE
9368 051106 010037 051116  MOV      RO,61$         ;;SETUP MESSAGE ADDRESS FOR APT
9369 051112 004737 051634  JSR     PC,$ATY3       ;;SPOOL MESSAGE TO APT
9370 051116 000000          .WORD    0              ;;MESSAGE ADDRESS
9371 051120 132737 000040 001231  62$:    BITB     #APTCSUP,$ENVM ;;APT CONSOLE SUPPRESSED
9372 051126 001003          BNE      60$           ;;YES,SKIP TYPE OUT
9373 051130 112046 2$:      MOVB     (RO)+,-(SP)    ;;PUSH CHARACTER TO BE TYPED ONTO STACK
9374 051132 001005          BNE      4$           ;;BR IF IT ISN'T THE TERMINATOR
9375 051134 005726          TST     (SP)+         ;;IF TERMINATOR POP IT OFF THE STACK
9376 051136 012600 60$:    MOV      (SP)+,RO      ;;RESTORE RO
9377 051140 062716 000002 3$:      ADD      #2,(SP)      ;;ADJUST RETURN PC
9378 051144 000002          RTI              ;;RETURN
9379 051146 122716 000011 4$:      CMPB     #HT,(SP)      ;;BRANCH IF <HT>
9380 051152 001430          BEQ      8$           ;;BRANCH IF NOT <CRLF>
9381 051154 122716 000200  CMPB     #CRLF,(SP)
9382 051160 001006          BNE      5$           ;;POP <CR><LF> EQUIV
9383 051162 005726          TST     (SP)+         ;;TYPE A CR AND LF
9384 051164 104401          TYPE
9385 051166 001205          $CRLF
9386 051170 105037 051376  CLRB     $CHARCNT      ;;CLEAR CHARACTER COUNT
9387 051174 000755          BR              ;;GET NEXT CHARACTER
9388 051176 004737 051260 5$:      JSR     PC,$TYPEPC     ;;GO TYPE THIS CHARACTER
9389 051202 123726 001156 6$:      CMPB     $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
9390 051206 001350          BNE      2$           ;;IF NO GO GET NEXT CHAR.
  
```

```
9391 051210 013746 001154          MOV    $NULL,-(SP)      ;;GET # OF FILLER CHARS. NEEDED
9392                                ;;AND THE NULL CHAR.
9393 051214 105366 000001          7$:   DECB    1(SP)      ;;DOES A NULL NEED TO BE TYPED?
9394 051220 002770                    BLT    6$               ;;BR IF NO--GO POP THE NULL OFF OF STACK
9395 051222 004737 051260          JSR    PC,$TYPEC      ;;GO TYPE A NULL
9396 051226 105337 051376          DECB    $CHARCNT      ;;DO NOT COUNT AS A COUNT
9397 051232 000770                    BR     7$               ;;LOOP
9398
9399                                ;HORIZONTAL TAB PROCESSOR
9400
9401 051234 112716 000040          8$:   MOVB    #' ,(SP)   ;;REPLACE TAB WITH SPACE
9402 051240 004737 051260          9$:   JSR    PC,$TYPEC   ;;TYPE A SPACE
9403 051244 132737 000007 051376  BITB    #7,$CHARCNT    ;;BRANCH IF NOT AT
9404 051252 001372                    BNE    9$               ;;TAB STOP
9405 051254 005726                    TST    (SP)+           ;;POP SPACE OFF STACK
9406 051256 000724                    BR     2$               ;;GET NEXT CHARACTER
9407 051260
9408 051260 105777 127660          $TYPEC: TSTB    @ $TKS        ;;CHAR IN KYBD BUFFER?           :MJD001
9409 051264 100022                    BPL    10$              ;;BR IF NOT                       :MJD001
9410 051266 017746 127654          MOV    @ $TKB,-(SP)    ;;GET CHAR                         :MJD001
9411 051272 042716 177600          BIC    #177600,(SP)   ;;STRIP EXTRANEIOUS BITS          :MJD001
9412 051276 122716 000023          CMPB    # $XOFF,(SP)  ;;WAS CHAR XOFF                   :MJD001
9413 051302 001012                    BNE    102$            ;;BR IF NOT                       :MJD001
9414 051304
9415 051304 105777 127634          101$: TSTB    @ $TKS        ;;WAIT FOR CHAR                    :MJD001
9416 051310 100375                    BPL    101$            ;;BR IF NOT                       :MJD001
9417 051312 117716 127630          MOVB    @ $TKB,(SP)   ;;GET CHAR                         :MJD001
9418 051316 042716 177600          BIC    #177600,(SP)   ;;STRIP IT                         :MJD001
9419 051322 122716 000021          CMPB    # $XON,(SP)  ;;WAS IT XON?                    :MJD001
9420 051326 001366                    BNE    101$            ;;BR IF NOT                       :MJD001
9421 051330
9422 051330 005726                    102$: TST    (SP)+       ;;FIX STACK                        :MJD001
9423 051332
9424 051332 105777 127612          10$:   TSTB    @ $TPS        ;;WAIT UNTIL PRINTER IS READY     :MJD001
9425 051336 100375                    BPL    10$              ;;BR IF NOT                       :MJD001
9426 051340 116677 000002 127604  MOVB    2(SP),@ $TPB   ;;LOAD CHAR TO BE TYPED INTO DATA REG.
9427 051346 122766 000015 000002  CMPB    #CR,2(SP)     ;;IS CHARACTER A CARRIAGE RETURN?
9428 051354 001003                    BNE    1$               ;;BRANCH IF NO
9429 051356 105037 051376          CLRB    $CHARCNT      ;;YES--CLEAR CHARACTER COUNT
9430 051362 000406                    BR     $TYPEX          ;;EXIT
9431 051364 122766 000012 000002  1$:   CMPB    #LF,2(SP)  ;;IS CHARACTER A LINE FEED?
9432 051372 001402                    BEQ    $TYPEX          ;;BRANCH IF YES
9433 051374 105227                    INCB    (PC)+          ;;COUNT THE CHARACTER
9434 051376 000000          $CHARCNT: .WORD    0  ;;CHARACTER COUNT STORAGE
9435 051400 000207          $TYPEX: RTS    PC
9436
9437                                .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
9438
9439                                ;*****
9440                                ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
9441                                ;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
9442                                ;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
9443                                ;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
9444                                ;REPLACED WITH SPACES.
9445                                ;*CALL:
9446                                ;*   MOV    NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
```

```

9447          ;*      TYPDS          ;;GO TO THE ROUTINE
9448
9449 051402          $TYPDS:
9450 051402 010046      MOV      R0,-(SP)          ;;PUSH R0 ON STACK
9451 051404 010146      MOV      R1,-(SP)          ;;PUSH R1 ON STACK
9452 051406 010246      MOV      R2,-(SP)          ;;PUSH R2 ON STACK
9453 051410 010346      MOV      R3,-(SP)          ;;PUSH R3 ON STACK
9454 051412 010546      MOV      R5,-(SP)          ;;PUSH R5 ON STACK
9455 051414 012746 020200  MOV      #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
9456 051420 016605 000020  MOV      20(SP),R5        ;;GET THE INPUT NUMBER
9457 051424 100004          BPL      1$                ;;BR IF INPUT IS POS.
9458 051426 005405          NEG      R5                ;;MAKE THE BINARY NUMBER POS.
9459 051430 112766 000055 000001  MOVVB   #'-,1(SP)        ;;MAKE THE ASCII NUMBER NEG.
9460 051436 005000          1$:    CLR      R0                ;;ZERO THE CONSTANTS INDEX
9461 051440 012703 051616      MOV      #SDBLK,R3        ;;SETUP THE OUTPUT POINTER
9462 051444 112723 000040      MOVVB   #' ,(R3)+        ;;SET THE FIRST CHARACTER TO A BLANK
9463 051450 005002          2$:    CLR      R2                ;;CLEAR THE BCD NUMBER
9464 051452 016001 051606      MOV      $DTBL(R0),R1     ;;GET THE CONSTANT
9465 051456 160105          3$:    SUB      R1,R5            ;;FORM THIS BCD DIGIT
9466 051460 002402          BLT     4$                ;;BR IF DONE
9467 051462 005202          INC     R2                ;;INCREASE THE BCD DIGIT BY 1
9468 051464 000774          BR      3$
9469 051466 060105          4$:    ADD      R1,R5            ;;ADD BACK THE CONSTANT
9470 051470 005702          TST     R2                ;;CHECK IF BCD DIGIT=0
9471 051472 001002          BNE     5$                ;;FALL THROUGH IF 0
9472 051474 105716          TSTB   (SP)              ;;STILL DOING LEADING 0'S?
9473 051476 100407          BMI     7$                ;;BR IF YES
9474 051500 106316          5$:    ASLB   (SP)              ;;MSD?
9475 051502 103003          BCC     6$                ;;BR IF NO
9476 051504 116663 000001 177777  MOVVB   1(SP),-1(R3)     ;;YES--SET THE SIGN
9477 051512 052702 000060 6$:    BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
9478 051516 052702 000040 7$:    BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
9479 051522 110223          MOVVB   R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
9480 051524 005720          TST     (R0)+            ;;JUST INCREMENTING
9481 051526 020027 000010      CMP     R0,#10          ;;CHECK THE TABLE INDEX
9482 051532 002746          BLT     2$                ;;GO DO THE NEXT DIGIT
9483 051534 003002          BGT     8$                ;;GO TO EXIT
9484 051536 010502          MOV     R5,R2            ;;GET THE LSD
9485 051540 000764          BR      6$                ;;GO CHANGE TO ASCII
9486 051542 105726          8$:    TSTB   (SP)+          ;;WAS THE LSD THE FIRST NON-ZERO?
9487 051544 100003          BPL     9$                ;;BR IF NO
9488 051546 116663 177777 177776  MOVVB   -1(SP),-2(R3)    ;;YES--SET THE SIGN FOR TYPING
9489 051554 105013          9$:    CLRB   (R3)            ;;SET THE TERMINATOR
9490 051556 012605          MOV     (SP)+,R5        ;;POP STACK INTO R5
9491 051560 012603          MOV     (SP)+,R3        ;;POP STACK INTO R3
9492 051562 012602          MOV     (SP)+,R2        ;;POP STACK INTO R2
9493 051564 012601          MOV     (SP)+,R1        ;;POP STACK INTO R1
9494 051566 012600          MOV     (SP)+,R0        ;;POP STACK INTO R0
9495 051570 104401 051616      TYPE   ,SDBLK           ;;NOW TYPE THE NUMBER
9496 051574 016666 000002 000004  MOV     2(SP),4(SP)      ;;ADJUST THE STACK
9497 051602 012616          MOV     (SP)+,(SP)
9498 051604 000002          RTI
9499 051606 023420          $DTBL: 10000.
9500 051610 001750          1000.
9501 051612 000144          100.
9502 051614 000012          10.

```



```
9503 051616 000004 $DBLK: .BLKW 4
9504 .SBTTL APT COMMUNICATIONS ROUTINE
9505
9506
9507 051626 112737 000001 052072 $ATY1: MOVB #1,$FFLG ;;TO REPORT FATAL ERROR
9508 051634 112737 000001 052070 $ATY3: MOVB #1,$MFLG ;;TO TYPE A MESSAGE
9509 051642 000403 BR $ATYC
9510 051644 112737 000001 052072 $ATY4: MOVB #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
9511 051652 $ATYC:
9512 051652 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
9513 051654 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
9514 051656 105737 052070 TSTB $MFLG ;;SHOULD TYPE A MESSAGE?
9515 051662 001450 BEQ 5$ ;;IF NOT: BR
9516 051664 122737 000001 001230 CMPB #APTENV,$ENV ;;OPERATING UNDER APT?
9517 051672 001031 BNE 3$ ;;IF NOT: BR
9518 051674 132737 000100 001231 BITB #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
9519 051702 001425 BEQ 3$ ;;IF NOT: BR
9520 051704 017600 000004 MOV @4(SP),R0 ;;GET MESSAGE ADDR.
9521 051710 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
9522 051716 005737 001210 1$: TST $MSGTYPE ;;SEE IF DONE W/ LAST XMISSION?
9523 051722 001375 BNE 1$ ;;IF NOT: WAIT
9524 051724 010037 001224 MOV R0,$MSGAD ;;PUT ADDR IN MAILBOX
9525 051730 105720 2$: TSTB (R0)+ ;;FIND END OF MESSAGE
9526 051732 001376 BNE 2$
9527 051734 163700 001224 SUB $MSGAD,R0 ;;SUB START OF MESSAGE
9528 051740 006200 ASR R0 ;;GET MESSAGE LNGTH IN WORDS
9529 051742 010037 001226 MOV R0,$MSGGLT ;;PUT LENGTH IN MAILBOX
9530 051746 012737 000004 001210 MOV #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
9531 051754 000413 BR 5$
9532 051756 017637 000004 052002 3$: MOV @4(SP),4$ ;;PUT MSG ADDR IN JSR LINKAGE
9533 051764 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDRESS
9534 051772 013746 177776 MOV 177776,-(SP) ;;PUSH 177776 ON STACK
9535 051776 004737 051046 JSR PC,$TYPE ;;CALL TYPE MACRO
9536 052002 000000 4$: .WORD 0
9537 052004 5$:
9538 052004 105737 052072 10$: TSTB $FFLG ;;SHOULD REPORT FATAL ERROR?
9539 052010 001416 BEQ 12$ ;;IF NOT: BR
9540 052012 005737 001230 TST $ENV ;;RUNNING UNDER APT?
9541 052016 001413 BEQ 12$ ;;IF NOT: BR
9542 052020 005737 001210 11$: TST $MSGTYPE ;;FINISHED LAST MESSAGE?
9543 052024 001375 BNE 11$ ;;IF NOT: WAIT
9544 052026 017637 000004 001212 MOV @4(SP),$FATAL ;;GET ERROR #
9545 052034 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
9546 052042 005237 001210 INC $MSGTYPE ;;TELL APT TO TAKE ERROR
9547 052046 105037 052072 12$: CLRB $FFLG ;;CLEAR FATAL FLAG
9548 052052 105037 052071 CLRB $LFLG ;;CLEAR LOG FLAG
9549 052056 105037 052070 CLRB $MFLG ;;CLEAR MESSAGE FLAG
9550 052062 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
9551 052064 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
9552 052066 000207 RTS PC ;;RETURN
9553 052070 000 $MFLG: .BYTE 0 ;;MESSG. FLAG
9554 052071 000 $LFLG: .BYTE 0 ;;LOG FLAG
9555 052072 000 $FFLG: .BYTE 0 ;;FATAL FLAG
9556 052074 .EVEN
9557 000200 APTSIZE=200
9558 000001 APTENV=001
```



```
9615 052232 005704          TST      R4          ;;SUPPRESS THIS 0?
9616 052234 001403          BEQ      5$          ;;BR IF YES
9617 052236 005204          4$: INC      R4          ;;DON'T SUPPRESS ANYMORE 0'S
9618 052240 052703 000060    BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
9619 052244 052703 000040    5$: BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY
9620 052250 110337 052314    MOV      R3,8$       ;;SAVE FOR TYPING
9621 052254 104401 052314    TYPE     8$         ;;GO TYPE THIS DIGIT
9622 052260 105337 052316    7$: DECB    $OCNT     ;;COUNT BY 1
9623 052264 003347          BGT      2$         ;;BR IF MORE TO DO
9624 052266 002402          BLT      6$         ;;BR IF DONE
9625 052270 005204          INC      R4         ;;INSURE LAST DIGIT ISN'T A BLANK
9626 052272 000744          BR       2$         ;;GO DO THE LAST DIGIT
9627 052274 012605          6$: MOV      (SP)+,R5  ;;RESTORE R5
9628 052276 012604          MOV      (SP)+,R4  ;;RESTORE R4
9629 052300 012603          MOV      (SP)+,R3  ;;RESTORE R3
9630 052302 016666 000002 000004  MOV      2(SP),4(SP) ;;SET THE STACK FOR RETURNING
9631 052310 012616          MOV      (SP)+,(SP)
9632 052312 000002          RTI             ;;RETURN
9633 052314          000          8$: .BYTE    0          ;;STORAGE FOR ASCII DIGIT
9634 052315          000          .BYTE    0          ;;TERMINATOR FOR TYPE ROUTINE
9635 052316          000          $OCNT: .BYTE    0          ;;OCTAL DIGIT COUNTER
9636 052317          000          $OFILL: .BYTE    0          ;;ZERO FILL SWITCH
9637 052320 000000          $OMODE: .WORD    0          ;;NUMBER OF DIGITS TO TYPE
9638          .SBTTL  TTY INPUT ROUTINE
9639
9640          ;;*****
9641          .ENABL  LSB
9642 052322 000000          $TKCNT: .WORD    0          ;;NUMBER OF ITEMS IN QUEUE
9643 052324 000000          $TKQIN: .WORD    0          ;;INPUT POINTER
9644 052326 000000          $TKQOUT: .WORD   0          ;;OUTPUT POINTER
9645 052330 000001          $TKQSRV: .BLKB   1          ;;TTY KEYBOARD QUEUE
9646          052331          $TKQEND=.
9647          052332          .EVEN
9648
9649          ;*TK INITIALIZE ROUTINE
9650          ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
9651          ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
9652          .
9653          ;*CALL:
9654          ;*      JSR      PC,$TKINT
9655          ;*      RETURN
9656          .
9657 052332 005037 052322          $TKINT: CLR      $TKCNT     ;;CLEAR COUNT OF ITEMS IN QUEUE
9658 052336 012737 052330 052324  MOV      #$TKQSRV,$TKQIN  ;;MOVE THE STARTING ADDRESS OF THE
9659 052344 013737 052324 052326  MOV      $TKQIN,$TKQOUT   ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
9660 052352 012737 052402 000060  MOV      #$TKSRV,@#TKVEC  ;;INITIALIZE THE KEYBOARD VECTOR
9661 052360 012737 000200 000062  MOV      #200,@#TKVEC+2   ;;'BR' LEVEL 4
9662 052366 005777 126554          TST      @$TKB           ;;CLEAR DONE FLAG
9663 052372 012777 000100 126544  MOV      #100,@$TKS       ;;ENABLE TTY KEYBOARD INTERRUPT
9664 052400 000207          RTS      PC             ;;RETURN TO CALLER
9665
9666          ;*TK SERVICE ROUTINE
9667          ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
9668          ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
9669          ;*IT IN THE QUEUE.
9670          ;*IF THE CHARACTER IS A "CONTROL-C" (^C) $TKINT IS CALLED AND
```

```
9671 ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
9672 .
9673 $TKSRV: MOVB @STKB, -(SP) ;: PICKUP THE CHARACTER
9674 BIC #^C177, (SP) ;: STRIP THE JUNK
9675 CMP (SP), #SXON ;: IS IT A RANDOM XON? ;:RAN001
9676 BNE 30$ ;: BRANCH IF NO ;:RAN001
9677 TST (SP)+ ;: CLEAN RANDOM XON OFF STACK ;:RAN001
9678 RTI ;: RETURN ;:RAN001
9679 30$:
9680 CMP (SP), #3 ;: IS IT A CONTROL C?
9681 BNE 1$ ;: BRANCH IF NO
9682 TYPE ,SCNTLC ;: TYPE A CONTROL-C (^C)
9683 JSR PC, $TKINT ;: INIT THE KEYBOARD
9684 TST (SP)+ ;: CLEAN UP STACK
9685 JMP STOP ;: CONTROL C RESTART
9686 1$: CMP (SP), #7 ;: IS IT A CONTROL G?
9687 BNE 2$ ;: BRANCH IF NO
9688 CMP #SWREG, SWR ;: IS SOFT-SWR SELECTED?
9689 BEQ 6$ ;: GO TO SWR CHANGE
9690
9691 2$:
9692 CMP #1, $TKCNT ;: IS THE QUEUE FULL?
9693 BNE 3$ ;: BRANCH IF NO
9694 TYPE ,SBELL ;: RING THE TTY BELL
9695 TST (SP)+ ;: CLEAN CHARACTER OFF OF STACK
9696 BR 5$ ;: EXIT
9697 3$: CMP (SP), #23 ;: IS IT A CONTROL-S?
9698 BNE 32$ ;: BRANCH IF NO
9699 CLR @STKS ;: DISABLE TTY KEYBOARD INTERRUPTS
9700 TST (SP)+ ;: CLEAN CHAR OFF STACK
9701 31$: TSTB @STKS ;: WAIT FOR A CHAR
9702 BPL 31$ ;: LOOP UNTIL ITS THERE
9703 MOVB @STKB, -(SP) ;: GET THE CHARACTER
9704 BIC #^C177, (SP) ;: MAKE IT 7-BIT ASCII
9705 CMP (SP)+, #21 ;: IS IT A CONTROL-Q?
9706 BNE 31$ ;: BRANCH IF NO
9707 MOV #100, @STKS ;: REENABLE TTY KEYBOARD INTERRUPTS
9708 RTI ;: RETURN
9709 32$: INC $TKCNT ;: COUNT THIS CHARACTER
9710 CMP (SP), #140 ;: IS IT UPPER CASE?
9711 BLT 4$ ;: BRANCH IF YES
9712 CMP (SP), #175 ;: IS IT A SPECIAL CHAR?
9713 BGT 4$ ;: BRANCH IF YES
9714 BIC #40, (SP) ;: MAKE IT UPPER CASE
9715 4$: MOVB (SP)+, @STKQIN ;: AND PUT IT IN QUEUE
9716 INC $TKQIN ;: UPDATE THE POINTER
9717 CMP $TKQIN, #STKQEND ;: GO OFF THE END?
9718 BNE 5$ ;: BRANCH IF NO
9719 MOV #STKQSR, $TKQIN ;: RESET THE POINTER
9720 5$: RTI ;: RETURN
9721
9722 ;:*****
9723 ;: *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
9724 ;: *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
9725 ;: *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
9726 ;: *CALL WHEN OPERATING IN TTY INTERRUPT MODE.
```

```

9727 052632 022737 000176 001140 $CKSWR: CMP #SWREG,SWR      ;; IS THE SOFT-SWR SELECTED
9728 052640 001124          BNE 15$           ;; EXIT IF NOT
9729 052642 105777 126276          TSTB @STKS        ;; IS A CHAR WAITING?
9730 052646 100121          BPL 15$           ;; IF NOT, EXIT
9731 052650 117746 126272          MOVB @STKB,-(SP)   ;; YES
9732 052654 042716 177600          BIC #^C177,(SP)  ;; MAKE IT 7-BIT ASCII
9733 052660 021627 000007          CMP (SP),#7      ;; IS IT A CONTROL-G?
9734 052664 001300          BNE 2$           ;; IF NOT, PUT IT IN THE TTY QUEUE
9735                                     ;; AND EXIT
9736
9737
9738
9739
9740

```

```

*****
;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.

```

```

9741 052666 123727 001134 000001 6$:  CMPB $AUTOB,#1    ;; ARE WE RUNNING IN AUTO-MODE?
9742 052674 001674          BEQ 2$           ;; BRANCH IF YES
9743 052676 005726          TST (SP)+       ;; CLEAR CONTROL-G OFF STACK
9744 052700 004737 052332          JSR PC,$TKINT   ;; FLUSH THE TTY INPUT QUEUE
9745 052704 005077 126234          CLR @STKS       ;; DISABLE TTY KEYBOARD INTERRUPTS
9746 052710 112737 000001 001135          MOVB #1,$INTAG  ;; SET INTERRUPT MODE INDICATOR
9747

```

```

9748 052716 104401 053554          TYPE .$CNTLG    ;; ECHO THE CONTROL-G (^G)
9749 052722 104401 053561          $GTSWR: TYPE .$MSWR  ;; TYPE CURRENT CONTENTS
9750 052726 013746 000176          MOV SWREG,-(SP) ;; SAVE SWREG FOR TYPEOUT
9751 052732 104402          TYPOC          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9752 052734 104401 053572          TYPE .$MNEW    ;; PROMPT FOR NEW SWR
9753 052740 005046          19$: CLR -(SP)   ;; CLEAR COUNTER
9754 052742 005046          CLR -(SP)      ;; THE NEW SWR
9755 052744 105777 126174          7$: TSTB @STKS   ;; CHAR THERE?
9756 052750 100375          BPL 7$         ;; IF NOT TRY AGAIN
9757

```

```

9758 052752 117746 126170          MOVB @STKB,-(SP) ;; PICK UP CHAR
9759 052756 042716 177600          BIC #^C177,(SP) ;; MAKE IT 7-BIT ASCII
9760
9761 052762 021627 000003          CMP (SP),#3     ;; IS IT A CONTROL-C?
9762 052766 001015          BNE 9$         ;; BRANCH IF NOT
9763 052770 104401 053542          TYPE .$CNTLC   ;; YES, ECHO CONTROL-C (^C)
9764 052774 062706 000006          ADD #6,SP      ;; CLEAN UP STACK
9765 053000 123727 001135 000001          CMPB $INTAG,#1 ;; REENABLE TTY KEYBOARD INTERRUPTS?
9766 053006 001003          BNE 8$        ;; BRANCH IF NO
9767 053010 012777 000100 126126          MOV #100,@STKS ;; ALLOW TTY KEYBOARD INTERRUPTS
9768 053016 000137 047330          8$: JMP STOP    ;; CONTROL-C RESTART
9769
9770

```

```

9771 053022 021627 000025          9$:  CMP (SP),#25   ;; IS IT A CONTROL-U?
9772 053026 001005          BNE 10$        ;; BRANCH IF NOT
9773 053030 104401 053547          TYPE .$CNTLU   ;; YES, ECHO CONTROL-U (^U)
9774 053034 062706 000006          20$: ADD #6,SP   ;; IGNORE PREVIOUS INPUT
9775 053040 000737          BR 19$        ;; LET'S TRY IT AGAIN
9776
9777

```

```

9778 053042 021627 000015          10$: CMP (SP),#15   ;; IS IT A <CR>?
9779 053046 001022          BNE 16$        ;; BRANCH IF NO
9780 053050 005766 000004          TST 4(SP)      ;; YES, IS IT THE FIRST CHAR?
9781 053054 001403          BEQ 11$        ;; BRANCH IF YES
9782 053056 016677 000002 126054          MOV 2(SP),@SWR ;; SAVE NEW SWR

```

```

9783 053064 062706 000006 11$: ADD #6,SP          ;;CLEAR UP STACK
9784 053070 104401 001205 14$: TYPE $CRLF        ;;ECHO <CR> AND <LF>
9785 053074 123727 001135 000001 CMPB $INTAG,#1        ;;RE-ENABLE TTY KBD INTERRUPTS?
9786 053102 001003 BNE 15$              ;;BRANCH IF NOT
9787 053104 012777 000100 126032 MOV #100,@$TKS        ;;RE-ENABLE TTY KBD INTERRUPTS
9788 053112 000002 15$: RTI                          ;;RETURN
9789 053114 004737 051260 16$: JSR PC,$TYPEC        ;;ECHO CHAR
9790 053120 021627 000060 CMP (SP),#60          ;;CHAR < 0?
9791 053124 002420 BLT 18$              ;;BRANCH IF YES
9792 053126 021627 000067 CMP (SP),#67          ;;CHAR > 7?
9793 053132 003015 BGT 18$              ;;BRANCH IF YES
9794 053134 042726 000060 BIC #60,(SP)+        ;;STRIP-OFF ASCII
9795 053140 005766 000002 TST 2(SP)            ;;IS THIS THE FIRST CHAR
9796 053144 001403 BEQ 17$              ;;BRANCH IF YES
9797 053146 006316 ASL (SP)              ;;NO, SHIFT PRESENT
9798 053150 006316 ASL (SP)              ;; CHAR OVER TO MAKE
9799 053152 006316 ASL (SP)              ;; ROOM FOR NEW ONE.
9800 053154 005266 000002 17$: INC 2(SP)            ;;KEEP COUNT OF CHAR
9801 053160 056616 177776 BIS -2(SP), (SP)     ;;SET IN NEW CHAR
9802 053164 000667 BR 7$                ;;GET THE NEXT ONE
9803 053166 104401 001204 18$: TYPE $QUES          ;;TYPE ?<CR><LF>
9804 053172 000720 BR 20$              ;;SIMULATE CONTROL-U
9805 .DSABL LSB
    
```

```

9808 *****
9809 *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
9810 *CALL:
9811 * RDCHR          ;;GET A CHARACTER FROM THE QUEUE
9812 * RETURN HERE    ;;CHARACTER IS ON THE STACK
9813 *               ;;WITH PARITY BIT STRIPPED OFF
9814 *
9815 *
    
```

```

9816 053174 011646 000004 000002 $RDCHR: MOV (SP),-(SP)    ;;PUSH DOWN THE PC AND
9817 053176 016666 000004 MOV 4(SP),2(SP)        ;;THE PS
9818 053204 005066 000004 CLR 4(SP)            ;;GET READY FOR A CHARACTER
9819 053210 005046 CLR -(SP)           ;;PUT NEW PS ON STACK
9820 053212 012746 053220 MOV #64$,-(SP)        ;;PUT NEW PC ON STACK
9821 053216 000002 RTI                          ;;POP NEW PC AND PS
9822 053220 64$:
9823 053220 005737 052322 1$: TST $TKCNT          ;;WAIT ON A CHARACTER
9824 053224 001775 BEQ 1$
9825 053226 005337 052322 DEC $TKCNT          ;;DECREMENT THE COUNTER
9826 053232 117766 177070 000004 MOVB @$TKQOUT,4(SP)    ;;GET ONE CHARACTER
9827 053240 005237 052326 INC $TKQOUT          ;;UPDATE THE POINTER
9828 053244 023727 052326 052331 CMP $TKQOUT,$$TKQEND ;;DID IT GO OFF OF THE END?
9829 053252 001003 BNE 2$                ;;BRANCH IF NO
9830 053254 012737 052330 052326 MOV $$TKQSRT,$TKQOUT ;;RESET THE POINTER
9831 053262 000002 2$: RTI                          ;;RETURN
    
```

```

9832 *****
9833 *THIS ROUTINE WILL INPUT A STRING FROM THE TTY
9834 *CALL:
9835 * RDLIN          ;;INPUT A STRING FROM THE TTY
9836 * RETURN HERE    ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
9837 *               ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
9838 *
    
```

```
9839 053264 010346          $RDLIN: MOV      R3,-(SP)          ;;SAVE R3
9840 053266 005046          CLR      -(SP)          ;;CLEAR THE RUBOUT KEY
9841 053270 012703 053520      1$:  MOV      $TTYIN,R3      ;;GET ADDRESS
9842 053274 022703 053542      2$:  CMP      $TTYIN+22,R3    ;;BUFFER FULL?
9843 053300 101456          BLOS     4$             ;;BR IF YES
9844 053302 104410          RDCHR                    ;;GO READ ONE CHARACTER FROM THE TTY
9845 053304 112613          MOVB    (SP)+,(R3)      ;;GET CHARACTER
9846 053306 122713 000177      10$: CMPB    #177,(R3)      ;;IS IT A RUBOUT
9847 053312 001022          BNE     5$             ;;BR IF NO
9848 053314 005716          TST     (SP)           ;;IS THIS THE FIRST RUBOUT?
9849 053316 001007          BNE     6$             ;;BR IF NO
9850 053320 112737 000134 053516      MOVB    #' \,9$        ;;TYPE A BACK SLASH
9851 053326 104401 053516          TYPE    ,9$
9852 053332 012716 177777          MOV     #-1,(SP)       ;;SET THE RUBOUT KEY
9853 053336 005303          6$:  DEC     R3          ;;BACKUP BY ONE
9854 053340 020327 053520      CMP     R3,$TTYIN      ;;STACK EMPTY?
9855 053344 103434          BLO     4$             ;;BR IF YES
9856 053346 111337 053516      MOVB    (R3),9$        ;;SETUP TO TYPEOUT THE DELETED CHAR.
9857 053352 104401 053516          TYPE    ,9$
9858 053356 000746          BR      2$             ;;GO TYPE
9859 053360 005716          5$:  TST     (SP)           ;;GO READ ANOTHER CHAR.
9860 053362 001406          BEQ     7$             ;;RUBOUT KEY SET?
9861 053364 112737 000134 053516      MOVB    #' \,9$        ;;BR IF NO
9862 053372 104401 053516          TYPE    ,9$           ;;TYPE A BACK SLASH
9863 053376 005016          CLR     (SP)           ;;CLEAR THE RUBOUT KEY
9864 053400 122713 000025      7$:  CMPB    #25,(R3)      ;;IS CHARACTER A CTRL U?
9865 053404 001003          BNE     8$             ;;BR IF NO
9866 053406 104401 053547          TYPE    ,SCNTLU       ;;TYPE A CONTROL 'U'
9867 053412 000726          BR      1$             ;;GO START OVER
9868 053414 122713 000022      8$:  CMPB    #22,(R3)      ;;IS CHARACTER A '^R'?
9869 053420 001011          BNE     3$             ;;BRANCH IF NO
9870 053422 105013          CLRB   (R3)           ;;CLEAR THE CHARACTER
9871 053424 104401 001205          TYPE    ,SCRLF        ;;TYPE A 'CR' & 'LF'
9872 053430 104401 053520          TYPE    ,TTYIN        ;;TYPE THE INPUT STRING
9873 053434 000717          BR      2$             ;;GO PICKUP ANOTHER CHACTER
9874 053436 104401 001204      4$:  TYPE    ,SQUES        ;;TYPE A '?'
9875 053442 000712          BR      1$             ;;CLEAR THE BUFFER AND LOOP
9876 053444 111337 053516      3$:  MOVB    (R3),9$        ;;ECHO THE CHARACTER
9877 053450 104401 053516          TYPE    ,9$
9878 053454 122723 000015          CMPB    #15,(R3)+     ;;CHECK FOR RETURN
9879 053460 001305          BNE     2$             ;;LOOP IF NOT RETURN
9880 053462 105063 177777          CLRB   -1(R3)        ;;CLEAR RETURN (THE 15)
9881 053466 104401 001206          TYPE    ,SLF          ;;TYPE A LINE FEED
9882 053472 005726          TST     (SP)+         ;;CLEAN RUBOUT KEY FROM THE STACK
9883 053474 012603          MOV     (SP)+,R3      ;;RESTORE R3
9884 053476 011646          MOV     (SP),-(SP)    ;;ADJUST THE STACK AND PUT ADDRESS OF THE
9885 053500 016666 000004 000002      MOV     4(SP),2(SP)    ;;FIRST ASCII CHARACTER ON IT
9886 053506 012766 053520 000004      MOV     $TTYIN,4(SP)
9887 053514 000002          RTI                    ;;RETURN
9888 053516 000          9$:  .BYTE   0             ;;STORAGE FOR ASCII CHAR. TO TYPE
9889 053517 000          .BYTE   0             ;;TERMINATOR
9890 053520 000022          $TTYIN: .BLKB  22      ;;RESERVE 22 BYTES FOR TTY INPUT
9891 053542 041536 005015 000      $CNTLC: .ASCIZ /^C/<15><12> ;;CONTROL 'C'
9892 053547 136 006525 000012      $CNTLU: .ASCIZ /^U/<15><12> ;;CONTROL 'U'
9893 053554 043536 005015 000      $CNTLG: .ASCIZ /^G/<15><12> ;;CONTROL 'G'
9894 053561 015 051412 051127      $MSWR:  .ASCIZ <15><12>/SWR = /
```

```

9895 053566 036440 000040
9896 053572 020040 042516 020127 $MNEW: .ASCIZ / NEW = /
9897 053600 020075 000
9898 053604
9899 .EVEN
9900 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
9901
9902 ::*****
9903 ::THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
9904 ::CHANGE IT TO BINARY.
9905 ::THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
9906 ::OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A '?' WILL BE TYPED
9907 ::FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
9908 ::THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
9909 ::CALL:
9910 ::* RDOCT ::READ AN OCTAL NUMBER
9911 ::* RETURN HERE ::LOW ORDER BITS ARE ON TOP OF THE STACK
9912 ::* ::HIGH ORDER BITS ARE IN $HIOCT
9913 053604 011646 $RDCCT: MOV (SP),-(SP) ::PROVIDE SPACE FOR THE
9914 053606 016666 000004 000002 MOV 4(SP),2(SP) ::INPUT NUMBER
9915 053614 010046 MOV R0,-(SP) ::PUSH R0 ON STACK
9916 053616 010146 MOV R1,-(SP) ::PUSH R1 ON STACK
9917 053620 010246 MOV R2,-(SP) ::PUSH R2 ON STACK
9918 053622 104411 1$: RDLIN ::READ AN ASCIZ LINE
9919 053624 012600 MOV (SP)+,R0 ::GET ADDRESS OF 1ST CHARACTER
9920 053626 010037 053732 MOV R0,5$ ::AND SAVE IT
9921 053632 005001 CLR R1 ::CLEAR DATA WORD
9922 053634 005002 CLR R2
9923 053636 112046 2$: MOV (R0)+,-(SP) ::PICKUP THIS CHARACTER
9924 053640 001420 BEQ 3$ ::IF ZERO GET OUT
9925 053642 122716 000060 CMPB #'0,(SP) ::MAKE SURE THIS CHARACTER
9926 053646 003026 BGT 4$ ::IS AN OCTAL DIGIT
9927 053650 122716 000067 CMPB #'7,(SP)
9928 053654 002423 BLT 4$
9929 053656 006301 ASL R1 ::*2
9930 053660 006102 ROL R2
9931 053662 006301 ASL R1 ::*4
9932 053664 006102 ROL R2
9933 053666 006301 ASL R1 ::*8
9934 053670 006102 ROL R2
9935 053672 042716 177770 BIC #'C7,(SP) ::STRIP THE ASCII JUNK
9936 053676 062601 ADD (SP)+,R1 ::ADD IN THIS DIGIT
9937 053700 000756 BR 2$ ::LOOP
9938 053702 005726 3$: TST (SP)+ ::CLEAN TERMINATOR FROM STACK
9939 053704 010166 000012 MOV R1,12(SP) ::SAVE THE RESULT
9940 053710 010237 053742 MOV R2,$HIOCT
9941 053714 012602 MOV (SP)+,R2 ::POP STACK INTO R2
9942 053716 012601 MOV (SP)+,R1 ::POP STACK INTO R1
9943 053720 012600 MOV (SP)+,R0 ::POP STACK INTO R0
9944 053722 000002 RTI ::RETURN
9945 053724 005726 4$: TST (SP)+ ::CLEAN PARTIAL FROM STACK
9946 053726 105010 CLR R0 ::SET A TERMINATOR
9947 053730 104401 TYPE ::TYPE UP THRU THE BAD CHAR.
9948 053732 000000 5$: .WORD 0
9949 053734 104401 001204 TYPE $QUES
9950 053740 000730 BR 1$ ::'?' 'CR' & 'LF'
::TRY AGAIN

```



```
9951 053742 000000 $HIOCT: .WORD 0 ::HIGH ORDER BITS GO HERE
9952 .SBTTL DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
9953
9954 ::*****
9955 ::THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
9956 ::UNSIGNED OCTAL ASCII NUMBER.
9957 ::CALL
9958 ::* MOV #PNTR,-(SP) ::POINTER TO LOW WORD OF BINARY NUMBER
9959 ::* JSR PC,@#$DB20 ::CALL THE ROUTINE
9960 ::* RETURN ::THE ADDRESS OF THE FIRST ASCII CHAR. IS ON THE STACK
9961
9962
9963 053744 104413 $DB20: SAVREG ::SAVE ALL REGISTERS
9964 053746 016601 000002 MOV 2(SP),R1 ::PICKUP THE POINTER TO LOW WORD
9965 053752 012705 054063 MOV #SOCTVL+13.,R5 ::POINTER TO DATA TABLE
9966 053756 012704 000014 MOV #12.,R4 ::DO ELEVEN CHARACTERS
9967 053762 012703 177770 MOV #^C7,R3 ::MASK
9968 053766 012100 MOV (R1)+,R0 ::LOWER WORD
9969 053770 012101 MOV (R1)+,R1 ::HIGH WORD
9970 053772 005002 CLR R2 ::TERMINATOR
9971 053774 110245 1$: MOVVB R2,-(R5) ::PUT CHARACTER IN DATA TABLE
9972 053776 010002 MOV R0,R2 ::GET THIS DIGIT
9973 054000 005304 DEC R4 ::COUNT THIS CHARACTER
9974 054002 003007 BGT 3$ ::BR IF NOT THE LAST DIGIT
9975 054004 001405 BEQ 2$ ::BR IF IT IS THE LAST DIGIT
9976 054006 005205 INC R5 ::ALL DIGITS DONE-ADJUST POINTER FOR FIRST
9977 054010 010566 000002 MOV R5,2(SP) ::ASCII CHAR. & PUT IT ON THE STACK
9978 054014 104414 RESREG ::RESTORE ALL REGISTERS
9979 054016 000207 RTS PC ::RETURN TO USER
9980 054020 006203 2$: ASR R3 ::POSITION THE MASK FOR THE LAST DIGIT
9981 054022 006001 3$: ROR R1 ::POSITION THE BINARY NUMBER FOR
9982 054024 006000 ROR R0 :: THE NEXT OCTAL DIGIT
9983 054026 006001 ROR R1
9984 054030 006000 ROR R0
9985 054032 006001 ROR R1
9986 054034 006000 ROR R0
9987 054036 040302 BIC R3,R2 ::MASK OUT ALL JUNK
9988 054040 062702 000060 AND #'0,R2 ::MAKE THIS CHAR. ASCII
9989 054044 000753 BR 1$ ::GO PUT IT IN THE DATA TABLE
9990 054046 000016 $SOCTVL: .BLKB 14. ::RESERVE DATA TABLE
9991 .SBTTL DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
9992
9993 ::*****
9994 ::THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
9995 ::DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
9996 ::POSITIVE.
9997 ::CALL
9998 ::* MOV #PNTR,-(SP) ::POINTER TO LOW WORD OF BINARY NUMBER
9999 ::* JSR PC,@#$DB2D ::CALL THE ROUTINE
10000 ::* RETURN ::THE FIRST ADDRESS OF ASCII
10001 :: IS ON THE STACK
10002
10003
10004 054064 104413 $DB2D: SAVREG ::SAVE REGISTERS
10005 054066 016602 000002 MOV 2(SP),R2 ::PICKUP THE DATA POINTER
10006 054072 012700 054244 MOV #SDECVL,R0 ::GET ADDRESS OF 'SDECVL' STRING
```

```

10007 054076 010066 000002          MOV     R0,2(SP)      ;;PUT ADDRESS OF ASCIZ STRING ON STACK
10008 054102 012201          MOV     (R2)+,R1     ;;PICKUP THE BINARY NUMBER
10009 054104 012202          MOV     (R2)+,R2
10010 054106 012737 000012 054162    MOV     #10.,4$      ;;SET UP TO DO 10 CONVERSIONS
10011 054114 012704 054174    MOV     #$STNPWR,R4  ;;ADDRESS OF TEN POWER
10012 054120 012705 054176    MOV     #$STNPWR+2,R5
10013 054124 005003          1$:    CLR     R3           ;;CLEAR PARTIAL
10014 054126 161401          2$:    SUB     (R4),R1     ;;SUBTRACT TEN POWER
10015 054130 005602          SBC     R2
10016 054132 161502          SUB     (R5),R2
10017 054134 002402          BLT     3$           ;;BR IF TEN POWER TO LARGE
10018 054136 005203          INC     R3           ;;ADD 1 TO PARTIAL
10019 054140 000772          BR      2$           ;;LOOP
10020 054142 062401          3$:    ADD     (R4)+,R1  ;;RESTORE SUBTRACTED VALUE
10021 054144 005502          ADC     R2
10022 054146 062402          ADD     (R4)+,R2
10023 054150 022525          CMP     (R5)+,(R5)+  ;;MOVE TO NEXT TEN POWER
10024 054152 052703 000060    BIS     #'0,R3       ;;CHANGE PARTIAL TO ASCII
10025 054156 110320          MOVB    R3,(R0)+     ;;SAVE IT
10026 054160 005327          DEC     (PC)+        ;;DONE?
10027 054162 000000          4$:    .WORD  0
10028 054164 001357          BNE     1$           ;;BR IF NO
10029 054166 105020          CLRB   (R0)+        ;;TERMINATOR
10030 054170 104414          RESREG
10031 054172 000207          RTS     PC           ;;RESTORE REGISTERS
10032 054174 145000          $STNPWR: 145000      ;;RETURN
10033 054176 035632          35632              ;;1.0E09
10034 054200 160400          160400              ;;1.0E08
10035 054202 002765          2765                ;;1.0E07
10036 054204 113200          113200              ;;1.0E06
10037 054206 000230          230                 ;;1.0E05
10038 054210 041100          041100              ;;1.0E04
10039 054212 000017          17                  ;;1.0E03
10040 054214 103240          103240              ;;1.0E02
10041 054216 000001          1                   ;;1.0E01
10042 054220 023420          23420               ;;1.0E00
10043 054222 000000          0
10044 054224 001750          1750
10045 054226 000000          0
10046 054230 000144          144
10047 054232 000000          0
10048 054234 000012          12
10049 054236 000000          0
10050 054240 000001          1
10051 054242 000000          0
10052 054244 000014          $DECVL: .BLKB 12.  ;;RESERVE STORAGE FOR ASCIZ STRING
10053          .SBTTL SINGLE LENGTH BINARY TO DECIMAL ASCII ROUTINE
10054
10055          ;;*****
10056          ;;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
10057          ;;*UNSIGNED DECIMAL ASCII NUMBER.
10058          ;;*CALL
10059          ;;*   MOV     NUMBER,-(SP)  ;;PUT BINARY NUMBER ON THE STACK
10060          ;;*   JSR     PC,@#$SB2D  ;;CALL
10061          ;;*   RETURN                ;;ADDRESS OF THE 1ST ASCII CHAR.IS ON THE STACK
10062

```

```

10063
10064 054260 016637 000002 054310 $SB2D: MOV 2(SP),1$ ;;SAVE BINARY NUMBER
10065 054266 012746 054310 MOV #1$,-(SP) ;;SET POINTER
10066 054272 004737 054064 JSR PC,@#$DB2D ;;CALL DOUBLE LENGTH CONVERT
10067 054276 062716 000005 ADD #5,(SP) ;;ONLY ALLOW FIVE CHARACTERS
10068 054302 012666 000002 MCV (SP)+,2(SP) ;;PICKUP POINTER
10069 054306 000207 RTS PC ;;RETURN
10070 054310 000000 000000 1$: .WORD 0,0
10071 .SBTTL TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
10072
10073 *****
10074 *THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
10075 *LEADING NUMBERS.
10076 *CALL
10077 * MOV #NUMADR,-(SP) ;;FIRST ADDRESS OF ASCIZ STRING
10078 * JSR PC,@#$SUPRS
10079
10080
10081 054314 010046 $SUPRS: MOV R0,-(SP) ;;SAVE R0
10082 054316 016600 000004 MOV 4(SP),R0 ;;PICKUP THE POINTER
10083 054322 105710 1$: TSTB (R0) ;;TERMINATEOR?
10084 054324 001403 BEQ 2$ ;;BR IF YES
10085 054326 122720 000060 CMPB #'0,(R0)+ ;;IS THIS AN ASCII '0' ?
10086 054332 001773 BEQ 1$ ;;BR IF YES
10087 054334 005300 2$: DEC R0 ;;BACKUP BY '1'
10088 054336 010037 054344 MOV R0,3$ ;;SAVE FOR TYPING
10089 054342 104401 TYPE ;;GO TYPE
10090 054344 000000 3$: .WORD 0 ;;ASCIZ POINTER GOES HERE
10091 054346 012600 MOV (SP)+,R0 ;;RESTORE R0
10092 054350 012616 MOV (SP)+,(SP) ;;RESTORE THE STACK
10093 054352 000207 RTS PC ;;RETURN
10094 .SBTTL INTEGER MULTIPLY ROUTINE
10095
10096 *****
10097 *CALL
10098 * MOV MULTIPLER,-(SP)
10099 * MOV MULTIPLICAND,-(SP)
10100 * JSR PC,@#$MULT
10101 * RETURN ;;PRODUCT IS ON THE STACK
10102
10103 * STACK PRODUCT
10104 * -----
10105 * TOP LSB'S
10106 * +2 MSB'S
10107
10108 054354 $MULT:
10109 054354 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
10110 054356 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
10111 054360 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
10112 054362 005046 CLR -(SP) ;;CLEAR THE SIGN KEY
10113 054364 016601 000012 MOV 12(SP),R1 ;;GET THE MULTIPLICAND
10114 054370 100002 BPL 1$ ;;BR IF PLUS
10115 054372 005216 INC (SP) ;;SET THE SIGN KEY
10116 054374 005401 NEG R1 ;;MAKE THE MULTIPLICAND POSTIVE
10117 054376 016602 000014 1$: MOV 14(SP),R2 ;;GET THE MULTIPLIER
10118 054402 100002 BPL 2$ ;;BR IF PLUS
    
```

```
10119 054404 005316          DEC      (SP)          ;;UPDATE THE SIGN KEY
10120 054406 005402          NEG      R2            ;;MAKE THE MULTIPLIER POSTIVE
10121 054410 012746 000021  2$:      MOV      #17.,-(SP) ;;SET THE LOOP COUNT
10122 054414 005000          CLR      R0            ;;SETUP FOR THE MULTIPLY LOOP
10123 054416 103001          3$:      BCC      4$          ;;DON'T ADD IF MULTIPLICAND = 0
10124 054420 060200          ADD      R2,R0
10125 054422 006000          4$:      ROR      R0            ;;POSITION THE PARITIAL PRODUCT AND
10126 054424 006001          ROR      R1            ;;THE MULTIPLICAND
10127 054426 005316          DEC      (SP)          ;;HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
10128 054430 001372          BNE      3$            ;;BR IF NO
10129 054432 022616          CMP      (SP)+,(SP)    ;;SHOULD PRODUCT BE NEGATIVE?
10130 054434 001403          BEQ      5$            ;;GO TO EXIT IF NO
10131 054436 005400          NEG      R0            ;;YES--SO MAKE IT SO
10132 054440 005401          NEG      R1
10133 054442 005600          SBC      R0
10134 054444 005726          5$:      TST      (SP)+          ;;CLEAR SIGN INFO. OFF OF STACK
10135 054446 010066 000012  MOV      R0,12(SP)     ;;PUT THE PRODUCT ON THE STACK (MSB'S)
10136 054452 010166 000010  MOV      R1,10(SP)     ;;LSB'S
10137 054456 012602          MOV      (SP)+,R2     ;;POP STACK INTO R2
10138 054460 012601          MOV      (SP)+,R1     ;;POP STACK INTO R1
10139 054462 012600          MOV      (SP)+,R0     ;;POP STACK INTO R0
10140 054464 000207          RTS      PC
```

.SBTTL SAVE AND RESTORE R0-R5 ROUTINES

```
;;*SAVE R0-R5
;;*CALL:
;;* SAVREG
;;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
;;*
;;*TOP---(+16)
;;* +2---(+18)
;;* +4---R5
;;* +6---R4
;;* +8---R3
;;*+10---R2
;;*+12---R1
;;*+14---R0
```

\$SAVREG:

```
10158 054466          MOV      R0,-(SP)     ;;PUSH R0 ON STACK
10159 054466 010046          MOV      R1,-(SP)     ;;PUSH R1 ON STACK
10160 054470 010146          MOV      R2,-(SP)     ;;PUSH R2 ON STACK
10161 054472 010246          MOV      R3,-(SP)     ;;PUSH R3 ON STACK
10162 054474 010346          MOV      R4,-(SP)     ;;PUSH R4 ON STACK
10163 054476 010446          MOV      R5,-(SP)     ;;PUSH R5 ON STACK
10164 054500 010546          MOV      22(SP),-(SP) ;;SAVE PS OF MAIN FLOW
10165 054502 016646 000022  MOV      22(SP),-(SP) ;;SAVE PC OF MAIN FLOW
10166 054506 016646 000022  MOV      22(SP),-(SP) ;;SAVE PS OF CALL
10167 054512 016646 000022  MOV      22(SP),-(SP) ;;SAVE PC OF CALL
10168 054516 016646 000022  MOV      22(SP),-(SP)
10169 054522 000002          RTI
```

;;*RESTORE R0-R5

```
;;*CALL:
;;* RESREG
$RESREG:
```

10174 054524

10175 054524 012666 000022
10176 054530 012666 000022
10177 054534 012666 000022
10178 054540 012666 000022
10179 054544 012605
10180 054546 012604
10181 054550 012603
10182 054552 012602
10183 054554 012601
10184 054556 012600
10185 054560 000002

```
MOV (SP)+,22(SP) ;;RESTORE PC OF CALL
MOV (SP)+,22(SP) ;;RESTORE PS OF CALL
MOV (SP)+,22(SP) ;;RESTORE PC OF MAIN FLOW
MOV (SP)+,22(SP) ;;RESTORE PS OF MAIN FLOW
MOV (SP)+,R5 ;;POP STACK INTO R5
MOV (SP)+,R4 ;;POP STACK INTO R4
MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
MOV (SP)+,R0 ;;POP STACK INTO R0
```

.SBTTL TRAP DECODER

```
*****
*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
*GO TO THAT ROUTINE.
```

10194 054562 010046
10195 054564 016600 000002
10196 054570 005740
10197 054572 111000
10198 054574 006300
10199 054576 016000 054616
10200 054602 000200

```
$TRAP: MOV R0,-(SP) ;;SAVE R0
MOV 2(SP),R0 ;;GET TRAP ADDRESS
TST -(R0) ;;BACKUP BY 2
MOVB (R0),R0 ;;GET RIGHT BYTE OF TRAP
ASL R0 ;;POSITION FOR INDEXING
MOV $TRPAD(R0),R0 ;;INDEX TO TABLE
RTS R0 ;;GO TO ROUTINE
```

;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO

10205 054604 011646
10206 054606 016666 000004 000002
10207 054614 000002

```
$TRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
RTI ;;RESTORE THE PSW
```

.SBTTL TRAP TABLE

```
*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
*BY THE 'TRAP' INSTRUCTION.
```

10214
10215
10216 054616 054604
10217 054620 051046
10218 054622 052120
10219 054624 052074
10220 054626 052134
10221 054630 051402
10222
10223 054632 052722
10224
10225 054634 052632
10226 054636 053174
10227 054640 053264
10228 054642 053604
10229 054644 054466
10230 054646 054524

```
ROUTINE
-----
$TRPAD: .WORD $TRAP2
$TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
$TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
$TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
$TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
$TYPDS ;;CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)

$GTSWR ;;CALL=GTSWR TRAP+6(104406) GET SOFT-SWR SETTING

$CKSWR ;;CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
$RDCHR ;;CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
$RDLIN ;;CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE
$RDOCT ;;CALL=RDOCT TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
$SAVREG ;;CALL=SAVREG TRAP+13(104413) SAVE R0-R5 ROUTINE
$RESREG ;;CALL=RESREG TRAP+14(104414) RESTORE R0-R5 ROUTINE
```

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 B 16
TRAP TABLE PAGE 197

SEQ 0196

10231 054650 047276
10232

SCOP1\$;;CALL=SCOP1 TRAP+15(104415) INTERNAL LOOP ON ERROR

10233
10234
10235
10236 054652 005015 047125 041111
10237 054660 051525 051040 030113
10238 054666 026466 045522 033460
10239 054674 042040 040525 020114
10240 054702 047520 052122 042040
10241 054710 044522 042526 042040
10242 054716 040511 047107 051517
10243 054724 044524 103
10244 054727 015 041412 051132
10245 054734 043466 030103 005015
10246 054742 005015 025011 025052
10247 054750 025052 041440 052501
10248 054756 044524 047117 025040
10249 054764 025052 025052 005015
10250 054772 005015 044124 051511
10251 055000 050040 047522 051107
10252 055006 046501 051440 047510
10253 055014 046125 020104 041040
10254 055022 020105 040510 052114
10255 055030 042105 047440 046116
10256 055036 020131 054502 052040
10257 055044 050131 047111 020107
10258 055052 047503 052116 047522
10259 055060 026514 103
10260 055063 015 047412 044124
10261 055070 051105 044527 042523
10262 055076 020054 040503 052122
10263 055104 044522 043504 020105
10264 055112 047506 046522 052101
10265 055120 044524 043516 040440
10266 055126 042116 020054 051117
10267 055134 052040 042510 042040
10268 055142 044522 042526
10269 055146 005015 040515 020131
10270 055154 042502 046040 043105
10271 055162 020124 047111 040440
10272 055170 020116 047125 042504
10273 055176 042524 046522 047111
10274 055204 042105 051440 040524
10275 055212 042524
10276 055214 005015 047111 052111
10277 055222 040511 046114 026131
10278 055230 042040 044522 042526
10279 055236 020123 047524 041040
10280 055244 020105 042524 052123
10281 055252 042105 051440 047510
10282 055260 046125 020104 040510
10283 055266 042526 006472 012
10284 055273 015 040412 020056
10285 055300 053105 047105 047040
10286 055306 046525 042502 042522
10287 055314 020104 047125 052111
10288 055322 051440 046105 041505

.SBTTL SERVICE MESSAGES

MSG1: .ASCII <CR><LF>/UNIBUS RK06-RK07 DUAL PORT DRIVE DIAGNOSTIC/

.ASCII <CR><LF>/CZR6GCO/<CR><LF>

.ASCII <CR><LF>/ ***** CAUTION *****/<CR><LF>

.ASCII <CR><LF>/THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING CONTROL-C/

.ASCII <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND, OR THE DRIVE/

.ASCII <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/

.ASCII <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE: /<CR><LF>

.ASCII <CR><LF>/A. EVEN NUMBERED UNIT SELECT PLUGS ONLY/

10289	055330	020124	046120	043525	
10290	055336	020123	047117	054514	
10291	055344	005015	027102	044040	.ASCII <CR><LF>/B. HEADS MANUALLY LOADED/
10292	055352	040505	051504	046440	
10293	055360	047101	040525	046114	
10294	055366	020131	047514	042101	
10295	055374	042105			
10296	055376	005015	027103	041040	.ASCII <CR><LF>/C. BOTH PORTS SELECTED/
10297	055404	052117	020110	047520	
10298	055412	052122	020123	042523	
10299	055420	042514	052103	042105	
10300	055426	005015	027104	042040	.ASCII <CR><LF>/D. DUAL PORT TEST SWITCH ENABLED/
10301	055434	040525	020114	047520	
10302	055442	052122	052040	051505	
10303	055450	020124	053523	052111	
10304	055456	044103	042440	040516	
10305	055464	046102	042105		
10306	055470	005015	027105	053440	.ASCII <CR><LF>/E. WRITE LOCK DISABLED/
10307	055476	044522	042524	046040	
10308	055504	041517	020113	044504	
10309	055512	040523	046102	042105	
10310	055520	005015	027106	042040	.ASCII <CR><LF>/F. DRIVE READY INDICATOR ON/<CR><LF>
10311	055526	044522	042526	051040	
10312	055534	040505	054504	044440	
10313	055542	042116	041511	052101	
10314	055550	051117	047440	006516	
10315	055556	012			
10316	055557	015	042012	044522	.ASCII <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
10317	055564	042526	020123	047516	
10318	055572	020124	047524	041040	
10319	055600	020105	042524	052123	
10320	055606	042105	046440	051525	
10321	055614	020124	040510	042526	
10322	055622	005015	047502	044124	.ASCII <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
10323	055630	050040	051117	051524	
10324	055636	042040	051505	046105	
10325	055644	041505	042524	006504	
10326	055652	012			
10327	055653	015	042012	051511	.ASCII <CR><LF>/DISABLE DUAL PORT TEST SWITCH ON DUAL PORT MODULE/
10328	055660	041101	042514	042040	
10329	055666	040525	020114	047520	
10330	055674	052122	052040	051505	
10331	055702	020124	053523	052111	
10332	055710	044103	047440	020116	
10333	055716	052504	046101	050040	
10334	055724	051117	020124	047515	
10335	055732	052504	042514		
10336	055736	005015	042502	047506	.ASCIIZ <CR><LF>/BEFORE PROCEEDING TO ANY OTHER PROGRAM/<CR><LF>
10337	055744	042522	050040	047522	
10338	055752	042503	042105	047111	
10339	055760	020107	047524	040440	
10340	055766	054516	047440	044124	
10341	055774	051105	050040	047522	
10342	056002	051107	046501	005015	
10343	056010	000			
10344	056011	015	041012	020105	MSG2: .ASCIIZ <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/

10345	056016	052523	042522	052040	
10346	056024	020117	052520	020124	
10347	056032	041523	040522	041524	
10348	056040	020110	040520	045503	
10349	056046	044440	020116	051104	
10350	056054	053111	020105	000060	
10351	056062	005015	051104	053111	MSG3: .ASCIZ <CR><LF>/DRIVE(S) TO BE TESTED (EVEN NOS. ONLY): /
10352	056070	024105	024523	052040	
10353	056076	020117	042502	052040	
10354	056104	051505	042524	020104	
10355	056112	042450	042526	020116	
10356	056120	047516	027123	047440	
10357	056126	046116	024531	020072	
10358	056134	000			
10359	056135	015	041012	051525	MSG4: .ASCIZ <CR><LF>/BUSS ADDRESS (177440): /
10360	056142	020123	042101	051104	
10361	056150	051505	020123	030450	
10362	056156	033467	032064	024460	
10363	056164	020072	000		
10364	056167	015	041412	047117	MSG5: .ASCIZ <CR><LF>/CONTROLLER INTERRUPT VECTOR (210): /
10365	056174	051124	046117	042514	
10366	056202	020122	047111	042524	
10367	056210	051122	050125	020124	
10368	056216	042526	052103	051117	
10369	056224	024040	030462	024460	
10370	056232	020072	000		
10371	056235	015	044412	052116	MSG6: .ASCIZ <CR><LF>/INTERRUPT OCCURRED AT PC= /
10372	056242	051105	052522	052120	
10373	056250	047440	041503	051125	
10374	056256	042522	020104	052101	
10375	056264	050040	036503	000	
10376	056271	015	042012	044522	MSG7: .ASCIZ <CR><LF>/DRIVE 0 WILL NOT BE TESTED /
10377	056276	042526	030040	053440	
10378	056304	046111	020114	047516	
10379	056312	020124	042502	052040	
10380	056320	051505	042524	000104	
10381	056326	005015	005015	046101	MSG8: .ASCIZ <CR><LF><CR><LF>/ALL DRIVES TESTED/<CR><LF><LF>
10382	056334	020114	051104	053111	
10383	056342	051505	052040	051505	
10384	056350	042524	006504	005012	
10385	056356	000			
10386	056357	015	005012	044527	MSG10: .ASCIZ <CR><LF><LF>/WILL TEST DRIVES: /
10387	056364	046114	052040	051505	
10388	056372	020124	051104	053111	
10389	056400	051505	000072		
10390	056404	005015	050012	053517	MSG11: .ASCIZ <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
10391	056412	051105	052440	020120	
10392	056420	042522	052123	051101	
10393	056426	020124	047524	052040	
10394	056434	051505	020124	006461	
10395	056442	000012			
10396	056444	005015	047516	046040	MSG13: .ASCII <CR><LF>/NO L OR P CLOCKS PRESENT /
10397	056452	047440	020122	020120	
10398	056460	046103	041517	051513	
10399	056466	050040	042522	042523	
10400	056474	052116			

10401	056476	005015	046101	020114		.ASCIZ	<CR><LF>/ALL TESTS BYPASSED/
10402	056504	042524	052123	020123			
10403	056512	054502	040520	051523			
10404	056520	042105	000				
10405	056523	015	041012	050131	MSG14:	.ASCIZ	<CR><LF>/BYPASSING DRIVE /
10406	056530	051501	044523	043516			
10407	056536	042040	044522	042526			
10408	056544	000040					
10409	056546	005015	042012	044522	MSG15:	.ASCIZ	<CR><LF><LF>/DRIVE /
10410	056554	042526	000040				
10411	056560	005015	042012	044522	MSG16:	.ASCIZ	<CR><LF><LF>/DRIVE SERIAL NO. /
10412	056566	042526	051440	051105			
10413	056574	040511	020114	047516			
10414	056602	020056	000				
10415	056605	015	041412	051101	MSG17:	.ASCIZ	<CR><LF>/CARTRIDGE SERIAL NO. /
10416	056612	051124	042111	042507			
10417	056620	051440	051105	040511			
10418	056626	020114	047516	020056			
10419	056634	000					
10420	056635	015	050012	051117	MSG19:	.ASCII	<CR><LF>/PORT /
10421	056642	020124					
10422	056644	006440	000012		MSG19A:	.ASCIZ	/ /<CR><LF>
10423	056650	005015	047520	052122	MSG20:	.ASCIZ	<CR><LF>/PORT A TIMEOUT: /
10424	056656	040440	052040	046511			
10425	056664	047505	052125	020072			
10426	056672	000					
10427	056673	015	040412	047502	MSG21:	.ASCIZ	<CR><LF>/ABORTING BALANCE OF TESTS/<CR><LF>
10428	056700	052122	047111	020107			
10429	056706	040502	040514	041516			
10430	056714	020105	043117	052040			
10431	056722	051505	051524	005015			
10432	056730	000					
10433	056731	040	051515	000	MSG22:	.ASCIZ	/ MS/
10434	056735	015	050012	051117	MSG23:	.ASCIZ	<CR><LF>/PORT B TIMEOUT: /
10435	056742	020124	020102	044524			
10436	056750	042515	052517	035124			
10437	056756	000040					
10438	056760	005015	051120	043517	MSG74:	.ASCIZ	<CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
10439	056766	040522	020115	041101			
10440	056774	051117	020124	042520			
10441	057002	042116	047111	027107			
10442	057010	027056	046120	040505			
10443	057016	042523	053440	044501			
10444	057024	000124					
10445	057026	005015	040510	052114	MSG75:	.ASCIZ	<CR><LF>/HALT PENDING...PLEASE WAIT/
10446	057034	050040	047105	044504			
10447	057042	043516	027056	050056			
10448	057050	042514	051501	020105			
10449	057056	040527	052111	000			
10450	057063	015	050012	047522	MSG76:	.ASCIZ	<CR><LF>/PROGRAM ABORTED/
10451	057070	051107	046501	040440			
10452	057076	047502	052122	042105			
10453	057104	000					
10454	057105	015	041412	052520	MSG77:	.ASCIZ	<CR><LF>/CPU HALTED/
10455	057112	044040	046101	042524			
10456	057120	000104					

10457
10458
10459
10460 057122 005015 047523 051122 EM1: .ASCIZ <CR><LF>/SORRY, ONLY 0,2,4,6 ALLOWED, TRY AGAIN/<CR><LF>
10461 057130 026131 047440 046116
10462 057136 020131 026060 026062
10463 057144 026064 020066 046101
10464 057152 047514 042527 026104
10465 057160 052040 054522 040440
10466 057166 040507 047111 005015
10467 057174 000
10468 057175 104 044522 042526 EM2: .ASCIZ /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
10469 057202 021440 044440 020116
10470 057210 045522 051503 020062
10471 057216 040503 047116 052117
10472 057224 041040 020105 042522
10473 057232 042101 041040 041501
10474 057240 020113 047503 051122
10475 057246 041505 046124 020131
10476 057254 047111 051040 046513
10477 057262 031122 000
10478 057265 015 040412 047502 EM3: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
10479 057272 052122 052040 051505
10480 057300 051524 027056 052456
10481 057306 042516 050130 041505
10482 057314 042524 020104 044524
10483 057322 042515 047440 052125
10484 057330 040440 020124 041520
10485 057336 000075
10486 057340 005015 041101 051117 EM4: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/
10487 057346 020124 042524 052123
10488 057354 027123 027056 047125
10489 057362 054105 042520 052103
10490 057370 042105 044440 052116
10491 057376 051105 052522 052120
10492 057404 040440 020124 041520
10493 057412 000075
10494 057414 042115 020123 042523 EM5: .ASCIZ /MDS SET IN RKCS2/
10495 057422 020124 047111 051040
10496 057430 041513 031123 000
10497 057435 125 042506 051440 EM6: .ASCIZ /UFE SET IN RKCS2/
10498 057442 052105 044440 020116
10499 057450 045522 051503 000062
10500 057456 047516 042040 040522 EM7: .ASCII /NO DRA IN RKDS & NO NED IN RKCS2/
10501 057464 044440 020116 045522
10502 057472 051504 023040 047040
10503 057500 020117 042516 020104
10504 057506 047111 051040 041513
10505 057514 031123
10506 057516 005015 044103 041505 .ASCII <CR><LF>/CHECK 1. THE PORT A USING DRIVE FLOP (DP1)/
10507 057524 020113 027061 052040
10508 057532 042510 050040 051117
10509 057540 020124 020101 051525
10510 057546 047111 020107 051104
10511 057554 053111 020105 046106
10512 057562 050117 024040 050104

10513	057570	024461				
10514	057572	005015	027062	042040		.ASCIZ <CR><LF>/2. DRIVE AVAILABLE TO PORT A STATUS (DP2)/
10515	057600	044522	042526	040440		
10516	057606	040526	046111	041101		
10517	057614	042514	052040	020117		
10518	057622	047520	052122	040440		
10519	057630	051440	040524	052524		
10520	057636	020123	042050	031120		
10521	057644	000051				
10522	057646	051104	053111	020105	EM8:	.ASCIZ /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
10523	057654	051120	051505	047105		
10524	057662	020124	052502	020124		
10525	057670	047516	020124	050123		
10526	057676	041505	043111	042511		
10527	057704	020104	054502	047440		
10528	057712	042520	040522	047524		
10529	057720	000122				
10530	057722	051104	053111	020105	EM9:	.ASCIZ /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
10531	057730	047516	020124	051120		
10532	057736	051505	047105	020124		
10533	057744	052502	020124	050123		
10534	057752	041505	043111	042511		
10535	057760	020104	054502	047440		
10536	057766	042520	040522	047524		
10537	057774	000122				
10538	057776	041101	051117	020124	EM10:	.ASCIZ /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
10539	060004	042524	052123	027123		
10540	060012	027056	040503	047116		
10541	060020	052117	051040	043105		
10542	060026	051105	047105	042503		
10543	060034	041440	047117	051124		
10544	060042	046117	042514	020122		
10545	060050	042522	044507	052123		
10546	060056	051105	000			
10547	060061	104	040522	044440	EM11:	.ASCIZ /DRA IN RKDS & NED IN RKCS2 BOTH SET/
10548	060066	020116	045522	051504		
10549	060074	023040	047040	042105		
10550	060102	044440	020116	045522		
10551	060110	051503	020062	047502		
10552	060116	044124	051440	052105		
10553	060124	000				
10554	060125	103	047117	051124	EM12:	.ASCIZ /CONTROLLER NOT READY IN RKCS1/
10555	060132	046117	042514	020122		
10556	060140	047516	020124	042522		
10557	060146	042101	020131	047111		
10558	060154	051040	041513	030523		
10559	060162	000				
10560	060163	116	020117	052101	EM13:	.ASCIZ /NO ATTN IN RKASOF/
10561	060170	047124	044440	020116		
10562	060176	045522	051501	043117		
10563	060204	000				
10564	060205	125	042516	050130	EM14:	.ASCIZ /UNEXPECTED MEMORY PARITY TRAP/
10565	060212	041505	042524	020104		
10566	060220	042515	047515	054522		
10567	060226	050040	051101	052111		
10568	060234	020131	051124	050101		

10569	060242	000				
10570	060243	122	042113	020103	EM15:	.ASCII /RKDC & RKDA INDICATE THAT WCE OCCURRED AT/
10571	060250	020046	045522	040504		
10572	060256	044440	042116	041511		
10573	060264	052101	020105	044124		
10574	060272	052101	053440	042503		
10575	060300	047440	041503	051125		
10576	060306	042522	020104	052101		
10577	060314	005015	054503	020114		.ASCIZ <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
10578	060322	030464	026061	052040		
10579	060330	040522	045503	031040		
10580	060336	020054	042523	052103		
10581	060344	051117	031040	000061		
10582	060352	040503	047116	052117	EM16:	.ASCIZ /CANNOT READ BAD SECTOR INFORMATION/
10583	060360	051040	040505	020104		
10584	060366	040502	020104	042523		
10585	060374	052103	051117	044440		
10586	060402	043116	051117	040515		
10587	060410	044524	047117	000		
10588	060415	115	051505	040523	EM17:	.ASCIZ /MESSAGE A0 ERROR/
10589	060422	042507	040440	020060		
10590	060430	051105	047522	000122		
10591	060436	042515	051523	043501	EM18:	.ASCIZ /MESSAGE B0 ERROR/
10592	060444	020105	030102	042440		
10593	060452	051122	051117	000		
10594	060457	115	051505	040523	EM19:	.ASCIZ /MESSAGE A1 ERROR/
10595	060464	042507	040440	020061		
10596	060472	051105	047522	000122		
10597	060500	042515	051523	043501	EM20:	.ASCIZ /MESSAGE B1 ERROR/
10598	060506	020105	030502	042440		
10599	060514	051122	051117	000		
10600	060521	103	051105	020122	EM21:	.ASCIZ /CERR SET IN RKCS1/
10601	060526	042523	020124	047111		
10602	060534	051040	041513	030523		
10603	060542	000				
10604	060543	116	020117	051104	EM22:	.ASCII /NO DRIVES FOUND IN DEVICE MAP (\$DEV)/<CR><LF>
10605	060550	053111	051505	043040		
10606	060556	052517	042116	044440		
10607	060564	020116	042504	044526		
10608	060572	042503	046440	050101		
10609	060600	024040	042044	053105		
10610	060606	024515	005015			
10611	060612	042523	052524	020120		.ASCIZ /SETUP CORRECTLY AND RESTART/<CR><LF>
10612	060620	047503	051122	041505		
10613	060626	046124	020131	047101		
10614	060634	020104	042522	052123		
10615	060642	051101	006524	000012		
10616	060650	047516	042040	044522	EM23:	.ASCII /NO DRIVES FOUND ON BUSS/<CR><LF>
10617	060656	042526	020123	047506		
10618	060664	047125	020104	047117		
10619	060672	041040	051525	006523		
10620	060700	012				
10621	060701	123	052105	050125		.ASCIZ /SETUP CORRECTLY AND PRESS 'CONTINUE'/<CR><LF>
10622	060706	041440	051117	042522		
10623	060714	052103	054514	040440		
10624	060722	042116	050040	042522		

10625	060730	051523	023440	047503	
10626	060736	052116	047111	042525	
10627	060744	006447	000012		
10628	060750	047526	020114	040526	EM24: .ASCIZ /VOL VALID NOT SET IN RKMR2/
10629	060756	044514	020104	047516	
10630	060764	020124	042523	020124	
10631	060772	047111	051040	046513	
10632	061000	031122	000		
10633	061003	015	042012	052105	EM25: .ASCIZ <CR><LF>/DETECTED 10 BAD SECTORS...ABORTING TEST/
10634	061010	041505	042524	020104	
10635	061016	030061	041040	042101	
10636	061024	051440	041505	047524	
10637	061032	051522	027056	040456	
10638	061040	047502	052122	047111	
10639	061046	020107	042524	052123	
10640	061054	000			
10641	061055	104	052105	041505	EM26: .ASCIZ /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
10642	061062	042524	020104	051502	
10643	061070	020105	052502	020124	
10644	061076	047516	020124	044514	
10645	061104	052123	042105	044440	
10646	061112	020116	040502	020104	
10647	061120	042523	052103	051117	
10648	061126	043040	046111	000105	
10649	061134	042504	042524	052103	EM27: .ASCII /DETECTED BSE IN READ COMMAND/
10650	061142	042105	041040	042523	
10651	061150	044440	020116	042522	
10652	061156	042101	041440	046517	
10653	061164	040515	042116		
10654	061170	005015	052502	020124	.ASCIZ <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
10655	061176	047516	020124	047111	
10656	061204	050040	042522	044526	
10657	061212	052517	020123	051127	
10658	061220	052111	020105	047503	
10659	061226	046515	047101	020104	
10660	061234	047524	051440	046501	
10661	061242	020105	042523	052103	
10662	061250	051117	000		
10663	061253	117	046116	020131	EM28: .ASCII /ONLY DRIVES 0,2,4,6 ALLOWED IN \$DEVN/
10664	061260	051104	053111	051505	
10665	061266	030040	031054	032054	
10666	061274	033054	040440	046114	
10667	061302	053517	042105	044440	
10668	061310	020116	042044	053105	
10669	061316	115			
10670	061317	015	051012	046105	.ASCIZ <CR><LF>/RELOAD \$DEVN & PRESS CONTINUE/<CR><LF>
10671	061324	040517	020104	042044	
10672	061332	053105	020115	020046	
10673	061340	051120	051505	020123	
10674	061346	047503	052116	047111	
10675	061354	042525	005015	000	
10676	061361	120	051117	020124	EM29: .ASCIZ /PORT AVAILABLE...TIMERS NOT INHIBITED/
10677	061366	053101	044501	040514	
10678	061374	046102	027105	027056	
10679	061402	044524	042515	051522	
10680	061410	047040	052117	044440	

10681	061416	044116	041111	052111	
10682	061424	042105	000		
10683	061427	120	050111	051440	EM30: .ASCIZ /PIP SET IN RKMR2/
10684	061434	052105	044440	020116	
10685	061442	045522	051115	000062	
10686	061450	047520	052122	047040	EM31: .ASCIZ /PORT NOT AVAILABLE/
10687	061456	052117	040440	040526	
10688	061464	046111	041101	042514	
10689	061472	000			
10690	061473	120	051117	020124	EM32: .ASCIZ /PORT AVAILABLE/
10691	061500	053101	044501	040514	
10692	061506	046102	000105		
10693	061512	052101	047124	051440	EM33: .ASCIZ /ATTN SET IN RKASOF/
10694	061520	052105	044440	020116	
10695	061526	045522	051501	043117	
10696	061534	000			
10697	061535	101	052124	020116	EM34: .ASCIZ /ATTN CLEARED IN RKASOF/
10698	061542	046103	040505	042522	
10699	061550	020104	047111	051040	
10700	061556	040513	047523	000106	
10701	061564	042503	051122	047040	EM35: .ASCIZ /CERR NOT SET IN RKCS1/
10702	061572	052117	051440	052105	
10703	061600	044440	020116	045522	
10704	061606	051503	000061		
10705	061612	054503	020114	042101	EM36: .ASCIZ /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
10706	061620	051104	044440	020116	
10707	061626	045522	051115	020063	
10708	061634	047516	020124	040523	
10709	061642	042515	040440	020123	
10710	061650	045522	041504	000	
10711	061655	115	046125	044524	EM37: .ASCIZ /MULTIPLE ATTENTIONS SEEN/
10712	061662	046120	020105	052101	
10713	061670	042524	052116	047511	
10714	061676	051516	051440	042505	
10715	061704	000116			
10716	061706	044524	042515	052517	EM38: .ASCIZ /TIMEOUT DID NOT RE-TRIGGER FOR FULL SECOND/
10717	061714	020124	044504	020104	
10718	061722	047516	020124	042522	
10719	061730	052055	044522	043507	
10720	061736	051105	043040	051117	
10721	061744	043040	046125	020114	
10722	061752	042523	047503	042116	
10723	061760	000			
10724	061761	103	046131	042040	EM39: .ASCIZ /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
10725	061766	043111	020106	020046	
10726	061774	043117	051506	052105	
10727	062002	044440	020116	045522	
10728	062010	051115	020062	047516	
10729	062016	020124	046103	040505	
10730	062024	042522	000104		
10731	062030	054503	020114	042101	EM40: .ASCIZ /CYL ADDR IN RKMR3 NOT CLEARED/
10732	062036	051104	044440	020116	
10733	062044	045522	051115	020063	
10734	062052	047516	020124	046103	
10735	062060	040505	042522	000104	
10736	062066	052101	047124	047040	EM55: .ASCIZ /ATTN NOT CLEARED IN RKASOF/

10737	062074	052117	041440	042514		
10738	062102	051101	042105	044440		
10739	062110	020116	045522	051501		
10740	062116	043117	000			
10741	062121	104	052114	051440	EM63:	.ASCIZ /DLT SET IN RKCS2/
10742	062126	052105	044440	020116		
10743	062134	045522	051503	000062		
10744	062142	042522	042101	044040	EM65:	.ASCIZ /READ HEADER ERROR/
10745	062150	040505	042504	020122		
10746	062156	051105	047522	000122		
10747	062164	046101	043511	046516	EM69:	.ASCIZ /ALIGNMENT CARTRIDGE USED/
10748	062172	047105	020124	040503		
10749	062200	052122	044522	043504		
10750	062206	020105	051525	042105		
10751	062214	000				
10752	062215	103	047524	051440	EM73:	.ASCIZ /CTO SET IN RKCS1/
10753	062222	052105	044440	020116		
10754	062230	045522	051503	000061		
10755	062236	052122	020132	047516	EM74:	.ASCIZ /RTZ NOT SET IN RKMR2/
10756	062244	020124	042523	020124		
10757	062252	047111	051040	046513		
10758	062260	031122	000			
10759	062263	116	042105	051440	EM79:	.ASCIZ /NED SET IN RKCS2/
10760	062270	052105	044440	020116		
10761	062276	045522	051503	000062		
10762	062304	051127	052111	020105	EM80:	.ASCIZ /WRITE CHECK ERROR SET IN RKCS2/
10763	062312	044103	041505	020113		
10764	062320	051105	047522	020122		
10765	062326	042523	020124	047111		
10766	062334	051040	041513	031123		
10767	062342	000				
10768	062343	122	040505	020104	EM82:	.ASCIZ /READ DATA DID NOT COMPARE WITH WRITE DATA/
10769	062350	040504	040524	042040		
10770	062356	042111	047040	052117		
10771	062364	041440	046517	040520		
10772	062372	042522	053440	052111		
10773	062400	020110	051127	052111		
10774	062406	020105	040504	040524		
10775	062414	000				
10776	062415	104	052101	020101	EM83:	.ASCIZ /DATA CHECK ERROR SET IN RKER/
10777	062422	044103	041505	020113		
10778	062430	051105	047522	020122		
10779	062436	042523	020124	047111		
10780	062444	051040	042513	000122		
10781	062452	044127	046111	020105	EM84:	.ASCIZ /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
10782	062460	040527	052111	047111		
10783	062466	020107	047506	020122		
10784	062474	047503	052116	020122		
10785	062502	042522	042101	020131		
10786	062510	051117	040440	052106		
10787	062516	051105	041440	047117		
10788	062524	051124	051040	040505		
10789	062532	054504	051040	041505		
10790	062540	042047	000			
10791	062543	122	040505	044504	EM93:	.ASCIZ /READING WRONG CYLINDER # IN HEADER...MISPOSITION/
10792	062550	043516	053440	047522		

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 B 1
ERROR MESSAGES PAGE 208

SEQ 0207

10793 062556 043516 041440 046131
10794 062564 047111 042504 020122
10795 062572 020043 047111 044040
10796 062600 040505 042504 027122
10797 062606 027056 044515 050123
10798 062614 051517 052111 047511
10799 062622 000116
10800
10801
10802
10803 062624 042524 052123 047040
10804 062632 027117 020040 041520
10805 062640 000
10806 062641 122 046513 030522
10807 062646 051011 046513 031122
10808 062654 051011 046513 031522
10809 062662 051011 042513 004522
10810 062670 045522 051504 051011
10811 062676 041513 030523 051011
10812 062704 041513 031123 000
10813 062711 122 053513 004503
10814 062716 045522 040502 051011
10815 062724 042113 004501 045522
10816 062732 051501 043117 051011
10817 062740 042113 004503 045522
10818 062746 041505 051520 051011
10819 062754 042513 050103 000124
10820 062762 043101 042524 020122
10821 062770 052101 047124 051040
10822 062776 041505 042047 043040
10823 063004 047522 020115 042522
10824 063012 040503 020114 047503
10825 063020 046515 047101 000104
10826 063026 043101 042524 020122
10827 063034 047523 052106 040527
10828 063042 042522 052040 046511
10829 063050 047505 052125 000
10830 063055 106 047522 020115
10831 063062 054503 020114 052040
10832 063070 020117 054503 020114
10833 063076 041440 046131 042040
10834 063104 043111 000106
10835 063110 043101 042524 020122
10836 063116 052523 051502 051531
10837 063124 041440 042514 051101
10838 063132 005015 044103 041505
10839 063140 020113 027061 052040
10840 063146 042510 050040 051117
10841 063154 020124 020102 051525
10842 063162 047111 020107 051104
10843 063170 053111 020105 046106
10844 063176 050117 024040 050104
10845 063204 024461
10846 063206 005015 027062 042040
10847 063214 044522 042526 040440
10848 063222 040526 046111 041101

.SBTTL DATA HEADERS

DH1: .ASCIZ /TEST NO. PC/

DH2: .ASCIZ /RKMR1 RKMR2 RKMR3 RKER RKDS RKCS1 RKCS2/

DH3: .ASCIZ /RKWC RKBA RKDA RKASOF RKDC RKECPS RKECPT/

DH4: .ASCIZ /AFTER ATTN REC'D FROM RECAL COMMAND/

DH5: .ASCIZ /AFTER SOFTWARE TIMEOUT/

DH6: .ASCIZ /FROM CYL TO CYL CYL DIFF/

DH7: .ASCII /AFTER SUBSYS CLEAR/

.ASCII <CR><LF>/CHECK 1. THE PORT B USING DRIVE FLOP (DP1)/

.ASCIZ <CR><LF>/2. DRIVE AVAILABLE TO PORT B STATUS (DP2)/

10849	063230	042514	052040	020117		
10850	063236	047520	052122	041040		
10851	063244	051440	040524	052524		
10852	063252	020123	042050	031120		
10853	063260	000051				
10854	063262	042524	052123	047040	DH8:	.ASCIZ /TEST NO. TRAP PC/
10855	063270	027117	052011	040522		
10856	063276	020120	041520	000		
10857	063303	101	052106	051105	DH9:	.ASCIZ /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
10858	063310	051440	040524	052122		
10859	063316	051440	044520	042116		
10860	063324	042514	041440	046517		
10861	063332	040515	042116	051040		
10862	063340	041505	042047	041040		
10863	063346	020131	051104	053111		
10864	063354	000105				
10865	063356	052101	042440	042116	DH10:	.ASCIZ /AT END OF HEAD LOADING/
10866	063364	047440	020106	042510		
10867	063372	042101	046040	040517		
10868	063400	044504	043516	000		
10869	063405	105	050130	041505	DH11:	.ASCIZ /EXPECTED WAS/
10870	063412	042524	004504	040527		
10871	063420	000123				
10872	063422	043101	042524	020122	DH12:	.ASCII /AFTER SEEK CMD/
10873	063430	042523	045505	041440		
10874	063436	042115				
10875	063440	005015	044103	041505		.ASCIZ <CR><LF>/CHECK STAT ADD SIGNAL BEFORE & AFTER MUX/
10876	063446	020113	052123	052101		
10877	063454	040440	042104	051440		
10878	063462	043511	040516	020114		
10879	063470	042502	047506	042522		
10880	063476	023040	040440	052106		
10881	063504	051105	046440	054125		
10882	063512	000				
10883	063513	117	020116	042523	DH13:	.ASCIZ /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
10884	063520	052103	051117	020123		
10885	063526	030061	020054	031061		
10886	063534	020054	032061	020054		
10887	063542	033061	020054	034061		
10888	063550	047440	020122	030062		
10889	063556	041440	046131	032040		
10890	063564	030061	052040	040522		
10891	063572	045503	031040	000		
10892	063577	102	043105	051117	DH14:	.ASCIZ /BEFORE TIMEOUT OR RELEASE/
10893	063604	020105	044524	042515		
10894	063612	052517	020124	051117		
10895	063620	051040	046105	040505		
10896	063626	042523	000			
10897	063631	127	044510	042514	DH15:	.ASCIZ /WHILE PORT UNAVAILABLE/
10898	063636	050040	051117	020124		
10899	063644	047125	053101	044501		
10900	063652	040514	046102	000105		
10901	063660	047524	044440	042116	DH16:	.ASCII /TO INDICATE THAT REQUESTING PORT CAN SEIZE/
10902	063666	041511	052101	020105		
10903	063674	044124	052101	051040		
10904	063702	050505	042525	052123		

10905	063710	047111	020107	047520		
10906	063716	052122	041440	047101		
10907	063724	051440	044505	042532		
10908	063732	005015	044103	041505		.ASCIZ <CR><LF>/CHECK PORT REQ. FLOPS & 1 SEC SAFETY TIMER/
10909	063740	020113	047520	052122		
10910	063746	051040	050505	020056		
10911	063754	046106	050117	020123		
10912	063762	020046	020061	042523		
10913	063770	020103	040523	042506		
10914	063776	054524	052040	046511		
10915	064004	051105	000			
10916	064007	101	052106	051105	DH17:	.ASCIZ /AFTER RECAL COMMAND/
10917	064014	051040	041505	046101		
10918	064022	041440	046517	040515		
10919	064030	042116	000			
10920	064033	101	052106	051105	DH18:	.ASCIZ /AFTER UNLOAD COMMAND/
10921	064040	052440	046116	040517		
10922	064046	020104	047503	046515		
10923	064054	047101	000104			
10924	064060	043101	042524	020122	DH19:	.ASCIZ /AFTER PACK COMMAND/
10925	064066	040520	045503	041440		
10926	064074	046517	040515	042116		
10927	064102	000				
10928	064103	101	052106	051105	DH20:	.ASCIZ /AFTER SELECT DRIVE COMMAND/
10929	064110	051440	046105	041505		
10930	064116	020124	051104	053111		
10931	064124	020105	047503	046515		
10932	064132	047101	000104			
10933	064136	043101	042524	020122	DH21:	.ASCIZ /AFTER SUBSYSTEM CLEAR/
10934	064144	052523	051502	051531		
10935	064152	042524	020115	046103		
10936	064160	040505	000122			
10937	064164	043101	042524	020122	DH22:	.ASCIZ /AFTER DRIVE CLEAR COMMAND/
10938	064172	051104	053111	020105		
10939	064200	046103	040505	020122		
10940	064206	047503	046515	047101		
10941	064214	000104				
10942	064216	044527	044124	052517	DH23:	.ASCIZ /WITHOUT REQUEST PENDING/
10943	064224	020124	042522	052521		
10944	064232	051505	020124	042520		
10945	064240	042116	047111	000107		
10946	064246	054502	042040	044522	DH24:	.ASCIZ /BY DRIVE CLEAR COMMAND TO OTHER PORT/
10947	064254	042526	041440	042514		
10948	064262	051101	041440	046517		
10949	064270	040515	042116	052040		
10950	064276	020117	052117	042510		
10951	064304	020122	047520	052122		
10952	064312	000				
10953	064313	101	052106	051105	DH25:	.ASCIZ /AFTER SEEK COMMAND/
10954	064320	051440	042505	020113		
10955	064326	047503	046515	047101		
10956	064334	000104				
10957	064336	043101	042524	020122	DH26:	.ASCIZ /AFTER READ DATA COMMAND/
10958	064344	042522	042101	042040		
10959	064352	052101	020101	047503		
10960	064360	046515	047101	000104		

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 E 1 PAGE 211
DATA HEADERS

SEQ 0210

10961	064366	043101	042524	020122	DH27:	.ASCIZ	/AFTER WRITE DATA COMMAND/
10962	064374	051127	052111	020105			
10963	064402	040504	040524	041440			
10964	064410	046517	040515	042116			
10965	064416	000					
10966	064417	011	054105	042520	DH28:	.ASCIZ /	EXPECT/
10967	064424	052103	000				
10968	064427	011	041501	052524	DH29:	.ASCIZ /	ACTUAL/
10969	064434	046101	000				
10970	064437	101	052106	051105	DH30:	.ASCIZ	/AFTER READ HEADER COMMAND/
10971	064444	051040	040505	020104			
10972	064452	042510	042101	051105			
10973	064460	041440	046517	040515			
10974	064466	042116	000				
10975	064471	101	004460	030102	DH31:	.ASCIZ /A0	B0 A1 B1 A2 B2 B3/
10976	064476	040411	004461	030502			
10977	064504	040411	004462	031102			
10978	064512	041011	000063				
10979	064516	043101	042524	020122	DH32:	.ASCIZ	/AFTER WRITE CHECK COMMAND/
10980	064524	051127	052111	020105			
10981	064532	044103	041505	020113			
10982	064540	047503	046515	047101			
10983	064546	000104					
10984	064550	043101	042524	020122	DH35:	.ASCIZ	/AFTER RELEASE ISSUED/
10985	064556	042522	042514	051501			
10986	064564	020105	051511	052523			
10987	064572	042105	000				
10988	064575	101	052106	051105	DH36:	.ASCIZ	/AFTER SELECT DRIVE COMMAND & DRIVE NOT AVAILABLE/
10989	064602	051440	046105	041505			
10990	064610	020124	051104	053111			
10991	064616	020105	047503	046515			
10992	064624	047101	020104	020046			
10993	064632	051104	053111	020105			
10994	064640	047516	020124	053101			
10995	064646	044501	040514	046102			
10996	064654	000105					
10997	064656	043101	042524	020122	DH37:	.ASCIZ	/AFTER RELEASE ISSUED & PORT REQUEST PENDING/
10998	064664	042522	042514	051501			
10999	064672	020105	051511	052523			
11000	064700	042105	023040	050040			
11001	064706	051117	020124	042522			
11002	064714	052521	051505	020124			
11003	064722	042520	042116	047111			
11004	064730	000107					
11005	064732	043101	042524	020122	DH38:	.ASCIZ	/AFTER DELAY FROM SEEK & IMMEDIATE RELEASE/
11006	064740	042504	040514	020131			
11007	064746	051106	046517	051440			
11008	064754	042505	020113	020046			
11009	064762	046511	042515	044504			
11010	064770	052101	020105	042522			
11011	064776	042514	051501	000105			
11012	065004	043101	042524	020122	DH39:	.ASCIZ	/AFTER WRITE HEADER COMMAND/
11013	065012	051127	052111	020105			
11014	065020	042510	042101	051105			
11015	065026	041440	046517	040515			
11016	065034	042116	000				

CZR6GCO RK611 DU PORT LGC
CZR6GC.F11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 F 1 PAGE 212
DATA HEADERS

SEQ 0211

11017	065037	127	051117	021504	DH40: .ASCIZ /WORD# HEADER WAS SHOULD BE/
11018	065044	044011	040505	042504	
11019	065052	020122	040527	020123	
11020	065060	051440	047510	046125	
11021	065066	020104	042502	000	
11022	065073	104	051125	047111	DH41: .ASCII /DURING RECAL COMMAND/
11023	065100	020107	042522	040503	
11024	065106	020114	047503	046515	
11025	065114	047101	104		
11026	065117	015	041412	042510	.ASCIZ <CR><LF>/CHECK BUFF1 SIGNAL BEFORE & AFTER MUX/
11027	065124	045503	041040	043125	
11028	065132	030506	051440	043511	
11029	065140	040516	020114	042502	
11030	065146	047506	042522	023040	
11031	065154	040440	052106	051105	
11032	065162	046440	054125	000	
11033	065167	117	020116	042523	DH42: .ASCIZ /ON SECTORS 0,2,4,6 OR 8 CYL 410 TRACK 2/
11034	065174	052103	051117	020123	
11035	065202	026060	026062	026064	
11036	065210	020066	051117	034040	
11037	065216	020040	054503	020114	
11038	065224	030464	020060	051124	
11039	065232	041501	020113	000062	
11040	065240	043101	042524	020122	DH43: .ASCIZ /AFTER RE-SEIZE PORT MID-WAY IN NORMAL TIMEOUT/
11041	065246	042522	051455	044505	
11042	065254	042532	020040	047520	
11043	065262	052122	046440	042111	
11044	065270	053455	054501	044440	
11045	065276	020116	047516	046522	
11046	065304	046101	052040	046511	
11047	065312	047505	052125	000	
11048	065317	106	051117	040515	DH44: .ASCIZ /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
11049	065324	020124	020046	046101	
11050	065332	020114	042522	042101	
11051	065340	053455	044522	042524	
11052	065346	052040	051505	051524	
11053	065354	053440	046111	020114	
11054	065362	042502	041040	050131	
11055	065370	051501	042523	000104	
11056	065376	042502	047506	042522	DH45: .ASCIZ /BEFORE RELEASE WHILE HEADS UNLOADED/
11057	065404	051040	046105	040505	
11058	065412	042523	053440	044510	
11059	065420	042514	044040	040505	
11060	065426	051504	052440	046116	
11061	065434	040517	042504	000104	
11062	065442	051515	020107	020101	DH49: .ASCIZ /MSG A & B IN RKMR2 & RKMR3 RESP. ARE INVALID/
11063	065450	020046	020102	047111	
11064	065456	051040	046513	031122	
11065	065464	023040	051040	046513	
11066	065472	031522	051040	051505	
11067	065500	027120	040440	042522	
11068	065506	044440	053116	046101	
11069	065514	042111	000		
11070	065517	101	052106	051105	DH51: .ASCIZ /AFTER SEEK TO SELF COMMAND/
11071	065524	051440	042505	020113	
11072	065532	047524	051440	046105	

11073	065540	020106	047503	046515	
11074	065546	047101	000104		
11075	065552	054503	020114	004443	DH56: .ASCIZ /CYL # HEADER WORD 0/
11076	065560	042510	042101	051105	
11077	065566	053440	051117	020104	
11078	065574	000060			
11079					
11080					.SBTTL ERROR OUTPUT DATA
11081					
11082					.EVEN
11083	065576	001214	001116		DT1: \$TESTN,\$ERRPC
11084	065602	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11085	065610	005330	005326	005314	
11086	065616	005316			
11087	065620	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11088	065626	005332	005334	005346	
11089	065634	005350			
11090	065636	001214	001334		DT3: \$TESTN,TRAPPC
11091	065642	001214	001116	001344	DT4: \$TESTN,\$ERRPC,FRCYL,TOCYL,CALDIF
11092	065650	001346	001354		
11093	065654	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11094	065662	005330	005326	005314	
11095	065670	005316			
11096	065672	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11097	065700	005332	005334	005346	
11098	065706	005350			
11099	065710	001214	001116	001416	DT6: \$TESTN,\$ERRPC,WD2,WD1
11100	065716	001414			
11101	065720	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11102	065726	005330	005326	005314	
11103	065734	005316			
11104	065736	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11105	065744	005332	005334	005346	
11106	065752	005350			
11107	065754	001214	001116	001432	DT7: \$TESTN,\$ERRPC,WDCNT,HDWD,TEMP1
11108	065762	001450	005352		
11109	065766	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11110	065774	005330	005326	005314	
11111	066002	005316			
11112	066004	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11113	066012	005332	005334	005346	
11114	066020	005350			
11115	066022	001214	001116	001346	DT8: \$TESTN,\$ERRPC,TOCYL,FRCYL,CALDIF
11116	066030	001344	001354		
11117	066034	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11118	066042	005330	005326	005314	
11119	066050	005316			
11120	066052	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11121	066060	005332	005334	005346	
11122	066066	005350			
11123	066070	001214	001116	001346	DT9: \$TESTN,\$ERRPC,TOCYL,RHTAB
11124	066076	001666			
11125	066100	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11126	066106	005330	005326	005314	
11127	066114	005316			
11128	066116	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT

11129	066124	005332	005334	005346	
11130	066132	005350			
11131	066134	001214	001116	005404	DT13: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
11132	066142	005406	005410	005412	
11133	066150	005364	005366	005370	
11134	066156	005372			
11135	066160	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11136	066166	005330	005326	005314	
11137	066174	005316			
11138	066176	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11139	066204	005332	005334	005346	
11140	066212	005350			
11141	066214	001214	001116	005404	DT14: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
11142	066222	005406	005410	005412	
11143	066230	005414	005416		
11144	066234	005364	005366	005370	H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
11145	066242	005372	005374	005376	
11146	066250	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11147	066256	005330	005326	005314	
11148	066264	005316			
11149	066266	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11150	066274	005332	005334	005346	
11151	066302	005350			
11152	066304	001214	001116	005404	DT15: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
11153	066312	005406	005410	005412	
11154	066320	005414	005416	005422	
11155	066326	005364	005366	005370	H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
11156	066334	005372	005374	005376	
11157	066342	005402			
11158	066344	005340	005342	005344	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11159	066352	005330	005326	005314	
11160	066360	005316			
11161	066362	005320	005322	005324	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11162	066370	005332	005334	005346	
11163	066376	005350			

.SBTTL ERROR DATA FORMATS

11164					
11165					
11166					
11167	066400	000003			DF1: 3
11168	066402	002	000		.BYTE 2,0
11169	066404	062641			DH2
11170	066406	007	000		.BYTE 7,0
11171	066410	062711			DH3
11172	066412	007	000		.BYTE 7,0
11173					
11174	066414	000001			DF2: 1
11175	066416	002	000		.BYTE 2,0
11176					
11177	066420	000005			DF3: 5
11178	066422	000	000		.BYTE 0,0
11179	066424	062624			DH1
11180	066426	002	000		.BYTE 2,0
11181	066430	063405			DH11
11182	066432	002	000		.BYTE 2,0
11183	066434	062641			DH2
11184	066436	007	000		.BYTE 7,0

11185	066440	062711			DH3	
11186	066442	007	000		.BYTE	7.0
11187						
11188	066444	000003		DF4:	3	
11189	066446	002	000		.BYTE	2.0
11190	066450	062641			DH2	
11191	066452	007	000		.BYTE	7.0
11192	066454	062711			DH3	
11193	066456	007	000		.BYTE	7.0
11194						
11195	066460	000005		DF5:	5	
11196	066462	000	000		.BYTE	0.0
11197	066464	065442			DH49	
11198	066466	000	000		.BYTE	0.0
11199	066470	062624			DH1	
11200	066472	002	000		.BYTE	2.0
11201	066474	062641			DH2	
11202	066476	007	000		.BYTE	7.0
11203	066500	062711			DH3	
11204	066502	007	000		.BYTE	7.0
11205						
11206	066504	000005		DF6:	5	
11207	066506	000	000		.BYTE	0.0
11208	066510	062624			DH1	
11209	066512	002	000		.BYTE	2.0
11210	066514	063055			DH6	
11211	066516	003	000		.BYTE	3.0
11212	066520	062641			DH2	
11213	066522	007	000		.BYTE	7.0
11214	066524	062711			DH3	
11215	066526	007	000		.BYTE	7.0
11216						
11217						
11218	066530	000004		DF10:	4	
11219	066532	000	000		.BYTE	0.0
11220	066534	062624			DH1	
11221	066536	002	000		.BYTE	2.0
11222	066540	062641			DH2	
11223	066542	007	000		.BYTE	7.0
11224	066544	062711			DH3	
11225	066546	007	000		.BYTE	7.0
11226						
11227	066550	000004		DF14:	4	
11228	066552	002	000		.BYTE	2.0
11229	066554	065037			DH40	
11230	066556	003	000		.BYTE	3.0
11231	066560	062641			DH2	
11232	066562	007	000		.BYTE	7.0
11233	066564	062711			DH3	
11234	066566	007	000		.BYTE	7.0
11235						
11236						
11237	066570	000004		DF15:	4	
11238	066572	000	000		.BYTE	0.0
11239	066574	062624			DH1	
11240	066576	002	000		.BYTE	2.0

11241	066600	062641			DH2	
11242	066602	007	000		.BYTE	7.0
11243	066604	062711			DH3	
11244	066606	007	000		.BYTE	7.0
11245						
11246	066610	000005		DF17:	5	
11247	066612	000	000		.BYTE	0.0
11248	066614	065317			DH44	
11249	066616	000	000		.BYTE	0.0
11250	066620	062624			DH1	
11251	066622	002	000		.BYTE	2.0
11252	066624	062641			DH2	
11253	066626	007	000		.BYTE	7.0
11254	066630	062711			DH3	
11255	066632	007	000	DF20:	5	
11256	066634	000005			.BYTE	0.0
11257	066636	000	000		DH1	
11258	066640	062624			.BYTE	2.0
11259	066642	002	000		DH56	
11260	066644	065552			.BYTE	2.0
11261	066646	002	000		DH2	
11262	066650	062641			.BYTE	7.0
11263	066652	007	000		DH3	
11264	066654	062711			.BYTE	7.0
11265	066656	007	000			
11266				DF21:	7	
11267	066660	000007			.BYTE	0.0
11268	066662	000	000		DH1	
11269	066664	062624			.BYTE	2.0
11270	066666	002	000		DH28	
11271	066670	064417			.BYTE	0.0
11272	066672	000	000		DH31	
11273	066674	064471			.BYTE	4.0
11274	066676	004	000		DH29	
11275	066700	064427			.BYTE	4.0
11276	066702	004	000		DH2	
11277	066704	062641			.BYTE	7.0
11278	066706	007	000		DH3	
11279	066710	062711			.BYTE	7.0
11280	066712	007	000			
11281				DF22:	7	
11282	066714	000007			.BYTE	0.0
11283	066716	000	000		DH1	
11284	066720	062624			.BYTE	2.0
11285	066722	002	000		DH28	
11286	066724	064417			.BYTE	0.0
11287	066726	000	000		DH31	
11288	066730	064471			.BYTE	6.0
11289	066732	006	000		DH29	
11290	066734	064427			.BYTE	6.0
11291	066736	006	000		DH2	
11292	066740	062641			.BYTE	7.0
11293	066742	007	000		DH3	
11294	066744	062711			.BYTE	7.0
11295	066746	007	000			
11296						

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 K 1
ERROR DATA FORMATS PAGE 217

SEQ 0216

11297	066750	000007	
11298	066752	000	000
11299	066754	062624	
11300	066756	002	000
11301	066760	064417	
11302	066762	000	000
11303	066764	064471	
11304	066766	007	000
11305	066770	064427	
11306	066772	007	000
11307	066774	062641	
11308	066776	007	000
11309	067000	062711	
11310	067002	007	000
11311			
11312			
11313			
11314			
11315			
11316			
11317			
11318			
11319			
11320			
11321			
11322	067004	104413	
11323	067006	104401	056635
11324	067012	113700	001114
11325	067016	042700	177400
11326	067022	005300	
11327	067024	006300	
11328	067026	006300	
11329	067030	006300	

DF23: 7
.BYTE 0,0
DH1
.BYTE 2,0
DH28
.BYTE 0,0
DH31
.BYTE 7,0
DH29
.BYTE 7,0
DH2
.BYTE 7,0
DH3
.BYTE 7,0

.EVEN
:*****
:SBTTL TYPE ERROR ROUTINE
:*ENTRY JSR PC,TYP ERR
:*RETURN RTS PC
:*
:*THIS ROUTINE USES THE "ITEM CONTROL BYTE" (\$ITEMB) TO DETERMINE WHICH
:*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" (\$ERRTB)
:*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
:*THE ERROR.
:*****

TYPERR: SAVREG
TYPE ,MSG19 ;PORT A OR B
MOVB \$ITEMB,R0 ;ENTER ERROR NUMBER
BIC #177400,R0 ;CLEAR SIGN EXTENSION
DEC R0 ;FORM INDEX FOR ERROR TABLE
ASL R0
ASL R0
ASL R0

11330	067032	062700	005470	1\$:	ADD	#SERRTB,R0	:FORM ADDRESS OF ERROR ENTRY
11331	067036	012037	067052		MOV	(R0)+,2\$:GET EM POINTER
11332	067042	001404			BEQ	3\$:BRANCH IF THERE ISN'T ONE
11333	067044	104401	001205		TYPE	,\$CRLF	:TYPE CARRIAGE RETURN LINE FEED
11334	067050	104401			TYPE		:TYPE ERROR MESSAGE (EM)
11335	067052	000000		2\$:	.WORD	0	:EM POINTER GOES HERE
11336	067054	012037	067070	3\$:	MOV	(R0)+,4\$:GET DH POINTER
11337	067060	001404			BEQ	5\$:BRANCH IF THERE ISN'T ONE
11338	067062	104401	001205		TYPE	,\$CRLF	:TYPE CR-LF
11339	067066	104401			TYPE		:TYPE DATA HEADER
11340	067070	000000		4\$:	.WORD	0	:DH POINTER GOES HERE
11341	067072	012001		5\$:	MOV	(R0)+,R1	:GET DT POINTER
11342	067074	001455			BEQ	20\$:BRANCH IF THERE ARE NONE
11343	067076	005004			CLR	R4	:SET INDENT SWITCH
11344	067100	012000			MOV	(R0)+,R0	:GET DF POINTER
11345	067102	012002			MOV	(R0)+,R2	:STORE NUMBER OF DH'S
11346	067104	001446			BEQ	17\$:DH NUM IS 0-BRANCH
11347	067106	005104			COM	R4	:NO INDENT
11348	067110	104401	001205		TYPE	,\$CRLF	
11349	067114	112003		10\$:	MOVB	(R0)+,R3	:GET & STORE NUMBER OF DATA WORDS
11350	067116	105720			TSTB	(R0)+	:BUMP PAST FORMAT WORD
11351	067120	005703			TST	R3	:TEST IF ANY DATA FOR THIS HEADER
11352	067122	001407			BEQ	14\$:NO - SKIP DATA PRINT
11353	067124	013146		11\$:	MOV	@(R1)+,-(SP)	:PUT FIRST DATA WORD ON STACK
11354	067126	104402			TYPOC		:TYPE IT
11355	067130	005303			DEC	R3	:MORE DATA WORDS
11356	067132	001403			BEQ	14\$:NO-BRANCH

```
11357 067134 104401 067264          TYPE      ,SPACE2      ;TYPE SEPARATORS
11358 067140 000771          BR          11$        ;LOOP
11359 067142 005302          14$: DEC      R2        ;MORE DH'S?
11360 067144 003431          BLE      20$        ;NO-BRANCH
11361 067146 104401 001205          TYPE      ,SCRLF      ;
11362 067152 005760 000002          TST      2(R0)        ;ONLY A DH IN THIS REQUEST?
11363 067156 001404          BEQ      15$        ;YES-BRANCH BYPASS INDENT
11364 067160 005104          COM      R4        ;INDENT?
11365 067162 001002          BNE      15$        ;NO-BRANCH
11366 067164 104401 067264          TYPE      ,SPACE2      ;YES-TYPE SPACES
11367 067170 012037 067176          15$: MOV      (R0)+,16$ ;GET NEXT DH POINTER
11368 067174 104401          TYPE      ;TYPE DH
11369 067176 000000          16$: .WORD    0        ;DH POINTER GOES HERE
11370 067200 105710          TSTB     (R0)        ;TYPE A DT?
11371 067202 001003          BNE      21$        ;YES-BRANCH
11372 067204 062700 000002          ADD      #2,R0      ;INCREMENT DF POINTER
11373 067210 000754          BR       14$        ;SEE IF END OF DF BLOCK
11374 067212 104401 001205          21$: TYPE      ,SCRLF      ;
11375 067216 005704          TST      R4        ;INDENT?
11376 067220 001335          BNE      10$        ;NO-BRANCH
11377 067222 104401 067264          17$: TYPE      ,SPACE2      ;YES-TYPE SPACES
11378 067226 000732          BR       10$        ;LOOP
11379 067230 104414          20$: RESREG
11380
11381 067232 032777 010000 111700          BIT      #SW12,@SWR  ;SEE IF ABORT DRV AFTER 20 ERRORS
11382 067240 001410          BEQ      25$        ;BR IF NO
11383 067242 023727 001103 000024          CMP      $ERFLG,#20. ;ELSE SEE IF HAVE 20 ERRORS
11384 067250 001004          BNE      25$        ;BR IF NO
11385 067252 012706 001100          MOV      #STACK,SP  ;ELSE RESTORE STACK PTR
11386 067256 000137 042644          JMP      $EOP       ;AND GO TO NEXT DRV
11387
11388 067262 000207          25$: RTS      PC
11389 067264 020040 000          SPACE2: .ASCIZ/ /    ;2 SPACES
11390          ; ODT-11 -- V005A
11391
11392          ; DEC-11-UODPA-A-LA
11393
11394          ; COPYRIGHT 1969,1970,1972
11395          ; DIGITAL EQUIPMENT CORPORATION
11396          ; MAYNARD, MASSACHUSETTS 01754
11397          .ENABL ABS,AMA
11398          .EVEN
11399          .=. +60
11400          R0      =      %0      ; REGISTER
11401          R1      =      %1      ; NAMING
11402          R2      =      %2      ; CONVENTIONS
11403          R3      =      %3
11404          R4      =      %4
11405          R5      =      %5
11406          SP      =      %6
11407          PC      =      %7
11408          ST      =      177776 ;STATUS REGISTER
11409          ;
11410          O.TVEC =      14      ;TRT VECTOR LOCATION
11411          O.STM  =      340     ;PRIORITY MASK - STATUS REGISTER
11412          O.TBT  =      20      ;T-BIT MASK - STATUS REGISTER
```

```
11413      000003      TRT      =      000003      ;TRT INSTRUCTION
11414      000006      RTT      =      000006      ;RTT INSTRUCTION
11415
11416      ;
11417      ; R5 IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
11418      ; RESIDES IN IT. AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
11419      ; OPERATIONS LEAVE IT RANDOMLY FILLED. OTHERWISE, IT SHOULD NOT
11420      ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
11421      ;
11422
11423
11424      177562      O.RDB   =      177562      ;R DATA BUFFER
11425      177560      O.RCSR  =      177560      ;R C/SR
11426      177566      O.TDB   =      177566      ;T DATA BUFFER
11427      177564      O.TCSR  =      177564      ;T C/SR
11428
11429      ;
11430      ; INITIALIZE ODT
11431      ; USE O.ODT FOR A NORMAL ENTRY
11432      ; USE O.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
11433      ; USE O.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
11434      ;
11435      067350      000413      O.ODT:  BR      O.STRT      ;NORMAL ENTRY
11436      067352      000417      BR      O.RST        ;RESTART
11437      067354      013737      177776      067330      O.ENTR:  MOV     ST,O.UST    ;RE-ENTER -- SAVE STATUS
11438      067362      013737      000016      177776      MOV     O.TVEC+2,ST    ;SET UP LOCAL STATUS
11439      067370      010737      067326      MOV     PC,O.UPC      ;FAKE THE PC
11440      067374      000137      070526      JMP     O.BK1
11441      ;
11442      067400      012706      067310      O.STRT:  MOV     #O.URO,SP   ;SET UP STACK
11443      067404      010637      067324      MOV     SP,O.USR      ;FAKE THE SAVED STACK
11444      067410      000414      BR      O.RST1       ;CLEAR BREAKPOINT TABLES
11445      067412      004037      070734      O.RST:   JSR     O,O.SVR   ;SAVE REGISTERS
11446      067416      013777      067346      177716      MOV     O.UIN,@O.ADR1  ;REMOVE THE BREAKPOINT
11447      067424      113704      067332      MOV     O.PRI,R4      ;GET ODT PRIORITY
11448      067430      106004      RORB   R4            ;SHIFT
11449      067432      106004      RORB   R4            ; INTO
11450      067434      106004      RORB   R4            ; POSITION
11451      067436      110437      177776      MOV     R4,ST         ;STORE IN STATUS
11452      067442      000127      O.RST1:  JMP     (PC)+
11453      067444      000403      BR      O.45
11454      067446      012737      000002      070436      MOV     #RTI,O.RTIT    ;SET TO RTI IF 11/20 OR /05
11455      067454      105037      071355      O.45:   CLRB   O.P          ;DISALLOW PROCEED
11456      067460      012737      000340      000016      MOV     #O.STM,O.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
11457      067466      012737      070516      000014      MOV     #O.BRK,O.TVEC  ;PC TO TRT VECTOR
11458      067474      000447      BR      O.RALL        ;CLEAR BREAKPOINT TABLES
11459      ;
11460      ; SPECIAL NAME HANDLER
11461      ; DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URO
11462      ;
11463      067476      004537      071156      O.REGT:  JSR     S,O.GET     ;SPECIAL NAME, GET ONE MORE CHARACTER
11464      067502      012704      071401      MOV     #O.TL,R4      ;TABLE START ADDRESS
11465      067506      120024      O.RSP:   CMPB   RO,(R4)+    ;IS THIS THE CORRECT CHARACTER?
11466      067510      001413      BEQ     O.SP          ;JUMP IF YES
11467      067512      022704      071407      CMP     #O.TL+O.LG,R4 ;IS THE SEARCH DONE?
11468      067516      101373      BHI     O.RSP         ;BRANCH IF NOT
```

```
11469 067520 042700 177770          BIC    #177770,R0      :MASK OFF OCTAL
11470 067524 010004          MOV    R0,R4
11471 067526 006304          O.SP1: ASL    R4
11472 067530 062704 067310          ADD    #0,URO,R4     :GENERATE ADDRESS
11473 067534 005202          INC    R2             :SET FOUND FLAG
11474 067536 000444          BR     O.SCAN        :GO FIND NEXT CHARACTER
11475 067540 162704 071372          O.SP:  SUB   #0,TL-7,R4 :CORRECT CONSTANT
11476 067544 000770          BR     O.SP1
11477
11478          :
11479          : _ HANDLER - OPEN INDEXED ON THE PC
11480 067546 004737 071302          O.ORPC: JSR   PC,O.TCLS
11481 067552 010502          MOV    R5,R2         :CURRENT ADDRESS IN R2
11482 067554 061202          ADD    @R2,R2        :COMPUTE
11483 067556 006202          ASR    R2             :MOVE ONE BIT TO CARRY
11484 067560 103421          BCS    O.ERR         :ERROR IF ODD NUMBER
11485 067562 006302          ASL    R2             :RESTORE WORD
11486 067564 005722          TST    (R2)+         : AND INCREMENT BY TWO
11487 067566 010205          MOV    R2,R5         :UPDATE CAD
11488 067570 000137 070042          JMP    O.OP2         :GO FINISH UP
11489
11490          :
11491          : B HANDLER - SET AND REMOVE BREAKPOINTS
11492 067574 005702          O.BKPT: TST   R2         :IF NO NUMBER TYPED
11493 067576 001406          BEQ    O.RALL        : REMOVE BREAKPOINT
11494 067600 006204          ASR    R4             :CHECK IF ODD
11495 067602 103410          BCS    O.ERR         :JUMP IF ODD
11496 067604 006304          ASL    R4             :RESTORE ONE BIT
11497 067606 010437 067342          MOV    R4,O.ADR1    :SET A BREAKPOINT
11498 067612 000412          BR     O.DCD
11499 067614 012737 071416 067342 O.RALL: MOV   #0,TRTC,O.ADR1 :CLEAR BREAKPOINT
11500 067622 000406          BR     O.DCD
11501
11502          :
11503          : COMMAND DECODER - ODT11
11504          :
11505          : REGISTERS R0-R4 MAY BE USED,
11506          : REGISTER R5 WILL BE CONSIDERED SAFE
11507 067624 052705 000001          O.ERR: BIS    #1,R5     :CLOSE EVERYTHING
11508 067630 012700 000077          MOV    #'?,R0       : ? TO BE TYPED
11509 067634 004537 071234          JSR    5,O.FTYP     : OUTPUT ?
11510 067640 004537 071334          O.DCD: JSR    5,O.CRLS  :TYPE <CR><LF>*
11511 067644 005004          O.DCD1: CLR    R4         : R4 CONTAINS THE CONVERTED OCTAL
11512 067646 005002          CLR    R2           : R2 IS THE NUMBER FOUND FLAG
11513 067650 004537 071156          O.SCAN: JSR    5,O.GET   :GET A CHAR, RETURN IN R0
11514 067654 022700 000060          CMP    #'0,R0       :COMPARE WITH ASCII 0
11515 067660 101013          BHI    O.CLGL       :CHECK LEGALITY IF NON-NUMERIC
11516 067662 022700 000067          CMP    #'7,R0       :COMPARE WITH ASCII 7
11517 067666 103410          BLO    O.CLGL       :CHECK LEGALITY IF NOT OCTAL
11518 067670 042700 177770          BIC    #177770,R0   :CONVERT TO BCD
11519 067674 006304          ASL    R4           : MAKE ROOM
11520 067676 006304          ASL    R4           : IN
11521 067700 006304          ASL    R4           : IN
11522 067702 060004          ADD    R0,R4        :PACK THREE BITS IN R4
11523 067704 005202          INC    R2           :R2 HAS NUMERIC FLAG
11524 067706 000760          BR     O.SCAN       : AND TRY AGAIN
```

```
11525 067710 005001  
11526 067712 120061 071365  
11527 067716 001405  
11528 067720 005201  
11529 067722 020127 000014  
11530 067726 103336  
11531 067730 000770  
11532 067732 006301  
11533 067734 000171 067740  
11534  
11535 067740 067770  
11536 067742 070022  
11537 067744 067476  
11538 067746 070332  
11539 067750 070034  
11540 067752 067546  
11541 067754 070066  
11542  
11543 067756 070076  
11544 067760 070154  
11545 067762 070150  
11546 067764 067574  
11547 067766 070440  
11548 000030  
11549  
11550  
11551  
11552 067770 005702  
11553 067772 001410  
11554 067774 010405  
11555 067776 006205  
11556 070000 103711  
11557 070002 006305  
11558 070004 011500  
11559 070006 004537 071072  
11560 070012 000714  
11561 070014 042705 000001  
11562 070020 000766  
11563  
11564  
11565  
11566 070022 004737 071302  
11567 070026 052705 000001  
11568 070032 000702  
11569  
11570  
11571  
11572 070034 004737 071302  
11573 070040 005725  
11574 070042 004537 071326  
11575 070046 010500  
11576 070050 004537 071072  
11577 070054 012700 000057  
11578 070060 004537 071234  
11579 070064 000744  
11580  
:  
O.CLGL: CLR R1 ;CLEAR INDEX  
O.LGL1: CMPB R0,O.LGCH(R1) ;DO THE CODES MATCH?  
BEQ O.LGL2 ;JUMP IF YES  
INC R1 ;SET INDEX FOR NEXT SEARCH  
CMP R1,#O.CLGT ;IS THE SEARCH DONE?  
BHIS O.ERR ; OOPS!  
BR O.LGL1 ;RE-LOOP  
O.LGL2: ASL R1 ;MULTIPLY BY TWO  
JMP @O.LGDR(R1) ;GO TO PROPER ROUTINE  
:  
O.LGDR: O.WRD : / OPEN WORD  
O.CRET : CARRIAGE RETURN CLOSE  
O.REGT : $ REGISTER OPS  
O.GO : G GO TO ADDRESS K  
O.OP1 : <LF> MODIFY, CLOSE, OPEN NEXT  
O.ORPC : * OPEN RELATED, INDEX - PC  
O.BACK : * OPEN PREVIOUS  
:  
O.OFST : O OFFSET  
O.WSCH : W SEARCH WORD  
O.EFF : E SEARCH EFFECTIVE ADDRESS  
O.BKPT : B BREAKPOINTS  
O.PROC : P PROCEED  
O.LGL = -O.LGDR ;LGL MUST EQUAL 2X CHLGT ALWAYS  
:  
: PROCESS / - OPEN WORD  
:  
O.WRD: TST R2 ;GET VALUE IF R2 IS NON-ZERO  
BEQ O.WRDA ;SKIP OTHERWISE  
MOV R4,R5 ;PUT VALUE IN CAD  
O.WRD1: ASR R5 ;MOVE ONE BIT TO CARRY  
O.ERR2: BCS O.ERR ;JUMP IF ODD ADDRESS  
ASL R5 ;RESTORE THE CARRY BIT  
MOV @R5,R0 ;GET CONTENTS OF WORD  
JSR 5,O.CADV ;GO GET AND TYPE OUT @CAD  
BR O.DCD1 ;GO BACK TO DECODER  
O.WRDA: BIC #1,R5 ;CLEAR CLOSED BIT  
BR O.WRD1 ;GO BACK TO MAIN-LINE  
:  
: PROCESS CARRIAGE RETURN  
:  
O.CRET: JSR PC,O.TCLS ;CLOSE LOCATION  
BIS #1,R5 ;CLOSE EVERYTHING  
BR O.DCD ;RETURN TO DECODER  
:  
: PROCESS <LF>, OPEN NEXT WORD  
:  
O.OP1: JSR PC,O.TCLS ;CLOSE PRESENT CELL  
TST (R5)+ ;GENERATE NEW ADDRESS  
O.OP2: JSR 5,O.CRLF ;<CR><LF>  
MOV R5,R0 ;NUMBER TO TYPE  
JSR 5,O.CADV ;TYPE OUT ADDRESS  
MOV #1,R0 ;TYPE A /  
JSR 5,O.FTYP  
BR O.WRD1 ;GO PROCESS IT  
:
```

```
11581 ; PROCESS ^, OPEN PREVIOUS WORD
11582 ;
11583 070066 004737 071302 O.BACK: JSR PC,O.TCLS
11584 070072 005745 TST -(R5) ;GENERATE NEW ADDRESS
11585 070074 000762 BR 0.OP2 ;GO DO THE REST
11586 ;
11587 ; PROCESS O, COMPUTE OFFSET
11588 ;
11589 070076 006205 O.OFST: ASR R5 ;GET LOW ORDER BIT
11590 070100 103737 BCS O.ERR2 ;ERROR IF CLOSED
11591 070102 006305 ASL R5 ;RESTORE WORD
11592 070104 012700 000040 MOV #' ,R0 ;TYPE ONE BLANK
11593 070110 004537 071234 JSR 5,O.FTYP ; AS A SEPARATOR
11594 070114 160504 SUB R5,R4 ;COMPUTE
11595 070116 005304 DEC R4 ;
11596 070120 005304 DEC R4 ; 16 BIT OFFSET
11597 070122 010400 MOV R4,R0 ;TYPE A
11598 070124 010402 MOV R4,R2 ;SAVE R4
11599 070126 004537 071072 JSR 5,O.CADV ;NUMBER IN R0 - WORD MODE
11600 070132 010200 MOV R2,R0
11601 070134 006200 ASR R0 ;DIVIDE BY TWO
11602 070136 103402 BCS O.OF1 ;BRANCH IF ODD
11603 070140 004537 071072 JSR 5,O.CADV ;NUMBER IN R0 - BYTE MODE
11604 070144 000137 067644 O.OF1: JMP O.DCD1 ;ALL DONE
11605 ;
11606 ; SEARCHES - $MSK HAS THE MASK
11607 ; $MSK+2 HAS THE FWA
11608 ; $MSK+4 HAS THE LWA
11609 ;
11610 ;
11611 ;
11612 ;
11613 ;
11614 070150 005201 O.EFF: INC R1 ;SET EFFECTIVE SEARCH
11615 070152 000401 BR O.WDS
11616 070154 005001 O.WSCH: CLR R1 ;SET WORD SEARCH
11617 070156 005702 O.WDS: TST R2 ;CHECK FOR OBJECT FOUND
11618 070160 001621 O.ERR1: BEQ O.ERR ;ERROR IF NO OBJECT
11619 070162 013702 067336 MOV O.MSK+2,R2 ;SET ORIGIN
11620 070166 013705 067334 MOV O.MSK,R5 ;SET MASK
11621 070172 005105 COM R5 ;AND COMPLEMENT IT
11622 070174 020237 067340 O.WDS2: CMP R2,O.MSK+4 ; IS THE SEARCH ALL DONE?
11623 070200 101217 BHI O.DCD ; YES
11624 070202 011200 MOV @R2,R0 ; GET OBJECT
11625 070204 005701 TST R1 ;NO
11626 070206 001027 BNE O.EFF1 ;BRANCH IF EFFECTIVE SEARCH
11627 070210 010046 MOV R0,-(SP)
11628 070212 010403 MOV R4,R3 ;EXCLUSIVE OR
11629 070214 040400 BIC R4,R0 ; IS DONE
11630 070216 042603 BIC (SP)+,R3 ; IN A VERY
11631 070220 050003 BIS R0,R3 ; FANCY MANNER HERE
11632 070222 040503 BIC R5,R3 ;AND RESULT WITH MASK
11633 070224 001016 O.WDS3: BNE O.WDS4 ;RE-LOOP IF NO MATCH
11634 070226 010446 MOV R4,-(SP) ;REGISTERS R2,R4, AND R5 ARE SAFE
11635 070230 004537 071326 JSR 5,O.CRLF ;TYPE <CR,LF>
11636 070234 010200 MOV R2,R0 ;GET READY TO TYPE
```



```
11637 070236 004537 071072      JSR      5,0.CADV      ; TYPE ADDRESS
11638 070242 012700 000057      MOV      #1,R0        ; SLASH TO R0
11639 070246 004537 071234      JSR      5,0.FTYP     ; TYPE IT
11640 070252 011200              MOV      @R2,R0       ; GET CONTENTS
11641 070254 004537 071072      JSR      5,0.CADV     ; TYPE CONTENTS
11642 070260 012604              MOV      (SP)+,R4     ; RESTORE R4
11643 070262 005722      O.WDS4: TST      (R2)+ ; INCREMENT TO NEXT CELL AND
11644 070264 000743              BR       0.WDS2      ; RETURN
11645 070266 020004      O.EFF1: CMP      R0,R4 ; IS (X)=K?
11646 070270 001755              BEQ     0.WDS3      ; TYPE IF EQUAL
11647 070272 010003              MOV      R0,R3      ; (X) TO R3
11648 070274 060203              ADD      R2,R3      ; (X)+X
11649 070276 005203              INC      R3
11650 070300 005203              INC      R3          ; (X)+X+2
11651 070302 020304              CMP      R3,R4      ; IS (X)+X+2=K?
11652 070304 001747              BEQ     0.WDS3      ; BRANCH IF EQUAL
11653 070306 042700 177400      BIC      #177400,R0  ; WIPE OUT EXTRANEIOUS BITS
11654 070312 110000              MOV     R0,R0       ; EXTEND SIGN
11655 070314 000257              CCC
11656 070316 006300              ASL     R0          ; MULTIPLY BY TWO
11657 070320 005200              INC     R0          ; ADD TWO
11658 070322 005200              INC     R0
11659 070324 060200              ADD     R2,R0       ; ADD PC
11660 070326 020004              CMP     R0,R4      ; IS THE RESULT A PROPER REL. BRANCH?
11661 070330 000735              BR     0.WDS3
11662
11663      ; PROCESS G - GO
11664
11665 070332 105037 071355      O.GO:   CLRB    0.P   ; DISALLOW PROCEED
11666 070336 006204              ASR     R4          ; CHECK LOW ORDER BIT
11667 070340 103617              BCS    0.ERR2     ; ERROR IF ODD NUMBER
11668 070342 006304              ASL     R4          ; RESTORE WORD
11669 070344 010437 067326      MOV     R4,0.UPC  ; SET UP NEW PC
11670 070350 112737 000340 177776      MOV     #0,STM,ST ; SET HIGH PRIORITY
11671 070356 004537 071024      JSR     5,0.RSTT  ; RESTORE TELETYPE
11672 070362 105037 071354      O.TBIT: CLRB    0.T   ; CLEAR BOTH
11673 070366 042737 000020 067330      BIC     #0.TBT,0.UST ; T-BIT FLAGS
11674 070374 017737 176742 067346      MOV     @0.ADR1,0.UIN ; SAVE INSTRUCTION
11675 070402 013777 071416 176732      MOV     0.TRTC,@0.ADR1 ; REPLACE WITH TRAP
11676 070410 012600      O.G02:  MOV     (SP)+,R0 ; RESTORE
11677 070412 012601              MOV     (SP)+,R1   ; R0
11678 070414 012602              MOV     (SP)+,R2   ; THRU
11679 070416 012603              MOV     (SP)+,R3
11680 070420 012604              MOV     (SP)+,R4
11681 070422 012605              MOV     (SP)+,R5
11682 070424 012606              MOV     (SP)+,SP
11683 070426 013746 067330      MOV     0.UST,-(SP) ; AND SP
11684 070432 013746 067326      MOV     0.UPC,-(SP) ; AND STATUS
11685 070436 000006      O.RTIT: RTT      ; AND PC
11686
11687      ; PROCESS P - PROCEED
11688      ; ONLY ALLOWED AFTER A BREAKPOINT
11689
11690 070440 105737 071355      O.PROC: TSTB    0.P   ; CHECK LEGALITY OF PROCEED
11691 070444 001645              BEQ     0.ERR1     ; NOT LEGAL
11692 070446 105037 071355              CLRB    0.P       ; CLEAR PROCEED FLAG
```

```

11693 070452 005702          TST      R2          ;WAS COUNT SPECIFIED?
11694 070454 001402          BEQ      C.PR1      ;NO
11695 070456 010437 067344    MOV      R4,O.CT    ;YES, PUT AWAY COUNT
11696 070462 112737 000340 177776  O.PR1:  MOVB   #0,STM,ST   ;FORCE HIGH PRIORITY
11697 070470 004537 071024          JSR      5,O.RSTT   ;RESTORE TTY
11698 070474 112737 000340 177776  O.C1:   MOVB   #0,STM,ST   ;SET HIGH PRIORITY
11699 070502 105237 071354          INCB    O.T         ;SET T-BIT FLAG
11700 070506 052737 000020 067330    BIS     #O.TBT,O.UST ;SET T-BIT
11701 070514 000735          BR      O.G02
11702
11703          ;
11704          ; BREAKPOINT HANDLER
11705          ; A TRT BREAKPOINT CAUSES O.BRK TO BE ENTERED, WHICH SAVES
11706          ; VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
11707          ; AND GIVES CONTROL TO THE COMMAND DECODER
11708 070516 012637 067326    O.BRK:  MOV     (SP)+,O.UPC ;PRIORITY IS 7 UPCN ENTRY
11709 070522 012637 067330          MOV     (SP)+,O.UST   ;SAVE STATUS AND PC
11710 070526 004037 070734    O.BK1:  JSR     O,O.SVR   ;SAVE VARIOUS REGISTERS
11711 070532 105737 071354          TSTB   O.T          ;CHECK FOR T-BIT SET
11712 070536 001311          BNE     O.TBIT       ;JUMP IF SET
11713 070540 013777 067346 176574    MOV     O.UIN,#O.ADR1 ;REMOVE BREAKPOINTS
11714 070546 105737 067332          TSTB   O.PRI        ;CHECK IF PRIORITY
11715 070552 100003          BPL     O.BK2        ; IS AS SAME AS USER PGM
11716 070554 113705 067330          MOVB   O.UST,R5     ;PICK UP USER UST IF SO
11717 070560 000407          BR      O.BK3        ;AND DON'T COMPUTE THE PRIORITY
11718 070562 113705 067332    O.BK2:  MOVB   O.PRI,R5   ;OTHERWISE PICK UP ACTUAL PRIORITY
11719 070566 000257          CCC
11720 070570 106005          RORB   R5           ;SHIFT LOW ORDER BITS
11721 070572 106005          RORB   R5           ; INTO
11722 070574 106005          RORB   R5           ; HIGH ORDER
11723 070576 106005          RORB   R5           ; POSITION
11724 070600 110537 177776    O.BK3:  MOVB   R5,ST      ;PUT THE STATUS AWAY WHERE IT BELONGS
11725 070604 013705 067326          MOV     O.UPC,R5    ;GET PC, IT POINTS TO THE TRT
11726 070610 005745          TST    -(R5)        ;SUBTRACT TWO
11727 070612 010537 067326          MOV     R5,O.UPC   ;FROM THE USER'S PC
11728 070616 020537 067342          CMP     R5,O.ADR1  ;COMPARE WITH LIST
11729 070622 001417          BEQ     O.B2        ;JUMP IF FOUND
11730 070624 004537 070772          JSR     5,O.SVTT   ;SAVE TELETYPE STATUS
11731 070630 004537 071326          JSR     5,O.CRLF   ;
11732 070634 012704 071360          MOV     #O.BD,R4   ;ERROR, NOTHING FOUND
11733 070640 012703 071361          MOV     #O.BD+1,R3 ;
11734 070644 004537 071220          JSR     5,O.TYPE   ;OUTPUT 'BE' FOR BAD ENTRY
11735 070650 010500          MOV     R5,R0      ;
11736 070652 042737 000020 067330    BIC     #O.TBT,O.UST ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
11737 070660 000420          BR      O.B3        ; AND CONTINUE
11738 070662 005337 067344    O.B2:   DEC     O.CT    ;
11739 070666 003302          BGT     O.C1        ;JUMP IF REPEAT
11740 070670 012737 000001 067344    MOV     #1,O.CT    ;RESET COUNT TO 1
11741 070676 105237 071355          INCB   O.P         ;ALLOW PROCEED
11742 070702 004537 070772          JSR     5,O.SVTT   ;SAVE TELETYPE STATUS, R4 IS SAFE
11743 070706 012700 000102          MOV     #'B',R0   ;
11744 070712 004537 071234          JSR     5,O.FTYP   ;TYPE 'B'
11745 070716 013700 067342          MOV     O.ADR1,R0 ;GET ADDRESS OF BREAK
11746 070722 004537 071072    O.B3:   JSR     5,O.CADV   ;TYPE ADDRESS
11747 070726 005005          CLR    R5          ;CLEAR CAD
11748 070730 000137 067640          JMP     O.DCD      ;GO TO DECODER

```

```

11749
11750
11751
11752 070734 012637 071352
11753 070740 010637 067324
11754 070744 012706 067324
11755 070750 010546
11756 070752 010446
11757 070754 010346
11758 070756 010246
11759 070760 010146
11760 070762 013746 071352
11761 070766 005746
11762 070770 000200
11763
11764
11765
11766 070772 113737 177560 071356
11767 071000 113737 177564 071357
11768 071006 105037 177560
11769 071012 105037 177564
11770 071016 004537 071326
11771 071022 000205
11772
11773
11774
11775 071024 004537 071326
11776 071030 105737 177564
11777 071034 100375
11778 071036 032737 004000 177560
11779 071044 001403
11780 071046 105737 177560
11781 071052 100375
11782 071054 113737 071356 177560
11783 071062 113737 071357 177564
11784 071070 000205
11785
11786
11787
11788
11789 071072 010246
11790 071074 012704 071415
11791 071100 012746 000060
11792 071104 010002
11793 071106 042702 177770
11794 071112 061602
11795 071114 110244
11796 071116 006200
11797 071120 006200
11798 071122 006200
11799 071124 020427 071410
11800 071130 101365
11801 071132 042700 177776
11802 071136 062600
11803 071140 110044
11804 071142 012703 071415

:
: SAVE REGISTERS R0-R6 IN INTERNAL STACK
:
O.SVR: MOV (SP)+,O.XXX ;PICK REGISTER FROM STACK AND SAVE
MOV SP,O.USP ;SAVE USER STACK ADDRESS
MOV #O.USP,SP ;SET TO INTERNAL STACK
MOV R5,-(SP) ;SAVE
MOV R4,-(SP) ;REGISTERS
MOV R3,-(SP) ;1
MOV R2,-(SP) ;THRU
MOV R1,-(SP) ;5
MOV O.XXX,-(SP) ;PUT SAVED REGISTER ON STACK
TST -(SP)
RTS R0

:
: SAVE TELETYPE STATUS
:
O.SVTT: MOVB O.RCSR,O.CSR1 ;SAVE R C/SR
MOVB O.TCSR,O.CSR2 ;SAVE T C/SR
CLRB O.RCSR ;CLEAR ENABLE AND MAINTENANCE
CLRB O.TCSR ;BITS IN BOTH C/SR
JSR 5,O.CRLF ;TYPE <CR,LF>
RTS R5

:
: RESTORE TELETYPE STATUS
:
O.RSTT: JSR 5,O.CRLF ;<CR,LF> BEFORE RESTORING
TSTB O.TCSR ;WAIT READY ON PRINTER
BPL -4
BIT #4000,O.RCSR ;CHECK BUSY FLAG ON READER
BEQ O.RSE1 ;SKIP READY LOOP IF NOT BUSY
TSTB O.RCSR ;WAIT READY
BPL -4 ;ON READER
O.RSE1: MOVB O.CSR1,O.RCSR ;RESTORE
MOVB O.CSR2,O.TCSR ;THE STATUS REGISTERS
RTS R5

:
: TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
: WORD IS IN R0
:
O.CADV: MOV R2,-(SP) ;SAVE R2
MOV #O.BUF+6,R4 ;BUFFER START ADDRESS
MOV #'0,-(SP) ;CONSTANT ASCII 0
O.SPC: MOV R0,R2 ;GET
BIC #177770,R2 ;OCTAL CHARACTER
ADD @SP,R2 ;CONVERT TO ASCII
MOVB R2,-(R4) ;STORE IN BUFFER
ASR R0 ;SHIFT THIS MESS
ASR R0 ;RIGHT
ASR R0 ;THREE WHOLE PLACES
CMP R4,#O.BUF+1 ;DONE?
BHI O.SPC ;NO
BIC #177776,R0 ;GET LAST BIT
ADD (SP)+,R0 ;CONVERT TO ASCII
MOVB R0,-(R4) ;AND PUT IT AWAY
MOV #O.BUF+6,R3 ;LWA

```

```

11805 071146 004537 071220      JSR      5,O.TYPE      ;TYPE WHOLE STRING OF CHARACTERS
11806 071152 012602              MOV      (SP)+,R2      ;RESTORE R2
11807 071154 000205              RTS      R5
11808
11809      ; GENERAL CHARACTER INPUT ROUTINE
11810      ; CHARACTER INPUT GOES TO R0
11811
11812 071156 105737 177560      O.GET:  TSTB     0,RCSR      ;WAIT FOR
11813 071162 100375              BPL     -4              ; INPUT FROM KEYBOARD
11814 071164 113700 177562      MOVB     0,RDB,R0      ;GET A CHARACTER
11815 071170 004537 071234      JSR      5,O.FTYP      ;ECHO CHARACTER
11816 071174 042700 177600      BIC      #177600,R0    ;STRIP OFF PARITY FROM CHARACTER
11817 071200 001766              BEQ     0,GET          ;IGNORE NULLS
11818 071202 122700 000040      CMPB     #40,R0        ;CHECK FOR SPACES
11819 071206 001763              BEQ     0,GET          ;IGNORE NULLS
11820 071210 122700 000073      CMPB     #'',R0        ;CHECK FOR SEMI-COLON
11821 071214 001760              BEQ     0,GET          ;IGNORE THEM IF FOUND
11822 071216 000205              RTS      R5
11823
11824      ; GENERAL CHARACTER OUTPUT ROUTINE
11825      ; ADDRESS OF FIRST BYTE IN R4,
11826      ; ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
11827
11828 071220 020304      O.TYPE:  CMP      R3,R4      ;CHECK FOR COMPLETION
11829 071222 103426              BLO     0,TYP1         ; EXIT WHEN DONE
11830 071224 112400              MOVB     (R4)+,R0      ;GET A CHARACTER
11831 071226 004537 071234      JSR      5,O.FTYP      ;TYPE ONE CHARACTER
11832 071232 000772              BR      0,TYPE         ;LOOP UNTIL DONE
11833
11834      ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
11835
11836 071234 105737 177564      O.FTYP: TSTB     0,TCSR      ;CHECK STATUS
11837 071240 100375              BPL     -4              ;WAIT UNTIL READY
11838 071242 110037 177566      MOVB     R0,0,TDB      ;TYPE ONE CHARACTER
11839 071246 120037 000045      CMPB     R0,#45        ;IS CHAR TO BE FILLED?
11840 071252 001012              BNE     0,TYP1         ;NO
11841 071254 113746 000044      MOVB     @#44,-(SP)    ;YES, INIT THE COUNT
11842 071260 105737 177564      O.TYP2: TSTB     0,TCSR      ;CHECK STATUS
11843 071264 100375              BPL     0,TYP2         ;WAIT UNTIL READY
11844 071266 105037 177566      CLRB     0,TDB          ;GENERATE NULL FILLER
11845 071272 105316              DECB     @SP
11846 071274 003371              BGT     0,TYP2
11847 071276 005726              TST     (SP)+          ;POP STACK
11848 071300 000205      O.TYP1: RTS      R5
11849
11850      ; CLOSE WORD OR BYTE AND EXIT,
11851      ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
11852
11853 071302 006205      O.TCLS: ASR      R5          ;GET LOW ORDER BIT
11854 071304 103405              BCS     0,TC           ;JUMP IF ALREADY CLOSED
11855 071306 006305              ASL     R5
11856 071310 005702              TST     R2             ;IF NO NUMBER WAS TYPED THERE IS
11857 071312 001401              BEQ     0,CLS1         ;NO CHANGE TO THE OPEN CELL
11858 071314 010415              MOV     R4,@R5         ;STORE WORD
11859 071316 000207      O.CLS1: RTS      PC
11860 071320 005746      O.TC:   TST     -(SP)    ;POP EXTRA CELL FROM STACK

```

```
11861 071322 000137 067624          JMP      O.ERR          ;AND SCREAM BLOODY MURDER
11862
11863          : O.CRLF - TYPE <CR,LF>
11864          : O.CRLS - TYPE <CR,LF>*
11865
11866 071326 012703 071363          O.CRLF: MOV      #O.CR+1,R3      ;LWA <CR,LF>
11867 071332 000402          BR          O.CRS
11868 071334 012703 071364          O.CRLS: MOV      #O.CR+2,R3      ;LWA <CR,LF>*
11869 071340 012704 071362          O.CRS:  MOV      #O.CR,R4        ;FWA
11870 071344 004537 071220          JSR          5,O.TYPE          ;TYPE SOMETHING
11871 071350 000205          RTS          R5
11872
11873 071352 000000          O.XXX:  .WORD    0              ;TEMPORARY STORAGE
11874 071354          000          O.T:    .BYTE    0              ; T-BIT FLAG
11875 071355          000          O.P:    .BYTE    0              ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
11876                                     = 1 IF PROCEED ALLOWED
11877 071356          000          O.CSR1: .BYTE    0              ;SAVE CELL - R C/SR
11878 071357          000          O.CSR2: .BYTE    0              ;SAVE CELL - T C/SR
11879
11880
11881 071360 042502          O.BD:    .WORD    'BE
11882
11883 071362          015          O.CR:    .BYTE    015          ; <CR>
11884 071363          012          .BYTE    012          ; <LF>
11885 071364          052          .BYTE    '*'              ; *
11886
11887 071365          057          O.LGCH: .BYTE    '/'          ; /
11888 071366          015          .BYTE    015          ; CARRIAGE RETURN
11889 071367          044          .BYTE    '$'          ; $
11890 071370          107          .BYTE    'G'          ; G
11891 071371          012          .BYTE    012          ; <LF>
11892 071372          137          .BYTE    '*'          ; *
11893 071373          136          .BYTE    'O'          ; O
11894 071374          117          .BYTE    'W'          ; W
11895 071375          127          .BYTE    'E'          ; E
11896 071376          105          .BYTE    'B'          ; B
11897 071377          102          .BYTE    'P'          ; P
11898 071400          120
11899          000014          O.CLGT =      .-O.LGCH          ;TABLE LENGTH
11900
11901 071401          123          O.TL:    .BYTE    'S'          ;DO 1
11902 071402          120          .BYTE    'P'          ;NOT 2
11903 071403          115          .BYTE    'M'          ;CHANGE 3
11904 071404          000          .BYTE    0            ;THE 4
11905 071405          000          .BYTE    0            ;ORDER 5
11906 071406          102          .BYTE    'B'          ;HERE 6
11907          000006          O.LG =      .-O.TL
11908
11909 071407          O.BUF:          ;6 CHAR. BUFFER WITH
11910          071415          =          ;TRAILING BLANK
11911 071415          040          .BYTE    '+6
11912          .EVEN
11913
11914 071416 000003          O.TRTC: TRT          ;TRACE TRAP PROTOTYPE
11915
11916          ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
```

11917
11918 067310 067310
11919 067310 000000
11920 067312 000000
11921 067314 000000
11922 067316 000000
11923 067320 000000
11924 067322 000000
11925 067324 000000
11926 067326 000000
11927 067330 000000
11928 067332 000007
11929 067334 000000
11930 067336 000000
11931 067340 000000
11932
11933
11934
11935
11936 067342 000000
11937 067344 000000
11938 067346 000000
11939 000001

```
;  
O.UR0: 0 = O.ODT-40  
          0 :USER R0  
          0 :R1  
          0 :R2  
          0 :R3  
          0 :R4  
          0 :R5  
O.USP: 0 :USER SP  
O.UPC: 0 :USER PC  
O.UST: 0 :USER ST  
O.PRI: 7 :ODT PRIORITY  
O.MSK: 0 :MASK  
          0 :LOW LIMIT  
          0 :HIGH LIMIT  
:  
: BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT, CT = COUNT,  
: UIN = CONTENTS  
:  
O.ADR1: 0  
O.CT: 0  
O.UIN: 0  
.END
```

ABASE = 177440	2562	2603	2617#						
ACDW1 = 000000	2562	2605							
ACDW2 = 000000	2562	2606							
ACLO = 000010	1443#								
ACPUOP= 000000	2562	2577							
ACT11 005430	2772#	3743*							
ADDW0 = 000000	2562	2607							
ADDW1 = 000000	2562	2608							
ADDW10= 000000	2562								
ADDW11= 000000	2562								
ADDW12= 000000	2562								
ADDW13= 000000	2562								
ADDW14= 000000	2562								
ADDW15= 000000	2562								
ADDW2 = 000000	2562	2609							
ADDW3 = 000000	2562	2610							
ADDW4 = 000000	2562	2611							
ADDW5 = 000000	2562	2612							
ADDW6 = 000000	2562	2613							
ADDW7 = 000000	2562	2614							
ADDW8 = 000000	2562								
ADDW9 = 000000	2562								
ADEVCT= 000000	2562	2568							
ADEVN = 000000	2562	2604							
AENV = 000000	2562	2573							
AENVN = 000000	2562	2574							
AFATAL= 000000	2562	2565							
AMADR1= 000000	2562	2590							
AMADR2= 000000	2562	2594							
AMADR3= 000000	2562	2597							
AMADR4= 000000	2562	2600							
AMAMS1= 000000	2562	2584							
AMAMS2= 000000	2562	2592							
AMAMS3= 000000	2562	2595							
AMAMS4= 000000	2562	2598							
AMSGAD= 000000	2562	2570							
AMSGLG= 000000	2562	2571							
AMSGTY= 000000	2562	2564							
AMTYP1= 000000	2562	2585							
AMTYP2= 000000	2562	2593							
AMTYP3= 000000	2562	2596							
AMTYP4= 000000	2562	2599							
APASS = 000000	2562	2567							
APRIOR= 000000	2562								
APTCSU= 000040	9371	9560#							
APTENV= 000001	9318	9364	9516	9558#					
APTSIZ= 000200	3672	9557#							
APTSP0= 000100	9366	9518	9559#						
ASWREG= 000000	2562	2575							
ATESTN= 000000	2562	2566							
ATTN 005304	2707#	7150	7278	7741	7863	8270	8290	8317	8346
AUNIT = 000000	2562	2569							
AUSWR = 000000	2562	2576							
AVECT1= 000000	2562	2601							
AVECT2= 000000	2562	2602							
BADHDR 005300	2697#	3751*	9012						

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0233

DOCMD 043372	4245	4272	4312	4446	4605	4721	4767	4816	4862	4918	4970	5025	5093
	5145	5200	5277	5285	5359	5367	5448	5456	5482	5532	5540	5571	5579
	5610	5618	5630	5673	5681	5707	5757	5765	5796	5804	5835	5843	5855
	5903	5911	5957	5965	5978	5986	6036	6044	6090	6098	6111	6119	6186
	6194	6243	6251	6295	6303	6315	6323	6331	6363	6371	6415	6423	6435
	6443	6451	6483	6491	6544	6552	6616	6624	6695	6703	6726	6734	6778
	6786	6809	6817	6876	6884	6919	6927	6940	6968	6976	7011	7019	7032
	7068	7076	7090	7123	7196	7204	7218	7251	7341	7349	7468	7476	7559
	7567	7584	7592	7712	7720	7736	7837	7845	7857	7960	7996	8004	8188#
	8518	9002	9020	9036									
DFAT1 001442	2676#												
DPAT2 001444	2677#												
DRA = 000001	1440#	3971	4074										
DRAV 044172	4238	4263	4344	4365	4396	4401	4412	4503	4524	4555	4560	4571	4716
	4789	4811	4884	4914	4950	5002	5089	5125	5177	5269	5351	5513	5738
	5938	6071	6178	6235	6564	6584	6636	6656	6715	6743	6798	6826	6907
	6999	7155	7162	7174	7283	7290	7302	7745	7760	7867	7882	8335#	
DRDY = 000200	1447#												
DRIVS 005434	2774#	3862*	3875*	3892	3908*	3938*	3957	4061*	4128	8034	8129*		
DRIV0 005436	2780#	3755	3864	3910	4023	8110							
DRIV1 005440	2781#												
DRIV2 005442	2782#												
DRIV3 005444	2783#												
DRIV4 005446	2784#												
DRIV5 005450	2785#												
DRIV6 005452	2786#												
DRIV7 005454	2787#												
DROT = 000040	1445#												
DRPAR = 000010	1424#												
DRVMSK = 000007	1404#	3931	4044										
DRVPTR 001342	2635#	3755*	4135	4155*									
DSC = 040000	1451#												
DSWR = 177570	1256#	2536	3659										
DTE = 010000	1433#												
DTYE = 000040	1426#	3963	4065										
DT1 065576	2814	2820	2826	2832	2843	2849	2854	2859	2864	2869	2874	2879	2900
	2906	2916	2932	2992	2997	3002	3007	3042	3082	3097	3102	3107	3112
	3117	3122	3127	3152	3177	3182	3187	3192	3197	3202	3207	3212	3217
	3222	3227	3232	3237	3242	3247	3252	3257	3262	3267	3292	3317	3342
	3347	3352	3357	3362	3367	3372	3377	3422	3427	3432	3437	3442	3447
	3452	3457	3462	3472	3497	3552	3567	3572	3577	3592	3602	11083#	
DT13 066134	2911	2921	2926	2937	2942	2947	2952	2957	2962	2967	2972	2977	2982
	2987	3027	3032	3037	3047	3052	3057	3062	3067	3072	3077	3087	3092
	3132	3137	3142	3147	3157	3162	3167	3172	3272	3277	3282	3287	3297
	3302	3307	3312	3322	3327	3332	3337	3382	3387	3392	3397	3402	3407
	3412	3417	3502	3507	3512	3517	3542	3547	11131#				
DT14 066214	3012	3017	11141#										
DT15 066304	11152#												
DT3 065636	2889	11090#											
DT4 065642	3492	3557	11091#										
DT6 065710	2884	2894	11099#										
DT7 065754	3467	11107#											
DT8 066022	3487	11115#											
DT9 066070	3022	11123#											
D.ACLO = 000100	1500#												
D.BRHM = 000100	1487#	4286	4321	4350	4419	4455	4509	4578	4614	4736	4776	4831	4871

CZR6GCO RK611 DU PORT LGC
 CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 PAGE 237
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0235

D.SPIN= 010000	1479#	4284	4319	4348	4372	4417	4453	4507	4531	4576	4612	4734	4774
	4829	4869	4931	4954	4983	5006	5032	5056	5106	5129	5158	5181	5207
	5231	5297	5318	5379	5400	5464	5495	5551	5590	5637	5689	5720	5776
	5815	5862	5919	6001	6052	6134	6202	6259	6343	6384	6463	6504	6567
	6639	7103	7130	7231	7258	7381	7516	7535	7636	7655	7769	7808	7890
	7929	7973	9043										
D.SPLS= 010000	1506#												
D.SPOK= 001000	1490#	4286	4321	4350	4419	4455	4509	4578	4614	4736	4776	4831	4871
	4933	4956	4985	5008	5034	5058	5108	5131	5160	5183	5209	5233	5299
	5320	5381	5402	5466	5497	5553	5592	5639	5691	5722	5778	5817	5864
	5921	6003	6054	6136	6204	6261	6345	6386	6465	6506	6569	6641	7105
	7132	7233	7260	7383	7518	7537	7638	7657	7771	7810	7892	7931	7975
	9045												
D.SSP = 000020	1485#	4286	4321	4350	4419	4455	4509	4578	4614	4736	4776	4831	4871
	4933	4956	4985	5008	5034	5058	5108	5131	5160	5183	5209	5233	5299
	5320	5381	5402	5466	5497	5553	5592	5639	5691	5722	5778	5817	5864
	5921	6003	6054	6136	6204	6261	6345	6386	6465	6506	6569	6641	7105
	7132	7233	7260	7383	7518	7537	7638	7657	7771	7810	7892	7931	7975
	9045												
D.SUNS= 040000	1522#												
D.TIB = 002000	1518#												
D.UNLD= 040000	1495#												
D.UNS = 040000	1508#												
D.VV = 000100	1473#	4248	4284	4319	4348	4372	4417	4453	4507	4531	4576	4612	4734
	4774	4829	4869	4931	4954	4983	5006	5032	5056	5106	5129	5158	5181
	5207	5231	5297	5318	5379	5400	5464	5495	5551	5590	5637	5689	5720
	5776	5815	5862	5919	6001	6052	6134	6202	6259	6343	6384	6463	6504
	6567	6639	7103	7130	7231	7258	7381	7516	7535	7636	7655	7769	7808
	7890	7929	7973	9043									
D.WCUR= 000040	1513#												
D.WGAT= 000100	1514#												
D.WLE = 004000	1505#												
D.WRL = 004000	1478#												
D.XERR= 001000	1517#												
ECCW = 020000	1466#												
ECH = 000100	1427#												
EMTVEC= 000030	1345#	3643*	3644*										
EM1 057122	8138	10460#											
EM10 057776	2847	10538#											
EM11 060061	2852	10547#											
EM12 060125	2857	2867	2877	3205	3210	3215	3220	3235	3260	3340	3420	3455	3565
	10554#												
EM13 060163	3040	3110	3115	3175	3185	3225	3265	3345	3365	10560#			
EM14 060205	2887	10564#											
EM15 060243	2995	10570#											
EM16 060352	3575	3590	10582#										
EM17 060415	2950	2970	3025	3035	3050	3060	3130	3155	3270	3295	3320	3380	3400
	3500	3540	10588#										
EM18 060436	2909	2924	2940	2955	2975	3065	3085	3135	3160	3275	3300	3325	3385
	3405	3505	10591#										
EM19 060457	2960	2980	3030	3045	3055	3070	3140	3165	3280	3305	3330	3390	3410
	3510	3545	10594#										
EM2 057175	2812	10468#											
EM20 060500	2919	2935	2945	2965	2985	3075	3090	3145	3170	3285	3310	3335	3395
	3415	3515	10597#										
EM21 060521	2862	2872	2904	2914	3435	3460	3495	3550	3570	10600#			

EM22	060543	3120	10604#															
EM23	060650	3125	10616#															
EM24	060750	2930	10628#															
EM25	061003	3005	10633#															
EM26	061055	2990	10641#															
EM27	061134	3080	10649#															
EM28	061253	3872	10663#															
EM29	061361	3360	3450	10676#														
EM3	057265	9085	10478#															
EM30	061427	3105	10683#															
EM31	061450	3000	3100	3190	3370	3425	3470	10686#										
EM32	061473	3150	10690#															
EM33	061512	3180	3200	3375	10693#													
EM34	061535	3195	3230	10697#														
EM35	061564	3255	10701#															
EM36	061612	3485	3490	3555	10705#													
EM37	061655	3315	10711#															
EM38	061706	3350	10716#															
EM39	061761	3010	3290	3440	10724#													
EM4	057340	10486#																
EM40	062030	3015	3445	10731#														
EM5	057414	2818	3250	10494#														
EM55	062066	3355	10736#															
EM6	057435	2824	10497#															
EM63	062121	3430	10741#															
EM65	062142	3465	10744#															
EM69	062164	3600	10747#															
EM7	057456	2830	10500#															
EM73	062215	3240	10752#															
EM74	062236	3095	10755#															
EM79	062263	3245	10759#															
EM8	057646	10522#																
EM80	062304	2882	10762#															
EM82	062343	2892	10768#															
EM83	062415	2898	10776#															
EM84	062452	3241	3246	3251	10781#													
EM9	057722	2841	10530#															
EM93	062543	3020	10791#															
ERRVEC=	000004	1338#	3657	3658*	3669*	3676*	3677*	3690*	3691*	3760*	3764*	3771*	3808*	3826*				
E.A0	005404	3827*	9244	9245*	9247*	9250*												
		2753#	4284*	4319*	4348*	4372*	4373	4417*	4453*	4507*	4531*	4532	4576*	4612*				
		4734*	4774*	4829*	4869*	4931*	4954*	4983*	5006*	5032*	5056*	5106*	5129*	5158*				
		5181*	5207*	5231*	5297*	5318*	5379*	5400*	5464*	5495*	5551*	5590*	5637*	5689*				
		5720*	5776*	5815*	5862*	5919*	6001*	6052*	6134*	6202*	6259*	6343*	6384*	6463*				
		6504*	6567*	6639*	7103*	7130*	7231*	7258*	7381*	7516*	7535*	7636*	7655*	7769*				
		7808*	7890*	7929*	7973*	8382*	8383*	8390*	8394	8396*	8425	9043*	11131	11141				
E.A1	005410	11152																
		2755#	4286*	4321*	4350*	4373*	4374*	4419*	4455*	4509*	4532*	4533*	4578*	4614*				
		4736*	4776*	4831*	4871*	4933*	4956*	4985*	5008*	5034*	5058*	5108*	5131*	5160*				
		5183*	5209*	5233*	5299*	5320*	5381*	5402*	5466*	5497*	5553*	5592*	5639*	5691*				
		5722*	5778*	5817*	5864*	5921*	6003*	6054*	6136*	6204*	6261*	6345*	6386*	6465*				
		6506*	6569*	6641*	7105*	7132*	7233*	7260*	7383*	7518*	7537*	7638*	7657*	7771*				
E.A2	005414	7810*	7892*	7931*	7975*	8384*	8385*	8398	8400*	8445	9045*	11131	11141	11152				
		2757#	4288*	4323*	4352*	4421*	4457*	4511*	4580*	4616*	4738*	4778*	4833*	4873*				
		4935*	4958*	4987*	5010*	5036*	5060*	5110*	5133*	5162*	5185*	5211*	5235*	5301*				
		5322*	5383*	5404*	5468*	5499*	5555*	5594*	5641*	5693*	5724*	5780*	5819*	5866*				

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 PAGE 242
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0240

MSG4	056135	3724	10359#							
MSG5	056167	3726	10364#							
MSG6	056235	9130	10371#							
MSG7	056271	3741	10376#							
MSG74	056760	8993	10438#							
MSG75	057026	8995	10445#							
MSG76	057063	9063	10450#							
MSG77	057105	9068	10454#							
MSG8	056326	4130	10381#							
MSP =	000100	1459#								
MULT6	046760	8810	8836#							
M.ALGN=	040000	1532#								
M.DPID	050210	9173#								
M.DP40	050246	9184#	9206							
M.DP41	050302	9189	9197#							
M.DP42	050312	9195	9201#							
M.DP44	050344	9208	9214#							
M.DP50	050356	9182	9219#							
M.DRV =	000007	1526#								
M.HEAD=	007000	1534#	8673							
M.ID =	000003	1531#								
M.PAR =	100000	1535#	8955							
M.SECT=	000760	1533#	8602	8663						
M.SER =	077770	1527#								
NED =	010000	1414#	3973	3985	4076	4079	8498			
NEM =	004000	1413#								
NUDRV	012472	3999	4109#	8036						
NXF =	000004	1423#								
OFFSET=	000015	1381#								
OFST =	000004	1442#								
OPI =	020000	1434#								
OR =	000200	1409#								
O.ADR1	067342	11446*	11497*	11499*	11674	11675*	11713*	11728	11745	11936#
O.BACK	070066	11541	11583#							
O.BD	071360	11732	11733	11881#						
O.BKPT	067574	11492#	11546							
O.BK1	070526	11440	11710#							
O.BK2	070562	11715	11718#							
O.BK3	070600	11717	11724#							
O.BRK	070516	11457	11708#							
O.BUF	071407	11790	11799	11804	11909#					
O.B2	070662	11729	11738#							
O.B3	070722	11737	11746#							
O.CADV	071072	11559	11576	11599	11603	11637	11641	11746	11789#	
O.CLGL	067710	11515	11517	11525#						
O.CLGT=	000014	11529	11899#							
O.CLS1	071316	11857	11859#							
O.CR	071362	11866	11868	11869	11883#					
O.CRET	070022	11536	11566#							
O.CRLF	071326	11574	11635	11731	11770	11775	11866#			
O.CRLS	071334	11510	11868#							
O.CRS	071340	11867	11869#							
O.CSR1	071356	11766*	11782	11877#						
O.CSR2	071357	11767*	11783	11878#						
O.CT	067344	11695*	11738*	11740*	11937#					
O.C1	070474	11698#	11739							

O.DCD	067640	11498	11500	11510#	11568	11623	11748		
O.DCD1	067644	11511#	11560	11604					
O.EFF	070150	11545	11614#						
O.EFF1	070266	11626	11645#						
O.ENTR	067354	11437#							
O.ERR	067624	11484	11495	11507#	11530	11556	11618	11861	
O.ERR1	070160	11618#	11691						
O.ERR2	070000	11556#	11590	11667					
O.FTYP	071234	11509	11578	11593	11639	11744	11815	11831	11836#
O.GET	071156	11463	11513	11812#	11817	11819	11821		
O.GO	070332	11538	11665#						
O.GO2	070410	11676#	11701						
O.LG =	000006	11467	11907#						
O.LGCH	071365	11526	11887#	11899					
O.LGDR	067740	11533	11535#	11548					
O.LGL =	000030	11548#							
O.LGL1	067712	11526#	11531						
O.LGL2	067732	11527	11532#						
O.MSK	067334	11619	11620	11622	11929#				
O.ODT	067350	1552	11435#	11918					
O.OFST	070076	11543	11589#						
O.OF1	070144	11602	11604#						
O.OP*	070034	11539	11572#						
O.OP2	070042	11488	11574#	11585					
O.ORPC	067546	11480#	11540						
O.P	071355	11455*	11665*	11690	11692*	11741*	11875#		
O.PRI	067332	11447	11714	11718	11928#				
O.PROC	070440	11547	11690#						
O.PR1	070462	11694	11696#						
O.RALL	067614	11458	11493	11499#					
O.RCSR=	177560	11425#	11766	11768*	11778	11780	11782*	11812	
O.RDB =	177562	11424#	11814						
O.REGT	067476	11463#	11537						
O.RSE1	071054	11779	11782#						
O.RSP	067506	11465#	11468						
O.RST	067412	11436	11445#						
O.RSTT	071024	11671	11697	11775#					
O.RST1	067442	11444	11452#						
O.RTIT	070436	11454*	11685#						
O.SCAN	067650	11474	11513#	11524					
O.SP	067540	11466	11475#						
O.SPC	071104	11792#	11800						
O.SP1	067526	11471#	11476						
O.STM =	000340	11411#	11456	11670	11696	11698			
O.STRT	067400	11435	11442#						
O.SVR	070734	11445	11710	11752#					
O.SVTT	070772	11730	11742	11766#					
O.T	071354	11672*	11699*	11711	11874#				
O.TBIT	070362	11672#	11712						
O.TBT =	000020	11412#	11673	11700	11736				
O.TC	071320	11854	11860#						
O.TCLS	071302	11480	11566	11572	11583	11853#			
O.TCSR=	177564	11427#	11767	11769*	11776	11783*	11836	11842	
O.TDB =	177566	11426#	11838*	11844*					
O.TL	071401	11464	11467	11475	11901#	11907			
O.TRTC	071416	11499	11675	11914#					

O.TVEC=	000014	11410#	11438	11456*	11457*					
O.TYPE	071220	11734	11805	11828#	11832	11870				
O.TYP1	071300	11829	11840	11848#						
O.TYP2	071260	11842#	11843	11846						
O.UIN	067346	11446	11674*	11713	11933#					
O.UPC	067326	11439*	11669*	11684	11708*	11725	11727*	11926#		
O.URO	067310	11442	11472	11919#						
O.USP	067324	11443*	11753*	11754	11925#					
O.UST	067330	11437*	11673*	11683	11700*	11709*	11716	11736*	11927#	
O.WDS	070156	11615	11617#							
O.WDS2	070174	11622#	11644							
O.WDS3	070224	11633#	11646	11652	11661					
O.WDS4	070262	11633	11643#							
O.WRD	067770	11535	11552#							
O.WRDA	070014	11553	11561#							
O.WRD1	067776	11555#	11562	11579						
O.WSCH	070154	11544	11616#							
O.XXX	071352	11752*	11760	11873#						
O.45	067454	11453	11455#							
PACK =	000003	1376#	4244							
PARAM	001336	2632#	3607*	3610*	3719					
PARSRT	010040	1549	3607#							
PAT =	000020	1457#								
PCA =	004000	1464#								
PCD =	010000	1465#								
PCLKF	005460	2790#	3766*	3773*	8898	8913				
PCVEC	001332	2626#	3767	3774						
PCYL	001352	2641#								
PFSRT	013004	4187#	9160							
PGE =	002000	1412#								
PIP =	020000	1450#								
PIRQ =	177772	1255#								
PIRQVE=	000240	1349#								
PKRB	001324	2622#								
PKS	001320	2620#	3765	3772	8903*	8917*				
PKSB	001322	2621#	8902*							
PPTP	005432	2773#	3710*							
PRGSRT	010054	3608	3611#	3874	8139					
PRO =	000000	1272#	3613	3803						
PR1 =	000040	1273#								
PR2 =	000100	1274#								
PR3 =	000140	1275#								
PR4 =	000200	1276#								
PR5 =	000240	1277#	2619							
PR6 =	000300	1278#	3776							
PR7 =	000340	1279#	3677	3683	3827	9155	9156			
PS =	177776	1252#	1253							
PSW =	177776	1253#								
PWRVEC=	000024	1344#	3647*	3648*	9145*	9154*	9155*			
RCYLA	046100	8470	8652#							
RCYLD	046026	8467	8639#							
RDCHR =	104410	9844	10226#							
RDCYLA	045756	4749	4844	8627#						
RDCYLD	045672	8611#								
RDDATA=	000021	1383#	7373							
RDGATE=	100000	1468#								

RDHEAD=	000025	1385#	7790	7911										
RDLIN =	104411	8105	9918	10227#										
RDOCT =	104412	8152	8165	10228#										
RDSEC =	045622	8599#	8808											
RDTAB =	004276	2693#												
RDY =	000200	1391#	8214	8228										
RECAL =	000013	1380#	4271	7145	7273	9019								
RESREG=	104414	9978	10030	10230#	11379									
RESVEC=	000010	1339#												
RHEAD =	046174	8475	8672#											
RHTAB =	001666	2688#	7789	7823	7910	7944	8815	8817	8821	8824	11123			
RKASOF=	000016	1362#	3817	7150	7278	7741	7863	8253	8290	8317	8346			
RKBA =	000004	1357#	3813	7365*	7409*	7427*	7490*	7615*	8249					
RKCS1 =	000000	1355#	3810	3922*	4035*	4241*	4268*	4308*	4390*	4405*	4442*	4549*	4564*	4601*
		4763*	4858*	5021*	5196*	5525*	5626*	5750*	5851*	5950*	6083*	6719*	6747*	6802*
		6830*	6898*	6912*	6990*	7004*	7119*	7147*	7159*	7247*	7275*	7287*	7512*	7731*
		7750*	7852*	7872*	8189*	8200*	8214	8228	8246	8528*	8537*	8546*	8555*	8568*
		9016*	9032*											
RKCS2 =	000010	1359#	3811	3901*	3915*	3919*	4028*	4032*	4242*	4243*	4269*	4270*	4309*	4310*
		4443*	4444*	4602*	4603*	4764*	4765*	4859*	4860*	5022*	5023*	5197*	5198*	5271*
		5273*	5275*	5280*	5282*	5353*	5355*	5357*	5362*	5364*	5442*	5444*	5446*	5451*
		5453*	5526*	5528*	5530*	5535*	5537*	5565*	5567*	5569*	5574*	5576*	5604*	5606*
		5608*	5613*	5615*	5627*	5628*	5667*	5669*	5671*	5676*	5678*	5751*	5753*	5755*
		5760*	5762*	5790*	5792*	5794*	5799*	5801*	5829*	5831*	5833*	5838*	5840*	5852*
		5853*	5897*	5899*	5901*	5906*	5908*	5951*	5953*	5955*	5960*	5962*	5972*	5974*
		5976*	5981*	5983*	6030*	6032*	6034*	6039*	6041*	6084*	6086*	6088*	6093*	6095*
		6105*	6107*	6109*	6114*	6116*	6180*	6182*	6184*	6189*	6191*	6237*	6239*	6241*
		6246*	6248*	6289*	6291*	6293*	6298*	6300*	6317*	6319*	6321*	6326*	6328*	6357*
		6359*	6361*	6366*	6368*	6409*	6411*	6413*	6418*	6420*	6437*	6439*	6441*	6446*
		6448*	6477*	6479*	6481*	6486*	6488*	6538*	6540*	6542*	6547*	6549*	6610*	6612*
		6614*	6619*	6621*	6689*	6691*	6693*	6698*	6700*	6720*	6722*	6724*	6729*	6731*
		6772*	6774*	6776*	6781*	6783*	6803*	6805*	6807*	6812*	6814*	6870*	6872*	6874*
		6879*	6881*	6913*	6915*	6917*	6922*	6924*	6962*	6964*	6966*	6971*	6973*	7005*
		7007*	7009*	7014*	7016*	7062*	7064*	7066*	7071*	7073*	7120*	7121*	7190*	7192*
		7194*	7199*	7201*	7248*	7249*	7335*	7337*	7339*	7344*	7346*	7371*	7372*	7462*
		7464*	7466*	7471*	7473*	7492*	7553*	7555*	7557*	7562*	7564*	7578*	7580*	7582*
		7587*	7589*	7614*	7706*	7708*	7710*	7715*	7717*	7732*	7733*	7785*	7786*	7802*
		7831*	7833*	7835*	7840*	7842*	7853*	7854*	7906*	7907*	7923	7956*	7957*	7990*
		7992*	7994*	7999*	8001*	8247	8515*	8516*	8529*	8530*	8538*	8539*	8547*	8548*
		8556*	8557*	8582*	8586*	8587*	9017*	9018*	9033*	9034*				
RKDA =	000006	1358#	3814	7367*	7411*	7429*	7489*	7618*	8250					
RKDB =	000024	1364#	3819	7629	7797	7798	7799	7918	7919	7920				
RKDC =	000020	1363#	3818	4718*	4755*	4813*	4850*	6313*	6433*	7087*	7215*	7369*	7491*	7616*
		7734*	7784*	7855*	7905*	8254								
RKDS =	000012	1360#	3815	8251										
RKECPS=	000030	1368#	3823	8258										
RKECPT=	000032	1369#	3824	8259										
RKER =	000014	1361#	3816	8252										
RKMR1 =	000026	1365#	3820	4275*	4298*	4664*	8255	8296*	8300*	8321*	8325*	8531*	8540*	8549*
		8588*	8599*	8611*	8627*	8684*	8701*	9023*						
RKMR2 =	000034	1366#	3821	8232	8237	8256								
RKMR3 =	000036	1367#	3822	8233	8238	8257								
RKPRI =	001316	2619#	8181											
RKVEC =	001314	2618#	3746*	8169*	8172*	8179								
RKWC =	000002	1356#	3812	7370*	7493*	7617*	8248							
RLS =	000010	1405#	5275	5357	5446	5530	5569	5608	5671	5755	5794	5833	5901	5955

CZR6GCO RK611 DU PORT LGC
 CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 F 4 PAGE 252
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0250

\$\$SCOPE 050366
 \$\$SETUP= 000137
 \$\$STUP = 177777
 \$\$SUPRS 054314
 \$\$SVLAD 050574
 \$\$SVPC = 000244
 \$\$SWR = 167400

\$\$SWREG 001232
 \$\$SWRMK= 000000
 \$TESTN 001214
 \$TIMES 001174

\$TKB 001146
 \$TKCNT 052322
 \$TKINT 052332
 \$TKQEN= 052331
 \$TKQIN 052324
 \$TKQOU 052326
 \$TKQSR 052330
 \$TKS 001144

\$TKSRV 052402
 \$TMP0 001160
 \$TMP1 001162
 \$TMP2 001164
 \$TMP3 001166
 \$TMP4 001170

\$TMP5 001172
 \$TN = 000041

\$TNPWR 054174
 \$TPB 001152
 \$TPFLG 001157
 \$TPS 001150
 \$TRAP 054562
 \$TRAP2 054604
 \$TRP = 000016

\$TRPAD 054616
 \$STSM 001004
 \$STSNM 001102

3641	9237#													
3604#	3640	3641	3643	3645	3647	3649	3650	3651	3653	8038	8086	9238		
9302	9328	9336	9686	9691	9692	9722	9898							
3604#														
4479	4638	4684	7437	10081#										
9248	9277#													
1558#	1563													
1203#	1218	1223	1224	1225	1226	1227	1228	1229	1230	2552	2553	2554		
3650	3651	3653	3654	3800	3851	4020	4123	4227	4492	4654	4705	4800		
4903	5078	5258	5340	5440	5665	5895	6028	6167	6224	6287	6407	6534		
6606	6685	6768	6866	6958	7058	7186	7329	7459	7700	8024	8039	8054		
8060	8062	9229	9230	9231	9232	9233	9233	9251	9253	9254	9257	9258		
9259	9266	9267	9268	9280	9283	9286	9293	9294	9295	9296	9297	9306		
9313	9325	9329	9341											
2575#	3674													
1230	1231	9233	9234	9255										
2566#	4125*	9278*	11083	11090	11091	11099	11107	11115	11123	11131	11141	11152		
2552#	3650*	3800*	3851*	4020*	4123*	4227*	4492*	4654*	4705*	4800*	4903*	5078*		
5258*	5340*	5440*	5665*	5895*	6028*	6167*	6224*	6287*	6407*	6534*	6606*	6685*		
6768*	6866*	6958*	7058*	7186*	7329*	7459*	7700*	8031*	8039*	9266*	9273	9276*		
9286														
2539#	9410	9417	9437	9641	9662	9673	9703	9731	9758					
9642#	9657*	9692	9709*	9823	9825*									
3617	9657#	9683	9744											
9646#	9717	9828												
9643#	9658*	9659	9715*	9716*	9717	9719*								
9644#	9659*	9826	9827*	9828	9830*									
9645#	9658	9719	9830											
2538#	9408	9415	9437	9641	9663*	9699*	9701	9707*	9729	9745*	9755	9767*		
9787*														
9660	9673#													
2546#														
2547#														
2548#														
2549#														
2550#	3921	3940	3954*	3965*	4034	4049	4052*	4067*	4111*	4153*	4167	6897		
6989	7146	7274	8188	8199	8567									
2551#	3754*	8423*	8477	8481										
1204#	1218	3777	3789	3800#	3828	3835	3851#	4004	4020#	4055	4113	4123#		
4125	4126	4173	4179	4191	4227#	4485	4492#	4644	4654#	4658	4688	4705#		
4792	4800#	4887	4903#	5071	5078#	5246	5258#	5315	5333	5340#	5397	5415		
5440#	5655	5658	5665#	5880	5884	5895#	6018	6021	6028#	6151	6154	6167#		
6217	6224#	6274	6287#	6399	6407#	6520	6534#	6596	6599	6606#	6668	6671		
6685#	6758	6761	6768#	6841	6844	6866#	6951	6958#	7044	7058#	7179	7186#		
7308	7329#	7415	7433	7442	7459#	7674	7700#							
10011	10012	10032#												
2541#	9426*	9437												
2545#	9358	9437												
2540#	9424	9437												
3645	10194#													
10205#	10216													
10209#	10218#	10219#	10220#	10221#	10222#	10223	10224#	10225	10226#	10227#	10228#	10229#		
10230#	10231#	10232#												
10199	10216#													
1583#														
2518#	4126*	8038*	9228	9255	9277*	9278	9283	9287	9305	9341				

CALIB	1692#	4266	9015													
CHECK	1631#	4291	4326	4355	4378	4424	4460	4514	4537	4583	4619	4741	4781	4836	4876	
	4938	4961	4990	5013	5039	5063	5113	5136	5165	5188	5214	5238	5304	5325	5386	
	5407	5471	5502	5558	5597	5644	5696	5727	5783	5822	5869	5926	6008	6059	6141	
	6209	6266	6350	6391	6470	6511	6574	6646	7110	7137	7238	7265	7388	7523	7542	
	7643	7662	7776	7815	7897	7936	7980	9050								
COMMEN	1#	1350#														
CWD2	1646#	4298														
DRCLR	1669#	4307	4441	4600	4762	4857	5020	5195	5625	5850	7118	7246	9031			
ENDCOM	1#	1350#														
EOPGM	1954#	8028														
ERROR	1244#	3831	3896	3904	3918	3927	3950	3959	3978	3982	3993	3997	4031	4040	4058	
	4085	4089	4093	4097	4235	4239	4246	4250	4261	4264	4273	4279	4282	4294	4295	
	4296	4297	4302	4305	4313	4316	4329	4330	4331	4332	4341	4345	4358	4359	4360	
	4361	4367	4371	4381	4382	4383	4384	4397	4403	4409	4413	4427	4428	4429	4430	
	4436	4447	4450	4463	4464	4465	4466	4500	4504	4517	4518	4519	4520	4526	4530	
	4540	4541	4542	4543	4556	4562	4568	4572	4586	4587	4588	4589	4595	4606	4609	
	4622	4623	4624	4625	4661	4713	4717	4722	4726	4730	4744	4745	4746	4747	4756	
	4761	4768	4771	4784	4785	4786	4787	4790	4808	4812	4817	4821	4825	4839	4840	
	4841	4842	4851	4856	4863	4866	4879	4880	4881	4882	4885	4911	4915	4919	4923	
	4927	4941	4942	4943	4944	4951	4964	4965	4966	4967	4971	4975	4979	4993	4994	
	4995	4996	5003	5016	5017	5018	5019	5026	5029	5042	5043	5044	5045	5052	5066	
	5067	5068	5069	5086	5090	5094	5098	5102	5116	5117	5118	5119	5126	5139	5140	
	5141	5142	5146	5150	5154	5168	5169	5170	5171	5178	5191	5192	5193	5194	5201	
	5204	5217	5218	5219	5220	5227	5241	5242	5243	5244	5266	5270	5278	5286	5290	
	5294	5307	5308	5309	5310	5316	5328	5329	5330	5331	5348	5352	5360	5368	5372	
	5376	5389	5390	5391	5392	5398	5410	5411	5412	5413	5449	5457	5461	5474	5475	
	5476	5477	5479	5483	5487	5491	5505	5506	5507	5508	5515	5519	5523	5533	5541	
	5545	5548	5561	5562	5563	5564	5572	5580	5584	5587	5600	5601	5602	5603	5611	
	5619	5623	5631	5634	5647	5648	5649	5650	5656	5674	5682	5686	5699	5700	5701	
	5702	5704	5708	5712	5716	5730	5731	5732	5733	5740	5744	5748	5752	5766	5770	
	5773	5786	5787	5788	5789	5797	5805	5809	5812	5825	5826	5827	5828	5836	5844	
	5848	5856	5859	5872	5873	5874	5875	5881	5904	5912	5916	5929	5930	5931	5932	
	5934	5940	5944	5948	5958	5966	5970	5979	5987	5991	5997	6011	6012	6013	6014	
	6019	6037	6045	6049	6062	6063	6064	6065	6067	6073	6077	6081	6091	6099	6103	
	6112	6120	6124	6130	6144	6145	6146	6147	6152	6175	6179	6187	6195	6199	6212	
	6213	6214	6215	6232	6236	6244	6252	6256	6269	6270	6271	6272	6296	6304	6308	
	6311	6316	6324	6332	6336	6340	6353	6354	6355	6356	6364	6372	6376	6380	6394	
	6395	6396	6397	6416	6424	6428	6431	6436	6444	6452	6456	6460	6473	6474	6475	
	6476	6484	6492	6496	6500	6514	6515	6516	6517	6537	6545	6553	6557	6560	6565	
	6577	6578	6579	6580	6586	6591	6597	6609	6617	6625	6629	6632	6637	6649	6650	
	6651	6652	6658	6663	6669	6688	6696	6704	6708	6711	6717	6727	6735	6739	6745	
	6753	6759	6771	6779	6787	6791	6794	6800	6810	6818	6822	6828	6836	6842	6868	
	6877	6885	6889	6892	6901	6909	6920	6928	6932	6937	6941	6945	6960	6969	6977	
	6981	6984	6993	7001	7012	7020	7024	7029	7033	7037	7060	7069	7077	7081	7084	
	7091	7095	7099	7113	7114	7115	7116	7124	7127	7140	7141	7142	7143	7157	7163	
	7169	7175	7188	7197	7205	7209	7212	7219	7223	7227	7241	7242	7243	7244	7252	
	7255	7268	7269	7270	7271	7285	7291	7297	7303	7333	7342	7350	7354	7358	7375	
	7379	7391	7392	7393	7394	7397	7404	7406	7418	7469	7477	7481	7484	7497	7510	
	7514	7526	7527	7528	7529	7532	7545	7546	7547	7548	7560	7568	7572	7585	7593	
	7597	7623	7631	7634	7646	7647	7648	7649	7665	7666	7667	7668	7704	7713	7721	
	7725	7737	7747	7755	7761	7765	7779	7780	7781	7782	7792	7795	7805	7818	7819	
	7820	7821	7825	7829	7838	7846	7850	7858	7869	7877	7883	7886	7900	7901	7902	
	7903	7913	7916	7926	7939	7940	7941	7942	7946	7950	7961	7965	7969	7983	7984	
	7985	7986	7997	8005	8009	8495	8500	8504	8519	8533	8542	8551	8559	8585	8987	
	8999	9003	9007	9021	9027	9030	9037	9040	9053	9054	9055	9056	9113			

CZR6GCO RK611 DU PORT LGC
CZR6GC.P11 04-JAN-82 12:39

MACY11 30(1046) 04-JAN-82 12:59 PAGE 258
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0255

.\$READ	1#	1203#	9638
.\$R2AZ	1#		
.\$SAVE	1#	1203#	10141
.\$SB2D	1#	1203#	10053
.\$SB2O	1#		
.\$SCOP	1#	1203#	9223
.\$SIZE	1#		
.\$SUPR	1#	1203#	10071
.\$STRAP	1#	1203#	10186
.\$STYPB	1#		
.\$STYPD	1#	1203#	9437
.\$STYPE	1#	1203#	9341
.\$STYPO	1#	1203#	9561
.\$4OCA	1#		
.\$1170	1#		

. ABS. 071420 000

ERRORS DETECTED: 0

CZR6GC,CZR6GC.LST/SOL/CRF/NL:TOC=SYSMAC.SML,CZR6GC.P11
RUN-TIME: 24 30 3 SECONDS
RUN-TIME RATIO: 101/58=1.7
CORE USED: 43K (86 PAGES)