

RK611

UNIBUS RK6 DRIVE PART 1
CZR6HE0

AH-9118E-MC

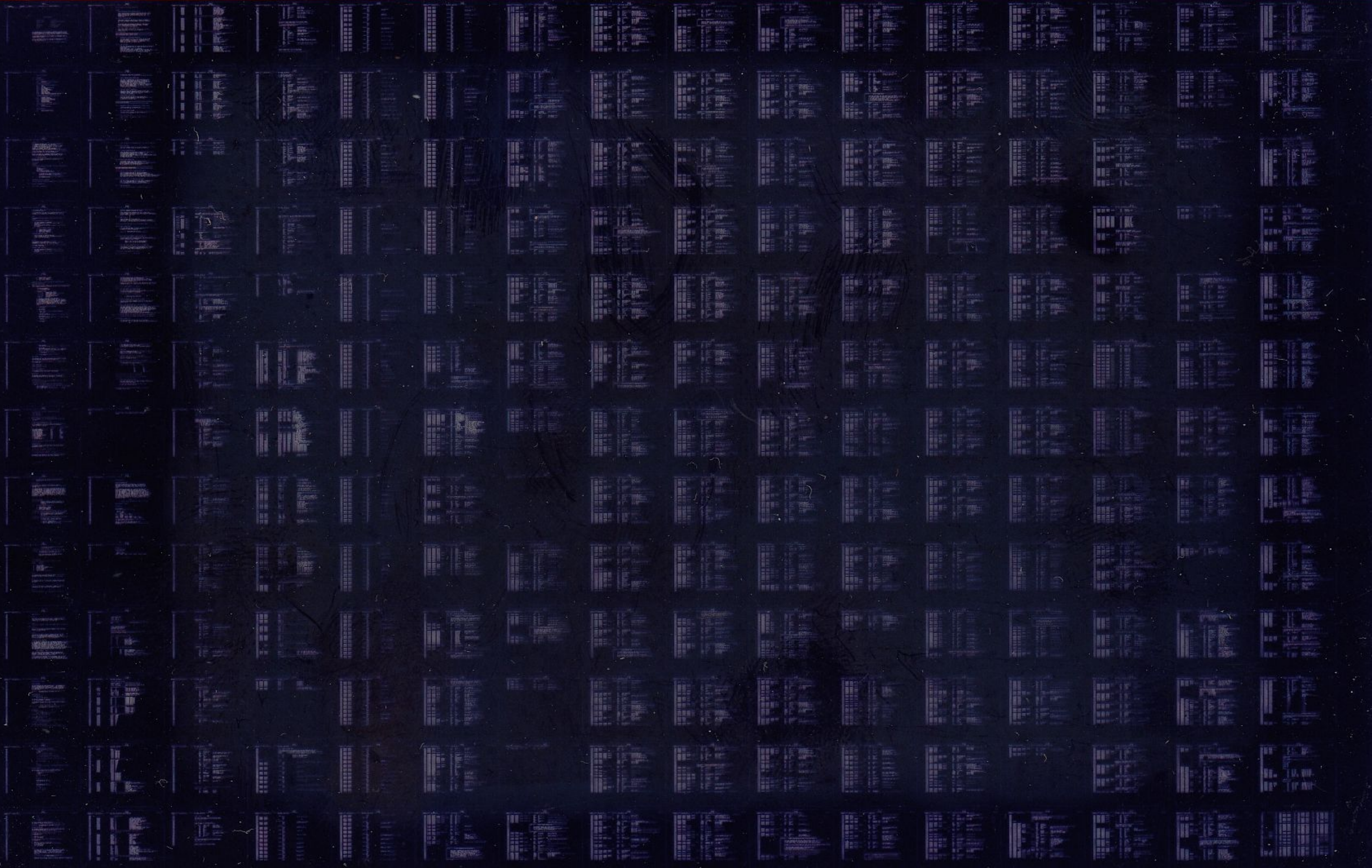
COPYRIGHT © 76-78

FICHE 1 OF 2

MAR 1978

digital

MADE IN USA



RK611

UNIBUS RK6 DRIVE PART 1
CZR6HE0

AH-9118E-MC

MAR 1978

COPYRIGHT © 76-78

digital

FICHE 2 OF 2

MADE IN USA

The image shows a grid of 15 columns and 15 rows of small, illegible data tables or charts. Each cell in the grid contains a small table with multiple columns and rows of text, which is too small to read. The tables appear to be organized in a regular pattern, possibly representing a series of test results or drive parameters. The overall appearance is that of a technical document or a data log.

43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

TABLE OF CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
 - 2.1 HARDWARE
 - 2.2 PRELIMINARY TESTING & PROGRAMS
- 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 DUAL ACCESS
 - 3.5 MEMORY MANAGEMENT
 - 3.6 PARITY CHECK ENABLED
 - 3.7 BAD SECTORS
 - 3.8 EXECUTION TIME
 - 3.9 FAULT ISOLATION
 - 3.10 ERROR CORRECTION & FAILURE RATE ANALYSIS
 - 3.11 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

91
92
93
94
95
96
97
98
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

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 1 OF THE DRIVE DIAGNOSTICS TO INSURE THAT THE RK06 & RK07 DISK DRIVE IS CAPABLE OF PERFORMING ALL STATIC & CYCLE UP TESTS. IT INSURES THAT THE DRIVE CAN WRITE AND READ HEADERS IN BOTH 20 & 22 SECTOR FORMATS. FINALLY, IT INSURES THAT THE DISK CAN PERFORM SEEK OPERATIONS BY DOING SEVERAL SEEK PATTERNS. ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING. AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF PART 1, THE DRIVE IS READY FOR PART 2 OF THE DRIVE DIAGNOSTICS.

TESTING IS BASED ON A HIERARCHY APPROACH STARTING WITH BASIC LOGIC TESTS AND PROCEEDING THRU DYNAMIC TESTING. THE TESTS WILL BE KEPT SMALL TO FACILITATE SCOPING LOOPS.

*****CAUTION*****

HALTING THIS PROGRAM ANYWHERE BUT AT THE END OF A PASS, MAY LEAVE THE HEADERS IN THE DISK CARTRIDGE IN AN UNDETERMINED STATE.

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 8 (TOTAL) RK06/RK07 DRIVES

- NOTES: 1. IF NEITHER KW11-L OR P CLOCK IS USED, ALL TIMING TESTS WILL BE BYPASSED. A MSG AT THE BEGINNING OF THE TESTS WILL CONFIRM THIS.
2. THE PROGRAM CAN WORK OFF EITHER FORMATTED OR NON-FORMATTED PACKS.

2.2 PRELIMINARY TESTING & PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFU

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202

THIS PROGRAM CAN BE USED BY THE PDP-11/04,05,10,20,
34,35,40,45,50, & 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 CAN BE CHAINED BY XXDP & WILL NOT OVERLAY THE
LOADER.

CHAIN MODE OPERATION (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS
DEFAULTED.
3. DRIVE 0 WILL NOT BE TESTED.
4. ALL OTHER DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL
BE TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

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

DUMP MODE OPERATION (MANUAL)

1. INPUT DIALOGUE IF STARTED FROM 220.
2. DRIVE 0 CAN BE TESTED, BUT THE OPERATOR IS FIRST GIVEN
A MSG TO REPLACE THE PACK IN DR0 WITH A SCRATCH
PACK & TYPE <CR> WHEN DONE.

3.3 ACT/APT

THIS PROGRAM IS ACT COMPATIBLE. IT IS APT
COMPATIBLE TO THE EXTENT THAT APT HOOKS WILL BE IN THE
PROGRAM & WILL WORK THRU THE 'UPTON INTERFACE'.

FOR OTHER INTERFACES, APT MAY ONLY LOAD & START THE PROGRAM.
I.E. LOAD & DUMP MODE.

AUTOMATIC MODE (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS
DEFAULTED.
3. ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL BE
TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

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

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:
BIT 7 = 1 ETABLE DOES SIZING
= 0 PROGRAM DOES SIZING
BIT 6 = 1 SPOOL MSGS 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
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

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

- 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.
- 13. CONTROLLER DESCRIPTOR WORDS:
NOT USED
- 14. DEVICE DESCRIPTOR CODES (IN WORDS):
NOT USED

3.4 DUAL ACCESS

THIS PROGRAM WILL NOT TEST OR SUPPORT DUAL-ACCESS. A DRIVE EQUIPED WITH DUAL ACCESS MUST BE SWITCHED TO THE PORT UNDER TEST TO PREVENT CONTENTION WITH THE OTHER PORT.

DUAL ACCESS TESTS WILL BE INCORPORATED IN A SEPARATE PROGRAM AT A LATER DATE.

3.5 MEMORY MANAGEMENT

MEMORY MANAGEMENT NOT USED

3.6 PARITY CHECK ENABLED

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

3.7 BAD SECTOR

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

3.8 EXECUTION TIME

THE EXECUTION TIMES SHOWN BELOW ARE BASED ON THE PDP 11/50.

TOTAL TIME: 5 MIN, 30 SEC

A BREAKDOWN OF THE MORE LENGTHY TESTS ARE SHOWN BELOW:

TEST 16	STATIC CYL ADDRESS & DIFF REGS-PART 2:	2 MIN, 15 SEC
TEST 34	FORMAT PACK	: 1 MIN
TEST 37	SEEK FROM CYL 0 TO ALL CYLS	: 40 SEC
TEST 40	SEEK FROM CYL 410 TO ALL CYLS	: 40 SEC

315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370

THE ABOVE TIMES ARE APPROX. DOUBLED FOR THE RK07.

3.9 FAULT ISOLATION
TO BE DETERMINED.

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

3.11 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 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 SET DRIVES TO BE TESTED IN THE 'LOAD' CONDITION & WITH THE APPROPRIATE PORT SELECTED & WRITE LOCK DISABLED. DRIVES NOT TO BE TESTED MUST HAVE BOTH PORTS DESELECTED.

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

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. CORRECT PORT SELECTED
- C. WRITE LOCK DISABLED
- D. DRIVE READY INDICATOR ON

LOCATION 204 - SAME AS 200 START BUT BYPASS TEST 16 (N SQUARE)

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

LOCATION 230 - SAME AS 220 START BUT BYPASS TEST 16 (N SQUARE)

LOCATION 260 - RUN MODULE TEST ... DEFAULT MODE ONLY.
THIS SKIPS OVER THE FOLLOWING TESTS:

- 1. TEST 35 FORMAT PACK
- 2. TEST 36 DECREMENT FROM CYL 410 TO 0 & READ HEADERS
- 3. TEST 40 SEEK FROM CYL 0 TO ALL
- 4. TEST 41 SEEK FROM CYL 410 TO ALL

371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426

THE PURPOSE OF BYPASSING IS TO PROVIDE
A QUICK MODULE TEST

LOCATION 270 - SAME AS 260 START BUT BYPASS TEST 16 ALSO.

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

ALSO, THE PROGRAM WILL DETERMINE WHETHER
THE DRIVE TO BE TESTED IS AN RK06 OR RK07.
OPERATOR INPUTS ARE NOT REG'D.

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

SWITCH	FUNCTION
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 MSG 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 MSGS. NORMALLY USED WHEN
LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9).
WITH SWITCH <13> SET, SWITCH <15> SHOULD NOT BE SET.

4.3.4 SW<12>

THIS SWITCH BYPASSES A GIVEN DRIVE AFTER 20 ERRORS HAVE

427
428
429
430
431
432
433
434
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
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

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 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 & 3.3.

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: 1,2,4,6

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

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.

539
540
541
542
543
544
545
546
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

ALL OPERATOR RESPONSES ARE UNDERLINED.

UNIBUS RK06-RK07 DRIVE DIAGNOSTIC
PART 1
CZR6HED

DRIVES TO BE TESTED: 1,3<CR>

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>

WILL TEST DRIVES:

1
3

DRIVE 1

(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 DRIVE DIAGNOSTIC
PART 1
CZR6HED

WILL TEST DRIVES:

0
1

DRIVE 0

DRIVE SERIAL NO. AAA
CARTRIDGE SERIAL NO. BBB

DRIVE 1

DRIVE SERIAL NO. CCC
CARTRIDGE SERIAL NO. DDD

END PASS #1

WILL TEST DRIVES:

0
1

DRIVE 0

595
596
597
598
599
600
601
602
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

DRIVE 1

END PASS # 2

(ETC)

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

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

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 & FORMATTING IS VALID,
THE PROGRAM WILL:

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

IF HEADS ARE NOT LOADED AND/OR FORMATTING IS INVALID,
THE PROGRAM WILL:

1. ECHO ↑C
2. TYPE 'HALT PENDING. PLEASE WAIT'
3. DO THE TEST(S) THAT LOADS HEADS AND/OR FORMATS
THE INVALID CYLS
4. TYPE 'CPU HALTED'
5. HALT THE PROGRAM

NOTES:

1. THE ABOVE EXAMPLE IS FOR THE PROGRAM RUNNING IN DUMP
MODE (MANUAL). IF THE PROGRAM IS RUNNING IN CHAIN/AUTO
MODE VIA XXDP,ACT,APT; IT WILL FIRST LOAD HEADS
AND/OR FORMAT CORRECTLY, IF REQ'D, THEN IT WILL
JUMP ON TO THE MONITOR WHERE THE NEXT PROGRAM CAN BE
CALLED IN.

THE TYPEOUTS WILL BE "ABORT PENDING - PLEASE WAIT"
& "PROGRAM ABORTING"

2. 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.

651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
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
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

5.0 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 SEEN ON THE LINE OR DEFAULTS TO TEST ALL THE DRIVES SEEN ON THE LINE. IT CHECKS THE EXISTENCE OF AN L OR P CLOCKS FOR USE IN THE TIMING TESTS.

B. STATIC & CYCLE UP TESTS

THESE TESTS CHECK OUT THE ABILITY TO SELECT & DESELECT THE DRIVE: TO DETECT PARITY, UNSAFE, AND FAULT CONDITIONS WITH THE DRIVE READY TO OPERATE BUT WITHOUT THE SPINDLE ON.

THE ENTIRE POWER UP SEQUENCE IS TESTED BY VERIFYING ALL STATUS BITS SET/RESET IN PROPER SEQUENCE: THE BRUSH CYCLE, INNER-OUTER LIMIT DETECTION, FORWARD, REVERSE, PIP...ETC STATUS BITS ARE CHECKED.

C. SEEK, WRITE HEADER, READ HEADER TESTS

THESE TESTS CHECK THE ABILITY OF THE DRIVE TO DO SEEKS, HEADER OPERATIONS & 20, 22 SECTOR FORMATTING.

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 MSG. 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 DRIVE FROM 0 THRU 7 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 IN-

763
764
765
766
767
768
769
770
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

DICATING THE OTHER PORT IS ACCESSED.
IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.

TEST 3 VERIFY OPERATOR DRIVE SELECTIONS

THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
DEFAULTED. EVERY DRIVE FROM 0 TO 7 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
NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF
NED ONLY, IT IS CHECKED AGAIN THE INPUTTED INFOR TO
VERIFY IT WAS NOT SPECIFIED.
IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.

TEST 4 FIND NEXT DRIVE TO BE TESTED

THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
ADDRESS IN 'DRVAD' & STMP4 IS SET TO CDT IF DRIVE IS RK07.
THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
THE DRIVE WHOSE ADDRESS IS IN 'DRVAD'.

TEST 5 UNLOAD DRIVE TO BE TESTED

THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE.
IT THEN WAITS FOR SPEED OK TO GO LOW BEFORE
PROCEEDING TO THE NEXT TEST.

STATIC & CYCLE UP TESTS

TEST 6 REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3

CHECKS THE ABILITY TO REFERENCE ALL
DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.

TEST 7 PRINT DRIVE SERIAL NUMBER

THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
IN DECIMAL & IS PERFORMED ON THE 1ST PASS ONLY

TEST 10 SET VV WITH PACK CMD

IF VV IS RESET, THE PACK CMD IS USED TO SET IT.

TEST 11 RELEASE DRIVE

819
820
821
822
823
824
825
826
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

TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK

TEST 12 DRIVE TYPE TEST

THIS TEST COMPARES DRIVE TYPE IN MSG A AGAINST 'DDT' IN RKDS.
WRONG CDT IN RKCS1 IS SENT & ERRORS ARE VERIFIED.

TEST 13 C-D PARITY ERROR DETECTION

TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
THE CONTROLLER BY SETTING 'PAT' ON RKMRI.
THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
AN ERROR AND SET SPAR.
THE ERROR CONDITION IS RESET WITH THE CLEAR CMD

TEST 14 VERIFY START SPINDLE CMD

THE PROGRAM CHECKS THE ENTIRE STARTUP SEQUENCE, IE:
BRUSH CYCLE, HEADS HOME, FWD, REV ETC.
BY VERIFYING ALL APPROPRIATE STATUS BITS FOR PROPER SEQUENCING.
THE CYL ADDRESS & CYL DIFFERENCE REGS ARE CHECKED
TO BE ZERO AT THE END OF THE SEQUENCE.

SEEK/READ HEADER/WRITE HEADER TESTS

TEST 15 STATIC CYL DIFF AND CYL ADDR REG TEST; PART 1

THIS TEST CHECKS EACH BIT OF THE CYL DIFFERENCE
AND CYL ADDRESS REGISTERS BY PERFORMING SEEKS TO ALL
MAJOR CYLS (0,1,2,4,8,16,32,64,128,256) (512 FOR THE RK07)
WITH EVEN PARITY SET, THIS FREEZES THE INFORMATION IN THE ABOVE REGISTER
AND ALLOWS FOR CHECKING. THIS TEST VERIFIES C-D PARITY ERROR BIT SET,
THAT HEADS DID NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.

TEST 16 STATIC CYL DIFF & CYL ADDR REG TEST-PART 2

THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
ALL CYLS FROM EVERY OTHER CYL. (N SQUARE SEEKS).
IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.

TEST 17 HEAD REGISTER TEST

THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)

875
876
877
878
879
880
881
882
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

VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.

SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK CMDS,
SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
ILLEGAL ADDRESS. IF NOT, THIS MEANS THAT CHANGING HEAD ADDRESSES
ARE NOT TIED TO SEEK CMDS

TEST 20 SEEK TO CYL 0

TESTS THE ABILITY TO DO A SEEK CMD.
VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
STATUS BITS. VERIFIES CMD COMPLETION BETWEEN 10-15USEC.
READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.

TEST 21 TEST SECTOR COUNT REG. FOR 22 & 20 SECTOR FORMAT

TEST 22 DETECT OUTER LIMIT

THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYL 0
BY DETECTING OUTER LIMIT AS THE ADJACENT CYL.
AN ERROR IN THIS TEST INDICATES:

AND/OR A. HEADS WERE NOT ON CYL 0
B. COULD NOT SEEK IN REVERSE DIRECTION.

TEST 23 BASIC WRITE/READ HEADER & HEAD SWITCHING TEST

THIS TEST CHECKS HEAD SWITCHING BY WRITING UNIQUE HEADERS
ON EACH TRACK OF CYL 0, READING BACK & VERIFYING THEY REMAINED
UNIQUE. 22 SECTOR FORMAT IS USED

I.E. TRACK 0: ALL 0'S FOR ALL SECTOR HEADERS
TRACK 1: 0101 FOR ALL SECTOR HEADERS
TRACK 2: ALL 1'S FOR ALL SECTOR HEADERS

TEST 24 BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS

USING HEAD 0, WRITE & READ 20 SECTOR HEADERS BY WRITING ALL
1'S AS HEADERS. ATTEMPT TO FIND SECTORS 20 & 21. VERIFY
THEY ARE NO LONGER THERE BY READING 22 SECTORS AND NOT
FINDING 0'S AS DATA FROM THE PREVIOUS TEST.

TEST 25 WRITE & READ HEADERS CYL 0, HEAD 0

TEST 26 SEEK FROM CYL 0 TO 1 & READ HEADERS

THIS TEST CHECKS MSG A & B WORDS 0, 1, 2 FOR CORRECT STATUS AFTER RDY IS RECEIVED FROM A SEEK CMD TO DETERMINE THAT THE HEADS ARE ACTUALLY MOVING & THE CYL DIFF IS 1. AFTER ATTN IS RECEIVED, CERR IS EXAMINED FOR ANY ERRORS. CYL DIFFERENCE IN MSG A2 IS VERIFIED TO BE 0 & CYL ADDR IN MSG B2 IS VERIFIED TO BE 1.

HEADERS ARE READ FROM 1 SECTOR, HEAD 0 & VERIFIED THAT THEY ARE DIFFERENT FROM CYL 0 TO SHOW THAT THE HEADS DID ACTUALLY MOVE.

TEST 27 WRITE & READ HEADERS CYL 1, HEAD 0

TEST 30 TEST RECALIBRATE CMD & READ HEADERS

THIS TEST DOES A RECALIBRATE & READS HEADERS. IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS TEST DID NOT OVERWRITE CYL 0 HEADERS.

AN ERROR IN THIS TEST INDICATES THAT HEADS:

OR A. MOVED TO A CYL OTHER THAN 1
 B. DID NOT GET BACK TO CYL 0

TEST 31 SINGLE INCREMENT SEEKS TO THE LAST CYLINDER

THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO THE LAST CYL WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY DESTROY DATA.

TEST 32 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. IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED, A MSG 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.

THIS IS THE FIRST TEST WHERE THE READ DATA CMD IS PERFORMED

TEST 33 DETECT INNER LIMIT

THIS TEST VERIFIES THAT THE LAST CYL IN THE ABOVE TEST WAS 632 (1456) BY DETECTING INNER LIMIT AS THE ADJACENT CYL. IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON THE LAST CYL

931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
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

& THAT BSE INFO IS NOT VALID. THE FORMAT PACK TEST
& ALL READ-WRITE TESTS ARE BYPASSED
TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.

TEST 34 FORMAT PACK

THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
DOING 1 CYL INCREMENTAL SEEKS
FROM 0 TO 632 (1456) WITH WRITE HEADER CMDS (ALL TRACKS).
HEADERS WILL BE READ IN THE NEXT TEST

TEST 35 DECREMENT FROM LAST CYL TO 0 & READ HEADERS

THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
SINGLE CYL INCREMENTAL SEEKS.

TEST 36 SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS

THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLS & READS HEADERS.
IT THEN SEEKS CYL 0 & READS HEADERS.

MAJOR CYLS ARE: 1 (DECIMAL) = 1 (OCTAL)

2	2
4	4
8	10
16	20
32	40
64	100
128	200
256	400
512	1000 (RK07)

TEST 37 SEEK TO ALL CYLS FROM 0 & READ HEADERS

TEST 40 SEEK TO ALL CYLS FROM LAST CYL & READ HEADERS

TEST 41 SEEK TO ALL INVALID CYLINDERS

THIS TEST VERIFIES THAT 'INV ADDR' & 'SEEK INCOMPLETE' IS
PRODUCED & THAT HEADS DO NOT MOVE OR UNLOAD IF AN ILLEGAL
CYL IS SPECIFIED IN A SEEK.

INVALID CYLS ARE 633 THRU 777 (8) FOR THE RK06
& 1457 THRU 1777 FOR THE RK07

THE PROGRAM DOES NOT REQUIRE FORMATTED PACKS AS FORMATTING
IS PERFORMED IN ANY CASE.

987
988
989
990
991
992
993
994
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

H02

CZR6HED UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 20

SEQ 0020

ANY TEST THAT MODIFIES STANDARD FORMATTING IS FOLLOWED BY A
'CLEAN UP' TEST TO PUT THOSE CYLS BACK TO STANDARD
FORMAT.

1043
1044
1045
1046
1047

1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
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

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION

WHENEVER AN ERROR MSG 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 MSG CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MSG, 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 MSGS ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLES:

EXAMPLE #1:

MSG AD ERROR
AFTER START SPINDLE CMD & FWD SET

TEST NO.	PC	EXPECT					
AD	BO	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 MSG REGISTERS AD, BO, A1 & B1.

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

1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120

EXAMPLE #2:

NO ATTN IN RKASOF
AFTER UNLOAD CMD

TEST NO.		PC					
000003	014330						
RKMR2	RKMR3	RKER	RKDS	RKCS1	RKCS2	RKASOF	
000144	100000	000000	100101	000206	000104	000000	

[END OF DOCUMENT]

%

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
1165
1166
1167
1168
1169
1170
1171
1172
1173167400
000001; *** PGM REV 039 ***
; NLIST CND,MC,MD
; LIST ME
; ENABL ABS,AMA

; DEFINE SYSMAC MACROS

\$SWR= 167400
\$TN= 1; DEFINE SWITCHES 15,14,13,11,10,9,8
; SET FIRST TEST NO. TO 1

.TITLE CZR6HEO UNIBUS RK6 DR PRT1

; *COPYRIGHT (C) 1976,1977

; *DIGITAL EQUIPMENT CORP.

; *MAYNARD, MASS. 01754

; *

; *PROGRAM BY GARY PAPAIZIAN

; *

; *THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC

; *PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.

; *

.SBTTL OPERATIONAL SWITCH SETTINGS

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

SWITCH

USE

15
14
13
12
11
10
9
8-----
HALT ON ERROR
LOOP ON TEST
INHIBIT ERROR TYPEOUTS
ABORT DRIVE AFTER 20 ERRORS
INHIBIT ITERATIONS
BELL ON ERROR
LOOP ON ERROR
LOOP ON TEST IN SWR<7:0>

.SBTTL SUMMARY OF STARTING LOCATIONS

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

; *

200
204
220
230
240
260
270DEFAULT PARAMETERS
DEFAULT PARAMETERS & BYPASS TEST 16
INPUT PARAMETERS
INPUT PARAMETERS & BYPASS TEST 16
ODT11
RUN MODULE TEST VERSION-DEFAULT MODE ONLY BYPASS
TESTS 35,36,40 & 41
SAME AS 260 START BUT BYPASS TEST 16 ALSO

.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

STACK= 1100

.EQUIV EMT,ERROR ;;;BASIC DEFINITION OF ERROR CALL

.EQUIV IOT,SCOPE ;;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS

HT= 11 ;;;CODE FOR HORIZONTAL TAB

LF= 12 ;;;CODE FOR LINE FEED

CR= 15 ;;;CODE FOR CARRIAGE RETURN

CRLF= 00 ;;;CODE FOR CARRIAGE RETURN-LINE FEED

PS= 177776 ;;;PROCESSOR STATUS WORD

.EQUIV PS,PSW

STKLMT= 177774 ;;;STACK LIMIT REGISTER

PIRQ= 177772 ;;;PROGRAM INTERRUPT REQUEST REGISTER

DSWR= 177570 ;;;HARDWARE SWITCH REGISTER

DDISP= 177570 ;;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS

R0= %0 ;;;GENERAL REGISTER

R1= %1 ;;;GENERAL REGISTER

R2= %2 ;;;GENERAL REGISTER

R3= %3 ;;;GENERAL REGISTER

R4= %4 ;;;GENERAL REGISTER

R5= %5 ;;;GENERAL REGISTER

R6= %6 ;;;GENERAL REGISTER

R7= %7 ;;;GENERAL REGISTER

SP= %6 ;;;STACK POINTER

PC= %7 ;;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS

PR0= 0 ;;;PRIORITY LEVEL 0

PR1= 40 ;;;PRIORITY LEVEL 1

PR2= 100 ;;;PRIORITY LEVEL 2

PR3= 140 ;;;PRIORITY LEVEL 3

PR4= 200 ;;;PRIORITY LEVEL 4

PR5= 240 ;;;PRIORITY LEVEL 5

PR6= 300 ;;;PRIORITY LEVEL 6

PR7= 340 ;;;PRIORITY LEVEL 7

;*SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000

SW14= 40000

SW13= 20000

SW12= 10000

SW11= 4000

SW10= 2000

SW09= 1000

SW08= 400

SW07= 200

SW06= 100

SW05= 40

SW04= 20

SW03= 10

SW02= 4

1174
1175
1176
1177 001100
1178
1179
1180
1181
1182 000011
1183 000012
1184 000015
1185 000200
1186 177776
1187
1188 177774
1189 177772
1190 177570
1191 177570
1192
1193
1194 000000
1195 000001
1196 000002
1197 000003
1198 000004
1199 000005
1200 000006
1201 000007
1202 000006
1203 000007
1204
1205
1206 000000
1207 000040
1208 000100
1209 000140
1210 000200
1211 000240
1212 000300
1213 000340
1214
1215
1216 100000
1217 040000
1218 020000
1219 010000
1220 004000
1221 002000
1222 001000
1223 000400
1224 000200
1225 000100
1226 000040
1227 000020
1228 000010
1229 000004

1230 000002
1231 000001
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244 100000
1245 040000
1246 020000
1247 010000
1248 004000
1249 002000
1250 001000
1251 000400
1252 000200
1253 000100
1254 000040
1255 000020
1256 000010
1257 000004
1258 000002
1259 000001
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272 000004
1273 000010
1274 000014
1275 000014
1276 000014
1277 000020
1278 000024
1279 000030
1280 000034
1281 000060
1282 000064
1283 000240
1284
1285

SW01= 2
SW00= 1
.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

.*BASIC "CPU" TRAP VECTOR ADDRESSES

ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS
RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC= 14 ;: "T" BIT
TRTVEC= 14 ;: TRACE TRAP
BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)
IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
PWRVEC= 24 ;: POWER FAIL
EMTVEC= 30 ;: EMULATOR TRAP (EMT) **ERROR**
TRAPVEC= 34 ;: "TRAP" TRAP
TKVEC= 60 ;: TTY KEYBOARD VECTOR
TPVEC= 64 ;: TTY PRINTER VECTOR
PIRQVEC= 240 ;: PROGRAM INTERRUPT REQUEST VECTOR

.SBTTL RK06 CONTROLLER REGISTER DEFINITION

```

1286
1287
1288
1289 000000
1290 000002
1291 000004
1292 000006
1293 000010
1294 000012
1295 000014
1296 000016
1297 000020
1298 000024
1299 000026
1300 000034
1301 000036
1302 000030
1303 000032
1304
1305 .SBTTL CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1306
1307 ;
1308 ; $BASE=177440
1309
1310 SELDRV= 1 ; SELECT DRIVE (GET STATUS)
1311 PACK= 3 ; PACK ACKNOWLEDGE
1312 CLEAR= 5 ; DRIVE CLEAR
1313 UNLOAD= 7 ; UNLOAD
1314 SRTSPL= 11 ; START SPINDLE
1315 RECAL= 13 ; RECALIBRATE
1316 OFFSET= 15 ; OFFSET
1317 SEEK= 17 ; SEEK
1318 RDDATA= 21 ; READ DATA
1319 WRDATA= 23 ; WRITE DATA
1320 RDHEAD= 25 ; READ HEADER
1321 WRHEAD= 27 ; WRITE HEADER AND DATA
1322 WRTCHK= 31 ; WRITE CHECK
1323
1324 GO= BIT0 ; GO BIT
1325 IE= BIT6 ; INTERRUPT ENABLE
1326 RDY= BIT7 ; CONTROLLER READY
1327 BA16= BIT8 ; BUS ADDRESS BIT 16
1328 BA17= BIT9 ; BUS ADDRESS BIT 17
1329 CDT= BIT10 ; CONTROLLER DRIVE TYPE (0=RK06, 1=RK07)
1330 CTO= BIT11 ; CONTROLLER TIMEOUT
1331 CFMT= BIT12 ; CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1332 DCPAR= BIT13 ; SERCON PARITY ERROR DETECTED BY CONTROLLER
1333 DI= BIT14 ; DRIVE INTERRUPT
1334 CERR= BIT15 ; CONTROLLER ERROR
1335 CCLR= BIT15 ; CONTROLLER CLEAR
1336
1337 .SBTTL CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1338
1339 DRVMSK= 7 ; MASK FOR DRIVE SELECTION CODE
1340 RLS= BIT3 ; DESELECT OR RELEASE DRIVE IN BITS 0-2
1341 BAI= BIT4 ; BUS ADDRESS INCREMENT INHIBIT
SCLR= BITS ; SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES

```

; \$BASE=177440

RKCS1= 0 ; CONTROL AND STATUS REGISTER 1
 RKWC= 2 ; WORD COUNT REGISTER
 RKBA= 4 ; BUS ADDRESS REGISTER
 RKDA= 6 ; DESIRED TRACK SECTOR REGISTER
 RKCS2= 10 ; CONTROL AND STATUS REGISTER 2
 RKDS= 12 ; DRIVE STATUS REGISTER
 RKER= 14 ; ERROR REGISTER
 RKASOF= 16 ; ATTENTION SUMMARY AND OFFSET REGISTER
 RKDC= 20 ; DESIRED CYL REGISTER
 RKDB= 24 ; DATA BUFFER
 RKMR1= 26 ; MAINTENANCE REGISTER 1
 RKMR2= 34 ; MAINTENANCE REGISTER 2 (MSG LINE A)
 RKMR3= 36 ; MAINTENANCE REGISTER 3 (MSG LINE B)
 RKECPS= 30 ; ECC POSITION INFORMATION
 RKECPT= 32 ; ECC PATTERN INFORMATION

.SBTTL CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)

; DRIVE CMDS

SELDRV= 1 ; SELECT DRIVE (GET STATUS)
 PACK= 3 ; PACK ACKNOWLEDGE
 CLEAR= 5 ; DRIVE CLEAR
 UNLOAD= 7 ; UNLOAD
 SRTSPL= 11 ; START SPINDLE
 RECAL= 13 ; RECALIBRATE
 OFFSET= 15 ; OFFSET
 SEEK= 17 ; SEEK
 RDDATA= 21 ; READ DATA
 WRDATA= 23 ; WRITE DATA
 RDHEAD= 25 ; READ HEADER
 WRHEAD= 27 ; WRITE HEADER AND DATA
 WRTCHK= 31 ; WRITE CHECK

GO= BIT0 ; GO BIT
 IE= BIT6 ; INTERRUPT ENABLE
 RDY= BIT7 ; CONTROLLER READY
 BA16= BIT8 ; BUS ADDRESS BIT 16
 BA17= BIT9 ; BUS ADDRESS BIT 17
 CDT= BIT10 ; CONTROLLER DRIVE TYPE (0=RK06, 1=RK07)
 CTO= BIT11 ; CONTROLLER TIMEOUT
 CFMT= BIT12 ; CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
 DCPAR= BIT13 ; SERCON PARITY ERROR DETECTED BY CONTROLLER
 DI= BIT14 ; DRIVE INTERRUPT
 CERR= BIT15 ; CONTROLLER ERROR
 CCLR= BIT15 ; CONTROLLER CLEAR

.SBTTL CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)

DRVMSK= 7 ; MASK FOR DRIVE SELECTION CODE
 RLS= BIT3 ; DESELECT OR RELEASE DRIVE IN BITS 0-2
 BAI= BIT4 ; BUS ADDRESS INCREMENT INHIBIT
 SCLR= BITS ; SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES

1342	000100	IR=	BIT6	: INPUT READY
1343	000200	OR=	BIT7	: OUTPUT READY
1344	000400	UFE=	BIT8	: UNIT FIELD ERROR
1345	001000	MDS=	BIT9	: MULTIPLE DRIVE SELECT
1346	002000	PGE=	BIT10	: PROGRAMMING ERROR
1347	004000	NEM=	BIT11	: NON-EXISTENT MEMORY
1348	010000	NED=	BIT12	: NON-EXISTENT DRIVE
1349	020000	UPE=	BIT13	: UNIBUS PARITY ERROR
1350	040000	WCE=	BIT14	: WRITE CHECK ERROR
1351	100000	DLT=	BIT15	: DATA LATE ERROR
1352				
1353		.SBTTL	ERROR REGISTER BIT DEFINITION (RKER:14)	
1354				
1355	000001	ILF=	BIT0	: ILLEGAL FUNCTION CODE
1356	000002	SKI=	BIT1	: SEEK INCOMPLETE
1357	000004	NXF=	BIT2	: NON-EXECUTABLE FUNCTION
1358	000010	DRPAR=	BIT3	: DRIVE DETECTED SERCON PARITY ERROR
1359	000020	FMTE=	BIT4	: FORMAT ERROR
1360	000040	DYE=	BIT5	: DRIVE TYPE ERROR
1361	000100	ECH=	BIT6	: ECC HARD
1362	000200	BSE=	BIT7	: BAD SECTOR ERROR
1363	000400	HVRC=	BIT8	: HEADER VRC ERROR
1364	001000	COE=	BIT9	: CYL ADDRESS OVERFLOW ERROR
1365	002000	IDAE=	BIT10	: INVALID DISK ADDRESS ERROR: HEAD/CYL
1366	004000	WLE=	BIT11	: WRITE LOCK ERROR
1367	010000	DTE=	BIT12	: DRIVE TIMING ERROR
1368	020000	OPI=	BIT13	: OPERATION (SEARCH) INCOMPLETE
1369	040000	UNS=	BIT14	: DRIVE UNSAFE
1370	100000	DCK=	BIT15	: DATA CHECK
1371				
1372		.SBTTL	STATUS REGISTER BIT DEFINITION (RKDS:12)	
1373				
1374	000001	DRA=	BIT0	: DRIVE AVAILABLE (CONTROLLER IS SET IF THIS BIT IS RESET)
1375				
1376	000004	OFST=	BIT2	: DRIVE OFFSET
1377	000010	ACLO=	BIT3	: AC LOW
1378	000020	DCLO=	BIT4	: DC LOW
1379	000040	DROT=	BIT5	: DRIVE OFF TRACK
1380	000100	VV=	BIT6	: VOLUME VALID
1381	000200	DRDY=	BIT7	: DRIVE READY
1382	000400	DDT=	BIT8	: DRIVE TYPE (0=RK06, 1=RK07)
1383	004000	WRL=	BIT11	: WRITE LOCK
1384	020000	PIP=	BIT13	: POSITIONING IN PROGRESS
1385	040000	DSC=	BIT14	: DRIVE STATUS CHANGE
1386	100000	SVAL=	BIT15	: STATUS VALID
1387				
1388		.SBTTL	MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)	
1389				
1390	000017	MESMSK=	17	: MSG MASK
1391	000020	PAT=	BIT4	: FORCE EVEN PARITY ON SERCON MSG LINES
1392	000040	DMD=	BIT5	: DIAGNOSTIC MODE
1393	000100	MSP=	BIT6	: MAINTENANCE SECTOR PULSE
1394	000200	MIND=	BIT7	: MAINTENANCE INDEX
1395	000400	MCLK=	BIT8	: MAINTENANCE CLOCK
1396	001000	MERD=	BIT9	: MAINTENANCE ENCODED READ DATA
1397	002000	MEWD=	BIT10	: MAINTENANCE ENCODED WRITE DATA

```

1398      004000      PCA=      BIT11      ;PRECOMPENSATION ADVANCE
1399      010000      PCD=      BIT12      ;PRECOMPENSATION DELAY
1400      020000      ECCW=     BIT13      ;ECC WORD IS BEING READ OR WRITTEN
1401      040000      WRTGAT=  BIT14      ;WRITE GATE
1402      100000      RDGATE=  BIT15      ;READ GATE
1403
1404      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MSG A (RKMR2:34)
1405
1406      000040      D.DRA=   BITS      ;DRIVE AVAILABLE
1407      000100      D.VV=    BIT6      ;VOLUME VALID
1408      000200      D.DRDY=  BIT7      ;DRIVE READY
1409      000400      D.DDT=   BIT8      ;DRIVE TYPE (0=RK06,1 = RK07)
1410      001000      D.FORM=  BIT9      ;DRIVE FORMAT
1411      002000      D.OFF=   BIT10     ;OFFSET ON
1412      004000      D.WRL=   BIT11     ;WRITE LOCK
1413      010000      D.SPIN=  BIT12     ;SPINDLE ON
1414      020000      D.PIP=   BIT13     ;POSITIONING IN PROGRESS
1415      040000      D.DSC=   BIT14     ;DRIVE STATUS CHANGE
1416
1417      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MSG A (RKMR2:34)
1418
1419      000020      D.SSP=   PIT4      ;SERVO SIG PRESENT
1420      000040      D.HDHM=  /BITS     ;HEADS HOME
1421      000100      D.BRHM=  BIT6      ;BRUSHES HOME
1422      000200      D.DOOR=  BIT7      ;DOOR INTERLOCKED
1423      000400      D.CART=  BIT8      ;CARTRAGE INTERLOCK
1424      001000      D.SPOK=  BIT9      ;SPEED OK
1425      002000      D.FWD=   BIT10     ;FORWARD
1426      004000      D.REV=   BIT11     ;REVERSE
1427      010000      D.LOAD=  BIT12     ;HEADS LOADING
1428      020000      D.RTZ=   BIT13     ;RETURN TO ZERO
1429      040000      D.UNLD=  BIT14     ;HEADS UNLOADING
1430
1431      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MSG B (RKMR3:36)
1432
1433      000040      D.IDAE=  BITS      ;INVALID DISK ADDRESS ERROR:HEAD/CYL
1434      000100      D.ACLO=  BIT6      ;AC LOW
1435      000200      D.FLT=   BIT7      ;DRIVE FAULT
1436      000400      D.ILF=   BIT8      ;ILLEGAL FUNCTION CODE
1437      001000      D.PAR=   BIT9      ;DRIVE DETECTED SERCON PARITY ERROR
1438      002000      D.SKI=   BIT10     ;SEEK INCOMPLETE
1439      004000      D.WLE=   BIT11     ;WRITE LOCK ERROR
1440      010000      D.SPLS=  BIT12     ;SPEED LOSS
1441      020000      D.DROT=  BIT13     ;DRIVE OFF TRACK
1442      040000      D.UNS=   BIT14     ;R/W UNSAFE
1443
1444      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MSG B (RKMR3:36)
1445
1446      000020      D.SECT=  BIT4      ;SECTOR ERROR
1447      000040      D.WCUR=  BITS      ;WRITE CURRENT AND NO WRITE GATE
1448      000100      D.WGAT=  BIT6      ;WRITE GATE AND NO TRANSISTIONS
1449      000200      D.HDFL=  BIT7      ;HEAD FAULT
1450      000400      D.MHD=   BIT8      ;MULTIPLE HEAD SELECT
1451      001000      D.XERROR=          BIT9      ;INDEX ERROR
1452      002000      D.TIB=   BIT10     ;TRIBIT ERROR
1453      004000      D.PLO=   BIT11     ;PLO ERROR

```

1454	010000	D.NMOV= BIT12	;SEEK AND NO MOTION
1455	020000	D.LIMD= BIT13	;LIMIT DETECT ON SEEK
1456	040000	D.SUNS= BIT14	;SERVO UNSAFE
1457			
1458		.SBTTL COMMON MASKS AND OTHER BITS: MSG A (RKMR2:34)	
1459			
1460	000007	M.DRV= 7	;DRIVE CODE, ALL BYTES
1461	077770	M.SER= 77770	;DRIVE SERIAL #, BYTE 11
1462			
1463		.SBTTL COMMON MASKS AND OTHER BITS: MSG B (RKMR3:36)	
1464			
1465	000003	M.ID= 3	;BYTE ID, ALL BYTES
1466	040000	M.ALGN= BIT14	;ALIGN SIGN, BYTE 10
1467	000760	M.SECT= 760	;SECTOR COUNT, BYTE 11
1468	007000	M.HEAD= 7000	;HEAD DECODE, BYTE 11
1469	100000	M.PAR= BIT15	;PARITY, MESS A/B, ALL BYTES

```

1470
1471
1472
1473      000000
1474
1475
1476
1477      000174
1478      000174 000000
1479      000176 000000
1480
1481      000200 000137 007120
1482      000204 000137 007014
1483      000220 000137 006774
1484      000220 000137 006774
1485      000230 000137 007034
1486      000230 000137 007034
1487      000240 000137 067744
1488      000240 000137 067744
1489      000260 000137 007056
1490      000260 000137 007056
1491      000270 000137 007076
1492      000270 000137 007076
1493
1494
1495
1496
1497
1498
1499      000274
1500      000046 043010
1501      000052 000052
1502      000052 100000
1503      000052 000274
1504      000052 001000
1505
1506
1507
1508
1509
1510
1511      001000
1512      000024 000200
1513      000024 000044
1514      000044 001000
1515      000044 001000
1516
1517
1518
1519
1520
1521      001000
1522      001000 000000
1523      001002 001210
1524      001004 000430
1525      001006 001130

.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
      .=204
      JMP  BYT16    ; BYPASS N-SQUARE TEST IN DEFAULT MODE
      .=220
      JMP  PARSRT   ; INPUT ALL PARAMETERS & START TESTING
      .=230
      JMP  BYT16A   ; BYPASS N-SQUARE TEST IN PARAM MODE
      .=240
      JMP  0.ODT    ; ENTER ODT11
      .=260
      JMP  MDTST    ; MODULE TESTS DEFAULT MODE ONLY
      .=270
      JMP  MDTSTA   ; BYPASS SEVERAL TESTS
      ; SAME AS 260 & BYPASS N-SQUARE TEST ALSO

.SBTTL ACT11 HOOKS
      ;*****
      ; HOOKS REQUIRED BY ACT11
      $SVPC=.      ; SAVE PC
      .=46
      $ENDAD      ;; 1) SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
      .=52
      .WORD 100000 ;; 2) SET LOC.52 TO 100000
      .=$SVPC     ;; RESTORE PC
      .=1000

.SBTTL APT PARAMETER BLOCK
      ;*****
      ; SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
      ;*****
      .SX=.      ; 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.
      $MADR:  .WORD $MAIL  ;; ADDRESS OF APT MAILBOX (BITS 0-15)
      $TSTM:  .WORD 280.   ;; RUN TIM OF LONGEST TEST
      $PASTM: .WORD 600.   ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)

```



```

1526 001010 001130 $UNITM: .WORD 600. ;:ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1527 001012 000042 .WORD $ETEND-$MAIL/2 ;:LENGTH MAILBOX-ETABLE(WORDS)
1528
1529
1530 .LIST MD
1531 ;:USE LOOP X TO OMIT JSR PC, SUBCLR
1532 ;:
1533
1534
1535 .MACRO LOOP A
1536 SCOP1
1537 MOV #STACK,SP ;RESTORE STK PTR
1538
1539 .IF B A
1540 JSR PC,SUBCLR
1541 ERROR 24 ;CERR AFTER SCLR
1542
1543 .ENDC
1544 .ENDM LOOP
1545
1546 ;:
1547 ;: THIS MACRO FILLS EXPECTED MSG AO,B0,A1,B1,A2,B2 & B3 WITH STANDARD BITS SET
1548 ;: A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
1549 ;: NOTE: A CAN BE ANY BIT COMBINATION DESIRED
1550
1551 .MACRO F.EAB A
1552
1553 MOV #<A!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG AO
1554 CLR E.B0 ;EXPECTED MSG B0
1555 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
1556 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
1557 CLR E.A2 ;EXPECTED MSG A2
1558 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
1559 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
1560 .ENDM F.EAB
1561
1562 ;:
1563 ;: THIS MACRO ASSUMES DRIVE MSG AO,B0,A1,B1 WILL ALWAYS BE TESTED
1564 ;: USE A,C,D,E FOR MSG AO,B0,A1,B1 ERROR NUMBERS RESP.
1565 ;: USE G=T.A2 TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
1566 ;: H=T.B2 TO READ MSG B2 & PUT INFOR INTO 'CYLADD'
1567 ;: I=T.B3 TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'
1568
1569 ;:
1570 ;: F= < ERROR DESCRIPTION>
1571
1572 .MACRO CHECK A,C,D,E,F,G,H,I
1573 JSR PC,CHKMSG ;CHECK MSGS AO,B0,A1,B1
1574 .WORD G!H!I ;& MSGS SPECIFIED HERE
1575 ERROR A ;MSG AO ERROR F
1576 ERROR C ;MSG B0 ERROR
1577 ERROR D ;MSG A1 ERROR
1578 ERROR E ;MSG B1 ERROR
1579 .ENDM CHECK
1580 ;:
1581 ;: A=CYL DIFF/OFFSET ERROR #
;: B=CYL ADDR ERROR #

```

```

1582 ; C= <ERROR DESCRIPTION>
1583 ;
1584 ;.MACRO CWD2 A,B,C,?D,?E
1585 ;
1586 ; MOV #2,RKMR1(R5) ;SELECT WORD 2
1587 ; JSR PC,GSTAT
1588 ; TST CYLDIF ;SEE IF MSG A2=0
1589 ; BEQ D ;BR IF YES
1590 ; ERROR A ;MSG A2 NOT CLEARED C
1591 D: ; TST CYLADD ;SEE IF MSG B2=0
1592 ; BEQ E ;BR IF YES
1593 ; ERROR B ;MSG B2 NOT CLEARED C
1594 ;
1595 ;.ENDM CWD2
1596 ;
1597 ;
1598 ;.MACRO LPCHK ?A
1599 ; CLR $ESCAPE
1600 ; TST LPFLG
1601 ; BEQ A
1602 ; JMP @SLPERR ;SW 9 WAS SET.
1603 A: ; JMP @SLPADR ;SW 14 OR 8 WAS SET
1604 ;.ENDM LPCHK
1605 ;
1606 ;.MACRO SWB14
1607 ; JSR PC,SWTST ;SEE IF SW 14 OR 8 IS SET
1608 ; SKIP R,<GO TO NEXT TEST> ;RETURN HERE IF NEITHER IS SET
1609 ; ;RETURN HERE IF SW 14 IS SET OR
1610 ; ;SW 8 WITH SWR <7:0> APPLY
1611 ;.ENDM SWB14
1612 ;
1613 ;
1614 ;:SWR9 (LOOP ON ERROR) TEST A=BRANCH POINT TO RECONDITION DRIVE
1615 ; B=JMP POINT TO RE-ENTER MAIN LINE
1616 ;.MACRO TSTSW9 A,B
1617 ; INC LPFLG
1618 ; BIT #SW9,@SWR ;LOOP ON ERROR?
1619 ; BNE A ;YES, RECONDITION DRIVE
1620 ; JMP B ;RETURN TO MAINLINE
1621 ;.ENDM TSTSW9
1622 ;
1623 ;
1624 ;
1625 ; USE DRCLR X TO OMIT CHECKING MSG A0,B0,A1 & B1
1626 ;.MACRO DRCLR A,?C
1627 ;
1628 ; MOV #CCLR,RKCS1(R5)
1629 ; MOV $UNIT,RKCS2(R5) ;DRIVE#
1630 ; MOV #CLEAR,HCSI
1631 ; JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
1632 ; ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
1633 ; JSR PC,TSTATN ;TEST FOR ATTN
1634 ; BR C
1635 ; ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1636 ;
1637 C:

```

```

1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693

```

```

      .IF B      A
                F.EAB      0
                CHECK      273,265,274,266,<AFTER DRIVE CLEAR CMD>,T.A2,T.B2,0
      .ENDC
      .ENDM      DRCLR

      ;A=BLANK TO CHECK A0 THRU B2
      ;A=NON BLANK TO OMIT CHECKING A0 THRU B2
      ;D=BLANK TO CHECK A0 THRU B2 IN DRCLR
      ;D=NON-BLANK TO OMIT CHECKING A0 THRU B2 IN DRCLR
      ;THE FOLLOWING MACRO DEFINITION IS CHANGED ON 24-OCT-77
      ;1. ADD NEW LABEL ?F
      ;2. DEL THE MACRO CALL CWD2
      ;3. ADD 5 LINES LONG HAND CODING FOR SUBSTITUTING ITEM 2
      ;4. SELECT MESSAGE 2

      .MACRO      CALIB      A,D,?C,?F

                MOV         #CCLR,RKCS1(R5)
                MOV         #UNIT,RKCS2(R5)
                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         C
                ERROR      244              ;RTZ NOT SET DURING RECAL CMD
      C:         MOV         T10,TEMP2      ;SETUP TIMEOUT
                JSR         PC,FATT1        ;FIND ATTN
                ERROR      55              ;NO ATTN AFTER RECAL CMD

      .IF B      A
                F.EAB      D.DSC
                CHECK      221,275,222,276,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
      ;         CWD2      47,50,<AFTER RECAL CMD>
                MOV         #2,RKMR1(R5)    ;SELECT THE MESSAGE
                JSR         PC,GSTAT        ;GET THE STATUS
                TST         CYLADD         ;RECAL SUCCESSFUL ?
                BEQ         F              ;BRANCH IF 50
                ERROR      50              ;REPORT THE ERROR
      F:
      .ENDC
      .ENDM      DRCLR      D

      .ENDM      CALIB

      ;IDAE IS CLEARED ONLY BY RECAL & DRIVE CLEAR
      .MACRO      CIDAE      ?A
                MOV         #CCLR,RKCS1(R5)

```

```

1694      MOV      $UNIT,RKCS2(R5)
1695      MOV      #RECAL,HCS1
1696      JSR      PC,DOCMD      ;DO RECAL CMD & GET CONTR RDY
1697      ERROR   124          ;RDY NOT FOUND AFTER RECAL CMD
1698      DRCLR   X
1699
1700      JSR      PC,GSTAT
1701      BIT      #D.IDAE,HMR3  ;SEE IF IDAE IS CLEARED
1702      BEQ     A           ;BR IF YES
1703      ERROR   155          ;IDAE NOT CLEARED AFTER RECAL CMD
1704
1705      A:      MOV      #CCLR,RKCS1(R5)
1706      MOV      T1,TEMP2     ;LOOK FOR ATTN FROM RECAL
1707      JSR      PC,FATT1
1708      ERROR   55           ;NO ATTN AFTER RECAL CMD
1709
1710      .ENDM   CIDAE
1711
1712      ; A=D.FWD/D.REV
1713      ;
1714      ;
1715      .MACRO  SKRDY A
1716
1717      MOV      #SEEK,HCS1
1718      JSR      PC,DOCMD     ;DO SEEK CMD & GET CONTR READY
1719      ERROR   131          ;NO RDY AFTER SEEK CMD
1720      MOV      #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED AO
1721      CLR     E.B0
1722      MOV      #<A!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
1723      MOV      #1,E.B1
1724      CHECK   203,204,205,206,<DURING SEEK CMD>,T.A2,T.B2,0
1725
1726      .ENDM   SKRDY
1727
1728      .MACRO  SKATN ?A,?B
1729
1730      JSR      PC,FATT2     ;FIND ATTN
1731      ERROR   132          ;NO ATTN AFTER SEEK CMD
1732      BIT      #CERR,HCS1
1733      BEQ     A
1734      ERROR   210          ;CERR AFTER SEEK CMD
1735      A:      F.EAB   D.DSC
1736      CHECK   133,134,135,136,<AFTER SEEK CMD>,T.A2,T.B2,0
1737      TST     CYLDIF
1738      BEQ     B
1739      ERROR   137          ;CYL DIFF NOT CLEARED AFTER SEEK CMD
1740
1741      B:      DRCLR
1742      .ENDM   SKATN
1743
1744      ; QUICK START SPINDLE.
1745      ;
1746      ;
1747      .MACRO  QKSRT  A
1748      JSR      PC,SUBCLR
1749

```

```

1750          ERROR      24          ;CERR AFTER SCLR
1751
1752          MOV        #SRTSPL,HCS1
1753          JSR        PC,DOCMD      ;DO START SPINDLE CMD & GET CONTR RDY
1754          ERROR      121         ;RDY NOT FOUND AFTER ST SPIN CMD.
1755
1756          MOV        T500,TEMP2    ;SETUP TIMEOUT
1757          JSR        PC,FATT1      ;FIND ATTN
1758          ERROR      67          ;NO ATTN AFTER ST SPIN CMD.
1759
1760          CLR        UNLD
1761          .IF B      A
1762          TSTSW9    10$,2$
1763          .ENDC
1764          .ENDM      QKSRT
1765
1766          ;QUICK SEEK.  ENTER WITH CYL # IN RKDC
1767          ;MACRO      QKSEEK ?A
1768
1769          MOV        #SEEK,HCS1
1770          JSR        PC,DOCMD      ;DO SEEK CMD & GET CONTR READY
1771          ERROR      131         ;NO RDY AFTER SEEK CMD.
1772
1773          MOV        T50000,TEMP1
1774          JSR        PC,FATT2      ;FIND ATTN
1775          ERROR      132         ;NO ATTN AFTER SEEK CMD
1776          BIT        #CERR,HCS1
1777          BEQ        A
1778          ERROR      210         ;CERR AFTER SEEK CMD.
1779
1780          A:         JSR        PC,SUBCLR
1781          ERROR      24          ;CERR AFTER SCLR
1782
1783          .ENDM      QKSEEK
1784
1785          ;QUICK REPETITIVE SEEKS
1786          ;A=INC/DEC  CYL#
1787          ;B=FINAL VALUE OF CYL# BEFORE EXITING
1788          ;MACRO      QKRPSK A,B,?C,?D
1789
1790          JSR        PC,SUBCLR
1791          ERROR      24          ;CERR AFTER SCRL
1792
1793          C:         MOV        TOCYL,RKDC(R5) ;CYL#
1794          QKSEEK
1795          CMP        TOCYL,B ;LAST CYL DONE?
1796          BEQ        D ;BR IF YES
1797          A         TOCYL ;ELSE DO ANOTHER
1798          BR        C
1799
1800          D:         JSR        PC,SUBCLR
1801          ERROR      24          ;CERR AFTER SCLR
1802
1803
1804
1805

```

```

1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861

```

```

                LPCHK
.ENDM QKRPSK

; QUICK UNLOAD
; D=BLANK TO DO SUBCLR & LPCHK
; D=NON-BLANK TO BYPASS
.MACRO QKUNLD D
                JSR     PC, SUBCLR
                ERROR   24          ;CERR AFTER SCLR

                MOV     #UNLOAD, HCS1
                JSR     PC, DOCMD    ; DO UNLOAD CMD & GET CONTR READY
                ERROR   11          ;RDY NOT SET AFTER UNLOAD CMD.
                JSR     PC, TSTATN
                ERROR   12          ;NO ATTN AFTER UNLOAD CMD

                JSR     PC, SUBCLR
                ERROR   24          ;CERR AFTER SCLR

                MOV     T10, TEMP2
                JSR     PC, FSPOK
                ERROR   315         ;SPEED NOT DOWN BY TIMEOUT
                B

                JSR     PC, SUBCLR
                ERROR   24          ;CERR AFTER SCLR

                LPCHK
.ENDC

.ENDM QKUNLD

; A=WRHEAD/<CFMT!WRHEAD>
; USE WRHDR <A>,X TO OMIT CHECKING AO,BO,A1,B1
.MACRO WRHDR A,C,?D
                MOV     #<A>, HCS1
                JSR     PC, DATCMD   ; DO DATA XFER CMD & GET CONTR RDY
                ERROR   200         ; NO RDY AFTER WRITE HEADER CMD
                JSR     PC, GSTAT    ; GET FRESH STATUS
                BIT     #CERR, HCS1
                BEQ     D
                ERROR   201         ; CERR AFTER WRITE HEADER CMD
                TYPE   MSG18
                JMP     $EOP        ; ABORTING BALANCE OF TESTS
                ; ABORT DRIVE

D:
; IF B C

```

```

1862          F.EAB  0
1863          CHECK  277,267,300,270,<AFTER WRITE HEADER CMD>,T.A2,T.B2,0
1864      .ENDC
1865
1866      .ENDM  WRHDR
1867
1868      ;
1869      ; A=RDHEAD/<CFMT!RDHEAD>
1870      ; USE RDHDR <A>,X TO OMIT CHECKING A0,B0,A1,B1
1871      ;
1872      .MACRO  RDHDR  A,C,?D,?E
1873
1874          MOV      #RHTAB,RO
1875          MOV      #<A>,HCS1
1876          JSR      PC,DATCMD          ; DO DATA XFER CMD & GET CONTR RDY
1877          ERROR    171                ; NO RDY AFTER READ HEADER CMD
1878          BIT      #CERR,HCS1
1879          BEQ      D
1880          ERROR    174                ; CERR AFTER READ HEADER CMD
1881          TYPE     .MSG18             ; ABORT BALANCE OF TESTS
1882          JMP      $EOP              ; ABORT DRIVE
1883
1884      D:      MOV      RKDB(R5),(RO)+  ; 1'ST WORD FROM SILO TO RHTAB
1885             MOV      RKDB(R5),(RO)+  ; 2'ND WORD
1886             MOV      RKDB(R5),(RO)+  ; 3'RD WORD
1887
1888
1889             BIT      #DLT,RKCS2(R5)
1890             BEQ      E
1891             JSR      PC,GSTAT
1892             ERROR    173                ; DLT AFTER READ HEADER CMD
1893             TYPE     .MSG18             ; ABORTING BALANCE OF TESTS
1894             JMP      $EOP              ; ABORT DRIVE
1895
1896      E:
1897      .IF     B
1898             F.EAB  0
1899             CHECK  301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
1900      .ENDC
1901      .ENDM  RDHDR
1902
1903      ;
1904      ; A=TOCYL/FRCYL
1905      ; B=310 FOR TOCYL/311 FOR FRCYL
1906      ;
1907
1908      .MACRO  HDCHK3  A,B,?C
1909
1910             RDHDR  RDHEAD,X
1911             CMP    RHTAB,A ;CHECK WORD 0 (CYL#) ONLY
1912             BEQ    C          ;BR IF SAME
1913             ERROR  B          ;READ CYL WORD HEADER ERROR
1914
1915      C:
1916      .ENDM  HDCHK3
1917

```

1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973

```

.MACRO RALLHD ?A,?B,?C,?D,?E
      MOV      #RHTAB,RO
A:    MOV      #RDHEAD,HCS1
      JSR      PC,DATCMD      ;DO READ HEADER CMD & GET CONTR RDY
      ERROR   171             ;NO RDY AFTER READ HEADER CMD
      BIT     #CERR,HCS1
      BEQ     B
      ERROR   174             ;CERR AFTER READ HEADER CMD
      TYPE   MSG18           ;ABORTING BALANCE OF TESTS
      JMP     $EOP           ;ABORT DRIVE
B:    MOV      RKDB(R5),(RO)+ ;1'ST WORD FROM SILO TO RHTAB
      MOV      RKDB(R5),(RO)+ ;2'ND WORD
      MOV      RKDB(R5),(RO)+ ;3'RD WORD
      BIT     #DLT,RKCS2(R5) ;SEE IF DATA LATE
      BEQ     C
      JSR      PC,GSTAT
      ERROR   173             ;DATA LATE ON READ HEADER
      TYPE   MSG18           ;ABORT BALANCE OF TESTS
      JMP     $EOP           ;ABORT DRIVE
C:    CMP      RO,#RHTAB+132. ;ALL 66 WORDS DONE?
      BNE     A              ;BR IF NO
      JSR      PC,SORT      ;SORT RHTAB INTO SRTTAB SO THAT IT
                          ;BEGINS WITH SECTOR 0
                          ;WORD COUNT
                          ;ACTUAL HEADER TABLE
                          ;CALC HEADER TABLE
D:    MOV      (RO)+,HDWD
      MOV      (R1)+,TEMP1
      CMP     HDWD,TEMP1    ;COMPARE ACTUAL WITH CALCULATED WORD
      BEQ     E              ;BR IF COMPARE
      ERROR   202           ;READ HEADER MISMATCH
E:    INC     WDCNT
      CMP     WDCNT,#66.    ;ALL WORDS DONE?
      BNE     D              ;BR IF NO
.ENDM RALLHD

; 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
    
```


1974
1975
1976
1977
1978
1979
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

```

.ENDM HDTBL
;USE FSECA FS022,RDSEC,22 FOR 22 SECTOR FORMAT
;USE FSECA FS020,R20SEC,20 FOR 20 SECTOR FORMAT.
;
.MACRO FSECA A,B,C

;FIND SECTOR 0 IN C SECTOR FORMAT.
;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
;
A:      MOV      TEMP1,-(SP)      ;SAVE TEMP1
        MOV      T5000,TEMP1    ;SETUP TIMEOUT
1$:     JSR      PC,B            ;READ SECTOR
        TST      SECTOR        ;LOOK FOR SECTOR 0
        BNE     2$
        JSR      PC,B
        TST      SECTOR
        BEQ     3$              ;BR IF SAME TWICE
2$:     DEC      TEMP1
        BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
        MOV      (SP)+,TEMP1    ;ELSE RESTORE TEMP1
        RTS     PC              ;EXIT
3$:     MOV      (SP)+,TEMP1
        ADD     #2,(SP)         ;SKIP OVER ERROR
        RTS     PC

.ENDM

;USE FSECB FNS22,RDSEC,22 FOR 22 SECTOR FORMAT
;USE FSECB FNS20,R20SEC,20 FOR 20 SECTOR FORMAT
;
.MACRO FSECB A,B,C

;FIND NEXT SECTOR IN C SECTOR FORMAT
;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
;

```

2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067

```

A:      MOV      TEMP1, -(SP)      ;SAVE TEMP 1
        MOV      T500, TEMP1      ;SETUP TIMEOUT
1$:      JSR      PC, B             ;READ SECTOR
        CMP      PSEC, SECTOR
        BEQ      3$              ;BR IF SAME
        JSR      PC, B             ;ELSE TRY READ DIFFERENT TWICE
        CMP      PSEC, SECTOR
        BNE      2$              ;BR IF DIFFERENT TWICE
3$:      DEC      TEMP1            ;ELSE TRY AGAIN IF TIME LEFT
        BNE      1$
        MOV      (SP)+, TEMP1     ;RESTORE TEMP 1
        RTS      PC
2$:      MOV      (SP)+, TEMP1     ;RESTORE TEMP 1
        ADD      #2, (SP)         ;SKIP OVER ERROR
        RTS      PC
.ENDM

;USE SECTST F5022, FNS22, RDSEC FOR 22 SECTOR FORMAT
;USE SECTST F5020, FNS20, R20SEC FOR 20 SECTOR FORMAT
;THE FOLLOWING MACRO DEFINITION IS CHANGED ON 24-OCT-77
;1. DETECT 8 LINES FROM THE CODING
;MACRO SECTST D,E,F?A,?B,?C
        JSR      PC, D             ;FIND SECTOR D
        ERROR   142              ;SECTOR D NOT FOUND BY TIMEOUT
        CLR     PSEC              ;PREVIOUS SECTOR
A:      JSR      PC, E             ;FIND NEXT SECTOR
        ERROR   143              ;DIFFERENT SECTOR NOT FOUND BY TIMEOUT
        MOV     PSEC, ESEC        ;SETUP EXPECTED SECTOR
        ADD    #1, ESEC           ;UPDATE PREV SECTOR
        MOV     SECTOR, PSEC      ;UPDATE PREV SECTOR
        JSR    PC, F             ;READ SECTOR
        CMP    SECTOR, PSEC
        BEQ    B                 ;BR IF READ SAME TWICE
        JSR    PC, F
        CMP    SECTOR, PSEC
        BEQ    B
        ERROR  144              ;TRY 1 MORE TIME
        ;MSG B3 ERROR, SECTOR REG UNSTABLE
        ;MAY BE DURING SECTOR PULSE TIME
B:      CMP     SECTOR, ESEC
        BEQ    C
        ERROR  145              ;MSG B3 ERROR BETWEEN SECTOR COUNTS
C:      DEC     SECNT
        BNE    A                 ;BR IF SECTOR COUNT NOT DONE
.ENDM SECTST

;DETECT OUTER LIMIT: FCP1, FC, D.REV, OUTER
;DETECT INNER LIMIT: LCM1, LC, D.FWD, INNER
;THE FOLLOWING MACRO DEFINITION IS CHANGED ON 24-OCT-77

```

```
2068 ;1. ADD NEW LOCAL LABEL ?F
2069 ;2. DELET CHECK MACRO CALL ON LIMIT DETECT
2070 ;3. DEL CDW2 MCARO CALL
2071 .MACRO LIMIT A,B,C,D,?F
2072
2073 JSR PC,SUBCLR ;SUBSYS CLEAR & GET STATUS
2074 ERROR 24 ;CERR AFTER SCLR
2075
2076 CLR LPFLG
2077 INC BYPCERR ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
2078 INC UNLD ;USED FOR VALID HALT
2079
2080 MOV #PAT,RKMR1(R5) ;PARITY & WORD 0
2081 MOV A,RKDC(R5)
2082 MOV #SEEK,HCS1
2083 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
2084 ERROR 122 ;NO RDY FROM SEEK WITH BAD PARITY
2085 JSR PC,TSTATN ;TEST FOR ATTN
2086 ERROR 125 ;NO ATTN FROM SEEK WITH BAD PARITY
2087 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED A0
2088 MOV #<D.FLT!D.PAR>,E.B0
2089 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
2090 MOV #1,E.B1
2091 CHECK 110,111,146,147,<AFTER SEEK WITH BAD PARITY>,0,0,0
2092 DRCLR
2093
2094 MOV B,RKDC(R5)
2095 MOV #SEEK,HCS1
2096 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
2097 ERROR 131 ;NO RDY AFTER SEEK CMD
2098 MOV #CCLR,RKCS1(R5)
2099 JSR PC,GSTAT
2100 JSR PC,FLIM ;FIND LIMIT DETECT
2101 ERROR 160 ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
2102
2103 BIT #D.UNLD,HMR2
2104 BNE 15
2105 ERROR 305 ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
2106 JMP 305 ;BYPASS REST OF TEST
2107
2108 15: MOV #205,SESCAPE ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
2109 MOV #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
2110 MOV #<D.SKI!D.FLT>,E.B0
2111 MOV #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
2112 MOV #<D.LIMD!D.NMOV!1>,E.B1
2113 ; CHECK 161,162,163,164,<AFTER D LIMIT DETECT>,0,0,0
2114 JSR PC,CHKMSG ;CHECK MESSAGE A0,B0,A1,B1
2115 .WORD 0!0!0
2116 ERROR 161 ;MAY BE A0 ERROR
2117 ERROR 162 ;MAY BE B0 ERROR
2118 ERROR 163 ;MAY BE B1 ERROR
2119 NOP ;NEED FOR THE CALLIN SEQ
2120 BIT #D.LIMD,H.B1 ;SEE IF LIMIT DETECT BIT SET IN B1
2121 BNE 35 ;BRANCH IF SO
2122 TYPE EM42 ;OTHERWISE REPORT ERROR
2123 ERROR 164
```

```
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
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
```

```
3$: JSR PC,TSTATN  
ERROR 165 ;NO ATTN AFTER D LIMIT DETECT  
CLR BYPCERR ;ALLOW CHECKING CERR IN GSTAT1  
  
JSR PC,SUBCLR ;SUBSYS CLR  
ERROR 24 ;CERR AFTER SCLR  
MOV T10,TEMP2 ;SET UP TIMEOUT  
JSR PC,FHDHM ;FIND HEAD HOME  
ERROR 166 ;HEAD HOME NOT FOUND BEFORE TIMEOUT  
JSR PC,FLOAD ;FIND LOAD HEADS  
ERROR 167 ;LOAD HEADS NOT FOUND BEFORE TIMEOUT  
MOV T100,TEMP2 ;SETUP TIMEOUT  
JSR PC,FATT1 ;FIND ATTN  
ERROR 67 ;ATTN NOT FOUND BEFORE TIMEOUT  
2$: CLR $ESCAPE  
CLR UNLD ;CLEAR FLAG  
F.EAB D,DSC  
CHECK 63,64,65,66,<AT END OF HEAD LOADING>,T.A2,T.B2,0  
; CWD2 175,176,<AT END OF HEAD LOADING>  
MOV #2,RKMR1(R5) ;SELECT MESSAGE  
JSR PC,GSTAT ;GET STATUS AND MR2,MR3  
TST CYLADD ;RECAL SUCCESSFUL ?  
BEQ F ;BRANCH IF SO  
ERROR 50 ;REPORT ERROR  
F: DRCLR  
SWB14  
  
.ENDM LIMIT  
  
; A=CYL#, B=HEAD#  
.MACRO HEADER A,B  
NEWTST <<WRITE & READ HEADERS CYL A, HEAD B>>,1  
MOV #STACK,SP ;RESTORE STK PTR  
  
JSR PC,SUBCLR  
ERROR 24 ;CERR AFTER SCLR  
  
INC BYPFMT ;SET BIT 14 & 15 IN HEADER  
  
MOV #HDTAB,RKBA(R5) ;HEADER WORD TABLE  
MOV #-66,RKWC(R5) ;WORD COUNT.  
MOV #A,TOCYL  
HDTBL TOCYL,0,0  
MOV #A,RKDC(R5) ;CYL#  
WRHDR WRHEAD  
CLR SECNT ;SECTOR COUNT  
LOOP  
MOV #A,RKDC(R5) ;CYL #  
RALLHD
```

2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235

```
CLR BYPFMT ;ALLOW CORRECT FORMATTING
.ENDM HEADER

;SEEK TO MAJOR CYL: FC,FCP1,TEMP3,TEMP4,D.FWD,D.REV,ASL,MC,DEC,FC
;SEEK 0 TO ALL CYL: FC,FCP1,TEMP3,TEMP4,D.FWD,D.REV,INC,LC,DEC,FC
;SEEK 410 TO ALL CYL: LC,LCM1,TEMP4,TEMP3,D.REV,D.FWD,DEC,FC,INC,LC
.MACRO SKOSC A,B,C,D,E,F,G,H,I,J
    MOV A,FRCYL ;SETUP FROM CYL
    MOV B,TOCYL ;SETUP TO CYL
1$: LOOP
    MOV #10$,SESCAPE
    MOV FRCYL,TEMP3 ;SETUP
    MOV TOCYL,TEMP4 ;CYL DIFF
    SUB C,D ;FOR
    MOV D,CALDIF ;ERROR PRINTOUT
    MOV TOCYL,RKDC(R5) ;GO TO CYL #
    SKRDY E
2$: MOV #12$,SESCAPE
    MOV T50000,TEMP1 ;SETUP TIMEOUT
    SKATN
    CMP CYLADD,TOCYL
    BEQ 3$
    ERROR 207 ;CYL ADDR IN RKMR3 NOT=RKDC
3$: LOOP
    CLR $ESCAPE
    MOV TOCYL,RKDC(R5) ;CYL #
    HDCHK3 TOCYL,310
    LOOP
    MOV #14$,SESCAPE
    MOV FRCYL,RKDC(R5) ;RETURN TO CYL #
    MOV FRCYL,CCYL ;CURRENT CYL FOR TRUERROR ROUTINE
    SKRDY F
4$: MOV #16$,SESCAPE
    MOV T50000,TEMP1 ;SETUP TIMEOUT
    SKATN
    CMP CYLADD,FRCYL
    BEQ 5$
    ERROR 243 ;CYL ADDR IN RKMR3 NOT=RKDC
5$: LOOP
    CLR $ESCAPE
    MOV FRCYL,RKDC(R5) ;CYL #
    HDCHK3 FRCYL,311
    CMP TOCYL,H ;ALL CYL DONE?
```

2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260

```

      BEQ      6$          ;BR IF YES
      G        TOCYL     ;ELSE DO ANOTHER
      JMP      1$
6$:   SWB14
8$:   QKRPSK  I,J
10$:  TSTSW9  8$,2$
12$:  TSTSW9  8$,3$
14$:  TSTSW9  8$,4$
16$:  TSTSW9  8$,5$
      .ENDM   SKOSC

      .MACRO  EOPGM

      SCOPE
      CLR    $ESCAPE
      MOV    #1,$TIMES
      MOV    #STACK,$SP
      INC    $DEVCT
      CMP    DRIVS,$DEVCT ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
      BEQ    $EOP1+2     ;ARE ALL DRIVES PRESINT TESTED?
      JMP    NUDRV      ;BR IF YES
      .ENDM   ;IF NOT , TEST NEXT DRIVE PRESENT

$EOP1:
      .ENDM   SCOPE
      .ENDM   EOPGM

      .NLIST MD

```

.SBTTL COMMON TAGS

*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
*USED IN THE PROGRAM.

.=1100

;; START OF COMMON TAGS

SCMTAG:	.WORD	0	;;	START OF COMMON TAGS
\$TSTNM:	.BYTE	0	;;	CONTAINS THE TEST NUMBER
\$ERFLG:	.BYTE	0	;;	CONTAINS ERROR FLAG
\$ICNT:	.WORD	0	;;	CONTAINS SUBTEST ITERATION COUNT
\$LPADR:	.WORD	0	;;	CONTAINS SCOPE LOOP ADDRESS
\$LPERR:	.WORD	0	;;	CONTAINS SCOPE RETURN FOR ERRORS
\$ERTTL:	.WORD	0	;;	CONTAINS TOTAL ERRORS DETECTED
\$ITEMB:	.BYTE	0	;;	CONTAINS ITEM CONTROL BYTE
\$ERMAX:	.BYTE	1	;;	CONTAINS MAX. ERRORS PER TEST
\$ERRPC:	.WORD	0	;;	CONTAINS PC OF LAST ERROR INSTRUCTION
\$GDADR:	.WORD	0	;;	CONTAINS ADDRESS OF 'GOOD' DATA
\$BDADR:	.WORD	0	;;	CONTAINS ADDRESS OF 'BAD' DATA
\$GDDAT:	.WORD	0	;;	CONTAINS 'GOOD' DATA
\$BDDAT:	.WORD	0	;;	CONTAINS 'BAD' DATA
	.WORD	0	;;	RESERVED--NOT TO BE USED
	.WORD	0	;;	
\$AUTOB:	.BYTE	0	;;	AUTOMATIC MODE INDICATOR
\$INTAG:	.BYTE	0	;;	INTERRUPT MODE INDICATOR
	.WORD	0	;;	
\$SWR:	.WORD	DSWR	;;	ADDRESS OF SWITCH REGISTER
\$DISPLAY:	.WORD	DDISP	;;	ADDRESS OF DISPLAY REGISTER
\$TKS:	177560		;;	TTY KBD STATUS
\$TKB:	177562		;;	TTY KBD BUFFER
\$TPS:	177564		;;	TTY PRINTER STATUS REG. ADDRESS
\$TPB:	177566		;;	TTY PRINTER BUFFER REG. ADDRESS
\$NULL:	.BYTE	0	;;	CONTAINS NULL CHARACTER FOR FILLS
\$FILLS:	.BYTE	2	;;	CONTAINS # OF FILLER CHARACTERS REQUIRED
\$FILLC:	.BYTE	12	;;	INSERT FILL CHARS. AFTER A "LINE FEED"
\$TPFLG:	.BYTE	0	;;	"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
\$TMP0:	.WORD	0	;;	USER DEFINED
\$TMP1:	.WORD	0	;;	USER DEFINED
\$TMP2:	.WORD	0	;;	USER DEFINED
\$TMP3:	.WORD	0	;;	USER DEFINED
\$TMP4:	.WORD	0	;;	USER DEFINED
\$TMPS:	.WORD	0	;;	USER DEFINED
\$TIMES:	0		;;	MAX. NUMBER OF ITERATIONS
\$ESCAPE:	0		;;	ESCAPE ON ERROR ADDRESS
\$BELL:	.ASCIZ	<207><377><377>	;;	CODE FOR BELL
\$QUES:	.ASCII	/?/	;;	QUESTION MARK
\$CRLF:	.ASCII	<15>	;;	CARRIAGE RETURN
\$LF:	.ASCIZ	<12>	;;	LINE FEED

.SBTTL APT MAILBOX-ETABLE

.EVEN
\$MAIL: ; ; APT MAILBOX
\$MSGTY: .WORD AMSGTY ; ; MESSAGE TYPE CODE

2261
2262
2263
2264
2265
2266
2267 001100
2268 001100 001100
2269 001100 000000
2270 001102 000
2271 001103 000
2272 001104 000000
2273 001106 000000
2274 001110 000000
2275 001112 000000
2276 001114 000
2277 001115 001
2278 001116 000000
2279 001120 000000
2280 001122 000000
2281 001124 000000
2282 001126 000000
2283 001130 000000
2284 001132 000000
2285 001134 000
2286 001135 000
2287 001136 000000
2288 001140 177570
2289 001142 177570
2290 001144 177560
2291 001146 177562
2292 001150 177564
2293 001152 177566
2294 001154 000
2295 001155 002
2296 001156 012
2297 001157 000
2298 001160 000000
2299 001162 000000
2300 001164 000000
2301 001166 000000
2302 001170 000000
2303 001172 000000
2304 001174 000000
2305 001176 000000
2306 001200 177607 000377
2307 001204 077
2308 001205 015
2309 001206 000012
2310
2311
2312
2313
2314
2315 001210
2316 001210 000000

2317	001212	000000	\$FATAL: .WORD	AFATAL	:: FATAL ERROR NUMBER
2318	001214	000000	\$TESTN: .WORD	ATESTN	:: TEST NUMBER
2319	001216	000000	\$PASS: .WORD	APASS	:: PASS COUNT
2320	001220	000000	\$DEVCT: .WORD	ADEVCT	:: DEVICE COUNT
2321	001222	000000	\$UNIT: .WORD	AUNIT	:: I/O UNIT NUMBER
2322	001224	000000	\$MSGAD: .WORD	AMSGAD	:: MESSAGE ADDRESS
2323	001226	000000	\$MSGLG: .WORD	AMSGLG	:: MESSAGE LENGTH
2324	001230		\$ETABLE: .WORD		:: APT ENVIRONMENT TABLE
2325	001230	000	\$ENV: .BYTE	AENV	:: ENVIRONMENT BYTE
2326	001231	000	\$ENVM: .BYTE	AENVM	:: ENVIRONMENT MODE BITS
2327	001232	000000	\$SWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
2328	001234	000000	\$USWR: .WORD	AUSWR	:: USER SWITCHES
2329	001236	000000	\$CPUOP: .WORD	ACPUOP	:: CPU TYPE, OPTIONS
2330			::*		BITS 15-11=CPU TYPE
2331			::*		11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
2332			::*		11/70=06, PDQ=07, Q=10
2333			::*		BIT 10=REAL TIME CLOCK
2334			::*		BIT 9=FLOATING POINT PROCESSOR
2335			::*		BIT 8=MEMORY MANAGEMENT
2336	001240	000	\$MAMS1: .BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
2337	001241	000	\$MTYP1: .BYTE	AMTYP1	:: MEM. TYPE, BLK#1
2338			::*		MEM. TYPE BYTE -- (HIGH BYTE)
2339			::*		900 NSEC CORE=001
2340			::*		300 NSEC BIPOLAR=002
2341			::*		500 NSEC MOS=003
2342	001242	000000	\$MADR1: .WORD	AMADR1	:: HIGH ADDRESS, BLK#1
2343			::*		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
2344	001244	000	\$MAMS2: .BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
2345	001245	000	\$MTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
2346	001246	000000	\$MADR2: .WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
2347	001250	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
2348	001251	000	\$MTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
2349	001252	000000	\$MADR3: .WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
2350	001254	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
2351	001255	000	\$MTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
2352	001256	000000	\$MADR4: .WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
2353	001260	000000	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
2354	001262	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
2355	001264	177440	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
2356	001266	000000	\$DEVN: .WORD	ADEVN	:: DEVICE MAP
2357	001270	000000	\$CDW1: .WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
2358	001272	000000	\$CDW2: .WORD	ACDW2	:: CONTROLLER DESCRIPTION WORD#2
2359	001274	000000	\$DDW0: .WORD	ADDW0	:: DEVICE DESCRIPTOR WORD#0
2360	001276	000000	\$DDW1: .WORD	ADDW1	:: DEVICE DESCRIPTOR WORD#1
2361	001300	000000	\$DDW2: .WORD	ADDW2	:: DEVICE DESCRIPTOR WORD#2
2362	001302	000000	\$DDW3: .WORD	ADDW3	:: DEVICE DESCRIPTOR WORD#3
2363	001304	000000	\$DDW4: .WORD	ADDW4	:: DEVICE DESCRIPTOR WORD#4
2364	001306	000000	\$DDW5: .WORD	ADDW5	:: DEVICE DESCRIPTOR WORD#5
2365	001310	000000	\$DDW6: .WORD	ADDW6	:: DEVICE DESCRIPTOR WORD#6
2366	001312	000000	\$DDW7: .WORD	ADDW7	:: DEVICE DESCRIPTOR WORD#7
2367	001314		\$ETEND:		
2368			.MEXIT		
2369		177440	ABASE=	177440	:: DEFAULT BUSS ADDRESS
2370	001314	000210	RKVEC:	210	:: DEFAULT CONTROLLER INTERRUPT VECTOR
2371	001316	000240	RKPRI:	PR5	:: PRIORITY
2372	001320	172540	PKS:	172540	:: P-CLOCK STATUS REG

2373	001322	172542	PKSB:	172542	:P-CLOCK SET BUFFER
2374	001324	172544	PKRB:	172544	:P-CLOCK READ BUFFER
2375	001326	177546	LKS:	177546	:L-CLOCK STATUS REG.
2376					
2377	001330	000100	LCVEC:	100	:L-CLOCK INTERRUPT VECTOR
2378	001332	000104	PCVEC:	104	:P-CLOCK INTERRUPT VECTOR.
2379					
2380		000114	MEMVEC=	114	:MEMORY PARITY VECTOR
2381		172100	MEMBAS=	172100	:MEMORY PARITY OPTION CSR START ADDR
2382	001334	000000	TRAPPC:	0	:PC FOR MEMORY CHECK ENABLE TRAP
2383					
2384	001336	000000	PARAM:	0	:1 FOR 220 OR 230 START, NO DEFAULT
2385	001340	000000	BYPT16:	0	:1 FOR 210, 230, 270 START
2386	001342	000000	MODTST:	0	:1 FOR 260 OR 270 START
2387	001344	000000	FTITLE:	0	:FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
2388					
2389	001346	000000	DRVPTR:	0	:CONTAINS THE POINTER TO THE DRIVE FLAG
2390					: (DRIVO-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
2391	001350	000000	FRCYL:	0	:FROM CYL
2392	001352	000000	TOCYL:	0	:TO CYL
2393	001354	000000	CCYL:	0	:CURRENT CYL, USED IN N SQUARE TEST
2394	001356	000000	PCYL:	0	:PREV CYL, USED IN N SQUARE TEST
2395	001360	000000	CALDIF:	0	:CALC CYL DIFF USED IN N SQUARE TEST
2396	001362	000000	CYLDIF:	0	:CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
2397	001364	000000	CYLOAD:	0	:CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
2398	001366	000000	CALADD:	0	:CYL ADDR USED IN FHDTAB ROUTINE
2399					
2400	001370	000074	HZ:	60.	:60 FOR 60 CPS
2401					:50 FOR 50 CPS
2402	001372	000000	COUNT:	0	:LOADED TO 50 OR 60 TO COUNT TO 1 SEC
2403					:OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
2404	001374	000000	SEC:	0	:SECOND COUNTER
2405	001376	000000	TIMUP:	0	:FLAG TO INDICATE TIME IS UP
2406	001400	000000	SECNT:	0	:SECTOR COUNT
2407	001402	000000	PSEC:	0	:PREVIOUS SECTOR
2408	001404	000000	ESEC:	0	:EXPECTED SECTOR
2409	001406	000000	SECTOR:	0	:SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
2410					
2411	001410	000000	LPFLG:	0	:SET TO 0 TO RETURN TO \$LPADR
2412					:IF SW14 OR SW8 SET
2413					:SET TO 1 TO RETURN TO \$LPERR
2414					:IF SW9 SET
2415	001412	000001	T1:	1	:TIMEOUT CONSTANTS
2416	001414	000012	T10:	10.	
2417	001416	000144	T100:	100.	
2418	001420	000764	T500:	500.	
2419	001422	004704	T2500:	2500.	
2420	001424	011610	T5000:	5000.	
2421	001426	141520	T50000:	50000.	
2422					
2423	001430	000000	HEAD:	0	:HEAD NUMBER
2424	001432	000000	HEAD#:	0	:HEAD # FROM H.B3 RIGHT JUSTIFIED
2425	001434	000000	HD1:	0	:SHIFTED HEAD# FOR FORMATTER ROUTINE
2426	001436	000000	FORMAT:	0	:FORMAT TYPE
2427	001440	000000	FMT1:	0	:SHIFTED FORMAT FOR FORMATTER ROUTINE
2428	001442	000000	WDCNT:	0	:WORD COUNT

```

2429
2430 001444 000000          DATA: 0                ;ALL 0'S
2431 001446 052525          DATA1: 52525           ;0101 PATT
2432 001450 177777          DATA1: 177777        ;ALL 1'S
2433
2434 001452 000000          WORD: 0                ;HEADER/DATA WORD
2435 001454 000000          HDWD: 0                ;HEADER WORD FROM RKDB
2436
2437 001456 000000          BSERR: 0              ;CANNOT READ BSE INFO WHEN SET
2438 001460 000000          LIMERR: 0            ;LIMIT DETECT ERROR FLAG
2439
2440 001462 000000          BYPCERR:0           ;SET TO 1 TO BYPASS CKCERR IN GSTAT1 ROUTINE
2441 001464 000000          BYPFMT: 0          ;BYPASS FORMAL FORMATTING OF HEADERS
2442
2443
2444
2445
2446 001466 000000          CHKFLG: 0           ;WORDS TO BE CHECKED
2447
2448 001470 000102          HDTAB: .BLKW 66.    ;CALCULATED HEADER WORD TABLE
2449 001674 000102          RHTAB: .BLKW 66.    ;FILLED AFTER READ HEADER CMD
2450 002100 000102          SRTTAB: .BLKW 66.   ;ABOVE RHTAB SORTED STARTING FORM
2451
2452 002304 000400          BSE22H: .BLKW 256.  ;SECTOR 0 BY SORT ROUTINE
2453
2454
2455
2456 003304 000000          UNLD: 0              ;22 SECTOR HARDWARE BSE INFO.
2457 003306 000000          BADHDR: 0           ;22 SECTOR SOFTWARE BSE INFO
2458 003310 000000          HPEND: 0            ;OVERLAYS MSG1
2459
2460
2461
2462
2463
2464
2465 003312 001 002 004 ATTN: .BYTE 1,2,4,10,20,40,100,200 ;ATN 0-7 RESP.
2466 003315 010 020 040
2467 003320 100 200
2468
2469
2470
2471
2472
2473
2474
2475
2476 003322 000000          HCS1: 0              ;THE ABOVE 3 FLAGS ARE USED
2477 003324 000000          HCS2: 0              ;BY 'STOP' ROUTINE TO BRING
2478 003326 000000          HWC: 0              ;THE CPU TO A VALID HALT.
2479 003330 000000          HBA: 0              ;
2480 003332 000000          HDA: 0              ;
2481 003334 000000          HDS: 0              ;
2482 003336 000000          HER: 0              ;
2483 003340 000000          HASOF: 0           ;HOLD RKCS1
2484 003342 000000          HDC: 0              ;HOLD RKCS2
                                     ;HOLD RKWC
                                     ;ETC.

```

```

; THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
; THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
;

```

2485 003344 000000
 2486 003346 000000
 2487 003350 000000
 2488 003352 000000
 2489 003354 000000
 2490 003356 000000
 2491
 2492 003360 000000
 2493 003362 000000
 2494 003364 000000
 2495 003366 000000
 2496 003370 000000
 2497
 2498
 2499
 2500 003372 000000
 2501 003374 000000
 2502 003376 000000
 2503 003400 000000
 2504 003402 000000
 2505 003404 000000
 2506 003406 000000
 2507 003410 000000
 2508
 2509
 2510
 2511 003412 000000
 2512 003414 000000
 2513 003416 000000
 2514 003420 000000
 2515 003422 000000
 2516 003424 000000
 2517 003426 000000
 2518 003430 000000
 2519
 2520
 2521
 2522
 2523
 2524
 2525
 2526
 2527
 2528
 2529
 2530
 2531 003432 000000
 2532 003434 000000
 2533 003436 000000
 2534 003440 000000
 2535 003442 000000
 2536
 2537
 2538
 2539
 2540 003444 000000

HDB: 0
 HMR1: 0
 HMR2: 0
 HMR3: 0
 HPOS: 0
 HPAT: 0

TEMP1: 0
 TEMP2: 0
 TEMP3: 0
 TEMP4: 0
 TEMPS: 0

; TEMPORARY STORAGE.

... THE FOLLOWING ARE HOLDING REGISTERS FOR MSG A (0-3) & MSG B (0-3)

H.A0: 0
 H.B0: 0
 H.A1: 0
 H.B1: 0
 H.A2: 0
 H.B2: 0
 H.A3: 0
 H.B3: 0

... THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE

E.A0: 0
 E.B0: 0
 E.A1: 0
 E.B1: 0
 E.A2: 0
 E.B2: 0
 E.A3: 0
 E.B3: 0

... THE FOLLOWING ARE IDENTITIES FOR DRIVE MSG WORDS TO BE TESTED

T.A2=BIT0
 T.B2=BIT1
 T.B3=BIT2

; TEST MSG A2 IF SET

... ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.

DDUMP: 0
 DDPCH: 0
 ACT11: 0
 PPTP: 0
 DRIVS: 0

; FLAG - SET WHEN IN DDP DUMP MODE
 ; FLAG - SET WHEN IN DDP CHAIN MODE
 ; FLAG - SET WHEN IN ACT11 MODE OF OPERATION
 ; FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
 ; CONTAINS THE NUMBER OF DRIVES PRESENT

... THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE IS PRESENT AND IS TO BE TESTED.

DRIVO: 0

; FLAG SET TO 1 WHEN DRIVE 0 PRESENT

2541 003446 000000
2542 003450 000000
2543 003452 000000
2544 003454 000000
2545 003456 000000
2546 003460 000000
2547 003462 000000
2548
2549 003464 000000
2550 003466 000000
2551 003470 000000
2552 003472 000000

DRIV1: 0
DRIV2: 0
DRIV3: 0
DRIV4: 0
DRIV5: 0
DRIV6: 0
DRIV7: 0

LCLKF: 0
PCLKF: 0
DOTIM: 0
SIZFLG: 0

:FOR DRIVE 1
:FOR DRIVE 2
:FOR DRIVE 3
:FOR DRIVE 4
:FOR DRIVE 5
:FOR DRIVE 6
:FOR DRIVE 7

:L-CLOCK FLAG PRESENT FLAG
:P-CLOCK FLAG PRESENT FLAG
:SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
:SET IF DEFAULT DO SIZING IN TEST 1

.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 SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
 ;*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
 ;*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:

2553				
2554				
2555				
2556				
2557				
2558				
2559				
2560				
2561				
2562				
2563				
2564				
2565				
2566				
2567	003474			
2568				
2569				
2570	003474	057016		
2571	003476	063767		
2572	003500	066076		
2573	003502	066744		
2574				
2575				
2576	003504	057266		
2577	003506	063767		
2578	003510	066076		
2579	003512	066744		
2580				
2581				
2582	003514	057307		
2583	003516	063767		
2584	003520	066076		
2585	003522	066744		
2586				
2587				
2588	003524	057330		
2589	003526	063767		
2590	003530	066076		
2591	003532	066744		
2592				
2593	003534	057417		
2594	003536	063767		
2595	003540	066076		
2596	003542	066744		
2597				
2598				
2599	003544	057451		
2600	003546	063767		
2601	003550	066076		
2602	003552	066744		
2603				
2604				
2605	003554	057521		
2606	003556	063767		
2607	003560	066076		
2608	003562	066744		

```

;ERROR 1
EM2 ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2
DH1
DT1
DF1

;ERROR 2
EM5 ;DETECTED MDS
DH1
DT1
DF1

;ERROR 3
EM6 ;DETECTED LIFE
DH1
DT1
DF1

;ERROR 4
EM7 ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
DH1
DT1
DF1

;ERROR 5
EM8 ;DDT NOT SET IN MR2
DH1
DT1
DF1

;ERROR 6
EM9 ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
DH1
DT1
DF1

;ERROR 7
EM10 ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
DH1
DT1
DF1
  
```

2609				
2610			;ERROR 10	
2611	003564	057564	EM11	;DRA & NED BOTH SET
2612	003566	063767	DH1	
2613	003570	066076	DT1	
2614	003572	066744	DF1	
2615				
2616			;ERROR 11	
2617	003574	057630	EM12	;CONTROLLER NOT READY
2618	003576	064560	DH18	;AFTER UNLOAD CMD.
2619	003600	066076	DT1	
2620	003602	067070	DF10	
2621				
2622			;ERROR 12	
2623	003604	057661	EM13	;NO ATTN
2624	003606	064560	DH18	;AFTER UNLOAD CMD
2625	003610	066076	DT1	
2626	003612	067070	DF10	
2627			;ERROR 13	
2628	003614	057703	EM14	;WRONG ATTN
2629	003616	064560	DH18	
2630	003620	066076	DT1	
2631	003622	067070	DF10	
2632			;ERROR 14	
2633	003624	057730	EM15	;DRDY NOT CLEARED
2634	003626	064560	DH18	
2635	003630	066076	DT1	
2636	003632	067070	DF10	
2637			;ERROR 15	
2638	003634	057762	EM16	;DSC NOT SET
2639	003636	064560	DH18	
2640	003640	066076	DT1	
2641	003642	067070	DF10	
2642			;ERROR 16	
2643	003644	060007	EM17	;MSG A0 ERROR
2644	003646	064160	DH8	;IN UNLD
2645	003650	066500	DT13	
2646	003652	067240	DF20	
2647			;ERROR 17	
2648	003654	060024	EM18	;MSG B0 ERROR
2649	003656	064160	DH8	;IN UNLD
2650	003660	066500	DT13	
2651	003662	067240	DF20	
2652			;ERROR 20	
2653	003664	060041	EM19	;MSG A1 ERROR
2654	003666	064160	DH8	;IN UNLD
2655	003670	066500	DT13	
2656	003672	067240	DF20	
2657			;ERROR 21	
2658	003674	060056	EM20	;MSG B1 ERROR
2659	003676	064160	DH8	;IN UNLD
2660	003700	066500	DT13	
2661	003702	067240	DF20	
2662			;ERROR 22	
2663	003704	061471	EM46	;MSG A2 ERROR
2664	003706	064160	DH8	;IN UNLD

2665	003710	066560	DT14	
2666	003712	067314	DF22	
2667			; ERROR 23	
2668	003714	061504	EM47	; MSG B2 ERROR
2669	003716	064160	DH8	; IN UNLD
2670	003720	066560	DT14	
2671	003722	067314	DF22	
2672				
2673			; ERROR 24	
2674	003724	060073	EM21	; CERR SET
2675	003726	064647	DH21	; AFTER SCLR
2676	003730	066076	DT1	
2677	003732	067070	DF10	
2678			; ERROR 25	
2679	003734	060115	EM22	; RLS DID NOT SET CERR
2680	003736	063767	DH1	
2681	003740	066076	DT1	
2682	003742	066744	DF1	
2683				
2684			; ERROR 26	
2685	003744	060154	EM23	; SACK SET AFTER RLS SENT
2686	003746	063767	DH1	
2687	003750	066076	DT1	
2688	003752	066744	DF1	
2689				
2690			; ERROR 27	
2691	003754	060234	EM24	; VOL VALID NOT SET
2692	003756	064601	DH19	; AFTER PACK CMD
2693	003760	066076	DT1	
2694	003762	067070	DF10	
2695			; ERROR 30	
2696	003764	060260	EM25	; DRIVE TYPE SET IN MR2
2697	003766	063767	DH1	
2698	003770	066076	DT1	
2699	003772	066744	DF1	
2700			; ERROR 31	
2701	003774	060306	EM26	; DDT SET IN RKDS
2702	003776	063767	DH1	
2703	004000	066076	DT1	
2704	004002	066744	DF1	
2705			; ERROR 32	
2706	004004	060326	EM27	; DTYE SET IN RKER
2707	004006	063767	DH1	
2708	004010	066076	DT1	
2709	004012	066744	DF1	
2710			; ERROR 33	
2711	004014	060347	EM28	; DTYE NOT SET IN RKER
2712	004016	063767	DH1	
2713	004020	066076	DT1	
2714	004022	066744	DF1	
2715			; ERROR 34	
2716	004024	060425	EM29	; DTYE DID NOT SET CERR
2717	004026	063767	DH1	
2718	004030	066076	DT1	
2719	004032	066744	DF1	
2720			; ERROR 35	

2721	004034	060474	EM30	;C-D PARITY ERROR SET IN MR3
2722	004036	063767	DH1	
2723	004040	066076	DT1	
2724	004042	066744	DF1	
2725			;ERROR 36	
2726	004044	060525	EM31	;D-C PARITY SET IN CS1
2727	004046	063767	DH1	
2728	004050	066076	DT1	
2729	004052	066744	DF1	
2730			;ERROR 37	
2731	004054	060552	EM32	;FAULT NOT SET IN MR3
2732	004056	063767	DH1	
2733	004060	066076	DT1	
2734	004062	066744	DF1	
2735			;ERROR 40	
2736	004064	060577	EM33	;C-D PARITY ERROR NOT SET IN MR3
2737	004066	063767	DH1	
2738	004070	066076	DT1	
2739	004072	066744	DF1	
2740			;ERROR 41	
2741	004074	060634	EM34	;D-C PARITY NOT SET IN CS1
2742	004076	063767	DH1	
2743	004100	066076	DT1	
2744	004102	066744	DF1	
2745			;ERROR 42	
2746	004104	060665	EM35	;DCPAR DID NOT SET CERR
2747	004106	063767	DH1	
2748	004110	066076	DT1	
2749	004112	066744	DF1	
2750			;ERROR 43	
2751	004114	060740	EM36	;CYL ADDR IN B2 NOT = RKDC
2752	004116	064446	DH14	;AFTER SEEK WITH BAD PARITY
2753	004120	066560	DT14	
2754	004122	067314	DF22	
2755			;ERROR 44	
2756	004124	061000	EM37	;CYL DIFF IN A2 NOT=RKDC
2757	004126	064446	DH14	
2758	004130	066560	DT14	
2759	004132	067314	DF22	
2760			;ERROR 45	
2761	004134	060740	EM36	;CYL ADDR IN RKMR3 NOT=RKDC
2762	004136	064446	DH14	
2763	004140	066136	DT4	
2764	004142	067044	DF6	
2765			;ERROR 46	
2766	004144	061040	EM38	;CYL DIFF IN RKMR2 NOT=CALDIF
2767	004146	064446	DH14	
2768	004150	066136	DT4	
2769	004152	067044	DF6	
2770			;ERROR 47	
2771	004154	061111	EM39	;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2772	004156	064540	DH17	;AFTER RECAL CMD
2773	004160	066560	DT14	
2774	004162	067314	DF22	
2775			;ERROR 50	
2776	004164	061146	EM40	;CYL ADDR IN RKMR3 NOT CLEARED

2777	004166	064540	DH17	;AFTER RECAL CMD
2778	004170	066560	DT14	
2779	004172	067314	DF22	
2780			;ERROR 51	
2781	004174	060007	EM17	;AO ERROR
2782	004176	064763	DH26	;AFTER READ DATA CMD
2783	004200	066500	DT13	
2784	004202	067240	DF20	
2785			;ERROR 52	
2786	004204	060024	EM18	;BO ERROR
2787	004206	064763	DH26	
2788	004210	066500	DT13	
2789	004212	067240	DF20	
2790			;ERROR 53	
2791	004214	061265	EM43	;HEAD DECODE IN B3 NOT CLEARED
2792	004216	064540	DH17	;AFTER RECAL CMD
2793	004220	066650	DT15	
2794	004222	067350	DF23	
2795			;ERROR 54	
2796	004224	061314	EM44	;B3 HEAD DECODE INCORRECT
2797	004226	064476	DH16	
2798	004230	066650	DT15	
2799	004232	067350	DF23	
2800			;ERROR 55	
2801	004234	057661	EM13	;NO ATTN
2802	004236	064540	DH17	;AFTER RECAL CMD
2803	004240	066076	DT1	
2804	004242	067070	DF10	
2805			;ERROR 56	
2806	004244	062436	EM64	;MSG B3 HEAD REG NOT CLEARED
2807	004246	064160	DH8	;IN UNLOAD
2808	004250	066650	DT15	
2809	004252	067350	DF23	
2810			;ERROR 57	
2811	004254	060007	EM17	;MSG AO ERROR
2812	004256	064204	DH9	;AFTER START SPINDLE CMD REC'D BY DRIVE
2813	004260	066500	DT13	
2814	004262	067240	DF20	
2815			;ERROR 60	
2816	004264	060024	EM18	;MSG BO ERROR
2817	004266	064204	DH9	
2818	004270	066500	DT13	
2819	004272	067240	DF20	
2820			;ERROR 61	
2821	004274	060041	EM19	;MSG A1 ERROR
2822	004276	064204	DH9	
2823	004300	066500	DT13	
2824	004302	067240	DF20	
2825			;ERROR 62	
2826	004304	060056	EM20	;MSG B1 ERROR
2827	004306	064204	DH9	
2828	004310	066500	DT13	
2829	004312	067240	DF20	
2830			;ERROR 63	
2831	004314	060007	EM17	
2832	004316	064250	DH10	;AT END OF HEAD LOADING

2833	004320	066500	DT13
2834	004322	067240	DF20
2835			;ERROR 64
2836	004324	060024	EM18
2837	004326	064250	DH10
2838	004330	066500	DT13
2839	004332	067240	DF20
2840			;ERROR 65
2841	004334	060041	EM19
2842	004336	064250	DH10
2843	004340	066500	DT13
2844	004342	067240	DF20
2845			;ERROR 66
2846	004344	060056	EM20
2847	004346	064250	DH10
2848	004350	066500	DT13
2849	004352	067240	DF20
2850			;ERROR 67
2851	004354	057661	EM13
2852	004356	064250	DH10
2853	004360	066076	DT1
2854	004362	067070	DF10
2855			;ERROR 70
2856	004364	061620	EM50
2857	004366	063767	DH1
2858	004370	066076	DT1
2859	004372	066744	DF1
2860			;ERROR 71
2861	004374	060007	EM17
2862	004376	064277	DH11
2863	004400	066500	DT13
2864	004402	067240	DF20
2865			;ERROR 72
2866	004404	060024	EM18
2867	004406	064277	DH11
2868	004410	066500	DT13
2869	004412	067240	DF20
2870			;ERROR 73
2871	004414	060041	EM19
2872	004416	064277	DH11
2873	004420	066500	DT13
2874	004422	067240	DF20
2875			;ERROR 74
2876	004424	060056	EM20
2877	004426	064277	DH11
2878	004430	066500	DT13
2879	004432	067240	DF20
2880			;ERROR 75
2881	004434	061671	EM51
2882	004436	063767	DH1
2883	004440	066076	DT1
2884	004442	066744	DF1
2885			;ERROR 76
2886	004444	060007	EM17
2887	004446	064336	DH12
2888	004450	066500	DT13

;NO ATTN
;AT END OF HEAD LOADING.

;FWD NOT SET WITHIN 60 SEC FROM
;START SPINDLE CMD.

;AFTER START SPINDLE CMD & FWD SET.

;FWD NOT CLEARED WITHIN 5 SEC OF MOTION
;FROM START SPINDLE CMD.

;AT INNER LIMIT FROM START SPINDLE CMD.

2889	004452	067240		DF20
2890			;ERROR 77	
2891	004454	060024		EM18
2892	004456	064336		DH12
2893	004460	066500		DT13
2894	004462	067240		DF20
2895			;ERROR 100	
2896	004464	060041		EM19
2897	004466	064336		DH12
2898	004470	066500		DT13
2899	004472	067240		DF20
2900			;ERROR 101	
2901	004474	060056		EM20
2902	004476	064336		DH12
2903	004500	066500		DT13
2904	004502	067240		DF20
2905			;ERROR 102	
2906	004504	061532		EM49
2907	004506	063767		DH1
2908	004510	066076		DT1
2909	004512	066744		DF1
2910			;ERROR 103	
2911	004514	060007		EM17
2912	004516	064377		DH13
2913	004520	066500		DT13
2914	004522	067240		DF20
2915			;ERROR 104	
2916	004524	060024		EM18
2917	004526	064377		DH13
2918	004530	066500		DT13
2919	004532	067240		DF20
2920			;ERROR 105	
2921	004534	060041		EM19
2922	004536	064377		DH13
2923	004540	066500		DT13
2924	004542	067240		DF20
2925			;ERROR 106	
2926	004544	060056		EM20
2927	004546	064377		DH13
2928	004550	066500		DT13
2929	004552	067240		DF20
2930			;ERROR 107	
2931	004554	061350		EM45
2932	004556	063767		DH1
2933	004560	066076		DT1
2934	004562	066744		DF1
2935			;ERROR 110	
2936	004564	060007		EM17
2937	004566	064446		DH14
2938	004570	066560		DT14
2939	004572	067314		DF22
2940			;ERROR 111	
2941	004574	060024		EM18
2942	004576	064446		DH14
2943	004600	066560		DT14
2944	004602	067314		DF22

;FWD NOT SET WITHIN 4 SEC IN RTZ PORTION
;OF START SPIN CMD.

;FROM OUTER LIMIT TO CYL 0 DURING LOADING

;DRIVE READY NOT SET WITHIN 1 SEC
;FROM FWD IN RTZ PORTION OF START SPIN CMD.

;MSG A0 ERROR
;AFTER SEEK WITH BAD PARITY

;MSG B0 ERROR
;AFTER SEEK WITH BAD PARITY

2945			;ERROR 112	
2946	004604	060041	EM19	;A1 ERROR
2947	004606	064763	DH26	;AFTER READ DATA CMD
2948	004610	066500	DT13	
2949	004612	067240	DF20	
2950			;ERROR 113	
2951	004614	060056	EM20	;B1 ERROR
2952	004616	064763	DH26	
2953	004620	066500	DT13	
2954	004622	067240	DF20	
2955			;ERROR 114	
2956	004624	060007	EM17	
2957	004626	064476	DH16	;AFTER LOADING HEAD REGISTER & SEEK CMD
2958	004630	066500	DT13	
2959	004632	067240	DF20	
2960			;ERROR 115	
2961	004634	060024	EM18	
2962	004636	064476	DH16	
2963	004640	066500	DT13	
2964	004642	067240	DF20	
2965			;ERROR 116	
2966	004644	057630	EM12	;CONT NOT RDY
2967	004646	064601	DH19	;AFTER PACK CMD
2968	004650	066076	DT1	
2969	004652	067070	DF10	
2970			;ERROR 117	
2971	004654	057630	EM12	;CONT NOT RDY
2972	004656	064620	DH20	;AFTER SEL DR CMD
2973	004660	066076	DT1	
2974	004662	067070	DF10	
2975			;ERROR 120	
2976	004664	057630	EM12	
2977	004666	064647	DH21	;AFTER SUBSYS CLEAR
2978	004670	066076	DT1	
2979	004672	067070	DF10	
2980			;ERROR 121	
2981	004674	057630	EM12	
2982	004676	064204	DH9	;AFTER START SPINDLE CMD
2983	004700	066076	DT1	
2984	004702	067070	DF10	
2985			;ERROR 122	
2986	004704	057630	EM12	
2987	004706	064446	DH14	;AFTER SEEK WITH BAD PARITY
2988	004710	066076	DT1	
2989	004712	067070	DF10	
2990			;ERROR 123	
2991	004714	063547	EM88	;NO DRIVES FOUND
2992	004716	063767	DH1	
2993	004720	066076	DT1	
2994	004722	066744	DF1	
2995			;ERROR 124	
2996	004724	057630	EM12	
2997	004726	064540	DH17	;AFTER RECAL CMD
2998	004730	066076	DT1	
2999	004732	067070	DF10	
3000			;ERROR 125	

3001	004734	057661	EM13	;NO ATTN
3002	004736	064446	DH14	;FROM SEEK WITH BAD PARITY
3003	004740	066076	DT1	
3004	004742	067070	DF10	
3005			;ERROR 126	
3006	004744	063640	EM89	;NO DRVS FOUND IN DEVICE MAP
3007	004746	063767	DH1	
3008	004750	066076	DT1	
3009	004752	066744	DF1	
3010			;ERROR 127	
3011	004754	061111	EM39	;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3012	004756	064647	DH21	;AFTER SCLR
3013	004760	066076	DT1	
3014	004762	067070	DF10	
3015			;ERROR 130	
3016	004764	061146	EM40	;CYL ADDR IN RKMR3 NOT CLEARED
3017	004766	064647	DH21	
3018	004770	066076	DT1	
3019	004772	067070	DF10	
3020			;ERROR 131	
3021	004774	057630	EM12	;NO RDY
3022	004776	064744	DH25	;AFTER SEEK CMD
3023	005000	066076	DT1	
3024	005002	067070	DF10	
3025			;ERROR 132	
3026	005004	057661	EM13	;NO ATTN
3027	005006	064744	DH25	
3028	005010	066076	DT1	
3029	005012	067070	DF10	
3030			;ERROR 133	
3031	005014	060007	EM17	;MSG A0 ERROR
3032	005016	064744	DH25	
3033	005020	066500	DT13	
3034	005022	067240	DF20	
3035			;ERROR 134	
3036	005024	060024	EM18	;MSG B0 ERROR
3037	005026	064744	DH25	
3038	005030	066500	DT13	
3039	005032	067240	DF20	
3040			;ERROR 135	
3041	005034	060041	EM19	;MSG A1 ERROR
3042	005036	064744	DH25	
3043	005040	066500	DT13	
3044	005042	067240	DF20	
3045			;ERROR 136	
3046	005044	060056	EM20	;MSG B1 ERROR.
3047	005046	064744	DH25	
3048	005050	066500	DT13	
3049	005052	067240	DF20	
3050			;ERROR 137	
3051	005054	061111	EM39	;CYL DIFF/OFFSET IN A2 NOT CLEARED
3052	005056	064744	DH25	
3053	005060	066560	DT14	
3054	005062	067314	DF22	
3055			;ERROR 140	
3056	005064	061146	EM40	;CYL ADDR IN B2 NOT CLEARED

3057	005066	064744	DH25	
3058	005070	066560	DT14	
3059	005072	067314	DF22	
3060			; ERROR 141	
3061	005074	061771	EM52	; 20 SECTOR FORMAT NOT SET IN RKMR2
3062	005076	063767	DH1	
3063	005100	066076	DT1	
3064	005102	066744	DF1	
3065			; ERROR 142	
3066	005104	062030	EM53	; SECTOR 0 NOT FOUND WITHIN 50 MS
3067	005106	063767	DH1	
3068	005110	066076	DT1	
3069	005112	066744	DF1	
3070			; ERROR 143	
3071	005114	062061	EM54	; DIFF SECTOR NOT FOUND WITHIN 3MS
3072	005116	063767	DH1	
3073	005120	066076	DT1	
3074	005122	066744	DF1	
3075			; ERROR 144	
3076	005124	061517	EM48	; MSG B3 ERROR
3077	005126	065122	DH34	; SECTOR REG UNSTABLE
3078	005130	066076	DT1	
3079	005132	067070	DF10	
3080			; ERROR 145	
3081	005134	061517	EM48	
3082	005136	065146	DH35	; BETWEEN SECTOR COUNTS
3083	005140	066204	DT6	
3084	005142	067110	DF12	
3085			; ERROR 146	
3086	005144	060041	EM19	; MSG A1 ERROR
3087	005146	064446	DH14	; AFTER SEEK WITH BAD PARITY
3088	005150	066560	DT14	
3089	005152	067314	DF22	
3090			; ERROR 147	
3091	005154	060056	EM20	; MSG B1 ERROR
3092	005156	064446	DH14	
3093	005160	066560	DT14	
3094	005162	067314	DF22	
3095			; ERROR 150	
3096	005164	060041	EM19	; MSG A1 ERROR
3097	005166	065217	DH37	
3098	005170	066076	DT1	
3099	005172	067070	DF10	
3100			; ERROR 151	
3101	005174	057630	EM12	; NO RDY
3102	005176	064675	DH22	; AFTER CLEAR CMD
3103	005200	066076	DT1	
3104	005202	067070	DF10	
3105			; ERROR 152	
3106	005204	063743	EM90	; DDT NOT SET IN RKDS
3107	005206	063767	DH1	
3108	005210	066076	DT1	
3109	005212	066744	DF1	
3110			; ERROR 153	
3111	005214	000000	0	
3112	005216	000000	0	

3113	005220	000000	0	
3114	005222	000000	0	
3115			; ERROR 154	
3116	005224	062114	EM55	; ATTN NOT CLEARED
3117	005226	064675	DH22	
3118	005230	066076	DT1	
3119	005232	067070	DF10	
3120			; ERROR 155	
3121	005234	063404	EM85	; IDAE NOT CLEARED
3122	005236	064540	DH17	; AFTER RECAL CMD
3123	005240	066076	DT1	
3124	005242	067070	DF10	
3125			; ERROR 156	
3126	005244	057630	EM12	; CONT NOT READY
3127	005246	065747	DH51	; AFTER SEEK TO SELF
3128	005250	066076	DT1	
3129	005252	067070	DF10	
3130			; ERROR 157	
3131	005254	057661	EM13	; NO ATTN
3132	005256	065747	DH51	
3133	005260	066076	DT1	
3134	005262	067070	DF10	
3135			; ERROR 160	
3136	005264	062274	EM59	; LIMIT DETECT NOT FOUND
3137	005266	063767	DH1	
3138	005270	066076	DT1	
3139	005272	066744	DF1	
3140			; ERROR 161	
3141	005274	060007	EM17	; MSG A0 ERROR
3142	005276	065263	DH38	; AFTER LIMIT DETECT
3143	005300	066500	DT13	
3144	005302	067240	DF20	
3145			; ERROR 162	
3146	005304	060024	EM18	; MSG B0 ERROR
3147	005306	065263	DH38	
3148	005310	066500	DT13	
3149	005312	067240	DF20	
3150			; ERROR 163	
3151	005314	060041	EM19	; MSG A1 ERROR
3152	005316	065263	DH38	
3153	005320	066500	DT13	
3154	005322	067240	DF20	
3155			; ERROR 164	
3156	005324	060056	EM20	; MSG B1 ERROR
3157	005326	065263	DH38	
3158	005330	066500	DT13	
3159	005332	067240	DF20	
3160			; ERROR 165	
3161	005334	057661	EM13	; NO ATTN
3162	005336	065263	DH38	
3163	005340	066076	DT1	
3164	005342	067070	DF10	
3165			; ERROR 166	
3166	005344	062325	EM60	; HEADS HOME NOT FOUND
3167	005346	065263	DH38	
3168	005350	066076	DT1	

3169	005352	067070		DF10	
3170			;ERROR	167	
3171	005354	062361		EM61	;LOAD HEADS NOT FOUND
3172	005356	065263		DH38	
3173	005360	066076		DT1	
3174	005362	067070		DF10	
3175			;ERROR	170	
3176	005364	057153		EM4	;FATAL ERROR
3177	005366	065515		DH45	;LIMIT DETECT ERROR ON PREVIOUS TEST
3178	005370	066076		DT1	
3179	005372	067174		DF16	
3180			;ERROR	171	
3181	005374	057630		EM12	;NO RDY
3182	005376	065027		DH30	;AFTER READ HEADER CMD
3183	005400	066076		DT1	
3184	005402	067070		DF10	
3185			;ERROR	172	
3186	005404	061111		EM39	;CYL DIFF/OFFSET NOT CLEARED
3187	005406	065027		DH30	;AFTER READ HEADER CMD
3188	005410	066560		DT14	
3189	005412	067314		DF22	
3190			;ERROR	173	
3191	005414	062415		EM63	;DLT SET
3192	005416	065027		DH30	
3193	005420	066076		DT1	
3194	005422	067154		DF15	
3195			;ERROR	174	
3196	005424	060073		EM21	;CERR SET
3197	005426	065027		DH30	
3198	005430	066076		DT1	
3199	005432	067154		DF15	
3200			;ERROR	175	
3201	005434	061111		EM39	;CYL DIFF NOT CLEARED
3202	005436	064250		DH10	;AT END OF HEAD LOADING
3203	005440	066560		DT14	
3204	005442	067314		DF22	
3205			;ERROR	176	
3206	005444	061146		EM40	;CYL ADDR NOT CLEARED.
3207	005446	064250		DH10	
3208	005450	066560		DT14	
3209	005452	067314		DF22	
3210			;ERROR	177	
3211	005454	062713		EM72	;FORMAT TEST BYPASSED
3212	005456	065574		DH46	;COULD NOT READ BSE INFO
3213	005460	066076		DT1	
3214	005462	067174		DF16	
3215			;ERROR	200	
3216	005464	057630		EM12	;NO RDY
3217	005466	065301		DH39	;AFTER WRITE HEADER CMD
3218	005470	066076		DT1	
3219	005472	067154		DF15	
3220			;ERROR	201	
3221	005474	060073		EM21	;CERR SET
3222	005476	065301		DH39	
3223	005500	066076		DT1	
3224	005502	067154		DF15	

3225			;ERROR 202	
3226	005504	062465	EM65	;READ HEADER ERROR
3227	005506	063767	DH1	
3228	005510	066250	DT7	
3229	005512	067134	DF14	
3230			;ERROR 203	
3231	005514	060007	EM17	;MSG A0 ERROR
3232	005516	065102	DH33	;DURING SEEK CMD
3233	005520	066500	DT13	
3234	005522	067240	DF20	
3235			;ERROR 204	
3236	005524	060024	EM18	;MSG B0 ERROR
3237	005526	065102	DH33	
3238	005530	066500	DT13	
3239	005532	067240	DF20	
3240			;ERROR 205	
3241	005534	060041	EM19	;MSG A1 ERROR
3242	005536	065102	DH33	
3243	005540	066500	DT13	
3244	005542	067240	DF20	
3245			;ERROR 206	
3246	005544	060056	EM20	;MSG B1 ERROR
3247	005546	065102	DH33	
3248	005550	066500	DT13	
3249	005552	067240	DF20	
3250			;ERROR 207	
3251	005554	060740	EM36	;CYL ADDR IN RKMR3 INCORRECT
3252	005556	064744	DH25	;AFTER SEEK CMD
3253	005560	066136	DT4	
3254	005562	067044	DF6	
3255			;ERROR 210	
3256	005564	060073	EM21	;CERR SET
3257	005566	064744	DH25	
3258	005570	066076	DT1	
3259	005572	067070	DF10	
3260			;ERROR 211	
3261	005574	062541	EM67	;READ CYL 0 HEADERS ON CYL 1
3262	005576	064744	DH25	
3263	005600	066076	DT1	
3264	005602	067070	DF10	
3265			;ERROR 212	
3266	005604	061040	EM38	;CYL DIFF IN RKMR2 NOT = CALDIF
3267	005606	065102	DH33	;DURING SEEK CMD
3268	005610	066136	DT4	
3269	005612	067044	DF6	
3270			;ERROR 213	
3271	005614	060007	EM17	;MSG A0 ERROR
3272	005616	065364	DH41	;DURING RECAL CMD
3273	005620	066500	DT13	
3274	005622	067240	DF20	
3275			;ERROR 214	
3276	005624	060024	EM18	;MSG B0 ERROR
3277	005626	065364	DH41	
3278	005630	066500	DT13	
3279	005632	067240	DF20	
3280			;ERROR 215	

3281	005634	060041	EM19	;MSG A1 ERROR
3282	005636	065364	DH41	
3283	005640	066500	DT13	
3284	005642	067240	DF20	
3285			;ERROR 216	
3286	005644	060056	EM20	;MSG B1 ERROR
3287	005646	065364	DH41	
3288	005650	066500	DT13	
3289	005652	067240	DF20	
3290			;ERROR 217	
3291	005654	061040	EM38	;CYL DIFF IN RKMR2 NOT=CALDIF
3292	005656	065364	DH41	
3293	005660	066136	DT4	
3294	005662	067044	DF6	
3295			;ERROR 220	
3296	005664	060073	EM21	;CERR SET
3297	005666	064540	DH17	;AFTER RECAL CMD
3298	005670	066076	DT1	
3299	005672	067070	DF10	
3300			;ERROR 221	
3301	005674	060007	EM17	;MSG A0 ERROR
3302	005676	064540	DH17	
3303	005700	066500	DT13	
3304	005702	067240	DF20	
3305			;ERROR 222	
3306	005704	060041	EM19	;MSG A1 ERROR
3307	005706	064540	DH17	
3308	005710	066500	DT13	
3309	005712	067240	DF20	
3310			;ERROR 223	
3311	005714	061111	EM39	;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
3312	005716	064540	DH17	
3313	005720	066076	DT1	
3314	005722	067070	DF10	
3315			;ERROR 224	
3316	005724	062505	EM66	;CYL ADDR IN RKMR3 INCORRECT
3317	005726	064540	DH17	
3318	005730	066076	DT1	
3319	005732	067070	DF10	
3320			;ERROR 225	
3321	005734	062600	EM68	;READING CYL 1 HEADERS ON CYL 0
3322	005736	064540	DH17	
3323	005740	066076	DT1	
3324	005742	067070	DF10	
3325			;ERROR 226	
3326	005744	057630	EM12	;NO RDY
3327	005746	064763	DH26	;AFTER READ DATA CMD
3328	005750	066076	DT1	
3329	005752	067070	DF10	
3330			;ERROR 227	
3331	005754	060073	EM21	;CERR SET
3332	005756	064763	DH26	
3333	005760	066076	DT1	
3334	005762	067154	DF15	
3335			;ERROR 230	
3336	005764	063522	EM87	;CANT READ BSE INFO

3337	005766	066026	DH53	; ON SECT 10,12,14,16,18,20
3338	005770	066076	DT1	
3339	005772	067214	DF17	
3340			; ERROR 231	
3341	005774	000000	0	
3342	005776	000000	0	
3343	006000	000000	0	
3344	006002	000000	0	
3345			; ERROR 232	
3346	006004	060740	EM36	; CYL ADDR IN RKMR3 NOT=RKDC
3347	006006	064744	DH25	; AFTER SEEK CMD
3348	006010	066076	DT1	
3349	006012	067024	DF5	
3350			; ERROR 233	
3351	006014	063522	EM87	; CANNOT READ BSE INFO
3352	006016	065405	DH42	; ON SECT 0,2,4,6,8
3353	006020	066076	DT1	
3354	006022	067214	DF17	
3355			; ERROR 234	
3356	006024	000000	0	
3357	006026	000000	0	
3358	006030	000000	0	
3359	006032	000000	0	
3360			; ERROR 235	
3361	006034	062637	EM69	; ALIGN CARTRIDGE USED
3362	006036	065445	DH44	; WILL BYPASS FORMAT & ALL R/W TESTS
3363	006040	066076	DT1	
3364	006042	067070	DF10	
3365			; ERROR 236	
3366	006044	062147	EM56	; UNEXP MEM PARITY TRAP
3367	006046	064723	DH23	; TEST #, TRAP PC
3368	006050	066474	DT11	
3369	006052	067004	DF3	
3370			; ERROR 237	
3371	006054	062672	EM71	; DSC SET
3372	006056	064675	DH22	; AFTER DRIVE CLEAR CMD
3373	006060	066076	DT1	
3374	006062	067070	DF10	
3375			; ERROR 240	
3376	006064	062600	EM68	; READ CYL 1 HEADERS ON CYL 0
3377	006066	064540	DH17	; AFTER RECAL CMD
3378	006070	066076	DT1	
3379	006072	067070	DF10	
3380			; ERROR 241	
3381	006074	060740	EM36	; RKMR3 NOT = RKDC
3382	006076	064446	DH14	; AFTER SEEK WITH BAD PARITY
3383	006100	066316	DT8	
3384	006102	067044	DF6	
3385			; ERROR 242	
3386	006104	061040	EM38	; CYL DIFF IN RKMR2 INCORRECT
3387	006106	064446	DH14	
3388	006110	066316	DT8	
3389	006112	067044	DF6	
3390			; ERROR 243	
3391			EM36	
3392	006114	060740		; CYL ADDR IN RKMR3 INCORRECT

3393	006116	064744	DH25	; AFTER SEEK CMD
3394	006120	066316	DT8	
3395	006122	067044	DF6	
3396			; ERROR 244	
3397	006124	062761	EM74	; RTZ NOT SET
3398	006126	065364	DH41	; DURING RECAL CMD
3399	006130	066076	DT1	
3400	006132	067070	DF10	
3401			; ERROR 245	
3402	006134	057661	EM13	; NO ATTN
3403	006136	065641	DH48	; AFTER SEEK TO INVALID CYL
3404	006140	066076	DT1	
3405	006142	067070	DF10	
3406			; ERROR 246	
3407	006144	063006	EM75	; IDAE NOT SET
3408	006146	065641	DH48	
3409	006150	066136	DT4	
3410	006152	067044	DF6	
3411			; ERROR 247	
3412	006154	060552	EM32	; FAULT NOT SET
3413	006156	065641	DH48	
3414	006160	066136	DT4	
3415	006162	067044	DF6	
3416			; ERROR 250	
3417	006164	063034	EM76	; PIP SET
3418	006166	063641	DH48	
3419	006170	066136	DT4	
3420	006172	067044	DF6	
3421			; ERROR 251	
3422	006174	057762	EM16	; DSC NOT SET
3423	006176	065641	DH48	
3424	006200	066136	DT4	
3425	006202	067044	DF6	
3426			; ERROR 252	
3427	006204	060007	EM17	; MSG A0 ERROR
3428	006206	065641	DH48	
3429	006210	066500	DT13	
3430	006212	067240	DF20	
3431			; ERROR 253	
3432	006214	060024	EM18	; MSG B0 ERROR
3433	006216	065641	DH48	
3434	006220	066500	DT13	
3435	006222	067240	DF20	
3436			; ERROR 254	
3437	006224	060041	EM19	; MSG A1 ERROR
3438	006226	065641	DH48	
3439	006230	066500	DT13	
3440	006232	067240	DF20	
3441			; ERROR 255	
3442	006234	060056	EM20	; MSG B1 ERROR
3443	006236	065641	DH48	
3444	006240	066500	DT13	
3445	006242	067240	DF20	
3446			; ERROR 256	
3447	006244	061040	EM38	; CYL DIFF IN RKMR2 NOT='CYL DIF'
3448	006246	065641	DH48	

3449	006250	066136	DT4	
3450	006252	067044	DF6	
3451			; ERROR 257	
3452	006254	060740	EM36	; CYL ADDR IN RKMR3 NOT=RKDC
3453	006256	065641	DH48	
3454	006260	066136	DT4	
3455	006262	067044	DF6	
3456			; ERROR 260	
3457	006264	000000	0	
3458	006266	000000	0	
3459	006270	000000	0	
3460	006272	000000	0	
3461			; ERROR 261	
3462	006274	000000	0	
3463	006276	000000	0	
3464	006300	000000	0	
3465	006302	000000	0	
3466			; ERROR 262	
3467	006304	063055	EM77	; FAULT NOT CLEARED
3468	006306	064675	DH22	; AFTER DRIVE CLEAR CMD
3469	006310	066076	DT1	
3470	006312	067070	DF10	
3471			; ERROR 263	
3472	006314	063103	EM78	; CYL DIFF IN RKMR2 NOT=1 IN SEEK TO SELF
3473	006316	064446	DH14	; AFTER SEEK WITH BAD PARITY
3474	006320	066316	DT8	
3475	006322	067044	DF6	
3476			; ERROR 264	
3477	006324	061174	EM41	; CYL ADDR NOT CLEARED
3478	006326	065027	DH30	; AFTER READ HEADER CMD
3479	006330	066560	DT14	
3480	006332	067314	DF22	
3481			; ERROR 265	
3482	006334	060024	EM18	; MSG B0 ERROR
3483	006336	064675	DH22	; AFTER DRIVE CLEAR CMD
3484	006340	066500	DT13	
3485	006342	067240	DF20	
3486			; ERROR 266	
3487	006344	060056	EM20	; MSG B1 ERROR
3488	006346	064675	DH22	
3489	006350	066500	DT13	
3490	006352	067240	DF20	
3491			; ERROR 267	
3492	006354	060024	EM18	
3493	006356	065301	DH39	; AFTER WRITE HEADER CMD
3494	006360	066500	DT13	
3495	006362	067240	DF20	
3496			; ERROR 270	
3497	006364	060056	EM20	
3498	006366	065301	DH39	
3499	006370	066500	DT13	
3500	006372	067240	DF20	
3501			; ERROR 271	
3502	006374	060024	EM18	
3503	006376	065027	DH30	; AFTER READ HEADER CMD
3504	006400	066500	DT13	

3505	006402	067240		DF20	
3506			; ERROR	272	
3507	006404	060056		EM20	
3508	006406	065027		DH30	
3509	006410	066500		DT13	
3510	006412	067240		DF20	
3511			; ERROR	273	
3512	006414	060007		EM17	; MSG A0 ERROR
3513	006416	064675		DH22	; AFTER DRV CLR CMD
3514	006420	066500		DT13	
3515	006422	067240		DF20	
3516			; ERROR	274	
3517	006424	060041		EM19	; MSG A1 ERROR
3518	006426	064675		DH22	
3519	006430	066500		DT13	
3520	006432	067240		DF20	
3521			; ERROR	275	
3522	006434	060024		EM18	; MSG B0 ERROR
3523	006436	064540		DH17	; AFTER RECAL CMD
3524	006440	066500		DT13	
3525	006442	067240		DF20	
3526			; ERROR	276	
3527	006444	060056		EM20	; MSG B1 ERROR
3528	006446	064540		DH17	
3529	006450	066500		DT13	
3530	006452	067240		DF20	
3531			; ERROR	277	
3532	006454	060007		EM17	; MSG A0 ERROR
3533	006456	065301		DH39	; AFTER WRITE HEADER CMD
3534	006460	066500		DT13	
3535	006462	067240		DF20	
3536			; ERROR	300	
3537	006464	060041		EM19	; MSG A1 ERROR
3538	006466	065301		DH39	
3539	006470	066500		DT13	
3540	006472	067240		DF20	
3541			; ERROR	301	
3542	006474	060007		EM17	
3543	006476	065027		DH30	; AFTER READ HEADER CMD
3544	006500	066500		DT13	
3545	006502	067240		DF20	
3546			; ERROR	302	
3547	006504	060041		EM19	
3548	006506	065027		DH30	
3549	006510	066500		DT13	
3550	006512	067240		DF20	
3551			; ERROR	303	
3552	006514	061111		EM39	; CYL DIFF/OFFSET NOT CLEARED
3553	006516	065301		DH39	; AFTER WRITE HEADER CMD
3554	006520	066560		DT14	
3555	006522	067314		DF22	
3556			; ERROR	304	
3557	006524	061174		EM41	; CYL ADDR NOT CLEARED
3558	006526	065301		DH39	
3559	006530	066560		DT14	
3560	006532	067314		DF22	

3561					
3562			; ERROR 305		
3563	006534	063211	EM80	; UNLD NOT SET	
3564	006536	065263	DH38	; AFTER LIMIT DETECT	
3565	006540	066076	DT1		
3566	006542	067070	DF10		
3567			; ERROR 306		
3568	006544	063241	EM81	; SPIN NOT SET	
3569	006546	064204	DH9	; AFTER START SPIN CMD.	
3570	006550	066076	DT1		
3571	006552	067070	DF10		
3572			; ERROR 307		
3573	006554	063267	EM82	; RTZ NOT SET	
3574	006556	065364	DH41	; DURING RECAL CMD	
3575	006560	066076	DT1		
3576	006562	067070	DF10		
3577			; ERROR 310		
3578	006564	063314	EM83	; READ HEADER ERROR	
3579	006566	063767	DH1		
3580	006570	066364	DT9		
3581	006572	067274	DF21		
3582			; ERROR 311		
3583	006574	063314	EM83		
3584	006576	063767	DH1		
3585	006600	066430	DT10		
3586	006602	067274	DF21		
3587			; ERROR 312		
3588	006604	063354	EM84	; FORMAT NOT SET	
3589	006606	065301	DH39	; AFTER WRITE HEADER CMD	
3590	006610	066076	DT1		
3591	006612	067070	DF10		
3592			; ERROR 313		
3593	006614	063354	EM84		
3594	006616	065027	DH30	; AFTER READ HEADER CMD	
3595	006620	066076	DT1		
3596	006622	067070	DF10		
3597			; ERROR 314		
3598	006624	062172	EM57	; WCE AT CYL 411, TRK 2, SEC 21	
3599	006626	063767	DH1		
3600	006630	066076	DT1		
3601	006632	067010	DF4		
3602			; ERROR 315		
3603	006634	062226	EM58	; SPOK NOT CLEARED	
3604	006636	064560	DH18	; AFTER UNLD CMD	
3605	006640	066076	DT1		
3606	006642	067070	DF10		
3607			; ERROR 316		
3608	006644	062657	EM70	; UNEXP ATTN	
3609	006646	064204	DH9	; AFTER START SPIN CMD	
3610	006650	066076	DT1		
3611	006652	067070	DF10		
3612			; ERROR 317		
3613	006654	062657	EM70		
3614	006656	064277	DH11	; AFT SPIN CMD & FWD SET	
3615	006660	066076	DT1		
3616	006662	067070	DF10		

3617			;ERROR 320	
3618	006664	062657	EM70	
3619	006666	064336	DH12	;AT INNER LIMIT FROM ST SPIN CMD
3620	006670	066076	DT1	
3621	006672	067070	DF10	
3622			;ERROR 321	
3623	006674	062657	EM70	
3624	006676	064377	DH13	;FROM OUTER LIM TO CYL 0 DURING LOADING
3625	006700	066076	DT1	
3626	006702	067070	DF10	
3627			;ERROR 322	
3628	006704	060041	EM19	;MSG A1 ERROR
3629	006706	064476	DH16	;AFT LD HEAD REG & SEEK
3630	006710	066500	DT13	
3631	006712	067240	DF20	
3632			;ERROR 323	
3633	006714	060056	EM20	;MSG B1 ERROR.
3634	006716	064476	DH16	
3635	006720	066500	DT13	
3636	006722	067240	DF20	
3637			;ERROR 324	
3638	006724	061471	EM46	;MSG A2 ERROR
3639	006726	064476	DH16	
3640	006730	066560	DT14	
3641	006732	067314	DF22	
3642			;ERROR 325	
3643	006734	061504	EM47	;MSG B2 ERROR
3644	006736	064476	DH16	
3645	006740	066560	DT14	
3646	006742	067314	DF22	
3647			;ERROR 326	
3648	006744	062740	EM73	;CTO SET
3649	006746	063435	EM86	;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
3650	006750	066076	DT1	
3651	006752	066760	DF2	
3652			;ERROR 327	
3653	006754	063170	EM79	;NED SET
3654	006756	063435	EM86	
3655	006760	066076	DT1	
3656	006762	066760	DF2	
3657			;ERROR 330	
3658			EM5	;MDS SET
3659	006764	057266	EM86	
3660	006766	063435	DT1	
3661	006770	066076	DF2	
3662	006772	066760		


```

3663
3664
3665
3666 006774 012737 000001 001336 PARSRT: MOV #1,PARAM ;SET FLAG FOR 220 START
3667 007002 005037 001340 CLR BYPT16
3668 007006 005037 001342 CLR MODTST
3669 007012 000450 BR PRGSRT ;START PROGRAM
3670
3671 007014 005037 001336
3672 007020 012737 000001 001340 BYT16: CLR PARAM ;SET FLAG TO BYPASS TEST 16
3673 007026 005037 001342 MOV #1,BYPT16
3674 007032 000440 CLR MODTST
3675 BR PRGSRT
3676 007034 012737 000001 001336 BYT16A: MOV #1,PARAM
3677 007042 012737 000001 001340 MOV #1,BYPT16
3678 007050 005037 001342 CLR MODTST
3679 007054 000427 BR PRGSRT
3680
3681 007056 005037 001336
3682 007062 005037 001340 MDTST: CLR PARAM
3683 007066 012737 000001 001342 CLR BYPT16
3684 007074 000417 MOV #1,MDTST
3685 BR PRGSRT
3686 007076 005037 001336
3687 007102 012737 000001 001340 MDTSTA: CLR PARAM
3688 007110 012737 000001 001342 MOV #1,BYPT16
3689 007116 000406 BR PRGSRT
3690 007120 005037 001336 START: CLR PARAM ;CLEAR FOR 200 START
3691 007124 005037 001340 CLR BYPT16
3692 007130 005037 001342 CLR MODTST
3693 007134 000005
3694 007136 012706 001100 PRGSRT: RESET ;CLEAR ALL INT ENABLE & INIT
3695 007142 012746 000000 MOV #STACK,SP ;SETUP STACK POINTER
3696 007146 012746 007154 MOV #PRO,-(SP) ;PSW LOADED TO BE
3697 007152 000002 RTI ;LSI-11 COMPATABLE
3698
3699 007154 004737 052156 1S: JSR PC,$TKINT ;ENABLE ALL INTERRUPTS
3700
3701
3702
3703 ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3704 ;*** ANY DEVICE WHICH SETS ITS ***
3705 ;*** INTERRUPT ENABLE BIT WILL ***
3706 ;*** SERVICED. ***
3707
3708 ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'ST5')
3709 ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3710 ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3711
3712 ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3713
3714 ;SYSMAC 'SETUP'
3715 .SBTTL INITIALIZE THE COMMON TAGS
3716 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3717 007160 012706 001100 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
3718 007164 005026 CLR (R6)+ ;;CLEAR MEMORY LOCATION

```

```

3719 007166 022706 001140      CMP      #SWR,R6 ;;DONE?
3720 007172 001374              BNE      .-6      ;;LOOP BACK IF NO
3721 007174 012706 001100      MOV      #STACK,SP ;;SETUP THE STACK POINTER
3722              ;;INITIALIZE A FEW VECTORS
3723 007200 012737 050264 000020      MOV      #SSCOPE,#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3724 007206 012737 000340 000022      MOV      #340,#IOTVEC+2 ;;LEVEL 7
3725 007214 012737 050544 000030      MOV      #ERROR,#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3726 007222 012737 000340 000032      MOV      #340,#EMTVEC+2 ;;LEVEL 7
3727 007230 012737 054262 000034      MOV      #STRAP,#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3728 007236 012737 000340 000036      MOV      #340,#TRAPVEC+2;LEVEL 7
3729 007244 012737 050176 000024      MOV      #SPWRON,#PWRVEC ;;POWER FAILURE VECTOR
3730 007252 012737 000340 000026      MOV      #340,#PWRVEC+2 ;;LEVEL 7
3731 007260 013737 042756 042750      MOV      SENDCT,SEOPCT ;;SETUP END-OF-PROGRAM COUNTER
3732 007266 005037 001174              CLR      STIMES      ;;INITIALIZE NUMBER OF ITERATIONS
3733 007272 005037 001176              CLR      SESCAPE     ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3734 007276 112737 000001 001115      MOV      #1,SEMAX    ;;ALLOW ONE ERROR PER TEST
3735 007304 012737 007304 001106      MOV      #.,SLPADR   ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3736 007312 012737 007312 001110      MOV      #.,SLPERR   ;;SETUP THE ERROR LOOP ADDRESS
3737              ;;SIZE FOR A HARDWARE SWITCH REGISTER, IF NOT FOUND OR IT IS
3738              ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
3739 007320 013746 000004              MOV      #ERRVEC,-(SP) ;;SAVE ERROR VECTOR
3740 007324 012737 007360 000004      MOV      #64S,#ERRVEC ;;SET UP ERROR VECTOR
3741 007332 012737 177570 001140      MOV      #DSWR,SWR   ;;SETUP FOR A HARDWARE SWICH REGISTER
3742 007340 012737 177570 001142      MOV      #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
3743 007346 022777 177777 171564      CMP      #-1,#SWR   ;;TRY TO REFERENCE HARDWARE SWR
3744 007354 001012              BNE      66S        ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3745              ;;AND THE HARDWARE SWR IS NOT = -1
3746 007356 000403              BR       65S        ;;BRANCH IF NO TIMEOUT
3747 007360 012716 007366 64S:      MOV      #65S,(SP)  ;;SET UP FOR TRAP RETURN
3748 007364 000002              RTI
3749 007366 012737 000176 001140 65S:      MOV      #SWREG,SWR ;;POINT TO SOFTWARE SWR
3750 007374 012737 000174 001142      MOV      #DISPREG,DISPLAY
3751 007402 012637 000004 66S:      MOV      (SP)+,#ERRVEC ;;RESTORE ERROR VECTOR
3752
3753 007406 005037 001216              CLR      SPASS      ;;CLEAR PASS COUNT
3754 007412 132737 000200 001231      BITB    #APTSIZE,SENVM ;;TEST USER SIZE UNDER APT
3755 007420 001403              BEQ     67S        ;;YES,USE NON-APT SWITCH
3756 007422 012737 001232 001140      MOV     #SSWREG,SWR ;;NO,USE APT SWITCH REGISTER
3757 007430 67S:
3758 007430 012737 007474 000004 MEMPAR: MOV     #1S,ERRVEC  ;;SET TIMEOUT VECTOR
3759 007436 012737 000340 000006      MOV     #PR7,ERRVEC+2
3760
3761 007444 012701 172100              MOV     #MEMBAS,R1  ;;ADDR OF MEM CSR
3762 007450 005011 3S:      CLR     (R1)        ;;SEE IF CAN REFERENCE
3763 007452 012711 000001              MOV     #1,(R1)    ;;SET ENABLE BIT IF YES
3764 007456 012737 050114 000114      MOV     #MEMERR,MEMVEC ;;LOAD MEMORY CHECK VECTOR IF NO TIMEOUT
3765 007464 012737 000340 000116      MOV     #PR7,MEMVEC+2
3766 007472 000401              BR      2S
3767
3768 007474 022626 1S:      CMP     (SP)+,(SP)+ ;;ADJ STACK
3769 007476 062701 000002 2S:      ADD     #2,R1      ;;TRY NEXT CSR
3770 007502 020127 172140      CMP     R1,#MEMBAS+40 ;;SEE IF TRIED ALL
3771 007506 001360              BNE    3S          ;;BR IF NO
3772 007510 012737 000006 000004      MOV     #ERRVEC+2,ERRVEC ;;RESTORE TRAP CATCHER
3773 007516 005037 000006              CLR     ERRVEC+2
3774

```

```

3775 007522 004737 043076      JSR    PC,CLRFLG      ;CLEAR DDUMP THRU SIZFLG
3776 007526 005037 001220      CLR    $DEVCT
3777 007532 005037 001222      CLR    $UNIT
3778
3779
3780      ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
3781
3782
3783 007536 005737 000042      START1: TST    42
3784 007542 001015 000000      BNE    1$           ;BR IF AUTO
3785 007544 004737 043116      JSR    PC,TITLE     ;MANUAL, TYPE PROG ID
3786 007550 123727 000041 000013  CMPB   41,#13       ;13=LOADED BY XXDP
3787 007556 001011 000000      BNE    2$
3788 007560 005237 003432      INC    DDUMP        ;SET RK06 DUMP MODE FLAG
3789 007564 104401 055405      TYPE  ,MSG2         ;REPLACE DRO PACK W/SCRATCH & DO<CR>
3790 007570 000000 000000      HALT
3791 007572 000137 007606      JMP    ST2          ;HALT HERE
3792 007576 000137 007652      1$:   JMP    ST3
3793 007602 005237 003440      2$:   INC    PPTP        ;SET ACT/APT/PTP DUMP MODE FLAG
3794
3795
3796      ;CHECK IF ALL PARAMETERS DEFAULTED. IF NOT, BEGIN INPUT DIALOGUE
3797      ;WITH OPERATOR. THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
3798      ;DRIVE NOS. SEPERATED BY COMMAS & TERMINATED BY <CR>
3799      ;EX:      DRIVES TO BE TESTED: 1,2,4<CR>
3800
3801
3802 007606 005737 001336      ST2:   TST    PARAM
3803 007612 001002 000000      BNE    1$           ;BR IF 220 START
3804 007614 000137 007704      JMP    ST4          ;200 START, DEFAULT & SIZE THE BUSS
3805 007620 104401 055466      1$:   TYPE  MSG3       ;DRIVES TO BE TESTED
3806 007624 004737 043176      JSR    PC,GDRVS     ;GET DR NOS.
3807 007630 104401 055520      TYPE  MSG4         ;BUSS ADDR
3808 007634 004737 043336      JSR    PC,GBA       ;GET BA
3809 007640 104401 055546      TYPE  MSG5         ;CONT INT VECTOR
3810 007644 004737 043364      JSR    PC,GINT      ;GET INT VECTOR
3811 007650 000427 000000      BR     ST5
3812
3813
3814      ;AUTO MODE
3815      ;CHECK IF LOADED BY XXDP OR OTHER. SET FLAGS & NO INPUT DIALOGUE.
3816      ;DEFAULT ALL PARAMETERS. TEST ONLY THOSE DRIVES THAT ARE READY
3817      ;ON THE BUSS
3818
3819
3820 007652 123727 000041 000013  ST3:   CMPB   41,#13       ;13=LOADED BY XXDP
3821 007660 001007 000000      BNE    1$
3822 007662 005237 003434      INC    DDPCH        ;SET RK06 CHAIN MODE FLAG
3823 007666 004737 043116      JSR    PC,TITLE
3824 007672 104401 055611      TYPE  MSG7         ;DRO NOT TSTD
3825 007676 000402 000000      BR     ST4
3826 007700 005237 003436      1$:   INC    ACT11     ;SET ACT AUTO FLAG.
3827
3828 007704 012737 177440 001264  ST4:   MOV    #177440,$BASE ;DEFAULT VALUE
3829 007712 012737 000210 001314  MOV    #210,RKVEC   ;DEFAULT VALUE
3830 007720 004737 043416      JSR    PC,SETINT

```

```

3831 007724 005237 003472          INC      SIZFLG          ;DO "SIZE THE BUSS" TEST
3832
3833 007730 005037 003304          STS:    CLR      UNLD          ;INITIALIZE FLAGS
3834 007734 005037 003306          CLR      BADHDR        ;USED IN 'STOP ROUTINE
3835 007740 005037 003310          CLR      HPEND         ;FOR VALID PROGRAM HALTS
3836 007744 005037 001176          CLR      $ESCAPE
3837 007750 005037 001170          CLR      $TMP4         ;CLR RK06 IND. FLAG
3838 007754 012737 003444          MOV      #DRIVO,DRVPT ;SETUP
001346                                ;NO. OF DRVS DONE
3839 007762 005037 001220          CLR      $DEVCT        ;CURRENT DRV UNDER TEST
3840 007766 005037 001222          CLR      $UNIT         ;SETUP TIMEOUT ERROR VECTOR
3841 007772 012737 010040          MOV      #1$,ERRVEC    ;SEE IF L-CLOCK THERE
000004                                ;PRESENT, SET FLAG.
3842 010000 005777 171322          TST      $LKS          ;VECTOR ADDR
3843 010004 005237 003464          INC      LCLKF         ;SEE IF P-CLOCK THERE
3844 010010 013700 001330          MOV      LCVEC,RO      ;PRESENT, SET FLAG
000004                                ;VECTOR ADDR
3845 010014 012737 010102          MOV      #2$,ERRVEC
3846 010022 005777 171272          TST      $PKS
3847 010026 005237 003466          INC      PCLKF
3848 010032 013700 001332          MOV      PCVEC,RO
3849 010036 000412          BR       3$
3850
3851 010040 022626          1$:    CMP      (SP)+,(SP)+ ;L-CLOCK NOT THERE, CLEAR STACK
000004                                ;4$,ERRVEC
3852 010042 012737 010106          MOV      #4$,ERRVEC
3853 010050 005777 171244          TST      $PKS          ;SEE IF P-CLOCK THERE
3854 010054 005237 003466          INC      PCLKF         ;PRESENT, SET FLAG
3855 010060 013700 001332          MOV      PCVEC,RO      ;VECTOR ADDR
3856 010064 005237 003470          3$:    INC      DOTIM     ;INDICATES TIMING TESTS CAN BE DONE
3857 010070 012720 047246          MOV      #CLOCK,(RO)+ ;SERVICE ROUTINE FOR CLOCKS
3858 010074 012710 000300          MOV      #PR6,(RO)
3859 010100 000407          BR       TST1         ;;GO TO NEXT TEST
3860
3861 010102 022626          2$:    CMP      (SP)+,(SP)+ ;P-CLOCK NOT THERE, CLEAR STACK
3862 010104 000767          BR       3$
3863
3864 010106 022626          4$:    CMP      (SP)+,(SP)+ ;NEITHER CLOCK THERE, CLEAR STACK
3865 010110 005037 003470          CLR      DOTIM         ;TIMING TESTS CANNOT BE DONE.
3866 010114 104401 056052          TYPE    ,MSG13        ;ALL TIMING TESTS BYPASSED
3867
3868

```

.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 MSG. ANY
* ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
* TESTS AND JUMPING TO 'END OF PASS'

```

↑ST1: SCOPE
MOV #1,STIMES ;:DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
MOV #PRO,-(SP) ;RESET PSW TO PRIORITY 0
MOV #SS,-(SP) ;& MAKE IT LSI COMPATABLE
RTI

SS:

MOV #IS,ERRVEC ;SETUP TIMEOUT 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

IS: 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 DRIVE FROM 0 THRU 7 IS ADDRESSED.
* CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE

3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881 010120 000004
3882 010122 012737 000001 001174
3883 010130 012706 001100
3884
3885 010134 012746 000000
3886 010140 012746 010146
3887 010144 000002
3888 010146
3889
3890 010146 012737 010272 000004
3891 010154 013705 001264
3892 010160 005765 000000
3893 010164 005765 000010
3894 010170 005765 000002
3895 010174 005765 000004
3896 010200 005765 000006
3897 010204 005765 000012
3898 010210 005765 000014
3899 010214 005765 000016
3900 010220 005765 000020
3901 010224 005765 000024
3902 010230 005765 000026
3903 010234 005765 000034
3904 010240 005765 000036
3905 010244 005765 000030
3906 010250 005765 000032
3907
3908 010254 012737 050026 000004
3909 010262 012737 000340 000006
3910 010270 000404
3911
3912 010272 022626
3913 010274 104007
3914 010276 000137 042722
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924

```

3925          : * DRIVE WILL BE TESTED AS AN RK06. IF SET, THE PROGRAM WILL BYPASS
3926          : * TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
3927          : * MDS, UFE OR NED BEING SET, OR BOTH NED & DRA RESET IN-
3928          : * DICATING THE OTHER PORT IS ACCESSED.
3929          : * IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.
3930          : *
3931          : * *****
3932 010302 000004 TST2: SCOPE
3933 010304 012737 000001 001174 MOV #1,STIMES ;DO 1 ITERATION
3934 010312 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
3935 010316 005237 001462 INC BYPCERR ;DO NOT TEST CERR IN 'FRDY'
3936
3937 010322 132737 000200 001231 BITB #BIT7,SENV ;SEE IF USE APT SELECTED DRIVES
3938 010330 001002 BNE 14$ ;BR IF YES
3939 010332 000137 010452 JMP 12$ ;ELSE DO NORM SIZING OR VERIFY
3940
3941 010336 104401 055732 14$: TYPE ,MSG10 ;WILL TEST DRIVES
3942 010342 005037 003442 CLR DRIVS ;# OF DRIVES PRESENT
3943 010346 005000 CLR RO ;DRV ADDR
3944 010350 012701 003444 MOV #DRIVO,R1 ;DRV FLAG
3945 010354 013702 001266 MOV $DEVN,R2 ;APT DEVICE MAP
3946
3947 010360 032702 000001 15$: BIT #BIT0,R2 ;SEE IF DRV IN DEVICE MAP
3948 010364 001410 BEQ 16$ ;BR IF NO
3949 010366 005237 003442 INC DRIVS ;ELSE INCR DRIVE COUNT
3950 010372 005211 INC (R1) ;& SET DRIVE PRESENT FLAG
3951 010374 104401 001205 TYPE ,SCLF
3952 010400 010046 MOV RO,-(SP) ;SAVE RO FOR TYPEOUT
3953 ;TYPE DRIVE #
3954 010402 104403 TYPOS ;GO TYPE--OCTAL ASCII
3955 010404 001 .BYTE 1 ;TYPE 1 DIGIT(S)
3956 010405 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
3957
3958 010406 005721 16$: TST (R1)+ ;ADV POINTER TO NEXT FLAG
3959 010410 005200 INC RO ;INC DRIVE #
3960 010412 022700 000010 CMP #8,R0 ;ALL 8 TESTED?
3961 010416 001402 BEQ 17$ ;BR IF YES
3962
3963 010420 006002 ROR R2 ;ELSE GET NEXT BIT OFF DEVICE MAP
3964 010422 000756 BR 15$ ;& TRY AGAIN
3965
3966 010424 005737 003442 17$: TST DRIVS ;SEE IF MORE DRIVES PRESENT
3967 010430 001402 BEQ 18$ ;BR IF NO
3968 010432 000137 011126 JMP VERIFY ;ELSE EXIT TEST & SETUP FOR RK07'S
3969
3970 010436 104126 18$: ERROR 126 ;NO DRIVES FOUND IN $DEVN
3971 010440 000000 HALT ;SETUP CORRECTLY & PRESS 'CONTINUE'
3972 010442 000137 007730 JMP STS ;TO TRY AGAIN
3973 010446 000137 011126 JMP VERIFY ;DO NOT SIZE, GO TO NEXT TEST
3974 010452 012765 000040 000010 20$: MOV #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3975 010460 013737 001414 003360 MOV T10,TEMP1
3976 010466 004737 043530 JSR PC,FRDY ;FIND RDY
3977 010472 104120 ERROR 120 ;RDY NOT SET BY END OF SCLR
3978 010474 005737 003472 TST SIZFLG ;SIZE BUS?
3979 010500 001762 BEQ 20$ ;BR IF NO
3980 010502 104401 055732 TYPE ,MSG10 ;WILL TEST DRIVES

```

```

3981 010506 005037 003442          CLR    DRIVS          ;# OF DRIVES PRESENT
3982 010512 005000                CLR    RO              ;DRV ADDR
3983 010514 012701 003444          MOV    #DRIVO,R1      ;DRV FLAG
3984 010520                1$:
3985 010520                SCOP1
3986 010522 104415 001100          MOV    #STACK,SP      ;RESTORE STK PTR
3987
3988 010526 012765 000040 000010  MOV    #SCLR,RKCS2(R5) ;SUBSYS CLEAR
3989 010534 013737 001414 003360  MOV    T10,TEMP1
3990 010542 004737 043530          JSR    PC,FRDY        ;FIND RDY
3991 010546 104120                ERROR 120             ;RDY NOT SET BY END OF SCLR
3992 010550 010065 000010          MOV    RO,RKCS2(R5)   ;SELECT THE DRIVE ADDR
3993 010554 012737 000001 003322  MOV    #SELDV,HCS1
3994 010562 053737 001170 003322  BIS    $TMP4,HCS1     ;ADD CDT IF RK07
3995 010570 013765 003322 000000  MOV    HCS1,RKCS1(R5) ;GET STATUS
3996 010576 013737 001426 003360  MOV    T50000,TEMP1
3997 010604 004737 044220          JSR    PC,DLY         ;DO DELAY TO CATCH MDS
3998 010610 013737 001414 003360  MOV    T10,TEMP1
3999 010616 004737 043530          JSR    PC,FRDY        ;FIND RDY
4000 010622 104117                ERROR 117             ;NO RDY AFTER SELECT DR. CMD
4001 010624 032737 100000 003322  BIT    #CERR,HCS1
4002 010632 001052                BNE
4003 010634 013737 003350 003360  MOV    HMR2,TEMP1
4004 010642 042737 177770 003360  BIC    #1<DRVMSK>,TEMP1
4005 010650 020037 003360          CMP    RO,TEMP1      ;S/B SAME
4006 010654 001020                BNE
4007 010656 005700                TST   RO
4008 010660 001003                BNE
4009 010662 005737 003434          TST   DDPCH          ;SEE IF XXDP CHAIN MODE
4010 010666 001016                BNE
4011 010670 005237 003442          4$: INC    DRIVS          ;INC DRIVE COUNT.
4012 010674 005211                INC    (R1)           ;SET DRIVE PRESENT FLAG
4013 010676 053711 001170          BIS    $TMP4,(R1)    ;ADD CDT IF SET
4014 010702 104401 001205          TYPE  $SCRLF
4015 010706 010046                MOV    RO,-(SP)      ;; SAVE RO FOR TYPEOUT
4016                ;; TYPE DR #
4017 010710 104403                TYPOS
4018 010712 001                .BYTE 1
4019 010713 000                .BYTE 0
4020 010714 000403                BR    5$
4021
4022 010716 004737 044236          3$: JSR    PC,BYP        ;TYPE BYPASS DR #
4023 010722 104001                ERROR 1              ;SELECTED DR # DOES NOT MATCH RKMR2 DR #
4024
4025 010724 005721                5$: TST    (R1)+        ;ADVANCE PTR TO NEXT DR. FLAG
4026 010726 005200                INC    RO             ;INC DR #
4027 010730 005037 001170          CLR    $TMP4         ;CLEAR RK07 FLAG FOR NEXT DRIVE
4028 010734 022700 000010          CMP    #8.,RO
4029 010740 001267                BNE    1$            ;MORE LEFT.
4030 010742 005737 003442          TST   DRIVS
4031 010746 001065                BNE    10$
4032 010750 104123                ERROR 123           ;NO DRIVES FOUND
4033 010752 000000                HALT
4034 010754 000137 007730          JMP    ST5           ;SETUP CORRECTLY
4035                ;& PRESS 'CONT'
4036 010760 032737 000040 003336  2$: BIT    #DTYE,HER

```

```

4037 010766 001405      BEQ      13$
4038 010770 012737 002000 001170    MOV      #CDT,STMP4      ;ADD CDT
4039 010776 000137 010520      JMP      1$              ;TRY AGAIN
4040 011002 032737 001000 003324 13$:    BIT      #MDS,HCS2
4041 011010 001015      BNE      6$
4042 011012 032737 000400 003324    BIT      #UFE,HCS2
4043 011020 001015      BNE      7$
4044 011022 032737 000001 003334    BIT      #DRA,HDS
4045 011030 001015      BNE      8$
4046 011032 032737 010000 003324    BIT      #NED,HCS2
4047 011040 001424      BEQ      9$
4048 011042 000730      BR       5$
4049
4050 011044 004737 044236      6$:    JSR      PC,BYP          ;TYPE BYP DR #
4051 011050 104002      ERROR   2$              ;MDS DETECTED
4052 011052 000724      BR       5$
4053
4054 011054 004737 044236      7$:    JSR      PC,BYP
4055 011060 104003      ERROR   3$              ;UFE DETECTED
4056 011062 000720      BR       5$
4057
4058 011064 032737 010000 003324 8$:    BIT      #NED,HCS2
4059 011072 001676      BEQ      4$
4060 011074 104401 056153      TYPE    MSG15          ;DRV#
4061 011100 010046      MOV     RD,-(SP)       ;SAVE RD FOR TYPEOUT
4062
4063 011102 104403      TYPOS
4064 011104 001      .BYTE   1              ;TYPE DR#
4065 011105 000      .BYTE   0              ;GO TYPE--OCTAL ASCII
4066 011106 104010      ERROR   10             ;TYPE 1 DIGIT(S)
4067 011110 000705      BR       5$            ;SUPPRESS LEADING ZEROS
4068
4069 011112 004737 044236      9$:    JSR      PC,BYP
4070 011116 104004      ERROR   4$              ;NO DRA & NO NED = OTHER PORT SELECTED
4071 011120 000701      BR       5$
4072 011122 000137 011520     10$:   JMP      NUDRV
4073
4074 011126      VERIFY:

```

```

*****
*TEST 3          VERIFY OPERATOR DRIVE SELECTIONS
*
* THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
* DEFAULTED. EVERY DRIVE FROM 0 TO 7 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
* NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF
* NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFO TO
* VERIFY IT WAS NOT SPECIFIED.
* IF CERR DUE TO DTYE , THE DRIVE WILL BE TESTED AS AN RK07.
*****
TST3:  SCOPE

```

4091 011126 000004


```

4093 011130 012737 000001 001174      MOV      #1,STIMES      ;;DO 1 ITERATION
4094 011136 012706 001100      MOV      #STACK,SP    ;RESTORE STK PTR
4095 011142 005000      CLR      RO           ;DRIVE ADDR
4096 011144 012701 003444      MOV      #DRIVO,R1    ;DRIVE FLAG
4097 011150      104415      1$:      SCOP1
4098 011150 012706 001100      MOV      #STACK,SP    ;RESTORE STK PTR
4100
4101 011156 012765 000040 000010      MOV      #SCLR,RKCS2(R5)
4102 011164 013737 001414 003360      MOV      T10,TEMP1
4103 011172 004737 043530      JSR      PC,FRDY      ;FIND RDY
4104 011176 104120      ERROR    120        ;NO RDY AFTER SCLR
4105 011200 010065 000010      MOV      RO,RKCS2(R5) ;DRV ADDR
4106 011204 012737 000001 003322      MOV      #SELDRV,HCS1
4107 011212 053737 001170 003322      BIS      $TMP4,HCS1   ;ADD CDT IF RK07
4108 011220 013765 003322 000000      MOV      HCS1,RKCS1(R5) ;GET STATUS
4109 011226 013737 001426 003360      MOV      T50000,TEMP1
4110 011234 004737 044220      JSR      PC,DLY      ;DO DELAY TO CATCH MDS
4111 011240 013737 001414 003360      MOV      T10,TEMP1
4112 011246 004737 043530      JSR      PC,FRDY      ;FIND RDY
4113 011252 104117      ERROR    117        ;NO RDY AFTER SELDRV CMD
4114 011254 032737 100000 003322      BIT      #CERR,HCS1
4115 011262 001036      BNE     2$
4116 011264 013737 003350 003360      MOV      HMR2,TEMP1
4117 011272 042737 177770 003360      BIC      #C<DRVMSK>,TEMP1
4118 011300 020037 003360      CMP      RO,TEMP1    ;S/B SAME
4119 011304 001014      BNE     3$
4120 011306 005711      11$:     TST      (R1)
4121 011310 001402      BEQ     4$
4122 011312 053711 001170      BIS      $TMP4,(R1)  ;ADD CDT IF SET
4123 011316 005721      4$:     TST      (R1)+      ;SHIFT PTR TO NEXT DR FLAG
4124 011320 005200      INC     RO           ;INC DR#
4125 011322 005037 001170      CLR     $TMP4       ;CLEAR CDT FLAG FOR NEXT DRIVE
4126 011326 022700 000010      CMP     #8.,RO
4127 011332 001306      BNE     1$          ;MORE LEFT
4128 011334 000475      BR      TST4        ;;GO TO NEXT TEST
4129
4130 011336 004737 044236      3$:     JSR      PC,BYP      ;TRY BYPASS DRIVE#
4131 011342 104001      ERROR    1          ;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
4132 011344 005711      TST     (R1)
4133 011346 001763      BEQ     4$          ;BRANCH IF NOT SPEC BY INPUT
4134 011350 005337 003442      12$:    DEC     DRIVS     ;DECREMENT TOTAL DRIVS
4135 011354 005011      CLR     (R1)       ;CLEAR DRIVE FLAG
4136 011356 000757      BR      4$
4137
4138 011360 032737 000040 003336      2$:     BIT     #DTYE,HER
4139 011366 001405      BEQ     13$
4140 011370 012737 002000 001170      MOV     #CDT,$TMP4   ;ADD CDT
4141 011376 000137 011150      JMP     1$          ;TRY AGAIN
4142
4143 011402 032737 001000 003324      13$:    BIT     #MDS,HCS2
4144 011410 001027      BNE     6$
4145 011412 032737 000400 003324      BIT     #UFE,HCS2
4146 011420 001027      BNE     7$
4147 011422 032737 000001 003334      BIT     #DRA,HDS
4148 011430 001005      BNE     8$

```

```

4149 011432 032737 010000 003324 BIT #NED,HCS2
4150 011440 001423 BEQ 9$
4151 011442 000404 BR 10$
4152 011444 032737 010J00 003324 8$: BIT #NED,HCS2
4153 011452 001715 BEQ 11$
4154 011454 005711 10$: TST (R1)
4155 011456 001717 BEQ 4$
4156
4157 011460 004737 044236 JSR PC,BYP ;TYPE BYPASS DRIVE#
4158 011464 104006 ERROR 6
4159 011466 000730 BR 12$
4160
4161 011470 004737 044236 6$: JSR PC,BYP ;TYPE BYPASS DRIVE#
4162 011474 104002 ERROR 2 ;MDS DETECTED
4163 011476 000724 BR 12$
4164
4165 011500 004737 044236 7$: JSR PC,BYP
4166 011504 104003 ERROR 3 ;UFE DETECTED
4167 011506 000720 BR 12$
4168
4169 011510 004737 044236 9$: JSR PC,BYP
4170 011514 104004 ERROR 4 ;DRA & NED RESET - OTHER PORT SELECTED
4171 011516 000714 BR 12$
4172
4173
4174
4175 ; THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
4176 ; DRIVE PRESENT
4177
4178 ; 'SUNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
4179 ; UNDER TEST
4180
4181 011520 005037 001462 NUDRV: CLR BYPCERR ;ENTER HERE FROM LAST TEST
4182 ;& TEST CERR IN 'FRDY'
4183 011524 005037 001170 CLR STMP4 ;CLEAR RK07 IND FLAG
4184
4185 ;*****
4186 ;TEST 4 FIND NEXT DRIVE TO BE TESTED
4187 ;
4188 ; THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
4189 ; ADDRESS IN 'SUNIT' & STMP4 IS SET TO CDT IF DRIVE IS RK07.
4190 ; THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
4191 ; THE DRIVE WHOSE ADDRESS IS IN 'SUNIT'.
4192 ;
4193 ;*****
4194 011530 000004 TST4: SCOPE
4195 011532 012737 000001 001174 MOV #1,STIMES ;:DO 1 ITERATION
4196 011540 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
4197 011544 012737 000004 001214 MOV #STN-1,STESTN
4198 011552 012737 000004 001102 MOV #STN-1,STSTNM
4199
4200 011560 005737 003442 TST DRIVS ;ANY DRIVES PRESENT?
4201 011564 001004 BNE 4$ ;YES BRANCH
4202 011566 104401 056271 TYPE MSG19 ;ALL DRIVES TESTED
4203 011572 000137 042722 JMP $EOP1 ;NO, GO TO END
4204

```

```

4205 011576 013701 001346      4$:  MOV  DRVPTR,R1      ;ADDR OF NEXT DRIVE FLAG
4206 011602 005737 001220      TST  $DEVCT          ;IS FIRST DRIVE BEING CHECKED
4207 011606 001402              BEQ  2$              ;YES BRANCH
4208 011610 005237 001222      1$:  INC  $UNIT        ;INCR DRIVE ADDR TO NEXT DRIVE
4209 011614 005711              2$:  TST  (R1)         ;IS DRIVE PRESENT?
4210 011616 001002              BNE  5$              ;BR IF YES
4211 011620 005721              TST  (R1)+          ;ELSE FIND NEXT DRIVE
4212 011622 000772              BR   1$
4213 011624 005737 003434      5$:  TST  DDPCH        ;DDP CHAIN MODE?
4214 011630 001405              BEQ  3$              ;BR IF NO
4215 011632 005737 001222      TST  $UNIT        ;ELSE IS IT DRIVE 0?
4216 011636 001002              BNE  3$              ;BR IF NO
4217 011640 005721              TST  (R1)+          ;ELSE FIND NEXT DRIVE PRESENT
4218 011642 000762              BR   1$
4219
4220 011644 032721 002000      3$:  BIT  #CDT,(R1)+   ;SEE IF DRIVE UNDER TEST IS RK07
4221 011650 001403              BEQ  6$              ;BR IF NO
4222 011652 012737 002000      MOV  #CDT,$TMP4     ;ELSE SET RK07 FLAG
4223 011660 010137 001346      6$:  MOV  R1,DRVPTR    ;STORE POINTER TO NEXT DR FLAG
4224 011664 104401 056153      TYPE MSG15         ;"DRIVE"
4225 011670 013700 001222      MOV  $UNIT,RO      ;SAVE RO FOR TYPEOUT
4226 011674 010046              MOV  RO,-($P)
4227
4228 011676 104403      TYPOS              ;DRIVE #
4229 011700 001              .BYTE 1            ;GO TYPE--OCTAL ASCII
4230 011701 000              .BYTE 0            ;TYPE 1 DIGIT(S)
4231
4232 ; TYPE ,$CRLF      ;SUPPRESS LEADING ZEROS
4233
4234 011702 005737 001170      TST  $TMP4         ;SEE IF RK07 UNDER TEST
4235 011706 001030              BNE  7$              ;BR IF YES
4236 011710 012737 000631 012052  MOV  #631,LCM1     ;ELSE LOAD RK06 PARAMETERS
4237 011716 012737 000632 012054  MOV  #632,LC
4238 011724 012737 000633 012056  MOV  #633,LCP1
4239 011732 005037 012070      CLR  E.DDT
4240 011736 012737 000400 012060  MOV  #400,MC
4241 011744 012737 001000 012062  MOV  #1000,MC1
4242 011752 012737 000777 012064  MOV  #777,MASK
4243 011760 012737 160017 012066  MOV  #160017,MASK1
4244 011766 000443              BR   TSTS           ;;GOTO NEXT TEST
4245
4246 011770 012737 001455 012052  7$:  MOV  #1455,LCM1   ;LOAD RK07 PARAMETERS
4247 011776 012737 001456 012054  MOV  #1456,LC
4248 012004 012737 001457 012056  MOV  #1457,LCP1
4249 012012 012737 000400 012070  MOV  #D.DDT,E.DDT
4250 012020 012737 001000 012060  MOV  #1000,MC
4251 012026 012737 002000 012062  MOV  #2000,MC1
4252 012034 012737 001777 012064  MOV  #1777,MASK
4253 012042 012737 140017 012066  MOV  #140017,MASK1
4254 012050 000412              BR   TSTS           ;;GOTO NEXT TEST
4255
4256 012052 000000      LCM1: 0            ;LC-1
4257 012054 000000      LC: 0              ;LAST CYL
4258 012056 000000      LCP1: 0            ;LC+1
4259 012060 000000      MC: 0              ;MAJ CYL
4260 012062 000000      MC1: 0             ;MAJ CYL + 1 SHIFT

```

4261 012064 000000
4262 012066 000000
4263 012070 000000
4264 012072 000000
4265 012074 000001

MASK: 0
MASK1: 0
E.DDT: 0 ; EXPECTED DRIVE TYPE TO E.AO
FC: 0 ; FIND CYL
FCP1: 1 ; FC+1

4266
4267
4268
4269
4270
4271
4272
4273
4274

*TEST 5 UNLOAD DRIVE TO BE TESTED
*
* THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
* WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE.
* IT THEN WAITS FOR SPEED OK TO GO LOW BEFORE
* PROCEEDING TO THE NEXT TEST

4275 012076 000004
4276 012100 012737 000001 001174
4277 012106 012706 001100
4278
4279 012112 005237 003304
4280
4281 012116 004737 045424
4282 012122 104024
4283
4284 012124 012737 000007 003322
4285 012132 004737 043434
4286 012136 104011
4287 012140 004737 044012
4288 012144 104012
4289
4290 012146 074737 045424
4291 012152 104024
4292
4293 012154 013737 001414 003362
4294 012162 004737 046460
4295 012166 104315
4296
4297
4298
4299
4300

TST5: SCOPE
MOV #1,STIMES ; DO 1 ITERATION
MOV #STACK,SP
INC UNLD ; USED TO CHECK VALID HALT
JSR PC,SUBCLR
ERROR 24 ; CERR AFTER SCLR
MOV #UNLOAD,HCS1
JSR PC,DOCMD ; DO UNLOAD CMD & GET CONTR READY
ERROR 11 ; RDY NOT SET AFTER UNLOAD CMD.
JSR PC,TSTATN
ERROR 12 ; NO ATTN AFTER UNLOAD CMD
JSR PC,SUBCLR
ERROR 24 ; CERR AFTER SCLR
MOV T10,TEMP2
JSR PC,FSPOK
ERROR 315 ; SPEED NOT DOWN BY TIMEOUT

4296
4297
4298
4299
4300

4301 012170
4302
4303
4304
4305
4306
4307
4308
4309
4310

PFSRT: ; ENTER HERE FOR POWER FAIL RESTART
.SBTTL STATIC & CYCLE UP TESTS

4311 012170 000004
4312 012172 012737 000001 001174
4313 012200 012706 001100
4314
4315 012204 004737 045424
4316 012210 104024

*TEST 6 REFERENCE & CHECK ALL STATUS BYTES IN RKMR2 & RKMR3
*
* CHECKS THE ABILITY TO REFERENCE ALL
* DRIVE REGISTERS AND THAT THEY CONTAIN CORRECT STATUS.
*

TST6: SCOPE
MOV #1,STIMES ; DO 1 ITERATION
MOV #STACK,SP ; RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ; CERR AFTER SCLR

```

4317
4318 012212 004737 045074 JSR PC,GSTAT
4319 012216 032737 000100 003350 BIT #D.VV,HMR2
4320 012224 001004 BNE 4$ ;BR IF VV SET
4321 012226 012737 000040 003412 MOV #D.DRA,E.AO ;LOAD EXPECTED VALUE FOR AO
4322 012234 000403 BR 5$
4323
4324 012236 012737 000140 003412 4$: MOV #<D.DRA!D.VV>,E.AO
4325 012244 005037 003414 5$: CLR E.B0 ;EXPECTED MSG B0
4326 012250 012737 000740 003416 MOV #<D.HDWM!D.BRHM!D.DOOR!D.CART>,E.A1 ;EXPECTED MSG A1
4327 012256 012737 000001 003420 MOV #1,E.B1 ;EXPECTED MSG B1
4328 012264 005037 003422 CLR E.A2 ;EXPECTED MSG A2
4329 012270 012737 000002 003424 MOV #2,E.B2 ;EXPECTED MSG B2
4330 012276 012737 000003 003430 MOV #3,E.B3 ;EXPECTED MSG B3
4331
4332 012304 004737 044252 JSR PC,CHKMSG ;CHECK MSGS AO,B0,A1,B1
4333 012310 000007 .WORD T.A2!T.B2!T.B3 ; & MSGS SPECIFIED HERE
4334 012312 104016 ERROR 16 ;MSG AO ERROR FOR DRIVE UNLOADED
4335 012314 104017 ERROR 17 ;MSG B0 ERROR
4336 012316 104020 ERROR 20 ;MSG A1 ERROR
4337 012320 104021 ERROR 21 ;MSG B1 ERROR
4338
4339 012322 012765 000002 000026 MOV #2,RKMR1(R5) ;SELECT WORD 2
4340 012330 004737 045074 JSR PC,GSTAT
4341 012334 005737 001362 TST CYLDIF ;SEE IF MSG A2=0
4342 012340 001401 BEQ 64$ ;BR IF YES
4343 012342 104022 ERROR 22 ;MSG A2 NOT CLEARED FOR DRIVE UNLOADED
4344 012344 005737 001364 64$: TST CYLADD ;SEE IF MSG B2=0
4345 012350 001401 BEQ 65$ ;BR IF YES
4346 012352 104023 ERROR 23 ;MSG B2 NOT CLEARED FOR DRIVE UNLOADED
4347 012354
4348 012354 023727 001432 000001 65$: CMP HEADA,#1 ;FOR HEAD 0, B3=1
4349 012362 001401 BEQ TST7 ;GO TO NXT TST IF YES
4350 012364 104056 ERROR 56 ;HEAD REG IN B3 NOT 0 IN UNLOAD
4351
4352
4353
4354
4355 *****
4356 *TEST 7 PRINT DRIVE SERIAL NUMBER
4357 *
4358 * THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
4359 * IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
4360 *****
4361 TST7: SCOPE
4362 012366 000004 MOV #1,$TIMES ;DO 1 ITERATION
4363 012370 012737 000001 001174 MOV #STACK,SP ;RESTORE STK PTR
4364 012376 012706 001100
4365 012402 005737 001216 TST $PASS
4366 012406 001046 BNE TST10 ;GO TO NEXT IF NOT FIRST PASS
4367 012410 004737 045424 JSR PC,SUBCLR ;DO SUBSYS CLEAR
4368 012414 104024 ERROR 24 ;CERR AFTER SCLR
4369
4370 012416 104401 056165 TYPE MSG16 ;DRIVE SERIAL NO.
4371 012422 012765 000003 000026 MOV #3,RKMR1(R5) ;SELECT BYTE 3
4372 012430 004737 045074 JSR PC,GSTAT ;GET STATUS

```

```

4373 012434 013701 003350
4374 012440 012704 053660
4375 012444 010446
4376 012446 012703 000003
4377 012452 006101
4378 012454 006101
4379 012456 006101
4380 012460 006101
4381 012462 006101
4382 012464 006101
4383 012466 010100
4384 012470 042700 177760
4385 012474 052700 000060
4386 012500 110024
4387 012502 005303
4388 012504 001364
4389 012506 105014
4390 012510 004737 054126
4391 012514 104401 001205
4392 012520 104401 001205

```

```

MOV HMR2,R1 ;GET SERIAL #
MOV #SOCTVL,R4 ;GET ADDR CHAR BUFF
MOV R4,-(SP) ;STORE ON STACK FOR $$SUPRS
MOV #3,R3 ;SETUP CHAR COUNT
ROL R1 ;INITIALIZE BIT POSITIONS
ROL R1
ROL R1 ;GET NEXT 4 BITS
ROL R1
ROL R1
MOV R1,R0 ;GET WORKING COPY
BIC #177760,R0 ;CLEAR ALL BUT LOW 4 BITS
BIS #60,R0 ;CONVERT TO ASCII DIGIT
MOVB R0,(R4)+ ;PUT ASCII DIGIT INTO CHAR BUFF
DEC R3
BNE IS ;BR IF ALL 3 CHARS NOT DONE
CLRB (R4) ;ELSE INSERT NULL TERMINATOR
JSR PC,$$SUPRS ;TYPE
TYPE ,SCLF
TYPE ,SCLF

```

```

4393
4394
4395
4396
4397
4398
4399
4400 012524 000004
4401 012526 012737 000001 001174
4402 012534 012706 001100
4403
4404 012540 005065 000026
4405 012544 004737 045074
4406 012550 032737 000100 003350
4407 012556 001021
4408
4409 012560 104415
4410 012562 012706 001100
4411
4412 012566 004737 045424
4413 012572 104024
4414
4415 012574 012737 000003 003322
4416 012602 004737 043434
4417 012606 104116
4418
4419 012610 032737 000100 003350
4420 012616 001001
4421 012620 104027
4422
4423
4424
4425
4426
4427
4428

```

```

*****
*TEST 10 SET VV WITH PACK CMD
*
* IF VV IS RESET, THE PACK CMD IS USED TO SET IT.
*
*****
TST10: SCOPE
MOV #1,$TIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
CLR RKMRI(R5) ;SELECT BYTE 0
JSR PC,GSTAT ;GET STATUS
BIT #D.VV,HMR2
BNE TST11 ;GO TO NEXT TEST IF VV SET
SCOP1
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
MOV #PACK,HCS1
JSR PC,DOCMD ;DO PACK CMD & GET CONTR READY
ERROR 116 ;RDY NOT SET AFTER PACK CMD
BIT #D.VV,HMR2
BNE TST11 ;GO TO NEXT TEST IF VV NOW SET
ERROR 27 ;PACK DID NOT SET V.V.
*****
*TEST 11 RELEASE DRIVE
*
* TESTS THE ABILITY TO RECOGNIZE THE RLS BIT AND NOT RAISE SACK
*
*****

```

```

4429 012622 000004
4430 012629 012737 000001 001174 TST11: SCOPE
4431 012632 012706 001100 MOV #1,STIMES ;:DO 1 ITERATION
4432 JSR PC,SUBCLR ;RESTORE STK PTR
4433 012636 004737 045424 ERROR 24 ;DO SUBSYS CLEAR & GET STATUS
4434 012642 104024 BIT #UFE,HCS2 ;CONTR ERROR SET AFTER SCLR
4435 012644 032737 000400 003324 BEQ 1$
4436 012652 001401 ERROR 3 ;UFE SET AFTER SCLR
4437 012654 104003
4438
4439 012656 1$: SCOP1
4440 012656 104415 MOV #STACK,SP ;RESTORE STK PTR
4441 012660 012706 001100
4442 012664 004737 045424 JSR PC,SUBCLR
4443 012670 104024 ERROR 24 ;CERR AFTER SCLR
4444 012672 062765 000010 000010 ADD #RLS,RKCS2(R5) ;ADD RELEASE BIT TO SUNIT
4445 012700 004737 045074 JSR PC,GSTAT ;GET STATUS
4446
4447 012704 032737 100000 003322 BIT #CERR,HCS1 ;CHECK FOR CONTR ERROR
4448 012712 001401 BEQ 2$
4449 012714 104025 ERROR 25 ;RLS SET CERR
4450 012716 032737 000400 003324 2$: BIT #UFE,HCS2
4451 012724 001401 BEQ TST12 ;:GO TO NEXT TEST IF SET
4452 012726 104026 ERROR 26 ;SACK SET AFTER RLS SENT

```

I07

CZR6HED UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 86
T11 RELEASE DRIVE

SEQ 0086

4455


```

4456 ;*****
4457 ;*TEST 12 DRIVE TYPE TEST
4458 ;*
4459 ;*
4460 ;* THIS TEST COMPARES DRIVE TYPE IN MSG A AGAINST 'DDT' IN RKDS.
4461 ;* WRONG CDT IN RKCS1 IS SENT & ERRORS ARE VERIFIED.
4462 ;*
4463 ;*****
4464 ST12: SCOPE
4465 MOV #1,STIMES ;DO 1 ITERATION
4466 MOV #STACK,SP ;RESTORE STK PTR
4467
4468 JSR PC,SUBCLR ;SUBSYS CLEAR & GET STATUS
4469 ERROR 24 ;CONT ERROR SET AFT SUBSYS CLEAR
4470 TST $TMP4
4471 BNE 1$
4472 BIT #D.DDT,HMR2
4473 BEQ 2$
4474 ERROR 30 ;DR TYPE SET IN MR2
4475 BIT #DDT,HDS
4476 BEQ 3$
4477 ERROR 31 ;DDT SET IN RKDS
4478 BR 3$
4479
4480 BIT #D.DDT,HMR2
4481 BNE 8$
4482 ERROR 5 ;DR TYPE NOT SET IN MR2
4483 BIT #DDT,HDS
4484 BNE 3$
4485 ERROR 152
4486 BIT #DTYE,HER
4487 BEQ 4$
4488 ERROR 32 ;DTYE SET IN RKER
4489
4490 4$:
4491 SCOP1
4492 MOV #STACK,SP ;RESTORE STK PTR
4493
4494 JSR PC,SUBCLR
4495 ERROR 24 ;CERR AFTER SCLR
4496
4497 TST $TMP4 ;SEE IF RK07
4498 BEQ 9$ ;BR IF NO
4499 MOV #SELDRV,RKCS1(R5)
4500 BR 10$
4501 MOV #<CDT!SELDRV>,RKCS1(R5) ;GET STATUS WITH CDT SET
4502 MOV T10,TEMP1
4503 JSR PC,FRDY ;FIND RDY
4504 ERROR 117 ;RDY NOT SET BY END OF SEL DRV CMD
4505 TST $TMP4
4506 BNE 11$
4507 BIT #D.DDT,HMR2
4508 BEQ 5$
4509 ERROR 30 ;DR TYPE SET IN MR2
4510 BIT #DDT,HDS
4511 BEQ 6$

```

```

4512 013150 104031          ERROR 31          ;DDT SET IN RKDS
4513 013152 000412          BR      6$
4514
4515 013154 032737 000400 003350 11$: BIT      #D.DDT,HMR2
4516 013162 001001          BNE     12$
4517 013164 104005          ERROR 5           ;DR TYPE NOT SET IN MR2
4518 013166 032737 000400 003334 12$: BIT      #DDT,HDS
4519 013174 001001          BNE     6$
4520 013176 104152          ERROR 152        ;DDT NOT SET IN RKDS
4521 013200 032737 000040 003336 6$: BIT      #DTYE,HER ;DTYE=DDT(NOT)*CDT
4522 013206 001001          BNE     7$
4523 013210 104033          ERROR 33        ;DTYE NOT SET AFT WRITING WRONG CDT
4524 013212 032737 100000 003322 7$: BIT      #CERR,HCS1
4525 013220 001001          BNE     TST13    ;GO TO NEXT TEST
4526 013222 104034          ERROR 34        ;CERR NOT SET AFT WRITING WRONG CDT

```

```

*****
:TEST 13          C-D PARITY ERROR DETECTION
:
:TESTS THE ABILITY OF THE DRIVE TO DETECT EVEN PARITY SENT BY
:THE CONTROLLER BY SETTING 'PAT' ON RKMR1.
:THE DRIVE SHOULD RESPOND WITH 'C-D PARITY ERROR'
:THE DRIVE STILL SENDS ODD PARITY TO THE CONTROLLER WHICH IS NOW
:CHECKING FOR EVEN PARITY THEREFORE THE CONTROLLER SHOULD DETECT
:AN ERROR AND SET DCPAR.
:THE ERROR CONDITION IS RESET WITH THE CLEAR CMD

```

```

4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540 013224 000004          TST13: SCOPE
4541 013226 012737 000001 001174 MOV      #1,STIMES ;DO 1 ITERATION
4542 013234 012706 001100 MOV      #STACK,SP ;RESTORE STK PTR
4543
4544 013240 004737 045424 JSR      PC,SUBCLR ;SUBSYS CLEAR & GET STATUS
4545 013244 104024          ERROR 24        ;CONT ERROR AFTER SUBSYS CLR
4546 013246 032737 001000 003352 BIT      #D.PAR,HMR3
4547 013254 001401          BEQ     25$
4548 013256 104035          ERROR 35
4549 013260 032737 020000 003322 25: BIT      #DCPAR,HCS1 ;C-D PARITY ERROR SET IN MR3

```

4550	013266	001401			BEQ	35		
4551	013270	104036			ERROR	36		;DCPAR SET IN CS1
4552								
4553	013272							
4554	013272	104415			SCOP1			
4555	013274	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
4556								
4557	013300	004737	045424		JSR	PC,SUBCLR		
4558	013304	104024			ERROR	24		;CERR AFTER SCLR
4559								
4560	013306	012765	000020	000026	MOV	#PAT,RKMR1(R5)		;SELECT BYTE 0 & EVEN PARITY
4561	013314	004737	045074		JSR	PC,GSTAT		;GET STATUS



M07

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 90
T13 C-D PARITY ERROR DETECTION

SEQ 0090

4562 013320 032737 000200 003352
4563 013326 001001

BIT #D.FLT,HMR3
BNE 4S

N07

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 91
T13 C-D PARITY ERROR DETECTION

SEQ 0091

```

4564 013330 104037          ERROR 37          ;FAULT NOT SET IN MR3
4565 013332 032737 001000 003352 4$: BIT #D.PAR,HMR3
4566 013340 001001          BNE 5$
4567 013342 104040          ERROR 40          ;C-D PARITY ERROR NOT SET IN MR3
4568 013344 032737 020000 003322 5$: BIT #DCPAR,HCS1
4569 013352 001001          BNE 6$
4570 013354 104041          ERROR 41          ;DCPAR NOT SET AFT WRITING PAT IN MR1
4571 013356 032737 100000 003322 6$: BIT #CERR,HCS1
4572 013364 001001          BNE TST14        ;:GO TO NEXT TEST
4573 013366 104042          ERROR 42          ;CERR NOT SET BY WRITING PAT IN MR1
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585 013370 000004          TST14: SCOPE
4586 013372 012737 000001 001174 MOV #1,STIMES ;:DO 1 ITERATION
4587 013400 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
4588
4589 013404 004737 045424 JSR PC,SUBCLR ;SUBSYS CLEAR & GET STATUS
4590 013410 104024          ERROR 24          ;CERR AFTER SCLR
4591
4592 013412 012737 000011 003322 MOV #SRTSPL,HCS1
4593 013420 004737 043434 JSR PC,DOCMD ;DO START SPINDLE CMD & GET CONTR RDY
4594 013424 104121          ERROR 121         ;RDY NOT SET AFTER START SPIN CMD
4595
4596 013426 004737 045074 JSR PC,GSTAT ;WORD 0
4597 013432 032737 010000 003350 BIT #D.SPIN,HMR2
4598 013440 001001          BNE 13$
4599 013442 104306          ERROR 306         ;SPIN NOT SET AFTER START SPIN CMD
4600
4601 013444 012737 014642 001176 13$: MOV #25$,SESCAPE
4602 013452 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
4603 013456 000401          BR 15$
4604 013460 104316          ERROR 316
4605 013462 012737 010140 003412 15$: MOV #<D.SPIN!D.VV!D.DRA>,E.A0 ;LOAD IN EXPECTED VALUES
4606 013470 005037 003414 CLR E.B0
4607 013474 012737 000740 003416 MOV #<D.CART!D.DOOR!D.HDHM!D.BRHM>,E.A1
4608 013502 012737 000001 003420 MOV #1,E.B1
4609
4610 013510 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
4611 013514 000000          .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4612 013516 104057          ERROR 57          ;MSG A0 ERROR AFTER START SPIN CMD REC'D BY DRIVE
4613 013520 104060          ERROR 60          ;MSG B0 ERROR
4614 013522 104061          ERROR 61          ;MSG A1 ERROR
4615 013524 104062          ERROR 62          ;MSG B1 ERROR
4616 013526 005737 003470 TST DOTIM
4617 013532 001126          BNE 3$
4618 013534 012737 014666 001176 1$: MOV #30$,SESCAPE ;BRANCH IF P OR L CLOCK PRESENT
4619 013542 012765 100000 000000 MOV #CCLR,RKCS1(R5)

```

```

4620 013550 013737 001416 003362 MOV T100,TEMP2 ;SETUP TIMEOUT
4621 013556 004737 044044 JSR PC,FATT1 ;FIND ATTN
4622 013562 104067 ERROR 67 ;NO ATTN AFTER HEAD LOADING
4623 013564 2S:
4624
4625 013564 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4626 013572 005037 003414 CLR E.B0 ;EXPECTED MSG B0
4627 013576 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4628 013604 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4629 013612 005037 003422 CLR E.A2 ;EXPECTED MSG A2
4630 013616 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4631 013624 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4632
4633 013632 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
4634 013636 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4635 013640 104063 ERROR 63 ;MSG A0 ERROR AT END OF HEAD LOAD
4636 013642 104064 ERROR 64 ;MSG B0 ERROR
4637 013644 104065 ERROR 65 ;MSG A1 ERROR
4638 013646 104066 ERROR 66 ;MSG B1 ERROR
4639 013650 005737 001364 TST CYLADD ;SEE IF MSG B2 = 0
4640 013654 001401 BEQ 16S ;BR IF YES
4641 013656 104176 ERROR 176 ;CYL ADDR NOT 0 AT END OF HEAD LOAD
4642 013660 16S:
4643
4644 013660 012765 100000 000000 MOV #CCLR,RKCS1(R5)
4645 013666 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
4646 013674 012737 000005 003322 MOV #CLEAR,HCS1
4647 013702 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
4648 013706 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
4649 013710 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
4650 013714 000401 BR 64S
4651 013716 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4652 013720 64S:
4653
4654 013720 012737 010340 003412 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4655 013726 005037 003414 CLR E.B0 ;EXPECTED MSG B0
4656 013732 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4657 013740 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4658 013746 005037 003422 CLR E.A2 ;EXPECTED MSG A2
4659 013752 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4660 013760 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4661
4662 013766 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
4663 013772 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4664 013774 104273 ERROR 273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4665 013776 104265 ERROR 265 ;MSG B0 ERROR
4666 014000 104274 ERROR 274 ;MSG A1 ERROR
4667 014002 104266 ERROR 266 ;MSG B1 ERROR
4668
4669 014004 000137 014462 JMP 12S
4670
4671 014010 012765 100000 000000 3S: MOV #CCLR,RKCS1(R5)
4672 014016 013737 001370 001372 MOV HZ,COUNT
4673 014024 012737 000074 001374 MOV #60.,SEC
4674 014032 004737 047206 JSR PC,CLKON ;TURN CLK INTR ON FOR 60 SEC MAX
4675 014036 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1

```

C08

CZR6HE0 UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 93
T14 VERIFY START SPINDLE CMD

SEQ 0093

4676	014044	004737	045074		4\$:	JSR	PC,GSTAT	
4677	014050	032737	002000	003350		BIT	#D.FWD,HMR2	
4678	014056	001004				BNE	5\$	
4679	014060	005737	001376			TST	TIMUP	; IS 60 SEC DELAY UP?
4680	014064	001767				BEQ	4\$; BRANCH IF NO & REPEAT
4681	014066	104070				ERROR	70	; FWD NOT SET WITHIN 60 SEC FROM
4682								; START SPINDLE CMD.
4683	014070	004737	047302		5\$:	JSR	PC,CLKOF	; TURN OFF CLOCK INTERRUPT
4684	014074	012765	100000	000000		MOV	#CCLR,RKCS1(R5)	
4685								
4686	014102	013737	001370	001372		MOV	HZ,COUNT	
4687	014110	012737	000005	001374		MOV	#5,SEC	
4688	014116	004737	047206			JSR	PC,CLKON	; TURN CLK INTR ON FOR 5 SEC MAX
4689	014122	012765	000001	000026	6\$:	MOV	#1,RKMR1(R5)	; WORD 1
4690	014130	004737	045074			JSR	PC,GSTAT	
4691	014134	032737	002000	003350		BIT	#D.FWD,HMR2	
4692	014142	001404				BEQ	7\$	
4693	014144	005737	001376			TST	TIMUP	
4694	014150	001764				BEQ	6\$	
4695	014152	104075				ERROR	75	; FWD NOT CLEARED WITHIN 5 SEC OF MOTION
4696								; FROM START SPINDLE CMD.
4697	014154	004737	047302		7\$:	JSR	PC,CLKOF	; TURN OFF CLK INTERRUPT
4698	014160	004737	044012			JSR	PC,TSTATN	; TEST FOR ATTN
4699	014164	000401				BR	17\$	
4700	014166	104320				ERROR	320	; UNEXP ATTN AFTER INNER LIM DETECT
4701	014170	012737	030140	003412	17\$:	MOV	#<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0	; EXPECTED A0
4702	014176	005037	003414			CLR	E.B0	
4703	014202	012737	025720	003416		MOV	#<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	
4704	014210	012737	000001	003420		MOV	#1,E.B1	
4705								
4706	014216	004737	044252			JSR	PC,CHKMSG	; CHECK MSGS A0,B0,A1,B1
4707	014222	000000				.WORD	0!0!0	; & MSGS SPECIFIED HERE
4708	014224	104076				ERROR	76	; MSG A0 ERROR AT INNER LIMIT DETECT
4709	014226	104077				ERROR	77	; MSG B0 ERROR
4710	014230	104100				ERROR	100	; MSG A1 ERROR
4711	014232	104101				ERROR	101	; MSG B1 ERROR
4712								
4713	014234	013737	001370	001372		MOV	HZ,COUNT	
4714	014242	012737	000004	001374		MOV	#4,SEC	
4715	014250	004737	047206			JSR	PC,CLKON	; TURN CLK INTR ON FOR 4 SEC MAX
4716	014254	012765	000001	000026	8\$:	MOV	#1,RKMR1(R5)	; WORD 1
4717	014262	004737	045074			JSR	PC,GSTAT	
4718	014266	032737	002000	003350		BIT	#D.FWD,HMR2	
4719	014274	001004				BNE	9\$	
4720	014276	005737	001376			TST	TIMUP	
4721	014302	001764				BEQ	8\$	
4722	014304	104102				ERROR	102	; FWD NOT DETECTED WITHIN 4 SEC IN RTZ PORTION OF
4723								; START SPINDLE CMD.
4724	014306	004737	047302		9\$:	JSR	PC,CLKOF	; TURN CLOCK INTR OFF.
4725	014312	004737	044012			JSR	PC,TSTATN	; TEST FOR ATTN
4726	014316	000401				BR	18\$	
4727	014320	104321				ERROR	321	; UNEXP ATTN AFTER OUTER LIM TO CYL 0
4728	014322	012737	030140	003412	18\$:	MOV	#<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0	; EXPECTED A0
4729	014330	005037	003414			CLR	E.B0	
4730	014334	012737	023720	003416		MOV	#<D.RTZ!D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	
4731	014342	012737	000001	003420		MOV	#1,E.B1	

4732										
4733	014350	004737	044252			JSR	PC,CHKMSG			;CHECK MSGS A0,B0,A1,B1
4734	014354	000000				.WORD	0!0!0			& MSGS SPECIFIED HERE
4735	014356	104103				ERROR	103			;MSG A0 ERROR FROM OUT LIM TO CYL 0 DURING LOAD
4736	014360	104104				ERROR	104			;MSG B0 ERROR
4737	014362	104105				ERROR	105			;MSG A1 ERROR
4738	014364	104106				ERROR	106			;MSG B1 ERROR
4739	014366	013737	001370	001372		MOV	HZ,COUNT			
4740	014374	012737	000001	001374		MOV	#1,SEC			
4741	014402	004737	047206			JSR	PC,CLKON			;TURN CLK INTR ON FOR 1 SEC MAX
4742	014406	005065	000026		10\$:	CLR	RKMR1(R5)			;WORD 0
4743	014412	004737	045074			JSR	PC,GSTAT			
4744	014416	032737	000200	003350		BIT	#D,DRDY,HMR2			;SEE IF DRIVE READY
4745	014424	001004				BNE	11\$			
4746	014426	005737	001376			TST	TIMUP			
4747	014432	001765				BEQ	10\$			
4748	014434	104107				ERROR	107			;DRIVE READY NOT SET WITHIN 1 SEC FROM
4749										;FWD IN RTZ PORTION OF START SPIN CMD
4750	014436	004737	047302		11\$:	JSR	PC,CLKOF			;TURN CLOCKS OFF
4751	014442	004737	044012			JSR	PC,TSTATN			;TEST ATTN
4752	014446	104067				ERROR	67			;NO ATTN AFTER START SPIN CMD
4753	014450	012737	014666	001176		MOV	#30\$, \$ESCAPE			
4754	014456	000137	013564			JMP	2\$;CHECK RKMR 2 & 3 WORDS 0 & 1
4755										
4756	014462	005037	001176		12\$:	CLR	\$ESCAPE			
4757	014466	012765	100000	000000		MOV	#CCLR,RKCS1(R5)			
4758	014474	005065	000026			CLR	RKMR1(R5)			
4759	014500	004737	045074			JSR	PC,GSTAT			
4760	014504	032737	010000	003350		BIT	#D,SPIN,HMR2			
4761	014512	001003				BNE	14\$			
4762	014514	104306				ERROR	306			;SPIN NOT SET AFTER ST. SPIN CMD
4763	014516	000137	042662			JMP	\$EOP			;ABORT DRIVE
4764	014522	005037	003304		14\$:	CLR	UNLD			;USED FOR VALID HALT
4765	014526	004737	047430			JSR	PC,SWTST			;SEE IF SW 14 OR 8 IS SET
4766	014532	000467				BR	TST15			;GO TO NEXT TEST
4767										;RETURN HERE IF SW 14 IS SET OR
4768										;SW 8 WITH SWR <7:0> APPLY
4769	014534	005237	003304		2C2:	INC	UNLD			
4770										
4771	014540	004737	045424			JSR	PC,SUBCLR			
4772	014544	104024				ERROR	24			;CERR AFTER SCLR
4773										
4774	014546	012737	000007	003322		MOV	#UNLOAD,HCS1			
4775	014554	004737	043434			JSR	PC,DOCMD			;DO UNLOAD CMD & GET CONTR READY
4776	014560	104011				ERROR	11			;RDY NOT SET AFTER UNLOAD CMD.
4777	014562	004737	044012			JSR	PC,TSTATN			
4778	014566	104012				ERROR	12			;NO ATTN AFTER UNLOAD CMD
4779										
4780	014570	004737	045424			JSR	PC,SUBCLR			
4781	014574	104024				ERROR	24			;CERR AFTER SCLR
4782										
4783	014576	013737	001414	003362		MOV	T10,TEMP2			
4784	014604	004737	046460			JSR	PC,FSPOK			
4785	014610	104315				ERROR	315			;SPEED NOT DOWN BY TIMEOUT
4786										
4787	014612	004737	045424			JSR	PC,SUBCLR			

E08

CZR6HE0 UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 95
T14 VERIFY START SPINDLE CMD

SEQ 0095

```

4788 014616 104024          ERROR 24          ;CERR AFTER SCLR
4789
4790 014620 005037 001176    CLR          $ESCAPE
4791 014624 005737 001410    TST          LPFLG
4792 014630 001402          BEQ          65$
4793 014632 000177 164252    JMP          $SLPERR          ;SW 9 WAS SET.
4794 014636 000177 164244    65$: JMP          $SLFADR          ;SW 14 OR 8 WAS SET
4795
4796
4797
4798 014642 004737 047302    25$: JSR          PC,CLKOF
4799 014646 005237 001410    INC          LPFLG
4800 014652 032777 001000 164260  BIT          $SW9,$SWR          ;LOOP ON ERROR?
4801 014660 001325          BNE          20$          ;YES, RECONDITION DRIVE
4802 014662 000137 013534    JMP          1$          ;RETURN TO MAINLINE
4803 014666 004737 047302    30$: JSR          PC,CLKOF
4804 014672 005237 001410    INC          LPFLG
4805 014676 032777 001000 164234  BIT          $SW9,$SWR          ;LOOP ON ERROR?
4806 014704 001313          BNE          20$          ;YES, RECONDITION DRIVE
4807 014706 000137 014462    JMP          12$          ;RETURN TO MAINLINE
4808 .SBTTL SEEK/READ HEADER/WRITE HEADER TESTS
4809
4810
4811 ;*****
4812 ;*TEST 15          STATIC CYL DIFF AND CYL ADDR REG TEST; PART 1
4813 ;*
4814 ;*          THIS TEST CHECKS EACH BIT OF THE CYL DIFFERENCE
4815 ;*          AND CYL ADDRESS REGISTERS BY PERFORMING SEEKS TO ALL
4816 ;*          MAJOR CYLS (0,1,2,4,8,16,32,64,128,256) (512 FOR RK07) WITH EVEN PARITY SET.
4817 ;*          THIS FREEZES THE INFORMATION IN THE ABOVE REGISTERS & ALLOWS FOR CHECKING.
4818 ;*          THIS TEST VERIFIES C-D PARITY ERROR BIT SET, THAT HEADS DID
4819 ;*          NOT MOVE & ALL OTHER APPLICABLE STATUS BITS & REGS.
4820 ;*
4821 ;*****
4821 014712 000004          TST15: SCOPE
4822 014714 012737 000001 001174  MOV          #1,$TIMES          ;DO 1 ITERATION
4823 014722 012706 001100          MOV          $STACK,SP          ;RESTORE STK PTR
4824
4825 014726 005000          CLR          R0          ;CYL # REGISTER
4826 014730 012737 100000 003370  MOV          $BIT15,TEMPS
4827
4828 014736          1$:
4829 014736 104415          SCOP1
4830 014740 012706 001100          MOV          $STACK,SP          ;RESTORE STK PTR
4831
4832 014744 004737 045424          JSR          PC,SUBCLR
4833 014750 104024          ERROR 24          ;CERR AFTER SCLR
4834
4835
4836 014752 012765 100000 000000  MOV          $CCLR,RKCS1(R5)
4837 014760 013765 001222 000010  MOV          $UNIT,RKCS2(R5)
4838 014766 012737 000013 003322  MOV          $RECAL,HCS1
4839 014774 004737 043434          JSR          PC,DOCMD          ;DO RECAL CMD & GET CONTR RDY
4840 015000 104124          ERROR 124          ;RDY NOT SET AFTER RECAL CMD
4841
4842 015002 012765 000001 000026  MOV          #1,RKMR1(R5)          ;SELECT WORD 1
4843 015010 004737 045074          JSR          PC,GSTAT

```

F08

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 96
T15 STATIC CYL DIFF AND CYL ADDR REG TEST; PART 1

SEQ 0096

4844	015014	032737	020000	003350	BIT	#D.RTZ,HMR2	
4845	015022	001001			BNE	64\$	
4846	015024	104244			ERROR	244	;RTZ NOT SET DURING RECAL CMD
4847	015026	013737	001414	003362	MOV	T10,TEMP2	;SETUP TIMEOUT
4848	015034	004737	044044		JSR	PC,FATT1	;FIND ATTN
4849	015040	104055			ERROR	55	;NO ATTN AFTER RECAL CMD
4850							
4851	015042	012737	050340	003412	MOV	#<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4852	015050	005037	003414		CLR	E.B0	;EXPECTED MSG B0
4853	015054	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4854	015062	012737	000001	003420	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4855	015070	005037	003422		CLR	E.A2	;EXPECTED MSG A2
4856	015074	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4857	015102	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4858							
4859	015110	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
4860	015114	000007			.WORD	T.A2!T.B2!T.B3	; & MSGS SPECIFIED HERE
4861	015116	104221			ERROR	221	;MSG A0 ERROR AFTER RECAL CMD
4862	015120	104275			ERROR	275	;MSG B0 ERROR
4863	015122	104222			ERROR	222	;MSG A1 ERROR
4864	015124	104276			ERROR	276	;MSG B1 ERROR
4865					CWD2	47,50,<AFTER RECAL CMD>	
4866	015126	012765	000002	000026	MOV	#2,RKMR1(R5)	;SELECT THE MESSAGE
4867	015134	004737	045074		JSR	PC,GSTAT	;GET THE STATUS
4868	015140	005737	001364		TST	CYLADD	;RECAL SUCCESSFUL ?
4869	015144	001401			BEQ	65\$;BRANCH IF SO
4870	015146	104050			ERROR	50	;REPORT THE ERROR
4871	015150						65\$:
4872							
4873	015150	012765	100000	000000	MOV	#CLR,RKCS1(R5)	
4874	015156	013765	001222	000010	MOV	SUNIT,RKCS2(R5)	;DRIVE#
4875	015164	012737	000005	003322	MOV	#CLEAR,HCS1	
4876	015172	004737	043434		JSR	PC,DOCMD	;DO DRIVE CLEAR CMD & GET CONTR RDY
4877	015176	104151			ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD
4878	015200	004737	044012		JSR	PC,TSTATN	;TEST FOR ATTN
4879	015204	000401			BR	66\$	
4880	015206	104154			ERROR	154	;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4881	015210						66\$:
4882							
4883	015210	012737	010340	003412	MOV	#<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4884	015216	005037	003414		CLR	E.B0	;EXPECTED MSG B0
4885	015222	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4886	015230	012737	000001	003420	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4887	015236	005037	003422		CLR	E.A2	;EXPECTED MSG A2
4888	015242	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4889	015250	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4890							
4891	015256	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
4892	015262	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4893	015264	104273			ERROR	273	;MSG A0 ERROR AFTER DRIVE CLEAR CMD
4894	015266	104265			ERROR	265	;MSG B0 ERROR
4895	015270	104274			ERROR	274	;MSG A1 ERROR
4896	015272	104266			ERROR	266	;MSG B1 ERROR
4897							
4898							
4899	015274	104415			SCOP1		

```

4900 015276 012706 001100      MOV      #STACK,SP      ;RESTORE STK PTR
4901
4902 015302 004737 045424      JSR      PC,SUBCLR      ;
4903 015306 104024                ERROR    24              ;CERR AFTER SCLR
4904
4905 015310 005237 001462      INC      BYPCERR        ;DO NOT TEST CERR IN GSTAT1
4906 015314 012765 000020 000026      MOV      #PAT,RKMR1(R5) ;EVEN PARITY
4907 015322 010065 000020                MOV      RO,RKDC(R5)    ;CYL ADDR
4908 015326 012737 000017 003322      MOV      #SEEK,HCS1
4909 015334 004737 043434      JSR      PC,DOCMD       ;DO SEEK CMD & GET CONTR RDY
4910 015340 104122                ERROR    122            ;NO RDY FROM SEEK WITH BAD PARITY
4911 015342 004737 044012      JSR      PC,TSTATN      ;TEST FOR ATTN
4912 015346 104125                ERROR    125            ;NO ATTN FROM SEEK & BAD PARITY
4913 015350 012737 050340 003412      MOV      #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED AO
4914 015356 012737 001200 003414      MOV      #<D.FLT!D.PAR>,E.B0
4915 015364 012737 001720 003416      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4916 015372 012737 000001 003420      MOV      #1,E.B1
4917 015400 010037 003422      MOV      RO,E.A2
4918 015404 006137 003422      ROL     E.A2
4919 015410 006137 003422      ROL     E.A2
4920 015414 006137 003422      ROL     E.A2
4921 015420 006137 003422      ROL     E.A2
4922 015424 013737 003422 003424      MOV      E.A2,E.B2
4923 015432 052737 000002 003424      BIS     #2,E.B2        ;ADD MSG ID
4924
4925 015440 004737 044252      JSR      PC,CHKMSG      ;CHECK MSGS AO,B0,A1,B1
4926 015444 000003                .WORD   T.A2!T.B2!0    ;& MSGS SPECIFIED HERE
4927 015446 104110                ERROR    110            ;MSG AO ERROR AFTER SEEK WITH BAD PARITY
4928 015450 104111                ERROR    111            ;MSG B0 ERROR
4929 015452 104146                ERROR    146            ;MSG A1 ERROR
4930 015454 104147                ERROR    147            ;MSG B1 ERROR
4931
4932 015456 020037 001364      CMP     RO,CYLADD
4933 015462 001401                BEQ     25
4934 015464 104043                ERROR    43              ;CYL ADDR IN B2 NOT=RKDC
4935
4936 015466 020037 001362      25:    CMP     RO,CYLDIF
4937 015472 001401                BEQ     35
4938 015474 104044                ERROR    44              ;CYL DIFF IN A2 NOT=RKDC
4939
4940 015476 005037 001462      35:    CLR     BYPCERR        ;ALLOW CHECKING FOR ANY CERR IN GSTAT1
4941 015502 006137 003370      ROL     TEMPS          ;SET CARRY ONLY ONCE
4942 015506 006100      ROL     RO              ;SELECT NEXT MAJOR CYL
4943 015510 020037 012062      CMP     RO,MC1         ;ALL MAJOR CYL DONE?
4944 015514 001001                BNE     45              ;BRANCH IF NO
4945 015516 000402                BR      TST16
4946 015520 000137 014736      45:    JMP     15              ;;GO TO NEXT TST
4947

```

```

4948 *****
4949 *TEST 16  STATIC CYL DIFF & CYL ADDR REG TEST-PART 2
4950 *
4951 * THIS TEST CHECKS THE ABILITY OF THE DRIVE TO PROPERLY SET THE CYL
4952 * DIFF. & CYL ADDR REGS FOR ALL COMBINATIONS BY SEEKING TO
4953 * ALL CYLS FROM EVERY OTHER CYL. (N SQUARE SEEKS).
4954 * IT IS PERFORMED IN THE SAME MANNER AS THE ABOVE TEST.
4955 *

```

H08

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 98
T16 STATIC CYL DIFF & CYL ADDR REG TEST-PART 2

SEQ 0098

```

*****
4956 015524 000004 000001 001174 TST16: SCOPE
4957 015526 012737 000001 001174 MOV #1,STIMES ;DO 1 ITERATION
4958 015526 012737 000001 001174 MOV #STACK,SP ;RESTORE STK PTR
4959 015534 012706 001100
4960
4961 015540 005737 001340 TST BYPT16
4962 015544 001404 BEQ 13$
4963 015546 104401 055704 TYPE MSG9 ;BYPASSING TEST 16
4964 015552 000137 016766 JMP 12$
4965
4966 015556 005737 001170 13$: TST $TMP4 ;SEE IF RK07
4967 015562 001403 BEQ 15$ ;BR OF NO
4968 015564 104401 056600 TYPE MSG23 ;10-12 MIN TEST
4969 015570 000402 BR 16$
4970 015572 104401 055646 15$: TYPE ,MSG8 ;2-4 MIN TEST
4971
4972 015576 005037 001350 16$: CLR FRCYL ;FROM CYL
4973 015602 005037 001352 CLR TOCYL ;TO CYL
4974 015606 005037 001354 CLR CCYL ;CURRENT CYL
4975 015612 005037 001356 CLR PCYL ;PREV CYL
4976
4977 015616 104415 SCOP1
4978 015620 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
4979
4980 015624 004737 045424 JSR PC,SUBCLR
4981 015630 104024 ERROR 24 ;CERR AFTER SCLR
4982
4983
4984 015632 012765 100000 000000 MOV #CCLR,RKCS1(R5)
4985 015640 013765 001222 000010 MOV $UNIT,RKCS2(R5)
4986 015646 012737 000013 003322 MOV #RECAL,HCS1
4987 015654 004737 043434 JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
4988 015660 104124 ERROR 124 ;RDY NOT SET AFTER RECAL CMD
4989
4990 015662 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
4991 015670 004737 045074 JSR PC,GSTAT
4992 015674 032737 020000 003350 BIT #D.RTZ,HMR2
4993 015702 001001 BNE 64$
4994 015704 104244 ERROR 244 ;RTZ NOT SET DURING RECAL CMD
4995 015706 013737 001414 003362 64$: MOV T10,TEMP2 ;SETUP TIMEOUT
4996 015714 004737 044044 JSR PC,FATT1 ;FIND ATTN
4997 015720 104055 ERROR 55 ;NO ATTN AFTER RECAL CMD
4998
4999 015722 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5000 015730 005037 003414 CLR E.B0 ;EXPECTED MSG B0
5001 015734 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5002 015742 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5003 015750 005037 003422 CLR E.A2 ;EXPECTED MSG A2
5004 015754 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5005 015762 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5006
5007 015770 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
5008 015774 000007 .WORD T.A2!T.B2!T.B3 ;& MSGS SPECIFIED HERE
5009 015776 104221 ERROR 221 ;MSG A0 ERROR AFTER RECAL CMD
5010 016000 104275 ERROR 275 ;MSG B0 ERROR
5011 016002 104222 ERROR 222 ;MSG A1 ERROR

```

5012	016004	104276			ERROR	276		:MSG B1 ERROR
5013					CWD2	47,50	<AFTER RECAL CMD>	
5014	016006	012765	000002	000026	MOV	#2,RKMR1(R5)		:SELECT THE MESSAGE
5015	016014	004737	045074		JSR	PC,GSTAT		:GET THE STATUS
5016	016020	005737	001364		TST	CYL,ADD		:RECAL SUCCESSFUL ?
5017	016024	001401			BEQ	65\$:BRANCH IF 50
5018	016026	104050			ERROR	50		:REPORT THE ERROR
5019	016030							
5020								
5021	016030	012765	100000	000000	MOV	#CCLR,RKCS1(R5)		
5022	016036	013765	001222	000010	MOV	SUNIT,RKCS2(R5)		:DRIVE#
5023	016044	012737	000005	003322	MOV	#CLEAR,HCS1		
5024	016052	004737	043434		JSR	PC,DOCMD		:DO DRIVE CLEAR CMD & GET CONTR RDY
5025	016056	104151			ERROR	151		:NO RDY AFTER DRIVE CLEAR CMD
5026	016060	004737	044012		JSR	PC,TSTATN		:TEST FOR ATTN
5027	016064	000401			BR	66\$		
5028	016066	104154			ERROR	154		:ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5029	016070							
5030								
5031	016070	012737	010340	003412	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		:EXPECTED MSG A0
5032	016076	005037	003414		CLR	E.B0		:EXPECTED MSG B0
5033	016102	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		:EXPECTED A1
5034	016110	012737	000001	003420	MOV	#1,E.B1		:MSG ID FOR EXPECTED MSG B1
5035	016116	005037	003422		CLR	E.A2		:EXPECTED MSG A2
5036	016122	012737	000002	003424	MOV	#2,E.B2		:MSG ID FOR EXPECTED MSG B2
5037	016130	012737	000003	003430	MOV	#3,E.B3		:MSG ID FOR EXPECTED MSG B3
5038								
5039	016136	004737	044252		JSR	PC,CHKMSG		:CHECK MSGS A0,B0,A1,B1
5040	016142	000003			.WORD	T.A2!T.B2!0		:& MSGS SPECIFIED HERE
5041	016144	104273			ERROR	273		:MSG A0 ERROR AFTER DRIVE CLEAR CMD
5042	016146	104265			ERROR	265		:MSG B0 ERROR
5043	016150	104274			ERROR	274		:MSG A1 ERROR
5044	016152	104266			ERROR	266		:MSG B1 ERROR
5045								
5046								
5047	016154	104415			SCOP1			
5048	016156	012706	001100		MOV	#STACK,SP		:RESTORE STK PTR
5049								
5050	016162	004737	045424		JSR	PC,SUBCLR		
5051	016166	104024			ERROR	24		:CERR AFTER SCLR
5052								
5053								
5054	016170	012765	000020	000026	MOV	#PAT,RKMR1(R5)		:EVEN PARITY
5055	016176	013765	001352	000020	MOV	TOCYL,RKDC(R5)		:SET TO CYL ADDR
5056	016204	013737	001352	001354	MOV	TOCYL,CCYL		:CURRENT CYL
5057	016212	013737	001354	003364	MOV	CCYL,TEMP3		
5058	016220	013737	001356	003366	MOV	PCYL,TEMP4		:PREV CYL
5059	016226	163737	003364	003366	SUB	TEMP3,TEMP4		
5060	016234	100002			BPL	2\$:BR IF TEMP4 IS POS
5061	016236	005437	003366		NEG	TEMP4		
5062	016242	013737	003366	001360	MOV	TEMP4,CALDIF		
5063	016250	013737	001354	001356	MOV	CCYL,PCYL		
5064	016256	012737	000017	003322	MOV	#SEEK,HCS1		
5065	016264	004737	043434		JSR	PC,DOCMD		:DO SEEK CMD & GET CONTR RDY
5066	016270	104122			ERROR	122		:NO RDY AFTER SEEK WITH BAD PARITY
5067	016272	004737	044012		JSR	PC,TSTATN		:TEST FOR ATTN

JOB

CZR6HED UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 100
T16 STATIC CYL DIFF & CYL ADDR REG TEST-PART 2

SEQ 0100

5068	016276	104125			ERROR	125		; NO ATTN FROM SEEK & BAD PARITY
5069	016300	012765	100000	000000	MOV	#CCLR,RKCS1(R5)		; CLEAR ERROR
5070	016306	013765	001352	000020	MOV	TOCYL,RKDC(R5)		; RESTOR RKDC AFT CCLR
5071	016314	004737	046014		JSR	PC,RDCYLA		; READ CYL ADDR
5072	016320	023737	001352	001364	CMP	TOCYL,CYLADD		; SEE IF TO CYL ECHOED OK
5073	016326	001401			BEQ	3\$		
5074	016330	104045			ERROR	45		; MR3 NOT=RKDC
5075								
5076	016332	004737	045730		JSR	PC,RDCYLD		; READ CYL DIFF
5077	016336	023737	001360	001362	CMP	CALDIF,CYLDIF		; SEE IF CYL DIFF CORRECT
5078	016344	001401			BEQ	4\$		
5079	016346	104046			ERROR	46		; CYL DIFF IN RKMR2 INCORRECT
5080								
5081	016350							4\$:
5082								
5083	016350	012765	100000	000000	MOV	#CCLR,RKCS1(R5)		
5084	016356	013765	001222	000010	MOV	SUNIT,RKCS2(R5)		; DRIVE#
5085	016364	012737	000005	003322	MOV	#CLEAR,HCS1		
5086	016372	004737	043434		JSR	PC,DOCMD		; DO DRIVE CLEAR CMD & GET CONTR RDY
5087	016376	104151			ERROR	151		; NO RDY AFTER DRIVE CLEAR CMD
5088	016400	004737	044012		JSR	PC,TSTATN		; TEST FOR ATTN
5089	016404	000401			BR	67\$		
5090	016406	104154			ERROR	154		; ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5091	016410							67\$:
5092								
5093								
5094	016410	104415			SCOP1			
5095	016412	012706	001100		MOV	#STACK,SP		; RESTORE STK PTR
5096								
5097	016416	004737	045424		JSR	PC,SUBCLR		
5098	016422	104024			ERROR	24		; CERR AFTER SCLR
5099								
5100	016424	012765	000020	000026	MOV	#PAT,RKMR1(R5)		; EVEN PARITY
5101	016432	013765	001350	000020	MOV	FRCYL,RKDC(R5)		; SET RETURN CYL ADDR
5102	016440	013737	001350	001354	MOV	FRCYL,CCYL		
5103	016446	013737	001354	003364	MOV	CCYL,TEMP3		
5104	016454	013737	001356	003366	MOV	PCYL,TEMP4		
5105	016462	163737	003364	003366	SUB	TEMP3,TEMP4		
5106	016470	100002			BPL	5\$; BR IF TEMP4 IS POS
5107	016472	005437	003366		NEG	TEMP4		
5108	016476	013737	003366	001360	MOV	TEMP4,CALDIF		5\$:
5109	016504	013737	001354	001356	MOV	CCYL,PCYL		
5110	016512	012737	000017	003322	MOV	#SEEK,HCS1		
5111	016520	004737	043434		JSR	PC,DOCMD		; DO SEEK CMD & GET CONTR RDY
5112	016524	104122			ERROR	122		; NO RDY AFTER SEEK WITH BAD PARITY
5113	016526	004737	044012		JSR	PC,TSTATN		; TEST FOR ATTN
5114	016532	104125			ERROR	125		; NO ATTN FROM SEEK & BAD PARITY
5115	016534	012765	100000	000000	MOV	#CCLR,RKCS1(R5)		; CLEAR ERROR
5116	016542	013765	001350	000020	MOV	FRCYL,RKDC(R5)		; RESTOR RKDC AFT CCLR
5117	016550	004737	046014		JSR	PC,RDCYLA		; READ CYL ADDR
5118	016554	023737	001350	001364	CMP	FRCYL,CYLADD		; SEE IF RETURN CYL ECHOED OK
5119	016562	001401			BEQ	6\$		
5120	016564	104241			ERROR	241		; MR3 NOT=RKDC
5121								
5122	016566	023737	001352	001350	CMP	TOCYL,FRCYL		; SEE IF TO=FROM
5123	016574	001022			BNE	10\$; DO NORMAL TEST IF NO

```

5124 016576 005737 001352 TST TOCYL ;SEE IF=0
5125 016602 001007 BNE 9$
5126 016604 004737 045730 JSR PC,RDCYLD ;CYL DIFF S/B 0 ON CYL 0
5127 016610 005737 001362 TST CYLDIF
5128 016614 001421 BEQ 7$
5129 016616 104242 ERROR 242 ;CYL DIFF IN RKMR2 INCORRECT
5130 016620 000417 BR 7$
5131
5132 016622 004737 045730 9$: JSR PC,RDCYLD ;CYL DIFF/OFFSET SHOULD NOT
5133 016626 023727 001362 000001 CMP CYLDIF,#1 ;CHANGE IN SEEK TO SELF
5134 016634 001411 BEQ 7$ ;SHOULD = 1 IN THIS TEST
5135 016636 104263 ERROR 263 ;CYL DIFF IN RKMR2 DID NOT REMAIN = 0
5136 016640 000407 BR 7$
5137 016642 004737 045730 10$: JSR PC,RDCYLD ;READ CYL DIFF
5138 016646 023737 001360 001362 CMP CALDIF,CYLDIF ;SEE IF CYL DIFF OK
5139 016654 001401 BEQ 7$
5140 016656 104242 ERROR 242 ;CYL DIFF IN RKMR2 INCORRECT
5141
5142 016660 7$:
5143
5144 016660 012765 100000 000000 MOV #CLR,RKCS1(R5)
5145 016666 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
5146 016674 012737 000005 003322 MOV #CLEAR,HCS1
5147 016702 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
5148 016706 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
5149 016710 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
5150 016714 000401 BR 68$
5151 016716 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5152 016720 68$:
5153
5154
5155 016720 005237 001352 INC TOCYL
5156 016724 023737 001352 012056 CMP TOCYL,LCP1 ;SEE IF SCANNED ALL CYLS
5157 016732 001402 BEQ 8$ ;BR IF YES
5158 016734 000137 016170 JMP 1$ ;ELSE REPEAT
5159
5160 016740 005237 001350 8$: INC FRCYL
5161 016744 023737 001350 012056 CMP FRCYL,LCP1 ;SEE IF ALL DONE
5162 016752 001405 BEQ TST17 ;GO TO NEXT TST
5163 016754 013737 001350 001352 MOV FRCYL,TOCYL ;FRCYL ALWAYS = OR > TOCYL
5164 016762 000137 016170 JMP 1$ ;ELSE REPEAT
5165 016766 12$:
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179

```

```

*****
:TEST 17 HEAD REGISTER TEST
:
: THIS TEST CHECKS THE ABILITY TO SELECT ALL HEADS (0,1,2)
: VIA RKDA & READING BACK FROM MSG B3 BY THE SELECT DRIVE CMD.
: HEAD 3 IS CHECKED TO PRODUCE INV. ADDR.
:
: SINCE CHANGING HEAD ADDRESSES ARE TIED TO SEEK CMDS,
: SELECTING HEAD 3 MUST RESULT IN A SEEK INCOMPLETE ALONG WITH
: ILLEGAL ADDRESS. IF NOT, THIS MEANS THAT CHANGING HEAD ADDRESSES
: ARE NOT TIED TO SEEK CMDS
:

```

```

*****
S180 016766 000004 000001 001174 ;ST17: SCOPE
S181 016770 012737 000001 001174 MOV #1,STIMES ;DO 1 ITERATION
S182 016776 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
S183 017002 005000 CLR R0 ;HEAD #
S184 017004 104415 001100 1S: SCOP1
S185 017006 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
S186 017012 004737 045424 JSR PC,SUBCLR
S187 017016 104024 ERROR 24 ;CERR AFTER SCLR
S188
S189
S190
S191
S192
S193
S194
S195 017020 012765 100000 000000 MOV #CCLR,RKCS1(R5)
S196 017026 013765 001222 000010 MOV #UNIT,RKCS2(R5)
S197 017034 012737 000013 003322 MOV #RECAL,HCS1
S198 017042 004737 043434 JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
S199 017046 104124 ERROR 124 ;RDY NOT SET AFTER RECAL CMD
S200
S201 017050 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
S202 017056 004737 045074 JSR PC,GSTAT
S203 017062 032737 020000 003350 BIT #D.RTZ,HMR2
S204 017070 001001 BNE 64$
S205 017072 104244 ERROR 244 ;RTZ NOT SET DURING RECAL CMD
S206 017074 013737 001414 003362 64$: MOV T10,TEMP2 ;SETUP TIMEOUT
S207 017102 004737 044044 JSR PC,FATT1 ;FIND ATTN
S208 017106 104055 ERROR 55 ;NO ATTN AFTER RECAL CMD
S209
S210 017110 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
S211 017116 005037 003414 CLR E.B0 ;EXPECTED MSG B0
S212 017122 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
S213 017130 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
S214 017136 005037 003422 CLR E.A2 ;EXPECTED MSG A2
S215 017142 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
S216 017150 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
S217
S218 017156 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0 B0 A1 B1
S219 017162 000007 .WORD T.A2!T.B2!T.B3 ;& MSGS SPECIFIED HERE
S220 017164 104221 ERROR 221 ;MSG A0 ERROR AFTER RECAL CMD
S221 017166 104275 ERROR 275 ;MSG B0 ERROR
S222 017170 104222 ERROR 222 ;MSG A1 ERROR
S223 017172 104276 ERROR 276 ;MSG B1 ERROR
S224
S225 017174 012765 000002 000026 ; CWD2 47,50,<AFTER RECAL CMD>
S226 017202 004737 045074 MOV #2,RKMR1(R5) ;SELECT THE MESSAGE
S227 017206 005737 001364 JSR PC,GSTAT ;GET THE STATUS
S228 017212 001401 TST CYLADD ;RECAL SUCCESSFUL ?
S229 017214 104050 BEQ 65$ ;BRANCH IF SO
S230 017216 65$: ERROR 50 ;REPORT THE ERROR
S231
S232 017216 012765 100000 000000 MOV #CCLR,RKCS1(R5)
S233 017224 013765 001222 000010 MOV #UNIT,RKCS2(R5) ;DRIVE#
S234 017232 012737 000005 003322 MOV #CLEAR,HCS1
S235 017240 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY

```



```

5236 017244 104151          ERROR 151          ;NO RDY AFTER DRIVE CLEAR CMD
5237 017246 004737 044012 JSR PC,TSTATN      ;TEST FOR ATTN
5238 017252 000401          BR 66$
5239 017254 104154          ERROR 154          ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5240 017256
5241
5242 017256 012737 010340 003412 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5243 017264 005037 003414 CLR E.B0           ;EXPECTED MSG B0
5244 017270 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5245 017276 012737 000001 003420 MOV #1,E.B1        ;MSG ID FOR EXPECTED MSG B1
5246 017304 005037 003422 CLR E.A2           ;EXPECTED MSG A2
5247 017310 012737 000002 003424 MOV #2,E.B2        ;MSG ID FOR EXPECTED MSG B2
5248 017316 012737 000003 003430 MOV #3,E.B3        ;MSG ID FOR EXPECTED MSG B3
5249
5250 017324 004737 044252 JSR PC,CHKMSG      ;CHECK MSGS A0,B0,A1,B1
5251 017330 000003          .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5252 017332 104273          ERROR 273         ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5253 017334 104265          ERROR 265         ;MSG B0 ERROR
5254 017336 104274          ERROR 274         ;MSG A1 ERROR
5255 017340 104266          ERROR 266         ;MSG B1 ERROR
5256
5257
5258 017342 023727 001432 000001 CMP HEAD0,#1       ;FOR HEAD 0, B3=1
5259 017350 001401          BEQ 3$
5260 017352 104053          ERROR 53          ;RECAL DID NOT RESET HEAD REG IN B3.
5261
5262
5263 017354          3$: SCOP1
5264 017354 104415          MOV #STACK,SP     ;RESTORE STK PTR
5265 017356 012706 001100
5266 017362 004737 045424 JSR PC,SUBCLR      ;CERR AFTER SCLR
5267 017366 104024          ERROR 24
5268
5269 017370 000300          SWAB R0
5270 017372 010065 000006 MOV R0,RKDA(R5)   ;HEAD #
5271 017376 000300          SWAB R0
5272
5273 017400 012737 000017 003322 MOV #SEEK,HCS1
5274 017406 004737 043434 JSR PC,DOCMD       ;DO SEEK CMD & GET CONTR RDY
5275 017412 104156          ERROR 156         ;NO RDY AFTER SEEK TO SELF
5276 017414 004737 044012 JSR PC,TSTATN
5277 017420 104157          ERROR 157         ;NO ATTN AFTER SEEK TO SELF
5278 017422 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED A0
5279 017430 020027 000003 CMP R0,#3
5280 017434 001403          BEQ 4$           ;BR FOR HEAD 3
5281 017436 005037 003414 CLR E.B0           ;FOR HEADS 0,1,2
5282 017442 000403          BR 5$
5283 017444 012737 002240 003414 4$: MOV #<D.SKI!D.FLT!D.IDAE>,E.B0 ;FOR HEAD 3
5284 017452 012737 001720 003416 5$: MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5285 017460 012737 000001 003420 MOV #1,E.B1
5286 017466 005037 003422 CLR E.A2
5287 017472 012737 000002 003424 MOV #2,E.B2
5288 017500 005700          TST R0           ;SEE IF HEAD 0
5289 017502 001004          BNE 6$          ;BR IF NO
5290 017504 012737 001003 003430 MOV #<BIT9!3>,E.B3 ;LOAD EXPECTED B3 FOR HEAD 0 & MSG ID
5291 017512 000412          BR 8$

```

```

5292 017514 020027 000001      6$:  CMP      RO,#1      ;SEE IF HEAD 1
5293 017520 001004                BNE      7$          ;BR IF NO
5294 017522 012737 002003 003430  MOV      #<BIT10!3>,E.B3 ;B3 FOR HEAD 1
5295 017530 000403                BR       8$          ;
5296 017532 012737 004003 003430  7$:  MOV      #<BIT11!3>,E.B3 ;B3 FOR HEAD 2
5297 017540                8$:
5298
5299 017540 004737 044252                JSR      PC,CHKMSG   ;CHECK MSGS A0,B0,A1,B1
5300 017544 000007                .WORD   T.A2!T.B2!T.B3 ;& MSGS SPECIFIED HERE
5301 017546 104114                ERROR   114         ;MSG A0 ERROR AFTER LOAD HEAD REG & SEEK CMD
5302 017550 104115                ERROR   115         ;MSG B0 ERROR
5303 017552 104322                ERROR   322         ;MSG A1 ERROR
5304 017554 104323                ERROR   323         ;MSG B1 ERROR
5305
5306 017556 012765 000002 000026  MOV      #2,RKMR1(R5) ;SELECT WORD 2
5307 017564 004737 045074                JSR      PC,GSTAT
5308 017570 005737 001362                TST     CYLDIF
5309 017574 001401                BEQ     67$         ;SEE IF MSG A2=0
5310 017576 104324                ERROR   324         ;BR IF YES
5311 017600 005737 001364 67$:  TST     CYLADD
5312 017604 001401                BEQ     68$         ;MSG A2 NOT CLEARED AFTER LOAD HEAD REG & SEEK CMD
5313 017606 104325                ERROR   325         ;SEE IF MSG B2=0
5314 017610                68$:
5315
5316 017610 020027 000003                CMP     RO,#3
5317 017614 001412                BEQ     9$          ;BR IF HEAD 3
5318
5319 017616 005037 003360                CLR     TEMP1
5320 017622 116037 003312 003360  MOV     ATTN(RO),TEMP1
5321 017630 023737 003360 001432  CMP     TEMP1,HEADA ;FOR RKDA=HEAD 0, HEAD=1 IN B3
5322                                     ;FOR RKDA=HEAD 1, HEAD=2 IN B3
5323                                     ;FOR RKDA=HEAD 2, HEAD=4 IN B3
5324 017636 001401                BEQ     9$
5325 017640 104054                ERROR   54          ;HEAD DECODE IN B3 INCORRECT
5326
5327
5328 017642 005200                9$:  INC     RO
5329 017644 020027 000004                CMP     RO,#4
5330 017650 001402                BEQ     10$         ;0 THRU 3 DONE?
5331 017652 000137 017004                JMP     1$          ;BR IF YES
5332                                     ;ELSE REPEAT
5333 017656                10$:
5334
5335 017656 012765 100000 000000  MOV     #CCLR,RKCS1(R5)
5336 017664 013765 001222 000010  MOV     $UNIT,RKCS2(R5)
5337 017672 012737 000013 003322  MOV     #RECAL,HCS1
5338 017700 004737 043434                JSR     PC,DOCMD
5339 017704 104124                ERROR   124        ;DO RECAL CMD & GET CONTR RDY
5340                                     ;RDY NOT FOUND AFTER RECAL CMD
5341 017706 012765 100000 000000  MOV     #CCLR,RKCS1(R5)
5342 017714 013765 001222 000010  MOV     $UNIT,RKCS2(R5) ;DRIVE#
5343 017722 012737 000005 003322  MOV     #CLEAR,HCS1
5344 017730 004737 043434                JSR     PC,DOCMD
5345 017734 104151                ERROR   151        ;DO DRIVE CLEAR CMD & GET CONTR RDY
5346 017736 004737 044012                JSR     PC,TSTATN  ;NO RDY AFTER DRIVE CLEAR CMD
5347 017742 000401                BR      70$        ;TEST FOR ATTN

```

```

5348 017744 104154          ERROR 154          ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5349 017746          70$:
5350
5351
5352 017746 004737 045074          JSR      PC,GSTAT
5353 017752 032737 000040 003352          BIT      #D.IDAE,HMR3 ;SEE IF IDAE IS CLEARED
5354 017760 001401          BEQ      69$          ;BR IF YES
5355 017762 104155          ERROR 155          ;IDAE NOT CLEARED AFTER RECAL CMD
5356
5357 017764 012765 100000 000000 69$:  MOV      #CCLR,RKCS1(R5)
5358 017772 013737 001412 003362          MOV      T1,TEMP2    ;LOOK FOR ATTN FROM RECAL
5359 020000 004737 044044          JSR      PC,FATT1
5360 020004 104055          ERROR 55           ;NO ATTN AFTER RECAL CMD
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372 020006 000004          *ST20: SCOPE
5373 020010 012737 000001 001174          MOV      #1,$TIMES    ;;DO 1 ITERATION
5374 020016 012706 001100          MOV      #STACK,SP   ;RESTORE STK PTR
5375
5376 020022 004737 045424          JSR      PC,SUBCLR   ;SUBSYS CLEAR & GET STATUS
5377 020026 104024          ERROR 24           ;CERR AFTER SCLR
5378 020030 004737 046014          JSR      PC,RDCYLA   ;READ CYL ADDR IN RKMR3
5379 020034 005737 001364          TST      CYLADD
5380 020040 001401          BEQ      1$
5381 020042 104130          ERROR 130          ;CYL ADDR NOT CLEARED AFTER SCLR
5382 020044          1$:
5383 020044 104415          SCOPI
5384 020046 012706 001100          MOV      #STACK,SP   ;RESTORE STK PTR
5385
5386 020052 004737 045424          JSR      PC,SUBCLR   ;CERR AFTER SCLR
5387 020056 104024          ERROR 24
5388
5389 020060 012737 000017 003322          MOV      #SEEK,HCS1
5390 020066 004737 043434          JSR      PC,DOCMD    ;DO SEEK CMD & GET CONTR RDY
5391 020072 104131          ERROR 131          ;RDY NOT SET AFTER SEEK CMD
5392 020074 012737 000005 003360          MOV      #5,TEMP1   ;SETUP 100US TIMEOUT
5393
5394 020102 004737 044140          JSR      PC,FATT2    ;FIND ATTN
5395 020106 104132          ERROR 132          ;NO ATTN AFTER SEEK CMD
5396 020110 032737 100000 003322          BIT      #CERR,HCS1
5397 020116 001401          BEQ      64$
5398 020120 104210          ERROR 210          ;CERR AFTER SEEK CMD
5399 020122          64$:
5400
5401 020122 012737 050340 003412          MOV      #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5402 020130 005037 003414          CLR      E.B0        ;EXPECTED MSG B0
5403 020134 012737 001720 003416          MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1

```

```

*****
*TEST 20      SEEK TO CYL 0
*
*      TESTS THE ABILITY TO DO A SEEK CMD.
*      VERIFIES THERE WAS NO MOVEMENT BY CHECKING ALL APPROPRIATE
*      STATUS BITS.  VERIFIES CMD COMPLETION BETWEEN 10-15USEC.
*      READ HEADER IS NOT PERFORMED AS THE PACK MAY NOT BE FORMATTED.
*
*****

```

```

5404 020142 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5405 020150 005037 003422 CLR E.A2 ;EXPECTED MSG B2
5406 020154 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5407 020162 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5408
5409 020170 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
5410 020174 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5411 020176 104133 ERROR 133 ;MSG A0 ERROR AFTER SEEK CMD
5412 020200 104134 ERROR 134 ;MSG B0 ERROR
5413 020202 104135 ERROR 135 ;MSG A1 ERROR
5414 020204 104136 ERROR 136 ;MSG B1 ERROR
5415 020206 005737 001362 TST CYLDIF
5416 020212 001401 BEQ 65$
5417 020214 104137 ERROR 137 ;CYL DIFF NOT CLEARED AFTER SEEK CMD
5418
5419 020216 65$:
5420
5421 020216 012765 100000 000000 MOV #CCLR,RKCS1(R5)
5422 020224 013765 001222 000010 MOV SUNIT,RKCS2(R5) ;DRIVE#
5423 020232 012737 000005 003322 MOV #CLEAR,HCS1
5424 020240 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
5425 020244 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
5426 020246 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
5427 020252 000401 BR 66$
5428 020254 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5429
5430 66$:
5431 020256 012737 010340 003412 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5432 020264 005037 003414 CLR E.B0 ;EXPECTED MSG B0
5433 020270 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
5434 020276 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5435 020304 005037 003422 CLR E.A2 ;EXPECTED MSG A2
5436 020310 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5437 020316 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5438
5439 020324 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
5440 020330 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5441 020332 104273 ERROR 273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5442 020334 104265 ERROR 265 ;MSG B0 ERROR
5443 020336 104274 ERROR 274 ;MSG A1 ERROR
5444 020340 104266 ERROR 266 ;MSG B1 ERROR
5445
5446 020342 005737 001364 TST CYLADD
5447 020346 001401 BEQ TST21 ;GO TO NEXT TEST
5448 020350 104140 ERROR 140 ;CYL ADDR IN B2 NOT CLEARED AFT SEEK CMD.
5449
5450
5451 ;*****
5452 ;*TEST 21 TEST SECTOR COUNT REG. IN MSG B3
5453 ;*****
5454 020352 000004 TST21: SCOPE
5455 020354 012737 000001 001174 MOV #1,$TIMES ;DO 1 ITERATION
5456 020362 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
5457
5458 020366 004737 045424 JSR PC,SUBCLR ;SUBSYS CLEAR & GET STATUS
5459 020372 104024 ERROR 24 ;CERR AFTER SCLR
020374 012737 020472 001176 MOV #2$, $ESCAPE ;GO TO NEXT TEST IF ANY ERROR DETECTED

```

```

5460 020402 012737 000025 001400      MOV      #21.,SECNT      ;22 SECTOR FORMAT TEST
5461
5462
5463 020410 004737 045554                JSR      PC,FS022        ;FIND SECTOR 0
5464 020414 104142                ERROR    142             ;SECTOR 0 NOT FOUND BY TIMEOUT
5465
5466 020416 005037 001402                CLR      PSEC           ;PREVIOUS SECTOR
5467 020422 004737 045640      64$: JSR      PC,FNS22        ;FIND NEXT SECTOR
5468 020426 104143                ERROR    143             ;DIFFERENT SECTOR NOT FOUND BY TIMEOUT
5469 020430 013737 001402 001404      MOV      PSEC,ESEC
5470 020436 062737 000001 001404      ADD      #1,ESEC        ;SETUP EXPECTED SECTOR
5471 020444 013737 001406 001402      MOV      SECTOR,PSEC    ;UPDATE PREV SECTOR
5472                                     JSR      PC,RDSEC       ;READ SECTOR
5473                                     CMP      SECTOR,PSEC
5474                                     BEQ      65$            ;BR IF READ SAME TWICE
5475                                     JSR      PC,RDSEC
5476                                     CMP      SECTOR,PSEC
5477                                     BEQ      65$
5478                                     ERROR    144            ;TRY 1 MORE TIME
5479                                     ;MSG B3 ERROR, SECTOR REG UNSTABLE
5480 020452 023737 001406 001404      65$: CMP      SECTOR,ESEC ;MAY BE DURING SECTOR PULSE TIME
5481 020460 001401                BEQ      66$
5482 020462 104145                ERROR    145            ;MSG B3 ERROR BETWEEN SECTOR COUNTS
5483 020464 005337 001400      66$: DEC      SECNT
5484 020470 001354                BNE     64$            ;BR IF SECTOR COUNT NOT DONE
5485
5486
5487 020472 005037 001176      2$: CLR      $ESCAPE
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500 020476 000004                ST22: SCOPE
5501 020500 012737 000001 001174      MOV      #1,$TIMES      ;DO 1 ITERATION
5502 020506 012706 001100      MOV      #STACK,SP      ;RESTORE STK PTR
5503
5504 020512 004737 045424                JSR      PC,SUBCLR      ;SUBSYS CLEAR & GET STATUS
5505 020516 104024                ERROR    24             ;CERR AFTER SCLR
5506
5507 020520 005037 001410                CLR      LPFLG
5508 020524 005237 001462                INC      BYPCERR        ;BYPASS CHECKING FOR ANY CERR IN GSTAT1
5509 020530 005237 003304                INC      UNLD           ;USED FOR VALID HALT
5510
5511 020534 012765 000020 000026      MOV      #PAT,RKMR1(R5) ;PARITY & WORD 0
5512 020542 013765 012074 000020      MOV      FCP1,RKDC(R5)
5513 020550 012737 000017 003322      MOV      #SEEK,HCS1
5514 020556 004737 043434                JSR      PC,DOCMD
5515 020562 104122                ERROR    122            ;DO SEEK CMD & GET CONTR READY
;NO RDY FROM SEEK WITH BAD PARITY

```

```

*****
*TEST 22      DETECT OUTER LIMIT
*
*      THIS TEST VERIFIES THAT THE ABOVE TEST DID ACTUALLY POSITION ON CYL 0
*      BY DETECTING OUTER LIMIT AS THE ADJACENT CYL.
*      AN ERROR IN THIS TEST INDICATES:
*
*      A. HEADS WERE NOT ON CYL 0
*      AND/OR B. COULD NOT SEEK IN REVERSE DIRECTION.
*****

```

```

5516 020564 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
5517 020570 104125 ERROR 125 ;NO ATTN FROM SEEK WITH BAD PARITY
5518 020572 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED A0
5519 020600 012737 001200 003414 MOV #<D.FLT!D.PAR>,E.B0
5520 020606 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5521 020614 012737 000001 003420 MOV #1,E.B1
5522
5523 020622 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
5524 020626 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
5525 020630 104110 ERROR 110 ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
5526 020632 104111 ERROR 111 ;MSG B0 ERROR
5527 020634 104146 ERROR 146 ;MSG A1 ERROR
5528 020636 104147 ERROR 147 ;MSG B1 ERROR
5529
5530 020640 012765 100000 000000 MOV #CLR,RKCS1(R5)
5531 020646 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
5532 020654 012737 000005 003322 MOV #CLR,HCS1
5533 020662 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
5534 020666 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
5535 020670 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
5536 020674 000401 BR 65$
5537 020676 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5538 020700
5539
5540 020700 012737 010340 003412 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5541 020706 005037 003414 CLR E.B0 ;EXPECTED MSG B0
5542 020712 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5543 020720 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5544 020726 005037 003422 CLR E.A2 ;EXPECTED MSG A2
5545 020732 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5546 020740 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5547
5548 020746 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
5549 020752 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5550 020754 104273 ERROR 273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5551 020756 104265 ERROR 265 ;MSG B0 ERROR
5552 020760 104274 ERROR 274 ;MSG A1 ERROR
5553 020762 104266 ERROR 266 ;MSG B1 ERROR
5554
5555
5556 020764 013765 012072 000020 MOV FC,RKDC(R5)
5557 020772 012737 000017 003322 MOV #SEEK,HCS1
5558 021000 004737 043434 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
5559 021004 104131 ERROR 131 ;NO RDY AFTER SEEK CMD
5560 021006 012765 100000 000000 MOV #CLR,RKCS1(R5)
5561 021014 004737 045074 JSR PC,GSTAT
5562 021020 004737 046260 JSR PC,FLIM ;FIND LIMIT DETECT
5563 021024 104160 ERROR 160 ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
5564
5565 021026 032737 040000 003350 BIT #D.UNLD,HMR2
5566 021034 001003 BNE 1$
5567 021036 104305 ERROR 305 ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
5568 021040 000137 021556 JMP 305 ;BYPASS REST OF TEST
5569
5570 021044 012737 021474 001176 1$: MOV #20$,SESCAPE ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
5571 021052 012737 070140 003412 MOV #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0

```

```

5572 021060 012737 002200 003414 MOV #<D.SKI!D.FLT> E.B0
5573 021066 012737 045720 003416 MOV #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
5574 021074 012737 030001 003420 MOV #<D.LIMD!D.NMOV!1>,E.B1
5575 ; CHECK 161,162,163,164,<AFTER OUTER LIMIT DETECT>,0,0,0
5576 021102 004737 044252 JSR PC,CHKMSG ;CHECK MESSAGE AD,B0,A1,B1
5577 021106 000000 .WORD 0!0!0
5578 021110 104161 ERROR 161 ;MAY BE A0 ERROR
5579 021112 104162 ERROR 162 ;MAY BE B0 ERROR
5580 021114 104163 ERROR 163 ;MAY BE B1 ERROR
5581 021116 000240 NOP ;NEED FOR THE CALLIN SEQ
5582 021120 032737 020000 003400 BIT #D.LIMD,H.B1 ;SEE IF LIMIT DETECT BIT SET IN B1
5583 021126 001003 BNE 3$ ;BRANCH IF SO
5584 021130 104401 061235 TYPE EM42 ;OTHERWISE REPORT ERROR
5585 021134 104164 ERROR 164
5586
5587 021136 004737 044012 3$: JSR PC,TSTATN
5588 021142 104165 ERROR 165 ;NO ATTN AFTER OUTER LIMIT DETECT
5589 021144 005037 001462 CLR BYPCERR ;ALLOW CHECKING CERR IN GSTAT1
5590
5591 021150 004737 045424 JSR PC,SUBCLR ;SUBSYS CLR
5592 021154 104024 ERROR 24 ;CERR AFTER SCLR
5593 021156 013737 001414 003362 MOV T10,TEMP2 ;SET UP TIMEOUT
5594 021164 004737 046336 JSR PC,FHDHM ;FIND HEAD HOME
5595 021170 104166 ERROR 166 ;HEAD HOME NOT FOUND BEFORE TIMEOUT
5596 021172 004737 046412 JSR PC,FLOAD ;FIND LOAD HEADS
5597 021176 104167 ERROR 167 ;LOAD HEADS NOT FOUND BEFORE TIMEOUT
5598 021200 013737 001416 003362 MOV T100,TEMP2 ;SETUP TIMEOUT
5599 021206 004737 044044 JSR PC,FATT1 ;FIND ATTN
5600 021212 104067 ERROR 67 ;ATTN NOT FOUND BEFORE TIMEOUT
5601 021214 005037 001176 2$: CLR $ESCAPE
5602 021220 005037 003304 CLR UNLD ;CLEAR FLAG
5603
5604 021224 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5605 021232 005037 003414 CLR E.B0 ;EXPECTED MSG B0
5606 021236 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5607 021244 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5608 021252 005037 003422 CLR E.A2 ;EXPECTED MSG A2
5609 021256 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5610 021264 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5611
5612 021272 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
5613 021276 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5614 021300 104063 ERROR 63 ;MSG A0 ERROR AT END OF HEAD LOADING
5615 021302 104064 ERROR 64 ;MSG B0 ERROR
5616 021304 104065 ERROR 65 ;MSG A1 ERROR
5617 021306 104066 ERROR 66 ;MSG B1 ERROR
5618 ; CWD2 175,176,<AT END OF HEAD LOADING>
5619 021310 012765 000002 000026 MOV #2,RKMR1(R5) ;SELECT MESSAGE
5620 021316 004737 045074 JSR PC,GSTAT ;GET STATUS AND MR2,MR3
5621 021322 005737 001364 TST CYLADD ;RECAL SUCCESSFUL?
5622 021326 001401 BEQ 64$ ;BRANCH IF SO
5623 021330 104050 ERROR 50 ;REPORT ERROR
5624 021332 64$:
5625
5626 021332 012765 100000 000000 MOV #CCLR,RKCS1(R5)
5627 021340 013765 001222 000010 MOV #SUNIT,RKCS2(R5) ;DRIVE#

```

```

5628 021346 012737 000005 003322      MOV      #CLEAR,HCS1
5629 021354 004737 043434      JSR      PC,DOCMD      ;DO DRIVE CLEAR CMD & GET CONTR RDY
5630 021360 104151      ERROR   151           ;NO RDY AFTER DRIVE CLEAR CMD
5631 021362 004737 044012      JSR      PC,TSTATN    ;TEST FOR ATTN
5632 021366 000401      BR       66$
5633 021370 104154      ERROR   154           ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5634 021372
5635
5636 021372 012737 010340 003412      MOV      #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
5637 021400 005037 003414      CLR      E.B0         ;EXPECTED MSG B0
5638 021404 012737 001720 003416      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5639 021412 012737 000001 003420      MOV      #1,E.B1      ;MSG ID FOR EXPECTED MSG B1
5640 021420 005037 003422      CLR      E.A2         ;EXPECTED MSG A2
5641 021424 012737 000002 003424      MOV      #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
5642 021432 012737 000003 003430      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
5643
5644 021440 004737 044252      JSR      PC,CHKMSG    ;CHECK MSGS A0,B0,A1,B1
5645 021444 000003      .WORD   T.A2!T.B2!0  ; & MSGS SPECIFIED HERE
5646 021446 104273      ERROR   273           ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
5647 021450 104265      ERROR   265           ;MSG B0 ERROR
5648 021452 104274      ERROR   274           ;MSG A1 ERROR
5649 021454 104266      ERROR   266           ;MSG B1 ERROR
5650
5651 021456 004737 047430      JSR      PC,SWTST     ;SEE IF SW 14 OR 8 IS SET
5652 021462 000435      BR       TST23        ;GO TO NEXT TEST
5653
5654
5655
5656
5657 021464 005037 001176 10$:      CLR      $ESCAPE
5658 021470 000177 157412 20$:      JMP      $SLPADR
5659 021474
5660
5661 021474 004737 045424      JSR      PC,SUBCLR   ;CERR AFTER SCLR
5662 021500 104024      ERROR   24
5663
5664 021502 012737 000011 003322      MOV      #SRTSPL,HCS1
5665 021510 004737 043434      JSR      PC,DOCMD    ;DO START SPINDLE CMD & GET CONTR RDY
5666 021514 104121      ERROR   121          ;RDY NOT FOUND AFTER ST SPIN CMD.
5667
5668 021516 013737 001420 003362      MOV      T500,TEMP2  ;SETUP TIMEOUT
5669 021524 004737 044044      JSR      PC,FATT1    ;FIND ATTN
5670 021530 104067      ERROR   67           ;NO ATTN AFTER ST SPIN CMD.
5671
5672 021532 005037 003304      CLR      UNLD
5673 021536 005237 001410      INC      LPFLG
5674 021542 032777 001000 157370      BIT      #SW9,$SWR   ;LOOP ON ERROR?
5675 021550 001345      BNE     10$         ;YES, RECONDITION DRIVE
5676 021552 000137 021214      JMP      2$         ;RETURN TO MAINLINE
5677 021556
5678
5679
5680
5681
5682
5683

```

::*****


```

; *TEST 23 BASIC WRITE/READ HEADER & HEAD SWITCHING TEST
; *
; * THIS TEST CHECKS HEAD SWITCHING BY WRITING UNIQUE HEADERS
; * ON EACH TRACK OF CYL 0, READING BACK & VERIFYING THEY REMAINED
; * UNIQUE. 22 SECTOR FORMAT IS USED
; *
; * I.E. TRACK 0: ALL 0'S FOR ALL SECTOR HEADERS
; * TRACK 1: 0101 FOR ALL SECTOR HEADERS
; * TRACK 2: ALL 1'S FOR ALL SECTOR HEADERS
; *
; * *****

```

TST23: SCOPE

```

MOV #1, $TIMES ; DO 1 ITERATION
MOV #STACK, SP ; RESTORE STK PTR
TST LIMERR ; CHK FOR LIMIT ERROR
BEQ $$ ; BR IF NO
ERROR 170 ; FATAL ERROR
JMP $EOP ; ABORT BAL OF TESTS

```

```

SS: INC BADHDR ; USED FOR VALID HALT
CLR HEAD ; HEAD CTR

```

```

SCOPI
MOV #STACK, SP ; RESTORE STK PTR

```

```

JSR PC, SUBCLR
ERROR 24 ; CERR AFTER SCLR

```

```

IS: BIS #BAI, RKCS2(R5) ; SET BUSS ADDR INCR INHIBIT
MOV #HDTAB, RKBA(R5) ; HEADER WORD TABLE
MOV #-66., RKWC(R5) ; WORD COUNT.
SWAB HEAD
MOV HEAD, RKDA(R5) ; SETUP HEAD ADDR
SWAB HEAD

```

```

MOV HEAD, RO
ASL RO ; DOUBLE RO
MOV DATA(R0), HDTAB ; SETUP HEADER WORD FOR RKBA

```

```

MOV #<WRHEAD>, HCS1
JSR PC, DATCMD ; DO DATA XFER CMD & GET CONTR RDY
ERROR 200 ; NO RDY AFTER WRITE HEADER CMD
JSR PC, GSTAT ; GET FRESH STATUS
BIT #CERR, HCS1

```

```

BEQ 64$
ERROR 201 ; CERR AFTER WRITE HEADER CMD
TYPE MSG18 ; ABORTING BALANCE OF TESTS
JMP $EOP ; ABORT DRIVE

```

```

64$: MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>, 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

```

5684									
5685									
5686									
5687									
5688									
5689									
5690									
5691									
5692									
5693									
5694									
5695	021556	000004							
5696	021560	012737	000001	001174					
5697	021566	012706	001100						
5698	021572	005737	001460						
5699	021576	001403							
5700	021600	104170							
5701	021602	000137	042662						
5702	021606								
5703									
5704	021606	005237	003306						
5705	021612	005037	001430						
5706									
5707	021616	104415							
5708	021620	012706	001100						
5709									
5710	021624	004737	045424						
5711	021630	104024							
5712									
5713	021632	052765	000020	000010	IS:				
5714	021640	012765	001470	000004					
5715	021646	012765	177676	000002					
5716	021654	000337	001430						
5717	021660	013765	001430	000006					
5718	021666	000337	001430						
5719									
5720	021672	013700	001430						
5721	021676	006300							
5722	021700	016037	001444	001470					
5723									
5724	021706	012737	000027	003322					
5725	021714	004737	043472						
5726	021720	104200							
5727	021722	004737	045074						
5728	021726	032737	100000	003322					
5729	021734	001405							
5730	021736	104201							
5731	021740	104401	056235						
5732	021744	000137	042662						
5733	021750				64\$:				
5734									
5735	021750	012737	010340	003412					
5736	021756	005037	003414						
5737	021762	012737	001720	003416					
5738	021770	012737	000001	003420					
5739	021776	005037	003422						

5740	022002	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5741	022010	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5742							
5743	022016	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
5744	022022	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5745	022024	104277			ERROR	277	;MSG A0 ERROR AFTER WRITE HEADER CMD
5746	022026	104267			ERROR	267	;MSG B0 ERROR
5747	022030	104300			ERROR	300	;MSG A1 ERROR
5748	022032	104270			ERROR	270	;MSG B1 ERROR
5749							
5750							
5751	022034	012765	000002	000026	MOV	#2,RKMR1(R5)	;SELECT WORD 2
5752	022042	004737	045074		JSR	PC,GSTAT	
5753	022046	005737	001362		TST	CYLDIF	;SEE IF MSG A2=0
5754	022052	001401			BEQ	65\$;BR IF YES
5755	022054	104303			ERROR	303	;MSG A2 NOT CLEARED AFTER WRITE HEADER CMD
5756	022056	005737	001364	65\$:	TST	CYLA00	;SEE IF MSG B2=0
5757	022062	001401			BEQ	66\$;BR IF YES
5758	022064	104304			ERROR	304	;MSG B2 NOT CLEARED AFTER WRITE HEADER CMD
5759	022066			66\$:			
5760							
5761	022066	005237	001430		INC	HEAD	
5762	022072	023727	001430	000003	CMP	HEAD,#3	
5763	022100	001254			BNE	1\$	
5764							
5765	022102	005037	001430		CLR	HEAD	;HEAD CTR
5766	022106	104415			SCOP1		
5767	022110	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
5768							
5769	022114	004737	045424		JSR	PC,SUBCLR	
5770	022120	104024			ERROR	24	;CERR AFTER SCLR
5771							
5772							
5773							
5774	022122	000337	001430	2\$:	SWAB	HEAD	
5775	022126	013765	001430	000006	MOV	HEAD,RKDA(R5)	;SETUP HEAD ADDR
5776	022134	000337	001430		SWAB	HEAD	
5777							
5778	022140	012700	001674		MOV	#RHTAB,RO	
5779	022144	012737	000025	003322	MOV	#<RDHEAD>,HCS1	
5780	022152	004737	043472		JSR	PC,DATCMD	;DO DATA XFER CMD & GET CONTR RDY
5781	022156	104171			ERROR	171	;NO RDY AFTER READ HEADER CMD
5782	022160	032737	100000	003322	BIT	#CERR,HCS1	
5783	022166	001405			BEQ	67\$	
5784	022170	104174			ERROR	174	;CERR AFTER READ HEADER CMD
5785	022172	104401	056235		TYPE	MSG18	;ABORT BALANCE OF TESTS
5786	022176	000137	042662		JMP	\$EOP	;ABORT DRIVE
5787							
5788	022202	016520	000024	67\$:	MOV	RKDB(R5),(RO)+	;1'ST WORD FROM SILO TO RHTAB
5789	022206	016520	000024		MOV	RKDB(R5),(RO)+	;2'ND WORD
5790	022212	016520	000024		MOV	RKDB(R5),(RO)+	;3'RD WORD
5791							
5792							
5793	022216	032765	100000	000010	BIT	#DLT,RKCS2(R5)	
5794	022224	001407			BEQ	68\$	
5795	022226	004737	045074		JSR	PC,GSTAT	

```

5796 022232 104173          ERROR 173          ;DLT AFTER READ HEADER CMD
5797 022234 104401 056235  TYPE    MSG18      ;ABORTING BALANCE OF TESTS
5798 022240 000137 042662  JMP     $EOP       ;ABORT DRIVE
5799 022244          68$:
5800
5801 022244 012737 010340 003412  MOV     #<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5802 022252 005037 003414          CLR     E.BC       ;EXPECTED MSG B0
5803 022256 012737 001720 003416  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5804 022264 012737 000001 003420  MOV     #1,E.B1     ;MSG ID FOR EXPECTED MSG B1
5805 022272 005037 003422          CLR     E.A2       ;EXPECTED MSG A2
5806 022276 012737 000002 003424  MOV     #2,E.B2     ;MSG ID FOR EXPECTED MSG B2
5807 022304 012737 000003 003430  MOV     #3,E.B3     ;MSG ID FOR EXPECTED MSG B3
5808
5809 022312 004737 044252          JSR     PC,CHKMSG  ;CHECK MSGS A0,B0,A1,B1
5810 022316 000003          .WORD  T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5811 022320 104301          ERROR 301          ;MSG A0 ERROR AFTER READ HEADER CMD
5812 022322 104271          ERROR 271          ;MSG B0 ERROR
5813 022324 104302          ERROR 302          ;MSG A1 ERROR
5814 022326 104272          ERROR 272          ;MSG B1 ERROR
5815
5816
5817 022330 012765 000002 000026  MOV     #2,RKMR1(R5) ;SELECT WORD 2
5818 022336 004737 045074          JSR     PC,GSTAT
5819 022342 005737 001362          TST     CYLDIF     ;SEE IF MSG A2=0
5820 022346 001401          BEQ     69$        ;BR IF YES
5821 022350 104172          ERROR 172          ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5822 022352 005737 001364          TST     CYLADD     ;SEE IF MSG B2=0
5823 022356 001401          BEQ     70$        ;BR IF YES
5824 022360 104264          ERROR 264          ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5825
5826 022362 000337 001430          SWAB   HEAD        ;RESTORE RKDA
5827 022366 013765 001430 000006  MOV     HEAD,RKDA(R5)
5828 022374 000337 001430          SWAB   HEAD
5829
5830 022400 012701 001674          MOV     #RHTAB,R1
5831
5832 022404 005037 001442          CLR     WDCNT      ;HEADER WORD COUNT
5833 022410 013700 001430          MOV     HEAD,RO
5834 022414 006300          ASL     RO         ;DOUBLE RO
5835 022416 016037 001444 003360  MOV     DATA(RO),TEMP1 ;GET THE 'SHOULD BE' DATA
5836 022424 012137 001454          MOV     (R1)+,HDWD ;READ HEADER WORD
5837 022430 023737 001454 003360  CMP     HDWD,TEMP1
5838 022436 001401          BEQ     4$         ;READ HEADER MISMATCH
5839 022440 104202          ERROR 202
5840 022442 005237 001442          INC     WDCNT      ;DO ONLY 1 SECTOR
5841 022446 023727 001442 000003  CMP     WDCNT,#3
5842 022454 001363          BNE     3$
5843
5844 022456 005237 001430          INC     HEAD       ;ALL 3 HEADS DONE?
5845 022462 023727 001430 000003  CMP     HEAD,#3   ;GO TO NXT TST IF YES
5846 022470 001402          BEQ     TST24
5847 022472 000137 022122          JMP     2$        ;ELSE REPEAT
5848
5849
5850
5851
:*****
:*TEST 24 BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS
:*

```

```

5852          : *      USING HEAD 0, WRITE & READ 20 SECTOR HEADERS BY WRITING ALL
5853          : *      1'S AS HEADERS.  ATTEMPT TO FIND SECTORS 20 & 21.  VERIFY
5854          : *      THEY ARE NO LONGER THERE BY READING 22 SECTORS AND NOT
5855          : *      FINDING 0'S AS DATA FROM THE PREVIOUS TEST.
5856          : *
5857          : * *****
5858          : *
5859          : *
5859 022476 000004          : ST24: SCOPE
5859 022500 012737 000001 001174      MOV      #1,STIMES          ; DO 1 ITERATION
5860 022506 012706 001100          MOV      #STACK,SP        ; RESTORE STK PTR
5861
5862 022512 004737 045424          JSR      PC,SUBCLR
5863 022516 104024          ERROR   24                ; CERR AFTER SCLR
5864 022520 052765 000020 000010      BIS      #BA1,RKCS2(R5)   ; SET BUSS ADDR INCR INHIBIT
5865 022526 012765 001450 000004      MOV      #DATA1,RKBA(R5) ; XFER 1'S ONLY
5866 022534 012765 177704 000002      MOV      #-60.,RKWC(R5)  ; WORD COUNT
5867
5868
5869 022542 012737 010027 003322      MOV      #<CFMT!WRHEAD>,HCS1
5870 022550 004737 043472          JSR      PC,DATCMD        ; DO DATA XFER CMD & GET CONTR RDY
5871 022554 104200          ERROR   200              ; NO RDY AFTER WRITE HEADER CMD
5872 022556 004737 045074          JSR      PC,GSTAT        ; GET FRESH STATUS
5873 022562 032737 100000 003322      BIT      #CERR,HCS1
5874 022570 001405          BEQ     64$
5875 022572 104201          ERROR   201              ; CERR AFTER WRITE HEADER CMD
5876 022574 104401 056235          TYPE   MSG18             ; ABORTING BALANCE OF TESTS
5877 022600 000137 042662          JMP     $EOP              ; ABORT DRIVE
5878          64$:
5879
5880 022604 012737 010001 003322      MOV      #<CFMT!SELDRV>,HCS1 ; GET 20 SECTOR STATUS
5881 022612 004737 043434          JSR      PC,DOCMD        ; DO COMMAND
5882 022616 104117          ERROR   117              ; NO RDY AFTER SELDRV CMD
5883 022620 032737 001000 003350      BIT      #D.FORM,HMR2
5884 022626 001001          BNE    1$
5885 022630 104312          ERROR   312              ; FORMAT NOT SET AFTER WRITE HDR CMD
5886
5887          1$:
5888
5889 022632 012737 010340 003412      MOV      #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ; EXPECTED MSG A0
5890 022640 005037 003414          CLR     E.B0              ; EXPECTED MSG B0
5891 022644 012737 001720 003416      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ; EXPECTED A1
5892 022652 012737 000001 003420      MOV      #1,E.B1          ; MSG ID FOR EXPECTED MSG B1
5893 022660 005037 003422          CLR     E.A2              ; EXPECTED MSG A2
5894 022664 012737 000002 003424      MOV      #2,E.B2          ; MSG ID FOR EXPECTED MSG B2
5895 022672 012737 000003 003430      MOV      #3,E.B3          ; MSG ID FOR EXPECTED MSG B3
5896
5897 022700 004737 044252          JSR      PC,CHKMSG       ; CHECK MSGS A0,B0,A1,B1
5898 022704 000000          .WORD  0!0!0            ; & MSGS SPECIFIED HERE
5899 022706 104277          ERROR   277              ; MSG A0 ERROR AFTER WRITE HEADER CMD
5900 022710 104267          ERROR   267              ; MSG B0 ERROR
5901 022712 104300          ERROR   300              ; MSG A1 ERROR
5902 022714 104270          ERROR   270              ; MSG B1 ERROR
5903
5904 022716 005037 001400          CLR     SECNT            ; SECTOR COUNT
5905          2$:
5906 022722 104415          SCOP1
5907 022724 012706 001100          MOV      #STACK,SP      ; RESTORE STK PTR

```

```

5908
5909 022730 004737 045424 JSR PC,SUBCLR
5910 022734 104024 ERROR 24 ;CERR AFTER SCLR
5911
5912
5913 022736 012700 001674 MOV #RHTAB,R0
5914 022742 012737 010025 003322 MOV #<CFMT!RDHEAD>,HCS1
5915 022750 004737 043472 JSR PC,DATCMD ;DO DATA XFER CMD & GET CONTR RDY
5916 022754 104171 ERROR 171 ;NO RDY AFTER READ HEADER CMD
5917 022756 032737 100000 003322 BIT #CERR,HCS1
5918 022764 001405 BEQ 65$
5919 022766 104174 ERROR 174 ;CERR AFTER READ HEADER CMD
5920 022770 104401 056235 TYPE MSG18 ;ABORT BALANCE OF TESTS
5921 022774 000137 042662 JMP $EOP ;ABORT DRIVE
5922
5923 023000 016520 000024 65$: MOV RKDB(R5),(R0)+ ;1'ST WORD FROM SILO TO RHTAB
5924 023004 016520 000024 MOV RKDB(R5),(R0)+ ;2'ND WORD
5925 023010 016520 000024 MOV RKDB(R5),(R0)+ ;3'RD WORD
5926
5927
5928 023014 032765 100000 000010 BIT #DLT,RKCS2(R5)
5929 023022 001407 BEQ 66$
5930 023024 004737 045074 JSR PC,GSTAT
5931 023030 104173 ERROR 173 ;DLT AFTER READ HEADER CMD
5932 023032 104401 056235 TYPE MSG18 ;ABORTING BALANCE OF TESTS
5933 023036 000137 042662 JMP $EOP ;ABORT DRIVE
5934 023042 66$:
5935
5936 023042 012737 010001 003322 MOV #<CFMT!SELDRV>,HCS1
5937 023050 004737 043434 JSR PC,DOCMD ;DO CMD.
5938 023054 104117 ERROR 117 ;NO RDY AFTER SELDRV CMD
5939 023056 032737 001000 003350 BIT #D.FORM,HMR2
5940 023064 001001 BNE 6$
5941 023066 104313 ERROR 313 ;FORMAT NOT SET AFTER READ HDR CMD
5942
5943 023070 6$:
5944
5945 023070 012737 010340 003412 MOV #<D!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5946 023076 005037 003414 CLR E.B0 ;EXPECTED MSG B0
5947 023102 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5948 023110 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5949 023116 005037 003422 CLR E.A2 ;EXPECTED MSG A2
5950 023122 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5951 023130 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5952
5953 023136 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
5954 023142 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5955 023144 104301 ERROR 301 ;MSG A0 ERROR AFTER READ HEADER CMD
5956 023146 104271 ERROR 271 ;MSG B0 ERROR
5957 023150 104302 ERROR 302 ;MSG A1 ERROR
5958 023152 104272 ERROR 272 ;MSG B1 ERROR
5959
5960 023154 012765 000002 000026 MOV #2,RKMR1(R5) ;SELECT WORD 2
5961 023162 004737 045074 JSR PC,GSTAT
5962 023166 005737 001362 TST CYLDIF ;SEE IF MSG A2=0
5963 023172 001401 BEQ 67$ ;BR IF YES

```

M09

CZR6HED UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 116
T24 BASIC WRITE/READ HEADER TEST; ALL 1'S, 20 SECTORS

SEQ 0116

```

5964 023174 104172          ERROR 172          ;MSG A2 NOT CLEARED AFTER READ HEADER CMD
5965 023176 005737 001364 67$:  TST  CYLADD      ;SEE IF MSG B2=0
5966 023202 001401          BEQ   68$          ;BR IF YES
5967 023204 104264          ERROR 264          ;MSG B2 NOT CLEARED AFTER READ HEADER CMD
5968 023206          68$:  MOV   #PHTAB,R1
5969 023206 012701 001674
5970
5971 023212 005037 001442 3$:  CLR   WDCNT        ;HEADER WORD COUNT
5972 023216 013737 001450 003360  MOV  DATA1,TEMP1 ;GET 'SHOULD BE' DATA
5973 023224 012137 001454          4$:  MOV  (R1)+,HDWD    ;READ HEADER WORD
5974 023230 023737 001454 003360  CMP  HDWD,TEMP1   ;MATCH OK?
5975 023236 001401          BEQ   5$           ;BR IF YES
5976 023240 104202          ERROR 202          ;READ HEADER MISMATCH
5977 023242 005237 001442 5$:  INC  WDCNT        ;JUST 1 SECTOR AND 1 HEAD
5978 023246 023727 001442 000003  CMP  WDCNT,#3
5979 023254 001363          BNE  4$
5980
5981
5982
5983
5984
5985
5986
5987 023256 000004          *****
5988 023260 012737 000001 001174  *TEST 25  WRITE & READ HEADERS CYL 0, HEAD 0
5989 023266 012706 001100          *****
5990
5991 023272 004737 045424          †ST25: SCOPE
5992 023276 104024          MOV  #1,STIMES    ;;DO 1 ITERATION
5993
5994 023300 005237 001464          MOV  #STACK,SP   ;RESTORE STK PTR
5995
5996 023304 012765 001470          JSR  PC,SUBCLR   ;CERR AFTER SCLR
5997 023312 012765 177676 000002          ERROR 24
5998 023320 012737 000000 001352          INC  BYPFMT      ;SET BIT 14 & 15 IN HEADER
5999
6000 023326 013737 001352 001366          MOV  #HDTAB,RKBA(R5) ;HEADER WORD TABLE
6001 023334 012737 000000 001430          MOV  #-66,RKWC(R5) ;WORD COUNT.
6002 023342 012737 000000 001436          MOV  #0,TOCYL
6003 023350 004737 046534          MOV  TOCYL,CALADD ;SETUP
6004
6005 023354 012765 000000 000020          MOV  #0,HEAD     ;TO FILL
6006
6007 023362 012737 000027 003322          MOV  #0,FORMAT   ;HEADER
6008 023370 004737 043472          JSR  PC,FHDTAB   ;TABLE
6009 023374 104200          MOV  #0,RKDC(R5) ;CYL#
6010 023376 004737 045074          MOV  #<WRHEAD>,HCS1
6011 023402 032737 100000 003322          JSR  PC,DAT/CMD ;DO DATA XFER CMD & GET CONTR RDY
6012 023410 001405          ERROR 200        ;NO RDY AFTER WRITE HEADER CMD
6013 023412 104201          JSR  PC,GSTAT    ;GET FRESH STATUS
6014 023414 104401 056235          BIT  #CERR,HCS1
6015 023420 000137 042662          BEQ  64$
6016 023424          ERROR 201        ;CERR AFTER WRITE HEADER CMD
6017
6018 023424 012737 010340 003412          TYPE #MSG18
6019 023432 005037 003414          JMP  $EOP        ;ABORTING BALANCE OF TESTS
                                ;ABORT DRIVE
                                ;EXPECTED MSG AD
                                ;EXPECTED MSG 80

```

N09

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 117
T25 WRITE & READ HEADERS CYL 0, HEAD 0

SEQ 0117

6020	023436	012737	001720	003416		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
6021	023444	012737	000001	003420		MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
6022	023452	005037	003422			CLR	E.A2	;EXPECTED MSG A2
6023	023456	012737	000002	003424		MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
6024	023464	012737	000003	003430		MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
6025								
6026	023472	004737	044252			JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
6027	023476	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
6028	023500	104277				ERROR	277	;MSG A0 ERROR AFTER WRITE HEADER CMD
6029	023502	104267				ERROR	267	;MSG B0 ERROR
6030	023504	104300				ERROR	300	;MSG A1 ERROR
6031	023506	104270				ERROR	270	;MSG B1 ERROR
6032								
6033	023510	005037	001400			CLR	SECNT	;SECTOR COUNT
6034	023514	104415				SCOP1		
6035	023516	012706	001100			MOV	#STACK,SP	;RESTORE STK PTR
6036								
6037	023522	004737	045424			JSR	PC,SUBCLR	
6038	023526	104024				ERROR	24	;CERR AFTER SCLR
6039								
6040	023530	012765	000000	000020		MOV	#0,RKDC(R5)	;CYL #
6041								
6042	023536	012700	001674			MOV	#RHTAB,RO	
6043								
6044	023542	012737	000025	003322	65\$:	MOV	#RDHEAD,HCS1	
6045	023550	004737	043472			JSR	PC,DATCMD	;DO READ HEADER CMD & GET CONTR RDY
6046	023554	104171				ERROR	171	;NO RDY AFTER READ HEADER CMD
6047	023556	032737	100000	003322		BIT	#CERR,HCS1	
6048	023564	001405				BEQ	66\$	
6049	023566	104174				ERROR	174	;CERR AFTER READ HEADER CMD
6050	023570	104401	056235			TYPE	MSG18	;ABORTING BALANCE OF TESTS
6051	023574	000137	042662			JMP	\$EOP	;ABORT DRIVE
6052								
6053	023600	016520	000024		66\$:	MOV	RKDB(R5),(RO)+	;1'ST WORD FROM SILO TO RHTAB
6054	023604	016520	000024			MOV	RKDB(R5),(RO)+	;2'ND WORD
6055	023610	016520	000024			MOV	RKDB(R5),(RO)+	;3'RD WORD
6056								
6057	023614	032765	100000	000010		BIT	#DLT,RKCS2(R5)	;SEE IF DATA LATE
6058	023622	001407				BEQ	67\$	
6059	023624	004737	045074			JSR	PC,GSTAT	
6060	023630	104173				ERROR	173	;DATA LATE ON READ HEADER
6061	023632	104401	056235			TYPE	MSG18	;ABORT BALANCE OF TESTS
6062	023636	000137	042662			JMP	\$EOP	;ABORT DRIVE
6063								
6064	023642	020027	002100		67\$:	CMP	RO,#RHTAB+132.	;ALL 66 WORDS DONE?
6065	023646	001335				BNE	65\$;BR IF NO
6066								
6067	023650	004737	047056			JSR	PC, SORT	;SORT RHTAB INTO SRTTAB SO THAT IT
6068								;BEGINS WITH SECTOR 0
6069	023654	005037	001442			CLR	WDCNT	;WORD COUNT
6070	023660	012700	002100			MOV	#SRTTAB,RO	;ACTUAL HEADER TABLE
6071	023664	012701	001470			MOV	#HDTAB,R1	;CALC HEADER TABLE
6072								
6073	023670	012037	001454		68\$:	MOV	(RO)+,HDWD	
6074	023674	012137	003360			MOV	(R1)+,TEMP1	
6075	023700	023737	001454	003360		CMP	HDWD,TEMP1	;COMPARE ACTUAL WITH CALCULATED WORD

6076	023706	001401			BEG	69\$;BR IF COMPARE
6077	023710	104202			ERROR	202		;READ HEADER MISMATCH
6078								
6079	023712	005237	001442		69\$: INC	WDCNT		
6080	023716	023727	001442	000102	CMP	WDCNT, #66.		;ALL WORDS DONE?
6081	023724	001361			BNE	68\$;BR IF NO
6082								
6083								
6084	023726	005037	001464		CLR BYPFMT			;ALLOW CORRECT FORMATTING
6085								
6086								
6087								
6088								
6089								
6090								
6091								
6092								
6093								
6094								
6095								
6096								
6097								
6098								
6099								
6100								
6101								
6102								

```
*****
*TEST 26          SEEK FROM CYL 0 TO 1 & READ HEADERS
*****
```

```
*
* THIS TEST CHECKS MSG A & B WORDS 0,1,2 FOR CORRECT STATUS AFTER RDY
* IS RECEIVED FROM A SEEK CMD TO DETERMINE
* THAT THE HEADS ARE ACTUALLY MOVING & THE CYL DIFF IS 1.
* AFTER ATTN IS RECEIVED, CERR IS EXAMINED FOR ANY ERRORS.
* CYL DIFFERENCE IN MSG A2 IS VERIFIED TO BE 0 & CYL ADDR
* IN MSG B2 IS VERIFIED TO BE 1.
*
* HEADERS ARE READ FROM 1 SECTOR, HEAD 0 & VERIFIED THAT THEY ARE
* DIFFERENT FROM CYL 0 TO SHOW THAT THE HEADS DID ACTUALLY MOVE.
*
*****
```

```
†ST26:  SCOPE
        MOV      #1,STIMES          ;;DO 1 ITERATION
        MOV      #STACK,SP         ;;RESTORE STK PTR
        JSR      PC,SUBCLR
        ERROR    24                  ;CERR AFTER SCLR
        CLR      FRCYL
        MOV      #1,TOCYL
        MOV      #1,CALDIF
        MOV      #1,RKDC(R5)        ;SET FOR CYL 1
        MOV      #10$,SESCAPE

        MOV      #SEEK,HCS1
        JSR      PC,DOCMD           ;;DO SEEK CMD & GET CONTR READY
        ERROR    131                ;NO RDY AFTER SEEK CMD
        MOV      #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
        CLR      E.B0
        MOV      #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
        MOV      #1,E.B1

        JSR      PC,CHKMSG          ;CHECK MSGS A0,B0,A1,B1
        .WORD    T.A2!T.B2!0        ;& MSGS SPECIFIED HERE
        ERROR    203                ;MSG A0 ERROR DURING SEEK CMD
        ERROR    204                ;MSG B0 ERROR
        ERROR    205                ;MSG A1 ERROR
        ERROR    206                ;MSG B1 ERROR

        CMP      CYLDIF,#1
        BEQ      1$
        ERROR    212                ;CYL DIFF INCORRECT DURING SEEK CMD.
```

6103	023732	000004		
6104	023734	012737	000001	001174
6105	023742	012706	001100	
6106	023746	004737	045424	
6107	023752	104024		
6108	023754	005037	001350	
6109	023760	012737	000001	001352
6110	023766	012737	000001	001360
6111	023774	012765	000001	000020
6112	024002	012737	024766	001176
6113				
6114	024010	012737	000017	003322
6115	024016	004737	043434	
6116	024022	104131		
6117	024024	012737	030140	003412
6118	024032	005037	003414	
6119	024036	012737	003720	003416
6120	024044	012737	000001	003420
6121				
6122	024052	004737	044252	
6123	024056	000003		
6124	024060	104203		
6125	024062	104204		
6126	024064	104205		
6127	024066	104206		
6128				
6129	024070	023727	001362	000001
6130	024076	001401		
6131	024100	104212		

C10

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 119
T26 SEEK FROM CYL 0 TO 1 & READ HEADERS

SEQ 0119

6132									
6133									
6134	024102	012737	025006	001176	1\$:	MOV	#12\$, \$ESCAPE		
6135	024110	013737	001422	003360		MOV	T2500, TEMP1	; SETUP TIMEOUT	
6136									
6137									
6138	024116	004737	044140			JSR	PC, FATT2	; FIND ATTN	
6139	024122	104132				ERROR	132	; NO ATTN AFTER SEEK CMD	
6140	024124	032737	100000	003322		BIT	#CERR, HCS1		
6141	024132	001401				BEQ	64\$		
6142	024134	104210				ERROR	210	; CERR AFTER SEEK CMD	
6143	024136				64\$:				
6144									
6145	024136	012737	050340	003412		MOV	#<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>, E.A0	; EXPECTED MSG A0	
6146	024144	005037	003414			CLR	E.B0	; EXPECTED MSG B0	
6147	024150	012737	001720	003416		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>, E.A1	; EXPECTED A1	
6148	024156	012737	000001	003420		MOV	#1, E.B1	; MSG ID FOR EXPECTED MSG B1	
6149	024164	005037	003422			CLR	E.A2	; EXPECTED MSG A2	
6150	024170	012737	000002	003424		MOV	#2, E.B2	; MSG ID FOR EXPECTED MSG B2	
6151	024176	012737	000003	003430		MOV	#3, E.B3	; MSG ID FOR EXPECTED MSG B3	
6152									
6153	024204	004737	044252			JSR	PC, CHKMSG	; CHECK MSGS A0, B0, A1, B1	
6154	024210	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE	
6155	024212	104133				ERROR	133	; MSG A0 ERROR AFTER SEEK CMD	
6156	024214	104134				ERROR	134	; MSG B0 ERROR	
6157	024216	104135				ERROR	135	; MSG A1 ERROR	
6158	024220	104136				ERROR	136	; MSG B1 ERROR	
6159	024222	005737	001362			TST	CYLDIF		
6160	024226	001401				BEQ	65\$		
6161	024230	104137				ERROR	137	; CYL. DIFF NOT CLEARED AFTER SEEK CMD	
6162									
6163	024232				65\$:				
6164									
6165	024232	012765	100000	000000		MOV	#CLR, RKCS1(R5)		
6166	024240	013765	001222	000010		MOV	\$UNIT, RKCS2(R5)	; DRIVE#	
6167	024246	012737	000005	003322		MOV	#CLEAR, HCS1		
6168	024254	004737	043434			JSR	PC, DOCMD	; DO DRIVE CLEAR CMD & GET CONTR RDY	
6169	024260	104151				ERROR	151	; NO RDY AFTER DRIVE CLEAR CMD	
6170	024262	004737	044012			JSR	PC, TSTATN	; TEST FOR ATTN	
6171	024266	000401				BR	66\$		
6172	024270	104154				ERROR	154	; ATTN NOT CLEARED AFTER DRIVE CLEAR CMD	
6173	024272				66\$:				
6174									
6175	024272	012737	010340	003412		MOV	#<D.SPIN!D.DRDY!D.VV!D.DRA>, E.A0	; EXPECTED MSG A0	
6176	024300	005037	003414			CLR	E.B0	; EXPECTED MSG B0	
6177	024304	012737	001720	003416		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>, E.A1	; EXPECTED A1	
6178	024312	012737	000001	003420		MOV	#1, E.B1	; MSG ID FOR EXPECTED MSG B1	
6179	024320	005037	003422			CLR	E.A2	; EXPECTED MSG A2	
6180	024324	012737	000002	003424		MOV	#2, E.B2	; MSG ID FOR EXPECTED MSG B2	
6181	024332	012737	000003	003430		MOV	#3, E.B3	; MSG ID FOR EXPECTED MSG B3	
6182									
6183	024340	004737	044252			JSR	PC, CHKMSG	; CHECK MSGS A0, B0, A1, B1	
6184	024344	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE	
6185	024346	104273				ERROR	273	; MSG A0 ERROR AFTER DRIVE CLEAR CMD	
6186	024350	104265				ERROR	265	; MSG B0 ERROR	
6187	024352	104274				ERROR	274	; MSG A1 ERROR	

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00MACY11 30A(1052) 25-JAN-78 12:08 PAGE 120
T26 SEEK FROM CYL 0 TO 1 & READ HEADERS

SEQ 0120

6188	024354	104266			ERROR	266		;MSG B1 ERROR
6189								
6190	024356	005737	001364		TST	CYLADD		
6191	024362	023727	001364	000001	CMP	CYLADD,#1		
6192	024370	001401			BEQ	25		
6193	024372	104207			ERROR	207		;CYL ADDR INCORRECT AFTER SEEK CMD
6194								
6195								
6196	024374				25:			
6197	024374	104415			SCOP1			
6198	024376	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
6199								
6200	024402	004737	045424		JSR	PC,SUBCLR		
6201	024406	104024			ERROR	24		;CERR AFTER SCLR
6202								
6203	024410	005037	001176		CLR	\$ESCAPE		
6204	024414	012765	000001	000020	MOV	#1,RKDC(R5)		;CYL #
6205								
6206	024422	012700	001674		MOV	#RHTAB,RO		
6207	024426	012737	000025	003322	MOV	#<RDHEAD>,HCS1		
6208	024434	004737	043472		JSR	PC,DATCMD		;DO DATA XFER CMD & GET CONTR RDY
6209	024440	104171			ERROR	171		;NO RDY AFTER READ HEADER CMD
6210	024442	032737	100000	003322	BIT	#CERR,HCS1		
6211	024450	001405			BEQ	675		
6212	024452	104174			ERROR	174		;CERR AFTER READ HEADER CMD
6213	024454	104401	056235		TYPE	MSG18		;ABORT BALANCE OF TESTS
6214	024460	000137	042662		JMP	\$EOP		;ABORT DRIVE
6215								
6216	024464	016520	000024		675:	MOV	RKDB(R5),(RO)+	;1'ST WORD FROM SILO TO RHTAB
6217	024470	016520	000024		MOV	RKDB(R5),(RO)+		;2'ND WORD
6218	024474	016520	000024		MOV	RKDB(R5),(RO)+		;3'RD WORD
6219								
6220								
6221	024500	032765	100000	000010	BIT	#DLT,RKCS2(R5)		
6222	024506	001407			BEQ	685		
6223	024510	004737	045074		JSR	PC,GSTAT		
6224	024514	104173			ERROR	173		;DLT AFTER READ HEADER CMD
6225	024516	104401	056235		TYPE	MSG18		;ABORTING BALANCE OF TESTS
6226	024522	000137	042662		JMP	\$EOP		;ABORT DRIVE
6227	024526				685:			
6228								
6229								
6230	024526	005737	001674		TST	RHTAB		;CHECK 1'ST WORD ONLY:CYL#
6231	024532	001001			BNE	35		
6232	024534	104211			ERROR	211		;CYL 0 HEADER ON CYL 1
6233								
6234	024536	013737	001674	001454	35:	MOV	RHTAB,HDWD	
6235	024544	012737	000001	003360	MOV	#1,TEMP1		
6236	024552	023737	001454	003360	CMP	HDWD,TEMP1		
6237	024560	001401			BEQ	45		
6238	024562	104202			ERROR	202		;READ CYL WORD HEADER ERROR
6239	024564				45:			
6240	024564	004737	047430		JSR	PC,SWTST		;SEE IF SW 14 OR 8 IS SET
6241	024570	000516			BR	TST27		::GO TO NEXT TEST
6242								;RETURN HERE IF SW 14 IS SET OR
6243								;SW 8 WITH SWR <7:0> APPLY


```

6300
6301
6302
6303
6304 025026 000004
6305 025030 012737 000001 001174
6306 025036 012706 001100
6307
6308 025042 004737 045424
6309 025046 104024
6310
6311 025050 005237 001464
6312
6313 025054 012765 001470 000004
6314 025062 012765 177676 000002
6315 025070 012737 000001 001352
6316
6317 025076 013737 001352 001366
6318 025104 012737 000000 001430
6319 025112 012737 000000 001436
6320 025120 004737 046534
6321
6322 025124 012765 000001 000020
6323
6324 025132 012737 000027 003322
6325 025140 004737 043472
6326 025144 104200
6327 025146 004737 045074
6328 025152 032737 100000 003322
6329 025160 001405
6330 025162 104201
6331 025164 104401 056235
6332 025170 000137 042662
6333 025174
6334
6335 025174 012737 010340 003412
6336 025202 005037 003414
6337 025206 012737 001720 003416
6338 025214 012737 000001 003420
6339 025222 005037 003422
6340 025226 012737 000002 003424
6341 025234 012737 000003 003430
6342
6343 025242 004737 044252
6344 025246 000003
6345 025250 104277
6346 025252 104267
6347 025254 104300
6348 025256 104270
6349
6350 025260 005037 001400
6351 025264 104415
6352 025266 012706 001100
6353
6354 025272 004737 045424
6355 025276 104024

```

```

*****
;TEST 27 WRITE & READ HEADERS CYL 1, HEAD 0
*****
†ST27: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
INC BYPFMT ;SET BIT 14 & 15 IN HEADER
MOV #HDTAB,RKBA(R5) ;HEADER WORD TABLE
MOV #-66,RKWC(R5) ;WORD COUNT.
MOV #1,TOCYL
MOV TOCYL,CALADD ;SETUP
MOV #0,HEAD ;TO FILL
MOV #0,FORMAT ;HEADER
JSR PC,FHDTAB ;TABLE
MOV #1,RKDC(R5) ;CYL#
MOV #<WRHEAD>,HCS1
JSR PC,DATCMD ;DO DATA XFER CMD & GET CONTR ROY
ERROR 200 ;NO ROY AFTER WRITE HEADER CMD
JSR PC,GSTAT ;GET FRESH STATUS
BIT #CERR,HCS1
BEQ 645
ERROR 201 ;CERR AFTER WRITE HEADER CMD
TYPE MSG18 ;ABORTING BALANCE OF TESTS
JMP $EOP ;ABORT DRIVE
645:
MOV #<O!D.SPIN!D.DRDY!D.VV!D.DRA>,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 277 ;MSG A0 ERROR AFTER WRITE HEADER CMD
ERROR 267 ;MSG B0 ERROR
ERROR 300 ;MSG A1 ERROR
ERROR 270 ;MSG B1 ERROR
CLR SECNT ;SECTOR COUNT
SCOP1
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR

```

G10

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 123
T27 WRITE & READ HEADERS CYL 1, HEAD 0

SEQ 0123

```

6356
6357 025300 012765 000001 000020      MOV      #1,RKDC(R5)      ;CYL #
6358
6359 025306 012700 001674                MOV      #RHTAB,R0
6360
6361 025312 012737 000025 003322 65$:  MOV      #RDHEAD,HCS1
6362 025320 004737 043472                JSR      PC,GATCMD      ;DO READ HEADER CMD & GET CONTR RDY
6363 025324 104171                ERROR   171            ;NO RDY AFTER READ HEADER CMD
6364 025326 032737 100000 003322      BIT      #CERR,HCS1
6365 025334 001405                BEQ      66$
6366 025336 104174                ERROR   174            ;CERR AFTER READ HEADER CMD
6367 025340 104401 056235                TYPE    MSG18          ;ABORTING BALANCE OF TESTS
6368 025344 000137 042662                JMP      $EOP          ;ABORT DRIVE
6369
6370 025350 016520 000024                66$:  MOV      RKDB(R5),(R0)+  ;1'ST WORD FROM SILO TO RHTAB
6371 025354 016520 000024                MOV      RKDB(R5),(R0)+  ;2'ND WORD
6372 025360 016520 000024                MOV      RKDB(R5),(R0)+  ;3'RD WORD
6373
6374 025364 032765 100000 000010      BIT      #DLT,RKCS2(R5) ;SEE IF DATA LATE
6375 025372 001407                BEQ      67$
6376 025374 004737 045074                JSR      PC,GSTAT
6377 025400 104173                ERROR   173            ;DATA LATE ON READ HEADER
6378 025402 104401 056235                TYPE    MSG18          ;ABORT BALANCE OF TESTS
6379 025406 000137 042662                JMP      $EOP          ;ABORT DRIVE
6380
6381 025412 020027 002100                67$:  CMP      R0,#RHTAB+132.  ;ALL 66 WORDS DONE?
6382 025416 001335                BNE     65$            ;BR IF NO
6383
6384 025420 004737 047056                JSR      PC, SORT      ;SORT RHTAB INTO SRTTAB SO THAT IT
6385                                ;BEGINS WITH SECTOR 0
6386 025424 005037 001442                CLR      WDCNT          ;WORD COUNT
6387 025430 012700 002100                MOV      #SRTTAB,R0    ;ACTUAL HEADER TABLE
6388 025434 012701 001470                MOV      #HDTAB,R1     ;CALC HEADER TABLE
6389
6390 025440 012037 001454                68$:  MOV      (R0)+,HDWD
6391 025444 012137 003360                MOV      (R1)+,TEMP1
6392 025450 023737 001454 003360      CMP      HDWD,TEMP1    ;COMPARE ACTUAL WITH CALCULATED WORD
6393 025456 001401                BEQ      69$
6394 025460 104202                ERROR   202            ;BR IF COMPARE
6395                                ;READ HEADER MISMATCH
6396 025462 005237 001442                69$:  INC      WDCNT
6397 025466 023727 001442 000102      CMP      WDCNT,#66.    ;ALL WORDS DONE?
6398 025474 001361                BNE     68$            ;BR IF NO
6399
6400
6401 025476 005037 001464                CLR      BYPFMT        ;ALLOW CORRECT FORMATTING
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411

```

```

*****
*TEST 30      TEST RECALIBRATE CMD & READ HEADERS
*
*      THIS TEST DOES A RECALIBRATE & READS HEADERS.
*      IT VERIFIES THAT WRITING HEADERS ON CYL 1 FROM THE PREVIOUS
*      TEST DID NOT OVERWRITE CYL 0 HEADERS.
*      AN ERROR IN THIS TEST INDICATES THAT HEADS:
*

```

```

6412
6413
6414
6415
6416
6417
6418 025502 000004
6419 025504 012737 000001 001174
6420 025512 012706 001100
6421
6422 025516 004737 045424
6423 025522 104024
6424 025524 012737 000001 001350
6425 025532 005037 001352
6426 025536 012737 000001 001360
6427 025544 012737 026542 001176
6428 025552 012737 000013 003322
6429 025560 004737 043434
6430 025564 104124
6431 025566 012765 100000 000000
6432 025574 012765 000001 000026
6433 025602 004737 045074
6434 025606 032737 020000 003350
6435 025614 001001
6436 025616 104307
6437
6438 025620 012737 030140 003412 1$:
6439 025626 005037 003414
6440 025632 012737 025720 003416
6441 025640 012737 000001 003420
6442
6443 025646 004737 044252
6444 025652 000001
6445 025654 104213
6446 025656 104214
6447 025660 104215
6448 025662 104216
6449
6450
6451
6452
6453 025664 012737 026562 001176 2$:
6454 025672 012737 177777 003360
6455 025700 004737 044140
6456 025704 104055
6457 025706 032737 100000 003322
6458 025714 001401
6459 025716 104220
6460 025720
6461
6462 025720 012737 050340 003412
6463 025726 005037 003414
6464 025732 012737 001720 003416
6465 025740 012737 000001 003420
6466 025746 005037 003422
6467 025752 012737 000002 003424

```

```

;*
;*
;* OR A. MOVED TO A CYL OTHER THAN 1
;* B. DID NOT GET BACK TO CYL 0
;*
*****
TST30: SCOPE
MOV #1,$TIMES ;:DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
MOV #1,FRCYL ;PARAMETERS
CLR TOCYL ;FOR
MOV #1,CALDIF ;ERROR TYPEOUTS
MOV #10,$$ESCAPE
MOV #RECAL,HCS1
JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
ERROR 124 ;NO RDY AFTER RECAL CMD
MOV #CCLR,RKCS1(R5)
MOV #1,RKMR1(R5) ;SELECT WORD 1
JSR PC,GSTAT
BIT #D,RTZ,HMR2
BNE 1$
ERROR 307 ;RTZ NOT SET DURING RECAL CMD
1$: MOV #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
CLR E.B0
MOV #<D.RTZ!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
MOV #1,E.B1
JSR PC,CHKMSG ;CHECK MSGS A0 B0 A1 B1
; & MSGS SPECIFIED HERE
WORD T,A2!0!0
ERROR 213 ;MSG A0 ERROR DURING RECAL CMD
ERROR 214 ;MSG B0 ERROR
ERROR 215 ;MSG A1 ERROR
ERROR 216 ;MSG B1 ERROR
TST CYLDIF
BEQ 2$
ERROR 217 ;CYL DIFF INCORRECT DURING RECAL CMD.
2$: MOV #12,$$ESCAPE
MOV #-1,TEMP1 ;SETUP TIMEOUT
JSR PC,FATT2 ;FIND ATTN
ERROR 55 ;NO ATTN AFTER RECAL CMD
BIT #CERR,HCS1
BEQ 3$
ERROR 220 ;CERR AFTER RECAL CMD
3$: MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,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

```

6468	025760	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
6469							
6470	025766	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
6471	025772	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
6472	025774	104221			ERROR	221	;MSG A0 ERROR AFTER RECAL CMD
6473	025776	104275			ERROR	275	;MSG B0 ERROR
6474	026000	104222			ERROR	222	;MSG A1 ERROR
6475	026002	104276			ERROR	276	;MSG B1 ERROR
6476							
6477	026004	012765	000002	000026	MOV	#2,RKMR1(R5)	;SELECT WORD 2
6478	026012	004737	045074		JSR	PC,GSTAT	
6479	026016	005737	001362		TST	CYLDIF	;SEE IF MSG A2=0
6480	026022	001401			BEQ	64\$;BR IF YES
6481	026024	104047			ERROR	47	;MSG A2 NOT CLEARED AFTER RECAL CMD
6482	026026	005737	001364	64\$:	TST	CYLADD	;SEE IF MSG B2=0
6483	026032	001401			BEQ	65\$;BR IF YES
6484	026034	104050			ERROR	50	;MSG B2 NOT CLEARED AFTER RECAL CMD
6485	026036			65\$:			
6486							
6487	026036	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	
6488	026044	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	;DRIVE#
6489	026052	012737	000005	003322	MOV	#CLEAR,HCS1	
6490	026060	004737	043434		JSR	PC,DOCMD	;DO DRIVE CLEAR CMD & GET CONTR RDY
6491	026064	104151			ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD
6492	026066	004737	044012		JSR	PC,TSTATN	;TEST FOR ATTN
6493	026072	000401			BR	66\$	
6494	026074	104154			ERROR	154	;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6495	026076			66\$:			
6496							
6497	026076	012737	010340	003412	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
6498	026104	005037	003414		CLR	E.B0	;EXPECTED MSG B0
6499	026110	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
6500	026116	012737	000001	003420	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
6501	026124	005037	003422		CLR	E.A2	;EXPECTED MSG A2
6502	026130	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
6503	026136	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
6504							
6505	026144	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
6506	026150	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
6507	026152	104273			ERROR	273	;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6508	026154	104265			ERROR	265	;MSG B0 ERROR
6509	026156	104274			ERROR	274	;MSG A1 ERROR
6510	026160	104266			ERROR	266	;MSG B1 ERROR
6511							
6512	026162	004737	045424	4\$:	JSR	PC,SUBCLR	
6513	026166	104024			ERROR	24	;CERR AFTER SCLR
6514							
6515	026170	005037	001176		CLR	\$ESCAPE	
6516							
6517	026174	012700	001674		MOV	#RHTAB,RO	
6518	026200	012737	000025	003322	MOV	#<RDHEAD>,HCS1	
6519	026206	004737	043472		JSR	PC,DATCMD	;DO DATA XFER CMD & GET CONTR RDY
6520	026212	104171			ERROR	171	;NO RDY AFTER READ HEADER CMD
6521	026214	032737	100000	003322	BIT	#CERR,HCS1	
6522	026222	001405			BEQ	67\$	
6523	026224	104174			ERROR	174	;CERR AFTER READ HEADER CMD

J10

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 126
T30 TEST RECALIBRATE CMD & READ HEADERS

SEQ 0126

6524	026226	104401	056235			TYPE	MSG18		;ABORT BALANCE OF TESTS
6525	026232	000137	042662			JMP	\$EOP		;ABORT DRIVE
6526									
6527	026236	016520	000024		67\$:	MOV	RKDB(R5),(R0)+		;1'ST WORD FROM SILO TO RHTAB
6528	026242	016520	000024			MOV	RKDB(R5),(R0)+		;2'ND WORD
6529	026246	016520	000024			MOV	RKDB(R5),(R0)+		;3'RD WORD
6530									
6531									
6532	026252	032765	100000	000010		BIT	#DLT,RKCS2(R5)		
6533	026260	001407				BEQ	68\$		
6534	026262	004737	045074			JSR	PC,GSTAT		
6535	026266	104173				ERROR	173		;DLT AFTER READ HEADER CMD
6536	026270	104401	056235			TYPE	MSG18		;ABORTING BALANCE OF TESTS
6537	026274	000137	042662			JMP	\$EOP		;ABORT DRIVE
6538	026300				68\$:				
6539									
6540	026300	023727	001674	000001		CMP	RHTAB,#1		;CHECK WORD 0, CYL # ONLY
6541	026306	001001				BNE	5\$		
6542	026310	104240				ERROR	240		;CYL 1 HEADERS ON CYL 0
6543									
6544	026312	005737	001674		5\$:	TST	RHTAB		
6545	026316	001401				BEQ	6\$		
6546	026320	104202				ERROR	202		;READ CYL WORD HEADER ERROR
6547	026322				6\$:				
6548	026322	004737	047430			JSR	PC,SWTST		;SEE IF SW 14 OR 8 IS SET
6549	026326	000525				BR	TST31		;GO TO NEXT TEST
6550									;RETURN HERE IF SW 14 IS SET OR
6551									;SW 8 WITH SWR <7:0> APPLY
6552	026330	004737	045424		8\$:	JSR	PC,SUBCLR		
6553	026334	104024				ERROR	24		;CERR AFTER SCLR
6554	026336	012765	000001	000020		MOV	#1,RKDC(R5)		;RECONDITION BACK TO CYL 1
6555									
6556	026344	012737	000017	003322		MOV	#SEEK,HCS1		
6557	026352	004737	043434			JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
6558	026356	104131				ERROR	131		;NO RDY AFTER SEEK CMD.
6559									
6560	026360	013737	001426	003360		MOV	T50000,TEMP1		
6561	026366	004737	044140			JSR	PC,FATT2		;FIND ATTN
6562	026372	104132				ERROR	132		;NO ATTN AFTER SEEK CMD
6563	026374	032737	100000	003322		BIT	#CERR,HCS1		
6564	026402	001401				BEQ	69\$		
6565	026404	104210				ERROR	210		;CERR AFTER SEEK CMD.
6566									
6567	026406	004737	045424		69\$:	JSR	PC,SUBCLR		
6568	026412	104024				ERROR	24		;CERR AFTER SCLR
6569									
6570									
6571									
6572	026414	012737	010340	003412		MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
6573	026422	005037	003414			CLR	E.B0		;EXPECTED MSG B0
6574	026426	012737	001720	003416		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
6575	026434	012737	000001	003420		MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
6576	026442	005037	003422			CLR	E.A2		;EXPECTED MSG A2
6577	026446	012737	000002	003424		MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
6578	026454	012737	000003	003430		MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
6579	026462	012737	000022	003424		MOV	#<BIT4!2>,E.B2		;EXPECTED MSG B2 & ID FOR CYL 1

K10

CZR6HEO UNIBUS RKG DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 127
T30 TEST RECALIBRATE CMD & READ HEADERS

SEQ 0127

```

6580
6581 026470 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
6582 026474 000003 .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
6583 026476 104133 ERROR 133 ;MSG A0 ERROR AFTER SEEK CMD
6584 026500 104134 ERROR 134 ;MSG B0 ERROR
6585 026502 104135 ERROR 135 ;MSG A1 ERROR
6586 026504 104136 ERROR 136 ;MSG B1 ERROR
6587 026506 023727 001364 000001 CMP CYLADD,#1
6588 026514 001401 BEQ 9$
6589 026516 104043 ERROR 43 ;CYL ADDR IN RKMR3 NOT=RKDC
6590 026520 9$:
6591 026520 005037 001176 CLR $ESCAPE
6592 026524 005737 001410 TST LPFLG
6593 026530 001402 BEQ 70$
6594 026532 000177 152352 JMP $SLPERR ;SW 9 WAS SET.
6595 026536 000177 152344 JMP $SLPADR ;SW 14 OR 8 WAS SET
6596 026542 10$:
6597 026542 005237 001410 INC LPFLG
6598 026546 032777 001000 152364 BIT #SW9,$SWR ;LOOP ON ERROR?
6599 026554 001265 BNE 8$ ;YES, RECONDITION DRIVE
6600 026556 000137 025664 JMP 2$ ;RETURN TO MAINLINE
6601 026562 12$:
6602 026562 005237 001410 INC LPFLG
6603 026566 032777 001000 152344 BIT #SW9,$SWR ;LOOP ON ERROR?
6604 026574 001255 BNE 8$ ;YES, RECONDITION DRIVE
6605 026576 000137 026162 JMP 4$ ;RETURN TO MAINLINE
6606 *****
6607 *TEST 31 SINGLE INCREMENT SEEKS TO LAST CYL
6608 *
6609 * THIS TEST DOES SINGLE INCREMENT SEEKS OUT TO THE LAST CYL
6610 * WITHOUT ANY WRITING OR READING SO AS NOT TO INADVERTENTLY
6611 * DESTROY DATA.
6612 *
6613 *****
6614 ST31: SCOPE
6615 026602 000004 MOV #1,$TIMES ;DO 1 ITERATION
6616 026604 012737 000001 001174 MOV #STACK,SP ;RESTORE STK PTR
6617 026612 012706 001100
6618 026616 004737 045424 JSR PC,SUBCLR
6619 026622 104024 ERROR 24 ;CERR AFTER SCLR
6620 026624 005037 001350 CLR FRCYL ;FROM CYL
6621 026630 012737 000001 001352 MOV #1,TOCYL ;TO CYL
6622 026636 012737 000001 001360 MOV #1,CALDIF ;CALCULATED DIFF.
6623
6624 026644 1$:
6625 026644 104415 SCOP1
6626 026646 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
6627
6628 026652 004737 045424 JSR PC,SUBCLR
6629 026656 104024 ERROR 24 ;CERR AFTER SCLR
6630
6631 026660 012737 027442 001176 MOV #10,$ESCAPE
6632 026666 013765 001352 000020 MOV TOCYL,RKDC(R5) ;CYL TO SEEK TO
6633
6634 026674 012737 000017 003322 MOV #SEEK,HCS1
6635 026702 004737 043434 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY

```

```

6636 026706 104131          ERROR 131          ;NO RDY AFTER SEEK CMD
6637 026710 012737 030140 003412  MOV   #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
6638 026716 005037 003414          CLR   E.B0
6639 026722 012737 003720 003416  MOV   #<D.FWD!D.SP0K!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
6640 026730 012737 000001 003420  MOV   #1,E.B1
6641
6642 026736 004737 044252          JSR   PC_CHKMSG          ;CHECK MSGS A0 B0 A1 B1
6643 026742 000003          .WORD T.A2!T.B2!0          ;& MSGS SPECIFIED HERE
6644 026744 104203          ERROR 203          ;MSG A0 ERROR DURING SEEK CMD
6645 026746 104204          ERROR 204          ;MSG B0 ERROR
6646 026750 104205          ERROR 205          ;MSG A1 ERROR
6647 026752 104206          ERROR 206          ;MSG B1 ERROR
6648
6649 026754 023727 001362 000001  CMP   CYLDIF,#1
6650 026762 001401          BEQ   25
6651 026764 104212          ERROR 212          ;CYL DIFF INCORRECT DURING SEEK
6652
6653 026766 012737 027462 001176 25:  MOV   #125,$ESCAPE
6654 026774 013737 001422 003360  MOV   T2500,TEMP1          ;SETUP TIMEOUT
6655
6656 027002 004737 044140          JSR   PC_FATT2          ;FIND ATTN
6657 027006 104132          ERROR 132          ;NO ATTN AFTER SEEK CMD
6658 027010 032737 100000 003322  BIT   #CERR,HCS1
6659 027016 001401          BEQ   64$
6660 027020 104210          ERROR 210          ;CERR AFTER SEEK CMD
6661
6662
6663 027022 012737 050340 003412  MOV   #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
6664 027030 005037 003414          CLR   E.B0          ;EXPECTED MSG B0
6665 027034 012737 001720 003416  MOV   #<D.SP0K!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1          ;EXPECTED A1
6666 027042 012737 000001 003420  MOV   #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
6667 027050 005037 003422          CLR   E.A2          ;EXPECTED MSG A2
6668 027054 012737 000002 003424  MOV   #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
6669 027062 012737 000003 003430  MOV   #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
6670
6671 027070 004737 044252          JSR   PC_CHKMSG          ;CHECK MSGS A0 B0 A1 B1
6672 027074 000003          .WORD T.A2!T.B2!0          ;& MSGS SPECIFIED HERE
6673 027076 104133          ERROR 133          ;MSG A0 ERROR AFTER SEEK CMD
6674 027100 104134          ERROR 134          ;MSG B0 ERROR
6675 027102 104135          ERROR 135          ;MSG A1 ERROR
6676 027104 104136          ERROR 136          ;MSG B1 ERROR
6677 027106 005737 001362  TST   CYLDIF
6678 027112 001401          BEQ   65$
6679 027114 104137          ERROR 137          ;CYL DIFF NOT CLEARED AFTER SEEK CMD
6680
6681 027116          65$:
6682
6683 027116 012765 100000 000000  MOV   #CLR,RKCS1(R5)
6684 027124 013765 001222 000010  MOV   $UNIT,RKCS2(R5) ;DRIVE#
6685 027132 012737 000005 003322  MOV   #CLEAR,HCS1
6686 027140 004737 043434          JSR   PC_DOCMD          ;DO DRIVE CLEAR CMD & GET CONTR RDY
6687 027144 104151          ERROR 151          ;NO RDY AFTER DRIVE CLEAR CMD
6688 027146 004737 044012  JSR   PC_TSTATN          ;TEST FOR ATTN
6689 027152 000401          BR    66$
6690 027154 104154          ERROR 154          ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6691 027156          66$:

```


N10

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 130
T31 SINGLE INCREMENT SEEKS TO LAST CYL

SEQ 0130

```

6748 027374 023737 001352 000000      CMP      TOCYL,0 ;LAST CYL DONE?
6749 027402 001403                BEQ      68$      ;BR IF YES
6750 027404 005337 001352                DEC      TOCYL    ;ELSE DO ANOTHER
6751 027410 000742                BR       67$
6752
6753 027412 004737 045424      68$:    JSR      PC,SUBCLR
6754 027416 104024                ERROR   24      ;CERR AFTER SCLR
6755
6756 027420 005037 001176      CLR      $ESCAPE
6757 027424 005737 001410      TST     LPFLG
6758 027430 001402                BEQ     70$
6759 027432 000177 151452      JMP     $SLPERR ;SW 9 WAS SET.
6760 027436 000177 151444      JMP     $SLPADR ;SW 14 OR 8 WAS SET
6761
6762
6763
6764 027442                10$:
6765 027442 005237 001410      INC     LPFLG
6766 027446 032777 001000 151464      BIT     #SW9,$SWR ;LOOP ON ERROR?
6767 027454 001315                BNE     6$      ;YES, RECONDITION DRIVE
6768 027456 000137 026766      JMP     2$      ;RETURN TO MAINLINE
6769
6770 027462                12$:
6771 027462 005237 001410      INC     LPFLG
6772 027466 032777 001000 151444      BIT     #SW9,$SWR ;LOOP ON ERROR?
6773 027474 001305                BNE     6$      ;YES, RECONDITION DRIVE
6774 027476 000137 027302      JMP     4$      ;RETURN TO MAINLINE
6775

```

*TEST 32 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 FOR BAD
* 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 MSG 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.

* THIS IS THE FIRST TEST WHERE THE READ DATA CMD IS PERFORMED

*ST32:

```

6798 027502 000004                SCOPE
6799 027504 012737 000001 001174      MOV     #1,$TIMES ;:DO 1 ITERATION
6800 027512 012706 001100                MOV     #STACK,SP ;RESTORE STK PTR
6801
6802 027516 004737 045424      JSR     PC,SUBCLR
6803 027522 104024                ERROR   24      ;CERR AFTER SCLR

```

6804	027524	005037	003362		CLR	TEMP2	;SECTOR CTR
6805	027530	005037	003364		CLR	TEMP3	;0=22 SECTOR HARDWARE DETECTED TABLE
6806							;1=22 SECTOR SOFTWARE DETECTED TABLE
6807							;2=DONE
6808	027534	012737	002304	003366	MOV	#BSE22H,TEMP4	;STORE 22 SECTOR HARDWARE BSE INFO
6809	027542	013765	003366	000004	MOV	TEMP4,RKBA(R5)	
6810	027550	012737	001000	003370	MOV	#1000,TEMP5	;TRACK 2, SECTOR 0
6811	027556	013765	003370	000006	MOV	TEMP5,RKDA(R5)	
6812							
6813	027564	013765	012054	000020	1\$: MOV	LC,RKDC(R5)	;LAST CYL
6814	027572	012765	177400	000002	MOV	#-256,RKWC(R5)	;LOAD WORD CT
6815	027600	012737	000021	003322	MOV	#RDATA,HCS1	
6816	027606	004737	043472		JSR	PC,DATCMD	;DO COMMAND
6817	027612	104226			ERROR	226	;NO RDY AFTER READ DATA CMD
6818	027614	004737	045074		JSR	PC,GSTAT	;GET FRESH STATUS
6819	027620	032737	100000	003322	BIT	#CERR,HCS1	
6820	027626	001470			BEQ	B\$	
6821	027630	104227			ERROR	227	;CERR AFTER READ DATA CMD
6822							
6823	027632	012737	010340	003412	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
6824	027640	005037	003414		CLR	E.B0	;EXPECTED MSG B0
6825	027644	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
6826	027652	012737	000001	003420	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
6827	027660	005037	003422		CLR	E.A2	;EXPECTED MSG A2
6828	027664	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
6829	027672	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
6830							
6831	027700	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
6832	027704	000000			.WORD	0!0!0	; & MSGS SPECIFIED HERE
6833	027706	104051			ERROR	51	;MSG A0 ERROR AFTER READ DATA CMD
6834	027710	104052			ERROR	52	;MSG B0 ERROR
6835	027712	104112			ERROR	112	;MSG A1 ERROR
6836	027714	104113			ERROR	113	;MSG B1 ERROR
6837							
6838	027716	004737	045424		JSR	PC,SUBCLR	
6839	027722	104024			ERROR	24	;CERR AFTER SUBCLR
6840							
6841	027724	005237	003362		INC	TEMP2	
6842	027730	023727	003362	000005	CMP	TEMP2,#5	;READ ALL 5 SECTORS?
6843	027736	001007			BNE	5\$	
6844	027740	005737	003364		TST	TEMP3	
6845	027744	001002			BNE	2\$	
6846	027746	104233			ERROR	233	;CANT READ SECTORS 0,2,4,6,8
6847	027750	000414			BR	3\$	
6848	027752	104230			ERROR	230	;CANT READ SECTORS 10,12,14,16,18,20
6849	027754	000412			BR	3\$	
6850							
6851	027756	013765	003366	000004	5\$: MOV	TEMP4,RKBA(R5)	;RESTORE TABLE ADDR
6852	027764	062737	000002	003370	ADD	#2,TEMP5	;SETUP TO READ 2 SECTORS FROM LAST
6853	027772	013765	003370	000006	MOV	TEMP5,RKDA(R5)	
6854	030000	000671			BR	1\$	
6855							
6856	030002	005237	001456		3\$: INC	BSEERR	;SET BSE FLAG
6857	030006	000454			BR	TST33	;GO TO NEXT TEST
6858							
6859	030010	005737	002312		8\$: TST	BSE22H+6	;TEST CARTRIDGE TYPE

```

6860 030014 001404          BEQ      9$          ;BRANCH IF DATA CARTRIDGE
6861 030016 104235          ERROR   235         ;ALIGNMENT CARTRIDGE USED
6862 030020 005237 001456    INC     BSERR       ;SET BSE ERROR FLAG
6863 030024 000426          BR      10$
6864
6865 030026 005237 003364    9$:      INC     TEMP3
6866 030032 023727 003364 000001    CMP     TEMP3,#1
6867 030040 001020          BNE     10$
6868 030042 005037 003362    CLR     TEMP2
6869 030046 012737 054352 003366    MOV     #BSE22S,TEMP4 ;STORE 22 SECTOR SOFTWARE BSE ADDR
6870 030054 013765 003366 000004    MOV     TEMP4,BKBA(R5)
6871 030062 012737 001012 003370    MOV     #1012,TEMP5 ;TRACK 2, SECTOR 12
6872 030070 013765 003370 000006    MOV     TEMPS,RKDA(R5)
6873 030076 000137 027564          JMP     1$          ;REPEAT
6874
6875 030102 005737 001216    10$:    TST     $PASS
6876 030106 001014          BNE     TST33      ;:GO TO NEXT TST IF NOT 1'ST PASS
6877 030110 104401 056206          TYPE   MSG17       ;:CART SERIAL #
6878 030114 012746 002304          MOV     #BSE22H, -(SP)
6879 030120 004737 053556          JSR     PC,$DB20   ;:CONVERT DBL BINARY WORD TO OCTAL
6880 030124 004737 054126          JSR     PC,$SUPRS ;:TYPE SERIAL #
6881 030130 104401 001205          TYPE   $SCLF
6882 030134 104401 001205          TYPE   $SCLF
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896 030140 000004          TST33: SCOPE
6897 030142 012737 000001 001174    MOV     #1,$TIMES ;:DO 1 ITERATION
6898 030150 012706 001100          MOV     #STACK,SP ;:RESTORE STK PTR
6899
6900 030154 004737 045424          JSR     PC,SUBCLR  ;:SUBSYS CLEAR & GET STATUS
6901 030160 104024          ERROR   24         ;:CERR AFTER SCLR
6902
6903 030162 005037 001410          CLR     LPFLG
6904 030166 005237 001462          INC     BYPCERR    ;:BYPASS CHECKING FOR ANY CERR IN GSTAT1
6905 030172 005237 003304          INC     UNLD       ;:USED FOR VALID HALT
6906
6907 030176 012765 000020 000026    MOV     #PAT,RKMR1(R5) ;:PARITY & WORD 0
6908 030204 013765 012052 000020    MOV     LCM1,RKDC(R5)
6909 030212 012737 000017 003322    MOV     #SEEK,HCS1
6910 030220 004737 043434          JSR     PC,DOCMD
6911 030224 104122          ERROR   122        ;:DO SEEK CMD & GET CONTR READY
6912 030226 004737 044012          JSR     PC,TSTATN ;:NO RDY FROM SEEK WITH BAD PARITY
6913 030232 104125          ERROR   125        ;:TEST FOR ATTN
6914 030234 012737 050340 003412    MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.AO ;:NO ATTN FROM SEEK WITH BAD PARITY
6915 030242 012737 001200 003414    MOV     #<D.FLT!D.PAR>,E.BO ;:EXPECTED AD

```

```

*****
*TEST 33          DETECT INNER LIMIT
*
* THIS TEST VERIFIES THAT THE LAST CYL IN THE ABOVE
* TEST WAS 632 (1456) BY DETECTING INNER LIMIT AS THE ADJACENT CYL.
* IF THIS TEST FAILS, IT INDICATES THAT HEADS WERE NOT ON THE LAST CYL
* & THAT BSE INFO IS NOT VALID. THE FORMAT PACK TEST
* & ALL READ-WRITE TESTS ARE BYPASSED
* TO AVOID DESTROYING BSE INFO OR AN ALIGNMENT CARTRIDGE
* SINCE THERE IS A SEEKING OR LIMIT DETECTION PROBLEM.
*
*****

```

```

6896 030140 000004          TST33: SCOPE
6897 030142 012737 000001 001174    MOV     #1,$TIMES ;:DO 1 ITERATION
6898 030150 012706 001100          MOV     #STACK,SP ;:RESTORE STK PTR
6899
6900 030154 004737 045424          JSR     PC,SUBCLR  ;:SUBSYS CLEAR & GET STATUS
6901 030160 104024          ERROR   24         ;:CERR AFTER SCLR
6902
6903 030162 005037 001410          CLR     LPFLG
6904 030166 005237 001462          INC     BYPCERR    ;:BYPASS CHECKING FOR ANY CERR IN GSTAT1
6905 030172 005237 003304          INC     UNLD       ;:USED FOR VALID HALT
6906
6907 030176 012765 000020 000026    MOV     #PAT,RKMR1(R5) ;:PARITY & WORD 0
6908 030204 013765 012052 000020    MOV     LCM1,RKDC(R5)
6909 030212 012737 000017 003322    MOV     #SEEK,HCS1
6910 030220 004737 043434          JSR     PC,DOCMD
6911 030224 104122          ERROR   122        ;:DO SEEK CMD & GET CONTR READY
6912 030226 004737 044012          JSR     PC,TSTATN ;:NO RDY FROM SEEK WITH BAD PARITY
6913 030232 104125          ERROR   125        ;:TEST FOR ATTN
6914 030234 012737 050340 003412    MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.AO ;:NO ATTN FROM SEEK WITH BAD PARITY
6915 030242 012737 001200 003414    MOV     #<D.FLT!D.PAR>,E.BO ;:EXPECTED AD

```

```

6916 030250 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
6917 030256 012737 000001 003420 MOV #1,E.B1
6918
6919 030264 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
6920 030270 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
6921 030272 104110 ERROR 110 ;MSG A0 ERROR AFTER SEEK WITH BAD PARITY
6922 030274 104111 ERROR 111 ;MSG B0 ERROR
6923 030276 104146 ERROR 146 ;MSG A1 ERROR
6924 030300 104147 ERROR 147 ;MSG B1 ERROR
6925
6926 030302 012765 100000 000000 MOV #CCLR,RKCS1(R5)
6927 030310 013765 001222 000010 MOV #UNIT,RKCS2(R5) ;DRIVE#
6928 030316 012737 000005 003322 MOV #CLEAR,HCS1
6929 030324 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
6930 030330 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
6931 030332 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
6932 030336 000401 BR 65$
6933 030340 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6934 030342
6935
6936 030342 012737 010340 003412 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
6937 030350 005037 003414 CLR E.B0 ;EXPECTED MSG B0
6938 030354 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
6939 030362 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
6940 030370 005037 003422 CLR E.A2 ;EXPECTED MSG A2
6941 030374 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
6942 030402 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
6943
6944 030410 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
6945 030414 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
6946 030416 104273 ERROR 273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
6947 030420 104265 ERROR 265 ;MSG B0 ERROR
6948 030422 104274 ERROR 274 ;MSG A1 ERROR
6949 030424 104266 ERROR 266 ;MSG B1 ERROR
6950
6951
6952 030426 013765 012054 000020 MOV LC,RKDC(R5)
6953 030434 012737 000017 003322 MOV #SEEK,HCS1
6954 030442 004737 043434 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
6955 030446 104131 ERROR 131 ;NO RDY AFTER SEEK CMD
6956 030450 012765 100000 000000 MOV #CCLR,RKCS1(R5)
6957 030456 004737 045074 JSR PC,GSTAT
6958 030462 004737 046260 JSR PC,FLIM ;FIND LIMIT DETECT
6959 030466 104160 ERROR 160 ;LIMIT DETECT NOT FOUND BEFORE TIMEOUT
6960
6961 030470 032737 040000 003350 BIT #D.UNLD,HMR2
6962 030476 001003 BNE 1$
6963 030500 104305 ERROR 305 ;DRIVE NOT UNLOADING AFTER LIMIT DETECT
6964 030502 000137 031342 JMP 30$ ;BYPASS REST OF TEST
6965
6966 030506 012737 031260 001176 1$: MOV #20$,SESCAPE ;MUST ESCAPE TO CYCLE UP DRIVE & TEST SWR
6967 030514 012737 070140 003412 MOV #<D.OSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
6968 030522 012737 002200 003414 MOV #<D.SKI!D.FLT>,E.B0
6969 030530 012737 045720 003416 MOV #<D.UNLD!D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
6970 030536 012737 030001 003420 MOV #<D.LIMD!D.NMOV!1>,E.B1
6971 ; CHECK 161,162,163,164,<AFTER INNER LIMIT DETECT> 0,0,0

```

6972	030544	004737	044252		JSR	PC,CHKMSG	;CHECK MESSAGE AO,BO,A1,B1
6973	030550	000000			.WORD	0!0!0	
6974	030552	104161			ERROR	161	;MAY BE AO ERROR
6975	030554	104162			ERROR	162	;MAY BE BO ERROR
6976	030556	104163			ERROR	163	;MAY BE B1 ERROR
6977	030560	000240			NOP		;NEED FOR THE CALLIN SEQ
6978	030562	032737	020000	003400	BIT	#D.LIMD,H.B1	;SEE IF LIMIT DETECT BIT SET IN B1
6979	030570	001003			BNE	3\$;BRANCH IF SO
6980	030572	104401	061235		TYPE	EM42	;OTHERWISE REPORT ERROR
6981	030576	104164			ERROR	164	
6982							
6983	030600	004737	044012		3\$: JSR	PC,TSTATN	
6984	030604	104165			ERROR	165	;NO ATTN AFTER INNER LIMIT DETECT
6985	030606	005037	001462		CLR	BYPCERR	;ALLOW CHECKING CERR IN GSTAT1
6986							
6987	030612	004737	045424		JSR	PC,SUBCLR	;SUBSYS CLR
6988	030616	104024			ERROR	24	;CERR AFTER SCLR
6989	030620	013737	001414	003362	MOV	T10,TEMP2	;SET UP TIMEOUT
6990	030626	004737	046336		JSR	PC,FHDHM	;FIND HEAD HOME
6991	030632	104166			ERROR	166	;HEAD HOME NOT FOUND BEFORE TIMEOUT
6992	030634	004737	046412		JSR	PC,FLOAD	;FIND LOAD HEADS
6993	030640	104167			ERROR	167	;LOAD HEADS NOT FOUND BEFORE TIMEOUT
6994	030642	013737	001416	003362	MOV	T100,TEMP2	;SETUP TIMEOUT
6995	030650	004737	044044		JSR	PC,FATT1	;FIND ATTN
6996	030654	104067			ERROR	67	;ATTN NOT FOUND BEFORE TIMEOUT
6997	030656	005037	001176		2\$: CLR	SESCAPE	
6998	030662	005037	003304		CLR	UNLD	;CLEAR FLAG
6999							
7000	030666	012737	050340	003412	MOV	#<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
7001	030674	005037	003414		CLR	E.B0	;EXPECTED MSG B0
7002	030700	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
7003	030706	012737	000001	003420	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
7004	030714	005037	003422		CLR	E.A2	;EXPECTED MSG A2
7005	030720	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
7006	030726	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
7007							
7008	030734	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS AO,BO,A1,B1
7009	030740	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
7010	030742	104063			ERROR	63	;MSG AO ERROR AT END OF HEAD LOADING
7011	030744	104064			ERROR	64	;MSG BO ERROR
7012	030746	104065			ERROR	65	;MSG A1 ERROR
7013	030750	104066			ERROR	66	;MSG B1 ERROR
7014					CWD2	175,176 <AT END OF HEAD LOADING>	
7015	030752	012765	000002	000026	MOV	#2,RKMR1(R5)	;SELECT MESSAGE
7016	030760	004737	045074		JSR	PC,GSTAT	;GET STATUS AND MR2,MR3
7017	030764	005737	001364		TST	CYLADD	;RECAL SUCCESSFUL ?
7018	030770	001401			BEG	64\$;BRANCH IF SO
7019	030772	104050			ERROR	50	;REPORT ERROR
7020	030774				64\$:		
7021							
7022	030774	012765	100000	000000	MOV	#CLR,RKCS1(R5)	
7023	031002	013765	001222	000010	MOV	SUNIT,RKCS2(R5)	;DRIVE#
7024	031010	012737	000005	003322	MOV	#CLEAR,HCS1	
7025	031016	004737	043434		JSR	PC,DOCMD	;DO DRIVE CLEAR CMD & GET CONTR RDY
7026	031022	104151			ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD
7027	031024	004737	044012		JSR	PC,TSTATN	;TEST FOR ATTN


```

7028 031030 000401 BR 66$
7029 031032 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7030 031034 66$:
7031
7032 031034 012737 010340 003412 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
7033 031042 005037 003414 CLR E.B0 ;EXPECTED MSG B0
7034 031046 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7035 031054 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7036 031062 005037 003422 CLR E.A2 ;EXPECTED MSG A2
7037 031066 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7038 031074 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7039
7040 031102 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
7041 031106 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7042 031110 104273 ERROR 273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7043 031112 104265 ERROR 265 ;MSG B0 ERROR
7044 031114 104274 ERROR 274 ;MSG A1 ERROR
7045 031116 104266 ERROR 266 ;MSG B1 ERROR
7046
7047 031120 004737 047430 JSR PC,SWTST ;SEE IF SW 14 OR 8 IS SET
7048 031124 000506 BR TST34 ;GO TO NEXT TEST
7049 ;RETURN HERE IF SW 14 IS SET OR
7050 ;SW 8 WITH SWR <7:0> APPLY
7051
7052
7053 031126 10$:
7054
7055 031126 004737 045424 JSR PC,SUBCLR
7056 031132 104024 ERROR 24 ;CERR AFTER SCRL
7057
7058 031134 013765 001352 000020 67$: MOV TOCYL,RKDC(R5) ;CYL#
7059
7060 031142 012737 000017 003322 MOV #SEEK,HCS1
7061 031150 004737 043434 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
7062 031154 104131 ERROR 131 ;NO RDY AFTER SEEK CMD.
7063
7064 031156 013737 001426 003360 MOV T5000,TEMP1
7065 031164 004737 044140 JSR PC,FATT2 ;FIND ATTN
7066 031170 104132 ERROR 132 ;NO ATTN AFTER SEEK CMD
7067 031172 032737 100000 003322 BIT #CERR,HCS1
7068 031200 001401 BEQ 69$
7069 031202 104210 ERROR 210 ;CERR AFTER SEEK CMD.
7070
7071 031204 004737 045424 69$: JSR PC,SUBCLR
7072 031210 104024 ERROR 24 ;CERR AFTER SCLR
7073
7074 031212 023737 001352 012054 CMP TOCYL,LC ;LAST CYL DONE?
7075 031220 001403 BEQ 68$ ;BR IF YES
7076 031222 005237 001352 INC TOCYL ;ELSE DO ANOTHER
7077 031226 000742 BR 67$
7078
7079 031230 004737 045424 68$: JSR PC,SUBCLR
7080 031234 104024 ERROR 24 ;CERR AFTER SCLR
7081
7082 031236 005037 001176 CLR $ESCAPE
7083 031242 005737 001410 TST LPFLG

```

```

7084 031246 001402      BEQ      70$
7085 031250 000177 147634    JMP      @SLPERR      ;SW 9 WAS SET.
7086 031254 000177 147626    70$:    JMP      @SLPADR      ;SW 14 OR 8 WAS SET
7087
7088 031260      20$:
7089
7090 031260 004737 045424    JSR      PC,SUBCLR
7091 031264 104024      ERROR    24      ;CERR AFTER SCLR
7092
7093 031266 012737 000011 003322    MOV      #SRTSPL,HCS1
7094 031274 004737 043434    JSR      PC,DOCMD      ;DO START SPINDLE CMD & GET CONTR RDY
7095 031300 104121      ERROR    121      ;RDY NOT FOUND AFTER ST SPIN CMD.
7096
7097 031302 013737 001420 003362    MOV      T500,TEMP2    ;SETUP TIMEOUT
7098 031310 004737 044044    JSR      PC,FATT1      ;FIND ATTN
7099 031314 104067      ERROR    67      ;NO ATTN AFTER ST SPIN CMD.
7100
7101 031316 005037 003304    CLR      UNLD
7102 031322 005237 001410    INC      LPFLG
7103 031326 032777 001000 147604    BIT      #SW9,@SWR      ;LOOP ON ERROR?
7104 031334 001274      BNE     10$      ;YES, RECONDITION DRIVE
7105 031336 000137 030656    JMP      2$      ;RETURN TO MAINLINE
7106 031342      30$:
7107
7108 031342      FORM:
7109      ;*****
7110      ;*TEST 34      FORMAT PACK
7111      ;*
7112      ;*      THIS TEST FORMATS THE ENTIRE PACK IN 22 SECTOR FORMAT BY
7113      ;*      DOING 1 CYL INCREMENTAL SEEKS
7114      ;*      FROM 0 TO 632 (1456 FOR RK07) WITH WRITE HEADER CMDS (ALL TRACKS).
7115      ;*      HEADERS WILL BE READ IN THE NEXT TEST
7116      ;*
7117      ;*****
7118 031342 000004      ST34:    SCOPE
7119 031344 012737 000001 001174    MOV      #1,$TIMES      ;;DO 1 ITERATION
7120
7121
7122 031352 012706 001100    MOV      #STACK,SP      ;RESTORE STK PTR
7123 031356 005737 001342    TST     MODTST          ;SEE IF MODULE TESTING
7124 031362 001402      BEQ     22$          ;BR IF NO
7125 031364 104401 056321    TYPE    ,MSG20          ;RUNNING MODIFIED VERSION OF TEST
7126
7127 031370 005737 001460    22$:    TST     LIMERR          ;CHECK IF FOUND LIMIT DETECT ERROR
7128 031374 001403      BEQ     1$          ;FATAL ERROR
7129 031376 104170      ERROR   170          ;ABORT BAL OF TESTS
7130 031400 000137 042662    JMP     $EOP          ;CHECK IF FOUND BSE INFO OK
7131 031404 005737 001456    1$:    TST     BSERR          ;FORMAT TEST BYPASSED-BSE ERROR
7132 031410 001403      BEQ     2$          ;
7133 031412 104177      ERROR   177          ;
7134 031414 000137 033016    JMP     7$          ;
7135
7136 031420 004737 045424    2$:    JSR     PC,SUBCLR      ;CERR AFTER SCLR
7137 031424 104024      ERROR    24          ;
7138
7139 031426 104401 056021    TYPE    ,MSG12          ;FORMATTING PACK, PLEASE WAIT

```

```

7140
7141 031432 005037 001352          CLR      TOCYL
7142
7143 031436 013737 001352 001366    MOV      TOCYL,CALADD      ;SETUP
7144 031444 012737 000000 001430    MOV      #0,HEAD          ;TO FILL
7145 031452 012737 000000 001436    MOV      #0,FORMAT        ;HEADER
7146 031460 004737 046534          JSR      PC,FHDTAB        ;TABLE
7147
7148
7149 031464 012765 001470 000004 9$:    MOV      #HDTAB,RKBA(R5) ;THIS SECTION
7150 031472 012765 177676 000002    MOV      #-66.,RKWC(R5) ;OF CODE
7151 031500 000337 001430          SWAB     HEAD             ;IS TO RESTORE STANDARD FORMAT
7152 031504 013765 001430 000006    MOV      HEAD,RKDA(R5)   ;TO CYL 0
7153 031512 000337 001430          SWAB     HEAD             ;HEAD 0,1 & 2
7154
7155 031516 012737 000027 003322    MOV      #<WRHEAD>,HCS1
7156 031524 004737 043472          JSR      PC,DATCMD        ;DO DATA XFER CMD & GET CONTR RDY
7157 031530 104200          ERROR    200             ;NO RDY AFTER WRITE HEADER CMD
7158 031532 004737 045074          JSR      PC,GSTAT        ;GET FRESH STATUS
7159 031536 032737 100000 003322    BIT      #CERR,HCS1
7160 031544 001405          BEQ     64$
7161 031546 104201          ERROR    201             ;CERR AFTER WRITE HEADER CMD
7162 031550 104401 056235          TYPE    MSG18            ;ABORTING BALANCE OF TESTS
7163 031554 000137 042662          JMP     $EOP             ;ABORT DRIVE
7164 031560
7165
7166 031560 005237 001430          INC     HEAD
7167 031564 023727 001430 000003    CMP     HEAD,#3
7168 031572 001403          BEQ     11$             ;BR IF ALL HEADS DONE
7169
7170 031574 004737 046534          JSR      PC,FHDTAB
7171 031600 000731          BR
7172
7173 031602 012737 000001 001366 11$:    MOV      #1,CALADD        ;SETUP
7174 031610 005037 001430          CLR     HEAD            ;FOR
7175 031614 005037 001436          CLR     FORMAT          ;FHDTAB ROUTINE
7176
7177 031620 012737 000001 001360    MOV      #1,CALDIF        ;SETUP
7178 031626 005037 001350          CLR     FRCYL           ;FOR
7179 031632 012737 000001 001352    MOV      #1,TOCYL        ;ERROR REPORT
7180
7181
7182
7183 031640
7184 031640 104415          3$:
7185 031642 012706 001100          SCOP1   MOV      #STACK,SP ;RESTORE STK PTR
7186
7187 031646 004737 045424          JSR      PC,SUBCLR
7188 031652 104024          ERROR    24             ;CERR AFTER SCLR
7189
7190 031654 005737 001342          TST     MODTST          ;SEE IF MODULE TESTING
7191 031660 001404          BEQ     18$             ;BR IF NO
7192 031662 012737 033352 001176    MOV      #16$,SESCAPE
7193 031670 000403          BR      19$
7194 031672 012737 033056 001176 18$:    MOV      #10$,SESCAPE
7195 031700 013765 001366 000020 19$:    MOV      CALADD,RKDC(R5) ;CYL #

```



```

7252 032164 66$:
7253
7254 032164 012765 100000 000000 MOV #CCLR,RKCS1(R5)
7255 032172 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
7256 032200 012737 000005 003322 MOV #CLEAR,HCS1
7257 032206 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
7258 032212 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
7259 032214 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
7260 032220 000401 BR 67$
7261 032222 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7262 032224 67$:
7263
7264 032224 012737 010340 003412 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
7265 032232 005037 003414 CLR E.B0 ;EXPECTED MSG B0
7266 032236 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7267 032244 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7268 032252 005037 003422 CLR E.A2 ;EXPECTED MSG A2
7269 032256 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7270 032264 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7271
7272 032272 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
7273 032276 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7274 032300 104273 ERROR 273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7275 032302 104265 ERROR 265 ;MSG B0 ERROR
7276 032304 104274 ERROR 274 ;MSG A1 ERROR
7277 032306 104266 ERROR 266 ;MSG B1 ERROR
7278
7279 032310 023737 001364 001366 CMP CYLADD,CALADD
7280 032316 001401 BEQ 5$
7281 032320 104232 ERROR 232 ;CYL ADDR IN RKMR2 NOT=RKDC
7282
7283 032322 5$:
7284 032322 104415 SCOP1
7285 032324 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
7286
7287 032330 004737 045424 JSR PC,SUBCLR
7288 032334 104024 ERROR 24 ;CERR AFTER SCLR
7289
7290 032336 005037 001176 CLR $ESCAPE
7291 032342 004737 046534 JSR PC,FHDTAB ;FILL HEADER TABLE
7292 032346 000337 001430 SWAB HEAD
7293 032352 013765 001430 000006 MOV HEAD,RKDA(R5) ;SET TRACK #
7294 032360 000337 001430 SWAB HEAD
7295 032364 012765 001470 000004 MOV #HDTAB,RKBA(R5) ;HEADER WORD TABLE
7296 032372 012765 177676 000002 MOV #-66,RKWC(R5) ;WORD CT
7297 032400 013765 001366 000020 MOV CALADD,RKDC(R5) ;CYL #
7298
7299
7300 032406 012737 000027 003322 MOV #<WRHEAD>,HCS1
7301 032414 004737 043472 JSR PC,DATCMD ;DO DATA XFER CMD & GET CONTR RDY
7302 032420 104200 ERROR 200 ;NO RDY AFTER WRITE HEADER CMD
7303 032422 004737 045074 JSR PC,GSTAT ;GET FRESH STATUS
7304 032426 032737 100000 003322 BIT #CERR,HCS1
7305 032434 001405 BEQ 68$
7306 032436 104201 ERROR 201 ;CERR AFTER WRITE HEADER CMD
7307 032440 104401 056235 TYPE ,MSG18 ;ABORTING BALANCE OF TESTS

```

K11

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 140
T34 FORMAT PACK

SEQ 0140

```
7308 032444 000137 042662          JMP      SEOP          ;ABORT DRIVE
7309 032450          68$:
7310
7311 032450 012737 010340 003412      MOV      #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0      ;EXPECTED MSG A0
7312 032456 005037 003414          CLR      E.B0          ;EXPECTED MSG B0
7313 032462 012737 001720 003416      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1      ;EXPECTED A1
7314 032470 012737 000001 003420      MOV      #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
7315 032476 005037 003422          CLR      E.A2          ;EXPECTED MSG A2
7316 032502 012737 000002 003424      MOV      #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
7317 032510 012737 000003 003430      MOV      #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
7318
7319 032516 004737 044252          JSR      PC,CHKMSG      ;CHECK MSGS A0,B0,A1,B1
7320 032522 000003          .WORD   T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
7321 032524 104277          ERROR   277          ;MSG A0 ERROR AFTER WRITE HEADER CMD
7322 032526 104267          ERROR   267          ;MSG B0 ERROR
7323 032530 104300          ERROR   300          ;MSG A1 ERROR
7324 032532 104270          ERROR   270          ;MSG B1 ERROR
7325
7326
7327 032534 005737 001342          TST     MODTST          ;SEE IF MODULE TESTING
7328 032540 001402          BEQ     23$            ;BR IF NO
7329 032542 000137 033116          JMP     14$            ;ELSE RESTORE HEADERS ONLY
```

7330									
7331	032546	005237	001430		23\$:	INC	HEAD		
7332	032552	023727	001430	000002		CMP	HEAD,#2		
7333	032560	001006				BNE	6\$		
7334	032562	023737	001366	012054		CMP	CALADD,LC		;HEAD 2, SEE IF ON LAST CYL
7335	032570	001002				BNE	6\$;DO NOT WRITE ON LAST CYL HEAD 2
7336	032572	000137	033016			JMP	7\$		
7337									
7338	032576	023727	001430	000003	6\$:	CMP	HEAD,#3		;ALL HEADS DONE?
7339	032604	001246				BNE	5\$;BR IF NO
7340	032606	005037	001430			CLR	HEAD		;ALL HEADS ON CYL DONE
7341	032612	005237	001366			INC	CALADD		;GO TO NEXT CYL
7342	032616	005237	001350			INC	FRCYL		;FOR ERROR REPORT
7343	032622	005237	001352			INC	TOCYL		;FOR ERROR REPORT
7344	032626	005737	003310			TST	HPEND		;SEE IF HALT PENDING
7345	032632	001002				BNE	24\$;BR IF YES
7346	032634	000137	031640			JMP	3\$;ELSE KEEP FORMATTING
7347									
7348	032640	005037	003310		24\$:	CLR	HPEND		;CLEAR FOR FUTURE FORMATTING
7349	032644	005037	003306			CLR	BADHDR		;HEADERS NOW OK
7350	032650	000137	047500			JMP	STOP		;GO & HALT THE CPU
7351									
7352	032654	005037	001366		8\$:	CLR	CALADD		
7353	032660	005037	001350			CLR	FRCYL		
7354									
7355	032664	004737	045424			JSR	PC,SUBCLR		
7356	032670	104024				ERROR	24		;CERR AFTER SCRL
7357									
7358	032672	013765	001352	000020	69\$:	MOV	TOCYL,RKDC(R5)		;CYL#
7359									
7360	032700	012737	000017	003322		MOV	#SEEK,HCS1		
7361	032706	004737	043434			JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
7362	032712	104131				ERROR	131		;NO RDY AFTER SEEK CMD.
7363									
7364	032714	013737	001426	003360		MOV	T50000,TEMP1		
7365	032722	004737	044140			JSR	PC,FATT2		;FIND ATTN
7366	032726	104132				ERROR	132		;NO ATTN AFTER SEEK CMD
7367	032730	032737	100000	003322		BIT	#CERR,HCS1		
7368	032736	001401				BEQ	71\$		
7369	032740	104210				ERROR	210		;CERR AFTER SEEK CMD.
7370									
7371	032742	004737	045424		71\$:	JSR	PC,SUBCLR		
7372	032746	104024				ERROR	24		;CERR AFTER SCLR
7373									
7374	032750	023737	001352	000000		CMP	TOCYL,0 ;LAST CYL DONE?		
7375	032756	001403				BEQ	70\$;BR IF YES
7376	032760	005337	001352			DEC	10CYL		;ELSE DO ANOTHER
7377	032764	000742				BR	69\$		
7378									
7379	032766	004737	045424		70\$:	JSR	PC,SUBCLR		
7380	032772	104024				ERROR	24		;CERR AFTER SCLR
7381									
7382	032774	005037	001176			CLR	\$ESCAPE		
7383	033000	005737	001410			TST	LPFLG		
7384	033004	001402				BEQ	72\$		
7385	033006	000177	146076			JMP	\$SLPERR		;SW 9 WAS SET.

7386	033012	000177	146070		72\$:	JMP	QSLPADR		;SW 14 OR 8 WAS SET
7387									
7388									
7389	033016	004737	050006		7\$:	JSR	PC,HPEN		;SEE IF HALT PENDING
7390	033022	000137	032640			JMP	24\$;RET HERE IF YES & EXIT
7391									;ELSE RET HERE
7392	033026	004737	047430			JSR	PC,SWTST		;SEE IF SW 14 OR 8 IS SET
7393	033032	000567				BR	TST35		;GO TO NEXT TEST
7394									;RETURN HERE IF SW 14 IS SET OR
7395									;SW 8 WITH SWR <7:0> APPLY
7396	033034	005037	001176			CLR	\$ESCAPE		
7397	033040	005737	001410			TST	LPFLG		
7398	033044	001402				BEQ	73\$		
7399	033046	000177	146036			JMP	QSLPERR		;SW 9 WAS SET
7400	033052	000177	146030		73\$:	JMP	QSLPADR		;SW 14 OR 8 WAS SET
7401									
7402	033056				10\$:				
7403	033056	005237	001410			INC	LPFLG		
7404	033062	032777	001000	146050		BIT	#SW9,QSWR		;LOOP ON ERROR?
7405	033070	001271				BNE	8\$;YES, RECONDITION DRIVE
7406	033072	000137	032016			JMP	4\$;RETURN TO MAINLINE
7407									
7408	033076				12\$:				
7409	033076	005237	001410			INC	LPFLG		
7410	033102	032777	001000	146030		BIT	#SW9,QSWR		;LOOP ON ERROR?
7411	033110	001261				BNE	8\$;YES, RECONDITION DRIVE
7412	033112	000137	032322			JMP	5\$;RETURN TO MAINLINE
7413									
7414									
7415	033116	005237	001430		14\$:	INC	HEAD		
7416	033122	023727	001430	000003		CMP	HEAD,#3		;SEE IF ALL HEADS DONE
7417	033130	001402				BEQ	15\$;BR IF YES TO GO BACK TO CYL 0
7418	033132	000137	032322			JMP	5\$;ELSE REPEAT FOR NEXT HEAD
7419									
7420	033136	005065	000006		15\$:	CLR	RKDA(R5)		;SEEK TO CYL 0 & READ HEADERS
7421	033142	005037	001352			CLR	TOCYL		;TO RECONDITION DRIVE
7422									
7423	033146	012737	000017	003322		MOV	#SEEK,HCS1		
7424	033154	004737	043434			JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
7425	033160	104131				ERROR	131		;NO RDY AFTER SEEK CMD.
7426									
7427	033162	013737	001426	003360		MOV	T50000,TEMP1		
7428	033170	004737	044140			JSR	PC,FATT2		;FIND ATTN
7429	033174	104132				ERROR	132		;NO ATTN AFTER SEEK CMD
7430	033176	032737	100000	003322		BIT	#CERR,HCS1		
7431	033204	001401				BEQ	74\$		
7432	033206	104210				ERROR	210		;CERR AFTER SEEK CMD.
7433									
7434	033210	004737	045424		74\$:	JSR	PC,SUBCLR		
7435	033214	104024				ERROR	24		;CERR AFTER SCLR
7436									
7437									
7438									
7439	033216	012700	001674			MOV	#RHTAB,RO		
7440	033222	012737	000025	003322		MOV	#<RDHEAD>,HCS1		
7441	033230	004737	043472			JSR	PC,DATCMD		;DO DATA XFER CMD & GET CONTR RDY


```

7442 033234 104171          ERROR 171          ;NO RDY AFTER READ HEADER CMD
7443 033236 032737 100000 003322  BIT    #CERR,HCS1
7444 033244 001405          BEQ    76$
7445 033246 104174          ERROR 174          ;CERR AFTER READ HEADER CMD
7446 033250 104401 056235  TYPE   MSG18      ;ABORT BALANCE OF TESTS
7447 033254 000137 042662  JMP    $EOP       ;ABORT DRIVE
7448
7449 033260 016520 000024 76$:  MOV   RKDB(R5),(R0)+ ;1'ST WORD FROM SILO TO RHTAB
7450 033264 016520 000024  MOV   RKDB(R5),(R0)+ ;2'ND WORD
7451 033270 016520 000024  MOV   RKDB(R5),(R0)+ ;3'RD WORD
7452
7453
7454 033274 032765 100000 000010  BIT    #DLT,RKCS2(R5)
7455 033302 001407          BEQ    77$
7456 033304 004737 045074  JSR   PC,GSTAT
7457 033310 104173          ERROR 173          ;DLT AFTER READ HEADER CMD
7458 033312 104401 056235  TYPE   MSG18      ;ABORTING BALANCE OF TESTS
7459 033316 000137 042662  JMP    $EOP       ;ABORT DRIVE
7460 033322
7461 77$:
7462 033322 023737 001674 001352  CMP   RHTAB,TOCYL ;CHECK WORD 0 (CYL#) ONLY
7463 033330 001401          BEQ    75$
7464 033332 104310          ERROR 310          ;BR IF SAME
7465 033334
7466 75$:
7467 033334 004737 050006  JSR   PC,HPEN
7468 033340 000137 032640  JMP   24$
7469
7470 033344 004737 047430  JSR   PC,SWTST
7471 033350 000420          BR    TST35
7472
7473
7474 033352
7475 033352 005237 001410 16$:  INC   LPFLG
7476 033356 032777 001000 145554  BIT   #SW9,JSWR ;LOOP ON ERROR?
7477 033364 001264          BNE   15$
7478 033366 000137 032016  JMP   4$         ;YES, RECONDITION DRIVE
7479 033372
7480 033372 005237 001410 17$:  INC   LPFLG
7481 033376 032777 001000 145534  BIT   #SW9,JSWR ;LOOP ON ERROR?
7482 033404 001254          BNE   15$
7483 033406 000137 032322  JMP   5$         ;YES, RECONDITION DRIVE
7484
7485
7486 033412
7487 13$:
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497 033412 000004

```

;TEST 35 DECREMENT FROM LAST CYL TO 0 & READ HEADERS
; THIS TEST VERIFIES MOTION IN THE NEGATIVE DIRECTION BY
; SINGLE CYL INCREMENTAL SEEKS.

TST35: SCOPE

```

7498 033414 012737 000001 001174 MOV #1,STIMES ;DO 1 ITERATION
7499 033422 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
7501 033426 104401 056550 TYPE MSG22 ;FORMATTING FINISHED
7502 033432 005737 001342 TST MODTST ;SEE IF MODULE TESTING
7503 033436 001404 BEQ 5$ ;BR IF NO
7504 033440 104401 056470 TYPE MSG21 ;BYP TESTS 36,40,41
7505 033444 000137 034470 JMP 13$
7506 033450 013737 012054 001350 5$: MOV LC,FRCYL ;FROM CYL
7507 033456 013737 012052 001352 MOV LCM1,TOCYL ;TO CYL
7509 033464 1$: SCOP1
7510 033464 104415 MOV #STACK,SP ;RESTORE STK PTR
7511 033466 012706 001100
7512 033472 004737 045424 JSR PC,SUBCLR
7514 033476 104024 ERROR 24 ;CERR AFTER SCLR
7516 033500 012737 034430 001176 MOV #10$,SESCAPE
7517 033506 013765 001352 000020 MOV TOCYL,RKDC(R5) ;CYL #
7519 033514 012737 000017 003322 MOV #SEEK,HCS1
7520 033522 004737 043434 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
7521 033526 104131 ERROR 131 ;NO RDY AFTER SEEK CMD
7522 033530 012737 030140 003412 MOV #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7523 033536 005037 003414 CLR E.B0
7524 033542 012737 005720 003416 MOV #<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7525 033550 012737 000001 003420 MOV #1,E.B1
7527 033556 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
7528 033562 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7529 033564 104203 ERROR 203 ;MSG A0 ERROR DURING SEEK CMD
7530 033566 104204 ERROR 204 ;MSG B0 ERROR
7531 033570 104205 ERROR 205 ;MSG A1 ERROR
7532 033572 104206 ERROR 206 ;MSG B1 ERROR
7534 033574 023727 001362 000001 CMP CYLDIF,#1
7535 033602 001406 BEQ 2$
7536 033604 012765 000002 000026 MOV #2,RKMR1(R5) ;SELECT WD 2
7537 033612 004737 045074 JSR PC,GSTAT
7538 033616 104212 ERROR 212 ;CYL DIFF INCORRECT DURING SEEK
7540 033620 012737 034450 001176 2$: MOV #2$,SESCAPE
7541 033626 012737 004704 003360 MOV #2500.,TEMP1 ;SETUP TIMEOUT
7543 033634 004737 044140 JSR PC,FATT2 ;FIND ATTN
7544 033640 104132 ERROR 132 ;NO ATTN AFTER SEEK CMD
7545 033642 032737 100000 003322 BIT #CERR,HCS1
7546 033650 001401 BEQ 64$
7547 033652 104210 ERROR 210 ;CERR AFTER SEEK CMD
7548 033654 64$:
7550 033654 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
7551 033662 005037 003414 CLR E.B0 ;EXPECTED MSG B0
7552 033666 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7553 033674 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1

```

7554	033702	005037	003422		CLR	E.A2	;EXPECTED MSG A2
7555	033706	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
7556	033714	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
7557							
7558	033722	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
7559	033726	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
7560	033730	104133			ERROR	133	;MSG A0 ERROR AFTER SEEK CMD
7561	033732	104134			ERROR	134	;MSG B0 ERROR
7562	033734	104135			ERROR	135	;MSG A1 ERROR
7563	033736	104136			ERROR	136	;MSG B1 ERROR
7564	033740	005737	001362		TST	CYLDIF	
7565	033744	001401			BEG	65\$	
7566	033746	104137			ERROR	137	;CYL DIFF NOT CLEARED AFTER SEEK CMD
7567							
7568	033750						
7569							
7570	033750	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	
7571	033756	013765	001222	000010	MOV	SUNIT,RKCS2(R5)	;DRIVE#
7572	033764	012737	000005	003322	MOV	#CLEAR,HCS1	
7573	033772	004737	043434		JSR	PC,DOCMD	;DO DRIVE CLEAR CMD & GET CONTR RDY
7574	033776	104151			ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD
7575	034000	004737	044012		JSR	PC,TSTATN	;TEST FOR ATTN
7576	034004	000401			BR	66\$	
7577	034006	104154			ERROR	154	;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7578	034010						
7579							
7580	034010	012737	010340	003412	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
7581	034016	005037	003414		CLR	E.B0	;EXPECTED MSG B0
7582	034022	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
7583	034030	012737	000001	003420	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
7584	034036	005037	003422		CLR	E.A2	;EXPECTED MSG A2
7585	034042	012737	000002	003424	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
7586	034050	012737	000003	003430	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
7587							
7588	034056	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
7589	034062	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
7590	034064	104273			ERROR	273	;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7591	034066	104265			ERROR	265	;MSG B0 ERROR
7592	034070	104274			ERROR	274	;MSG A1 ERROR
7593	034072	104266			ERROR	266	;MSG B1 ERROR
7594							
7595	034074	023737	001364	001352	CMP	CYLADD,TOCYL	
7596	034102	001401			BEG	3\$	
7597	034104	104207			ERROR	207	;CYL ADDR IN RKMR3 NOT = RKDC
7598							
7599	034106						
7600	034106	104415			SCOP1		
7601	034110	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
7602							
7603	034114	004737	045424		JSR	PC,SUBCLR	
7604	034120	104024			ERROR	24	;CERR AFTER SCLR
7605							
7606	034122	005037	001176		CLR	\$ESCAPE	
7607							
7608	034126	013765	001352	000020	MOV	TOCYL,RKDC(R5)	;CYL #
7609							

7610									
7611	034134	012700	001674			MOV	#RHTAB,RO		
7612	034140	012737	000025	003322		MOV	#(RDHEAD),HCS1		
7613	034146	004737	043472			JSR	PC,DATCMD		;DO DATA XFER CMD & GET CONTR RDY
7614	034152	104171				ERROR	171		;NO RDY AFTER READ HEADER CMD
7615	034154	032737	100000	003322		BIT	#CERR,HCS1		
7616	034162	001405				BEQ	68\$		
7617	034164	104174				ERROR	174		;CERR AFTER READ HEADER CMD
7618	034166	104401	056235			TYPE	,MSG18		;ABORT BALANCE OF TESTS
7619	034172	000137	042662			JMP	\$EOP		;ABORT DRIVE
7620									
7621	034176	016520	000024		68\$:	MOV	RKDB(R5),(R0)+		;1'ST WORD FROM SILO TO RHTAB
7622	034202	016520	000024			MOV	RKDB(R5),(R0)+		;2'ND WORD
7623	034206	016520	000024			MOV	RKDB(R5),(R0)+		;3'RD WORD
7624									
7625									
7626	034212	032765	100000	000010		BIT	#DLT,RKCS2(R5)		
7627	034220	001407				BEQ	69\$		
7628	034222	004737	045074			JSR	PC,GSTAT		
7629	034226	104173				ERROR	173		;DLT AFTER READ HEADER CMD
7630	034230	104401	056235			TYPE	,MSG18		;ABORTING BALANCE OF TESTS
7631	034234	000137	042662			JMP	\$EOP		;ABORT DRIVE
7632	034240				69\$:				
7633									
7634	034240	023737	001674	001352		CMP	RHTAB,TOCYL		;CHECK WORD 0 (CYL#) ONLY
7635	034246	001401				BEQ	67\$;BR IF SAME
7636	034250	104310				ERROR	310		;READ CYL WORD HEADER ERROR
7637	034252				67\$:				
7638									
7639	034252	005337	001350			DEC	FRCYL		
7640	034256	001404				BEQ	4\$		
7641	034260	005337	001352			DEC	TOCYL		
7642	034264	000137	033464			JMP	1\$		
7643									
7644	034270				4\$:				
7645	034270	004737	047430			JSR	PC,SWTST		;SEE IF SW 14 OR 8 IS SET
7646	034274	000475				BR	TST36		;GO TO NEXT TEST
7647									;RETURN HERE IF SW 14 IS SET OR
7648									;SW 8 WITH SWR <7:0> APPLY
7649									
7650									
7651	034276				6\$:				
7652									
7653	034276	004737	045424			JSR	PC,SUBCLR		
7654	034302	104024				ERROR	24		;CERR AFTER SCRL
7655									
7656	034304	013765	001352	000020	70\$:	MOV	TOCYL,RKDC(R5)		;CYL#
7657									
7658	034312	012737	000017	003322		MOV	#SEEK,HCS1		
7659	034320	004737	043434			JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
7660	034324	104131				ERROR	131		;NO RDY AFTER SEEK CMD.
7661									
7662	034326	013737	001426	003360		MOV	T50000,TEMP1		
7663	034334	004737	044140			JSR	PC,FATT2		;FIND ATTN
7664	034340	104132				ERROR	132		;NO ATTN AFTER SEEK CMD
7665	034342	032737	100000	003322		BIT	#CERR,HCS1		

```

7666 034350 001401      BEQ      72$
7667 034352 104210      ERROR    210          ;CERR AFTER SEEK CMD.
7668
7669 034354 004737 045424 72$:  JSR      PC,SUBCLR
7670 034360 104024      ERROR    24          ;CERR AFTER SCLR
7671
7672 034362 023737 001352 012054  CMP      TOCYL,LC    ;LAST CYL DONE?
7673 034370 001403      BEQ      71$          ;BR IF YES
7674 034372 005237 001352      INC      TOCYL      ;ELSE DO ANOTHER
7675 034376 000742      BR       70$
7676
7677 034400 004737 045424 71$:  JSR      PC,SUBCLR
7678 034404 104024      ERROR    24          ;CERR AFTER SCLR
7679
7680 034406 005037 001176      CLR      $ESCAPE
7681 034412 005737 001410      TST      LPFLG
7682 034416 001402      BEQ      73$
7683 034420 000177 144464      JMP      $SLPERR    ;SW 9 WAS SET.
7684 034424 000177 144456      JMP      $SLPADR    ;SW 14 OR 8 WAS SET
7685
7686
7687
7688
7689 034430 005237 001410 10$:  INC      LPFLG
7690 034434 032777 001000 144476  BIT      #SW9,$SWR   ;LOOP ON ERROR?
7691 034442 001315      BNE     6$          ;YES, RECONDITION DRIVE
7692 034444 000137 033620      JMP      2$          ;RETURN TO MAINLINE
7693
7694
7695 034450 005237 001410 12$:  INC      LPFLG
7696 034454 032777 001000 144456  BIT      #SW9,$SWR   ;LOOP ON ERROR?
7697 034462 001305      BNE     6$          ;YES, RECONDITION DRIVE
7698 034464 000137 034106      JMP      3$          ;RETURN TO MAINLINE
7699
7700 034470 13$:

```

```

7701
7702
7703
7704
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721 034470 000004

```

```

*****
*TEST 36          SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS
*
* THIS TEST SEEKS FROM CYL 0 TO ALL THE MAJOR CYLS & READS HEADERS.
* IT THEN SEEKS CYL 0 & READS HEADERS.
*
* MAJOR CYLS ARE: 1 (DECIMAL) = 1 (OCTAL)
*
*
*           2           2
*           4           4
*           8           10
*          16           20
*          32           40
*          64           100
*         128           200
*         256           400
*         512           1000 (RK07)
*
*****
TST36: SCOPE

```

```

7722 034472 012737 000001 001174 MOV #1,STIMES ;DO 1 ITERATION
7723 034500 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
7724
7725 034504 013737 012072 001350 MOV FC,FRCYL ;SETUP FROM CYL
7726 034512 013737 012074 001352 MOV FCP1,TOCYL ;SETUP TO CYL
7727
7728 034520 1S:
7729 034520 104415 SCOP1
7730 034522 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
7731
7732 034526 004737 045424 JSR PC,SUBCLR
7733 034532 104024 ERROR 24 ;CERR AFTER SCLR
7734
7735 034534 012737 036242 001176 MOV #10$,SESCAPE
7736 034542 013737 001350 003364 MOV FRCYL,TEMP3 ;SETUP
7737 034550 013737 001352 003366 MOV TOCYL,TEMP4 ;CYL DIFF
7738 034556 163737 003364 003366 SUB TEMP3,TEMP4 ;FOR
7739 034564 013737 003366 001360 MOV TEMP4,CALDIF ;ERROR PRINTOUT
7740
7741 034572 013765 001352 000020 MOV TOCYL,RKDC(R5) ;GO TO CYL #
7742
7743 034600 012737 000017 003322 MOV #SEEK,HCS1
7744 034606 004737 043434 JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
7745 034612 104131 ERROR 131 ;NO RDY AFTER SEEK CMD
7746 034614 012737 030140 003412 MOV #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
7747 034622 005037 003414 CLR E.B0
7748 034626 012737 003720 003416 MOV #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
7749 034634 012737 000001 003420 MOV #1,E.B1
7750
7751 034642 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
7752 034646 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7753 034650 104203 ERROR 203 ;MSG A0 ERROR DURING SEEK CMD
7754 034652 104204 ERROR 204 ;MSG B0 ERROR
7755 034654 104205 ERROR 205 ;MSG A1 ERROR
7756 034656 104206 ERROR 206 ;MSG B1 ERROR
7757
7758 034660 012737 036262 001176 2S: MOV #12$,SESCAPE
7759 034666 013737 001426 003360 MOV T5000,TEMP1 ;SETUP TIMEOUT
7760
7761 034674 004737 044140 JSR PC,FATT2 ;FIND ATTN
7762 034700 104132 ERROR 132 ;NO ATTN AFTER SEEK CMD
7763 034702 032737 100000 003322 BIT #CERR,HCS1
7764 034710 001401 BEQ 64$
7765 034712 104210 ERROR 210 ;CERR AFTER SEEK CMD
7766 64$:
7767
7768 034714 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
7769 034722 005037 003414 CLR E.B0 ;EXPECTED MSG B0
7770 034726 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7771 034734 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7772 034742 005037 003422 CLR E.A2 ;EXPECTED MSG A2
7773 034746 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7774 034754 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7775
7776 034762 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
7777 034766 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE

```

7778	034770	104133			ERROR	133		;MSG A0 ERROR AFTER SEEK CMD
7779	034772	104134			ERROR	134		;MSG B0 ERROR
7780	034774	104135			ERROR	135		;MSG A1 ERROR
7781	034776	104136			ERROR	136		;MSG B1 ERROR
7782	035000	005737	001362		TST	CYLDIF		
7783	035004	001401			BEQ	65\$		
7784	035006	104137			ERROR	137		;CYL DIFF NOT CLEARED AFTER SEEK CMD
7785								
7786	035010						65\$:	
7787								
7788	035010	012765	100000	000000	MOV	#CCLR,RKCS1(R5)		
7789	035016	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	;DRIVE#	
7790	035024	012737	000005	003322	MOV	#CLEAR,HCS1		
7791	035032	004737	043434		JSR	PC,DOCMD		;DO DRIVE CLEAR CMD & GET CONTR RDY
7792	035036	104151			ERROR	151		;NO RDY AFTER DRIVE CLEAR CMD
7793	035040	004737	044012		JSR	PC,TSTATN		;TEST FOR ATTN
7794	035044	000401			BR	66\$		
7795	035046	104154			ERROR	154		;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7796	035050						66\$:	
7797								
7798	035050	012737	010340	003412	MOV	#<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
7799	035056	005037	003414		CLR	E.B0		;EXPECTED MSG B0
7800	035062	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
7801	035070	012737	000001	003420	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
7802	035076	005037	003422		CLR	E.A2		;EXPECTED MSG A2
7803	035102	012737	000002	003424	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
7804	035110	012737	000003	003430	MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
7805								
7806	035116	004737	044252		JSR	PC,CHKMSG		;CHECK MSGS A0,B0,A1,B1
7807	035122	000003			.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
7808	035124	104273			ERROR	273		;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7809	035126	104265			ERROR	265		;MSG B0 ERROR
7810	035130	104274			ERROR	274		;MSG A1 ERROR
7811	035132	104266			ERROR	266		;MSG B1 ERROR
7812								
7813	035134	023737	001364	001352	CMP	CYLADD,TOCYL		
7814	035142	001401			BEQ	3\$		
7815	035144	104207			ERROR	207		;CYL ADDR IN RKMR3 NOT=RKDC
7816								
7817	035146						3\$:	
7818	035146	104415			SCOP1			
7819	035150	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
7820								
7821	035154	004737	045424		JSR	PC,SUBCLR		
7822	035160	104024			ERROR	24		;CERR AFTER SCLR
7823								
7824	035162	005037	001176		CLR	\$ESCAPE		
7825	035166	013765	001352	000020	MOV	TOCYL,RKDC(R5)		;CYL #
7826								
7827								
7828	035174	012700	001674		MOV	#RHTAB,RO		
7829	035200	012737	000025	003322	MOV	#<RDHEAD>,HCS1		
7830	035206	004737	043472		JSR	PC,DATCMD		;DO DATA XFER CMD & GET CONTR RDY
7831	035212	104171			ERROR	171		;NO RDY AFTER READ HEADER CMD
7832	035214	032737	100000	003322	BIT	#CERR,HCS1		
7833	035222	001405			BEQ	68\$		

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00MACY11 30A(1052) 25-JAN-78 12:08 PAGE 150
T36 SEEK FROM CYL 0 TO ALL MAJOR CYLS & READ HEADERS

SEQ 0150

7834	035224	104174			ERROR	174		;CERR AFTER READ HEADER CMD
7835	035226	104401	056235		TYPE	MSG18		;ABORT BALANCE OF TESTS
7836	035232	000137	042662		JMP	\$EOP		;ABORT DRIVE
7837								
7838	035236	016520	000024	68\$:	MOV	RKDB(R5),(R0)+		;1'ST WORD FROM SILO TO RHTAB
7839	035242	016520	000024		MOV	RKDB(R5),(R0)+		;2'ND WORD
7840	035246	016520	000024		MOV	RKDB(R5),(R0)+		;3'RD WORD
7841								
7842								
7843	035252	032765	100000	000010	BIT	#DLT,RKCS2(R5)		
7844	035260	001407			BEQ	69\$		
7845	035262	004737	045074		JSR	PC,GSTAT		
7846	035266	104173			ERROR	173		;DLT AFTER READ HEADER CMD
7847	035270	104401	056235		TYPE	MSG18		;ABORTING BALANCE OF TESTS
7848	035274	000137	042662		JMP	\$EOP		;ABORT DRIVE
7849	035300			69\$:				
7850								
7851	035300	023737	001674	001352	CMP	RHTAB,TOCYL		;CHECK WORD 0 (CYL#) ONLY
7852	035306	001401			BEQ	67\$;BR IF SAME
7853	035310	104310			ERROR	310		;READ CYL WORD HEADER ERROR
7854	035312			67\$:				
7855								
7856								
7857	035312	104415			SCOP1			
7858	035314	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
7859								
7860	035320	004737	045424		JSR	PC,SUBCLR		
7861	035324	104024			ERROR	24		;CERR AFTER SCLR
7862								
7863	035326	012737	036302	001176	MOV	#14\$,SESCAPE		
7864	035334	013765	001350	000020	MOV	FRCYL,RKDC(R5)		;RETURN TO CYL #
7865	035342	013737	001350	001354	MOV	FRCYL,CCYL		;CURRENT CYL FOR TRUERROR ROUTINE
7866								
7867	035350	012737	000017	003322	MOV	#SEEK,HCS1		
7868	035356	004737	043434		JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
7869	035362	104131			ERROR	131		;NO RDY AFTER SEEK CMD
7870	035364	012737	030140	003412	MOV	#<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0		;EXPECTED A0
7871	035372	005037	003414		CLR	E.B0		
7872	035376	012737	005720	003416	MOV	#<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		
7873	035404	012737	000001	003420	MOV	#1,E.B1		
7874								
7875	035412	004737	044252		JSR	PC,CHKMSG		;CHECK MSGS A0,B0,A1,B1
7876	035416	000003			.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
7877	035420	104203			ERROR	203		;MSG A0 ERROR DURING SEEK CMD
7878	035422	104204			ERROR	204		;MSG B0 ERROR
7879	035424	104205			ERROR	205		;MSG A1 ERROR
7880	035426	104206			ERROR	206		;MSG B1 ERROR
7881								
7882								
7883	035430	012737	036322	001176	MOV	#16\$,SESCAPE		
7884	035436	013737	001426	003360	MOV	T50000,TEMP1		;SETUP TIMEOUT
7885								
7886	035444	004737	044140		JSR	PC,FATT2		;FIND ATTN
7887	035450	104132			ERROR	132		;NO ATTN AFTER SEEK CMD
7888	035452	032737	100000	003322	BIT	#CERR,HCS1		
7889	035460	001401			BEQ	70\$		


```

7890 035462 104210
7891 035464
7892
7893 035464 012737 050340 003412 MOV #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
7894 035472 005037 003414 CLR E.B0 ;EXPECTED MSG B0
7895 035476 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7896 035504 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7897 035512 005037 003422 CLR E.A2 ;EXPECTED MSG A2
7898 035516 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7899 035524 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7900
7901 035532 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
7902 035536 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7903 035540 104133 ERROR 133 ;MSG A0 ERROR AFTER SEEK CMD
7904 035542 104134 ERROR 134 ;MSG B0 ERROR
7905 035544 104135 ERROR 135 ;MSG A1 ERROR
7906 035546 104136 ERROR 136 ;MSG B1 ERROR
7907 035550 005737 001362 TST CYLDIF
7908 035554 001401 BEQ 71$
7909 035556 104137 ERROR 137 ;CYL DIFF NOT CLEARED AFTER SEEK CMD
7910
7911 035560 71$:
7912
7913 035560 012765 100000 000000 MOV #CCLR,RKCS1(R5)
7914 035566 013765 001222 000010 MOV SUNIT,RKCS2(R5) ;DRIVE#
7915 035574 012737 000005 003322 MOV #CLEAR,HCS1
7916 035602 004737 043434 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
7917 035606 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
7918 035610 004737 044012 JSR PC,TSTATN ;TEST FOR ATTN
7919 035614 000401 BR 72$
7920 035616 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
7921 035620 72$:
7922
7923 035620 012737 010340 003412 MOV #<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
7924 035626 005037 003414 CLR E.B0 ;EXPECTED MSG B0
7925 035632 012737 001720 003416 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
7926 035640 012737 000001 003420 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
7927 035646 005037 003422 CLR E.A2 ;EXPECTED MSG A2
7928 035652 012737 000002 003424 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
7929 035660 012737 000003 003430 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
7930
7931 035666 004737 044252 JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
7932 035672 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
7933 035674 104273 ERROR 273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
7934 035676 104265 ERROR 265 ;MSG B0 ERROR
7935 035700 104274 ERROR 274 ;MSG A1 ERROR
7936 035702 104266 ERROR 266 ;MSG B1 ERROR
7937
7938 035704 023737 001364 001350 CMP CYLADD,FRCYL
7939 035712 001401 BEQ 5$
7940 035714 104243 ERROR 243 ;CYL ADDR IN RKMR3 NOT=RKDC
7941
7942 035716 5$:
7943 035716 104415 SCOP1
7944 035720 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
7945

```

7946	035724	004737	045424		JSR	PC, SUBCLR	
7947	035730	104024			ERROR	24	;CERR AFTER SCLR
7948							
7949	035732	005037	001176		CLR	\$ESCAPE	
7950	035736	013765	001350	000020	MOV	FRCYL, RKDC(R5)	;CYL #
7951							
7952							
7953	035744	012700	001674		MOV	#RHTAB, R0	
7954	035750	012737	000025	003322	MOV	#<RDHEAD>, HCS1	
7955	035756	004737	043472		JSR	PC, DATCMD	;DO DATA XFER CMD & GET CONTR RDY
7956	035762	104171			ERROR	171	;NO RDY AFTER READ HEADER CMD
7957	035764	032737	100000	003322	BIT	#CERR, HCS1	
7958	035772	001405			BEQ	74\$	
7959	035774	104174			ERROR	174	;CERR AFTER READ HEADER CMD
7960	035776	104401	056235		TYPE	MSG18	;ABORT BALANCE OF TESTS
7961	036002	000137	042662		JMP	\$EOP	;ABORT DRIVE
7962							
7963	036006	016520	000024		MOV	RKDB(R5), (R0)+	;1'ST WORD FROM SILO TO RHTAB
7964	036012	016520	000024		MOV	RKDB(R5), (R0)+	;2'ND WORD
7965	036016	016520	000024		MOV	RKDB(R5), (R0)+	;3'RD WORD
7966							
7967							
7968	036022	032765	100000	000010	BIT	#DLT, RKCS2(R5)	
7969	036030	001407			BEQ	75\$	
7970	036032	004737	045074		JSR	PC, GSTAT	
7971	036036	104173			ERROR	173	;DLT AFTER READ HEADER CMD
7972	036040	104401	056235		TYPE	MSG18	;ABORTING BALANCE OF TESTS
7973	036044	000137	042662		JMP	\$EOP	;ABORT DRIVE
7974	036050						
7975							
7976	036050	023737	001674	001350	CMP	RHTAB, FRCYL	;CHECK WORD 0 (CYL#) ONLY
7977	036056	001401			BEQ	73\$;BR IF SAME
7978	036060	104311			ERROR	311	;READ CYL WORD HEADER ERROR
7979	036062						
7980							
7981							
7982	036062	023737	001352	012060	CMP	TOCYL, MC	;ALL CYL DONE?
7983	036070	001404			BEQ	6\$;BR IF YES
7984	036072	006337	001352		ASL	TOCYL	;ELSE DO ANOTHER
7985	036076	000137	034520		JMP	1\$	
7986	036102						
7987	036102	004737	047430		JSR	PC, SWTST	;SEE IF SW 14 OR 8 IS SET
7988	036106	000515			BR	TST37	;GO TO NEXT TEST
7989							;RETURN HERE IF SW 14 IS SET OR
7990							;SW 8 WITH SWR <7:0> APPLY
7991	036110						
7992							
7993	036110	004737	045424		JSR	PC, SUBCLR	
7994	036114	104024			ERROR	24	;CERR AFTER SCRL
7995							
7996	036116	013765	001352	000020	MOV	TOCYL, RKDC(R5)	;CYL#
7997							
7998	036124	012737	000017	003322	MOV	#SEEK, HCS1	
7999	036132	004737	043434		JSR	PC, DOCMD	;DO SEEK CMD & GET CONTR READY
8000	036136	104131			ERROR	131	;NO RDY AFTER SEEK CMD.
8001							

```

8002 036140 013737 001426 003360      MOV      T50000,TEMP1
8003 036146 004737 044140      JSR      PC,FATT2      ;FIND ATTN
8004 036152 104132      ERROR    132          ;NO ATTN AFTER SEEK CMD
8005 036154 032737 100000 003322      BIT      #CERR,HCS1
8006 036162 001401      BEQ      78$
8007 036164 104210      ERROR    210          ;CERR AFTER SEEK CMD.
8008
8009 036166 004737 045424      78$:     JSR      PC,SUBCLR
8010 036172 104024      ERROR    24          ;CERR AFTER SCLR
8011
8012 036174 023737 001352 012072      CMP      TOCYL,FC      ;LAST CYL DONE?
8013 036202 001403      BEQ      77$          ;BR IF YES
8014 036204 005337 001352      DEC      TOCYL        ;ELSE DO ANOTHER
8015 036210 000742      BR
8016
8017 036212 004737 045424      77$:     JSR      PC,SUBCLR
8018 036216 104024      ERROR    24          ;CERR AFTER SCLR
8019
8020 036220 005037 001176      CLR      $ESCAPE
8021 036224 005737 001410      TST      LPFLG
8022 036230 001402      BEQ      79$
8023 036232 000177 142652      JMP      @SLPERR      ;SW 9 WAS SET.
8024 036236 000177 142644      79$:     JMP      @SLPADR      ;SW 14 OR 8 WAS SET
8025
8026 036242      10$:
8027 036242 005237 001410      INC      LPFLG
8028 036246 032777 001000 142664      BIT      #SW9,@SWR    ;LOOP ON ERROR?
8029 036254 001315      BNE      8$          ;YES, RECONDITION DRIVE
8030 036256 000137 034660      JMP      2$          ;RETURN TO MAINLINE
8031
8032 036262 005237 001410      12$:     INC      LPFLG
8033 036266 032777 001000 142644      BIT      #SW9,@SWR    ;LOOP ON ERROR?
8034 036274 001305      BNE      8$          ;YES, RECONDITION DRIVE
8035 036276 000137 035146      JMP      3$          ;RETURN TO MAINLINE
8036
8037 036302 005237 001410      14$:     INC      LPFLG
8038 036306 032777 001000 142624      BIT      #SW9,@SWR    ;LOOP ON ERROR?
8039 036314 001275      BNE      8$          ;YES, RECONDITION DRIVE
8040 036316 000137 035430      JMP      4$          ;RETURN TO MAINLINE
8041
8042 036322 005237 001410      16$:     INC      LPFLG
8043 036326 032777 001000 142604      BIT      #SW9,@SWR    ;LOOP ON ERROR?
8044 036334 001265      BNE      8$          ;YES, RECONDITION DRIVE
8045 036336 000137 035716      JMP      5$          ;RETURN TO MAINLINE
8046
8047
8048
8049
8050      ;*****
8051      ;*TEST 37      SEEK TO ALL CYLS FROM 0 & READ HEADERS
8052      ;*****
8052 036342 000004      †ST37:  SCOPE
8053 036344 012737 000001 001174      MOV      #1,$TIMES    ;DO 1 ITERATION
8054 036352 012706 001100      MOV      #STACK,SP    ;RESTORE STK PTR
8055
8056 036356 005737 001342      TST      MODTST        ;SEE IF MODULE TESTING
8057 036362 001402      BEQ      DOSEEK      ;BR IF NO

```

```

8058 036364 000137 042164
8059 036370
8060
8061 036370 013737 012072 001350
8062 036376 013737 012074 001352
8063
8064 036404
8065 036404 104415
8066 036406 012706 001100
8067
8068 036412 004737 045424
8069 036416 104024
8070
8071 036420 012737 040126 001176
8072 036426 013737 001350 003364
8073 036434 013737 001352 003366
8074 036442 163737 003364 003366
8075 036450 013737 003366 001360
8076
8077 036456 013765 001352 000020
8078
8079 036464 012737 000017 003322
8080 036472 004737 043434
8081 036476 104131
8082 036500 012737 030140 003412
8083 036506 005037 003414
8084 036512 012737 003720 003416
8085 036520 012737 000001 003420
8086
8087 036526 004737 044252
8088 036532 000003
8089 036534 104203
8090 036536 104204
8091 036540 104205
8092 036542 104206
8093
8094 036544 012737 040146 001176 25:
8095 036552 013737 001426 003360
8096
8097 036560 004737 044140
8098 036564 104132
8099 036566 032737 100000 003322
8100 036574 001401
8101 036576 104210
8102 036600
8103
8104 036600 012737 050340 003412
8105 036606 005037 003414
8106 036612 012737 001720 003416
8107 036620 012737 000001 003420
8108 036626 005037 003422
8109 036632 012737 000002 003424
8110 036640 012737 000003 003430
8111
8112 036646 004737 044252
8113 036652 000003

```

```

DOSEEK: JMP CYLINV ;ELSE BYPASS TESTS 40 & 41
MOV FC,FRCYL ;SETUP FROM CYL
MOV FCP1,TOCYL ;SETUP TO CYL
1$: SCOP1
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
MOV #10$,SESCAPE
MOV FRCYL,TEMP3 ;SETUP
MOV TOCYL,TEMP4 ;CYL DIFF
SUB TEMP3,TEMP4 ;FOR
MOV TEMP4,CALDIF ;ERROR PRINTOUT
MOV TOCYL,RKDC(R5) ;GO TO CYL #
MOV #SEEK,HCS1
JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
ERROR 131 ;NO RDY AFTER SEEK CMD
MOV #<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED A0
CLR E.B0
MOV #<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
MOV #1,E.B1
JSR PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
.WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
ERROR 203 ;MSG A0 ERROR DURING SEEK CMD
ERROR 204 ;MSG B0 ERROR
ERROR 205 ;MSG A1 ERROR
ERROR 206 ;MSG B1 ERROR
2$: MOV #12$,SESCAPE
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.SPIN!D.DRDY!D.VV!D.DRA>,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

```

M12

CZR6MED UNIBUS RK6 DR PRT1
CZR6ME.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 155
T37 SEEK TO ALL CYLS FROM 0 & READ HEADERS

SEQ 0155

8114	036654	104133			ERROR	133		;MSG A0 ERROR AFTER SEEK CMD
8115	036656	104134			ERROR	134		;MSG B0 ERROR
8116	036660	104135			ERROR	135		;MSG A1 ERROR
8117	036662	104136			ERROR	136		;MSG B1 ERROR
8118	036664	005737	001362		TST	CYLDIF		
8119	036670	001401			BEQ	65\$		
8120	036672	104137			ERROR	137		;CYL DIFF NOT CLEARED AFTER SEEK CMD
8121								
8122	036674						65\$:	
8123								
8124	036674	012765	100000	000000	MOV	#CCLR,RKCS1(R5)		
8125	036702	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	;DRIVE#	
8126	036710	012737	000005	003322	MOV	#CLEAR,HCS1		
8127	036716	004737	043434		JSR	PC,DOCMD		;DO DRIVE CLEAR CMD & GET CONTR RDY
8128	036722	104151			ERROR	151		;NO RDY AFTER DRIVE CLEAR CMD
8129	036724	004737	044012		JSR	PC,TSTATN		;TEST FOR ATTN
8130	036730	000401			BR	66\$		
8131	036732	104154			ERROR	154		;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8132	036734						66\$:	
8133								
8134	036734	012737	010340	003412	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
8135	036742	005037	003414		CLR	E.B0		;EXPECTED MSG B0
8136	036746	012737	001720	003416	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
8137	036754	012737	000001	003420	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
8138	036762	005037	003422		CLR	E.A2		;EXPECTED MSG A2
8139	036766	012737	000002	003424	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
8140	036774	012737	000003	003430	MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
8141								
8142	037002	004737	044252		JSR	PC,CHKMSG		;CHECK MSGS A0,B0,A1,B1
8143	037006	000003			.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
8144	037010	104273			ERROR	273		;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8145	037012	104265			ERROR	265		;MSG B0 ERROR
8146	037014	104274			ERROR	274		;MSG A1 ERROR
8147	037016	104266			ERROR	266		;MSG B1 ERROR
8148								
8149	037020	023737	001364	001352	CMP	CYLADD,TOCYL		
8150	037026	001401			BEQ	3\$		
8151	037030	104207			ERROR	207		;CYL ADDR IN RKMR3 NOT=RKDC
8152							3\$:	
8153	037032							
8154	037032	104415			SCOP1			
8155	037034	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
8156								
8157	037040	004737	045424		JSR	PC,SUBCLR		
8158	037044	104024			ERROR	24		;CERR AFTER SCLR
8159								
8160	037046	005037	001176		CLR	\$ESCAPE		
8161	037052	013765	001352	000020	MOV	TOCYL,RKDC(R5)		;CYL #
8162								
8163								
8164	037060	012700	001674		MOV	#RHTAB,RO		
8165	037064	012737	000025	003322	MOV	#<RDHEAD>,HCS1		
8166	037072	004737	043472		JSR	PC,DATCMD		;DO DATA XFER CMD & GET CONTR RDY
8167	037076	104171			ERROR	171		;NO RDY AFTER READ HEADER CMD
8168	037100	032737	100000	003322	BIT	#CERR,HCS1		
8169	037106	001405			BEQ	68\$		

8170	037110	104174			ERROR	174		;CERR AFTER READ HEADER CMD
8171	037112	104401	056235		TYPE	MSG18		;ABORT BALANCE OF TESTS
8172	037116	000137	042662		JMP	\$EOP		;ABORT DRIVE
8173								
8174	037122	016520	000024	68\$:	MOV	RKDB(R5),(R0)+		;1'ST WORD FROM SILO TO RHTAB
8175	037126	016520	000024		MOV	RKDB(R5),(R0)+		;2'ND WORD
8176	037132	016520	000024		MOV	RKDB(R5),(R0)+		;3'RD WORD
8177								
8178								
8179	037136	032765	100000	000010	BIT	#DLT,RKCS2(R5)		
8180	037144	001407			BEQ	69\$		
8181	037146	004737	045074		JSR	PC,GSTAT		
8182	037152	104173			ERROR	173		;DLT AFTER READ HEADER CMD
8183	037154	104401	056235		TYPE	MSG18		;ABORTING BALANCE OF TESTS
8184	037160	000137	042662		JMP	\$EOP		;ABORT DRIVE
8185	037164			69\$:				
8186								
8187	037164	023737	001674	001352	CMP	RHTAB,TOCYL		;CHECK WORD 0 (CYL#) ONLY
8188	037172	001401			BEQ	67\$;BR IF SAME
8189	037174	104310			ERROR	310		;READ CYL WORD HEADER ERROR
8190	037176			67\$:				
8191								
8192								
8193	037176	104415			SCOP1			
8194	037200	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
8195								
8196	037204	004737	045424		JSR	PC,SUBCLR		
8197	037210	104024			ERROR	24		;CERR AFTER SCLR
8198								
8199	037212	012737	040166	001176	MOV	#14\$, \$ESCAPE		
8200	037220	013765	001350	000020	MOV	FRCYL,RKDC(R5)		;RETURN TO CYL #
8201	037226	013737	001350	001354	MOV	FRCYL,CCYL		;CURRENT CYL FOR TRUERROR ROUTINE
8202								
8203	037234	012737	000017	003322	MOV	#SEEK,HCS1		
8204	037242	004737	043434		JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
8205	037246	104131			ERROR	131		;NO RDY AFTER SEEK CMD
8206	037250	012737	030140	003412	MOV	#<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0		;EXPECTED A0
8207	037256	005037	003414		CLR	E.B0		
8208	037262	012737	005720	003416	MOV	#<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		
8209	037270	012737	000001	003420	MOV	#1,E.B1		
8210								
8211	037276	004737	044252		JSR	PC,CHKMSG		;CHECK MSGS A0,B0,A1,B1
8212	037302	000003			.WORD	T.A2!T.B2!0		& MSGS SPECIFIED HERE
8213	037304	104203			ERROR	203		;MSG A0 ERROR DURING SEEK CMD
8214	037306	104204			ERROR	204		;MSG B0 ERROR
8215	037310	104205			ERROR	205		;MSG A1 ERROR
8216	037312	104206			ERROR	206		;MSG B1 ERROR
8217								
8218								
8219	037314	012737	040206	001176	4\$:	MOV	#16\$, \$ESCAPE	
8220	037322	013737	001426	003360	MOV	T5000,TEMP1		;SETUP TIMEOUT
8221								
8222	037330	004737	044140		JSR	PC,FATT2		;FIND ATTN
8223	037334	104132			ERROR	132		;NO ATTN AFTER SEEK CMD
8224	037336	032737	100000	003322	BIT	#CERR,HCS1		
8225	037344	001401			BEQ	70\$		

```

8226 037346 104210          ERROR 210          ;CERR AFTER SEEK CMD
8227 037350          70$:
8228
8229 037350 012737 050340 003412  MOV      #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
8230 037356 005037 003414          CLR      E.B0 ;EXPECTED MSG B0
8231 037362 012737 001720 003416  MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
8232 037370 012737 000001 003420  MOV      #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
8233 037376 005037 003422          CLR      E.A2 ;EXPECTED MSG A2
8234 037402 012737 000002 003424  MOV      #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
8235 037410 012737 000003 003430  MOV      #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
8236
8237 037416 004737 044252          JSR      PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
8238 037422 000003          .WORD   T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
8239 037424 104133          ERROR   133 ;MSG A0 ERROR AFTER SEEK CMD
8240 037426 104134          ERROR   134 ;MSG B0 ERROR
8241 037430 104135          ERROR   135 ;MSG A1 ERROR
8242 037432 104136          ERROR   136 ;MSG B1 ERROR
8243 037434 005737 001362          TST     CYLDIF
8244 037440 001401          BEQ     71$
8245 037442 104137          ERROR   137 ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8246
8247 037444          71$:
8248
8249 037444 012765 100000 000000  MOV      #CLR,RKCS1(R5)
8250 037452 013765 001222 000010  MOV      $UNIT,RKCS2(R5) ;DRIVE#
8251 037460 012737 000005 003322  MOV      #CLEAR,HCS1
8252 037466 004737 043434          JSR      PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
8253 037472 104151          ERROR   151 ;NO RDY AFTER DRIVE CLEAR CMD
8254 037474 004737 044012          JSR      PC,TSTATN ;TEST FOR ATTN
8255 037500 000401          BR      72$
8256 037502 104154          ERROR   154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8257 037504          72$:
8258
8259 037504 012737 010340 003412  MOV      #<D!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
8260 037512 005037 003414          CLR      E.B0 ;EXPECTED MSG B0
8261 037516 012737 001720 003416  MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
8262 037524 012737 000001 003420  MOV      #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
8263 037532 005037 003422          CLR      E.A2 ;EXPECTED MSG A2
8264 037536 012737 000002 003424  MOV      #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
8265 037544 012737 000003 003430  MOV      #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
8266
8267 037552 004737 044252          JSR      PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
8268 037556 000003          .WORD   T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
8269 037560 104273          ERROR   273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8270 037562 104265          ERROR   265 ;MSG B0 ERROR
8271 037564 104274          ERROR   274 ;MSG A1 ERROR
8272 037566 104266          ERROR   266 ;MSG B1 ERROR
8273
8274 037570 023737 001364 001350  CMP      CYLADD,FRCYL
8275 037576 001401          BEQ     5$
8276 037600 104243          ERROR   243 ;CYL ADDR IN RKMR3 NOT=RKDC
8277
8278 037602          5$:
8279 037602 104415          SCOP1
8280 037604 012706 001100  MOV      #STACK,SP ;RESTORE STK PTR
8281

```

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00MACY11 30A(1052) 25-JAN-78 12:08 PAGE 158
T37 SEEK TO ALL CYLS FROM 0 & READ HEADERS

SEQ 0158

8282	037610	004737	045424		JSR	PC, SUBCLR	
8283	037614	104024			ERROR	24	;CERR AFTER SCLR
8284							
8285	037616	005037	001176		CLR	\$ESCAPE	
8286	037622	013765	001350	000020	MOV	FRCYL, RKDC(R5)	;CYL #
8287							
8288							
8289	037630	012700	001674		MOV	#RHTAB, RO	
8290	037634	012737	000025	003322	MOV	#<RDHEAD>, HCS1	
8291	037642	004737	043472		JSR	PC, DATCMD	;DO DATA XFER CMD & GET CONTR RDY
8292	037646	104171			ERROR	171	;NO RDY AFTER READ HEADER CMD
8293	037650	032737	100000	003322	BIT	#CERR, HCS1	
8294	037656	001405			BEQ	74\$	
8295	037660	104174			ERROR	174	;CERR AFTER READ HEADER CMD
8296	037662	104401	056235		TYPE	MSG18	;ABORT BALANCE OF TESTS
8297	037666	000137	042662		JMP	\$EOP	;ABORT DRIVE
8298							
8299	037672	016520	000024	74\$:	MOV	RKDB(R5), (RO)+	;1'ST WORD FROM SILO TO RHTAB
8300	037676	016520	000024		MOV	RKDB(R5), (RO)+	;2'ND WORD
8301	037702	016520	000024		MOV	RKDB(R5), (RO)+	;3'RD WORD
8302							
8303							
8304	037706	032765	100000	000010	BIT	#DLT, RKCS2(R5)	
8305	037714	001407			BEQ	75\$	
8306	037716	004737	045074		JSR	PC, GSTAT	
8307	037722	104173			ERROR	173	;DLT AFTER READ HEADER CMD
8308	037724	104401	056235		TYPE	MSG18	;ABORTING BALANCE OF TESTS
8309	037730	000137	042662		JMP	\$EOP	;ABORT DRIVE
8310	037734			75\$:			
8311							
8312	037734	023737	001674	001350	CMP	RHTAB, FRCYL	;CHECK WORD 0 (CYL#) ONLY
8313	037742	001401			BEQ	73\$;BR IF SAME
8314	037744	104311			ERROR	311	;READ CYL WORD HEADER ERROR
8315	037746			73\$:			
8316							
8317							
8318	037746	023737	001352	012054	CMP	TOCYL, LC	;ALL CYL DONE?
8319	037754	001404			BEQ	6\$;BR IF YES
8320	037756	005237	001352		INC	TOCYL	;ELSE DO ANOTHER
8321	037762	000137	036404		JMP	1\$	
8322	037766			6\$:			
8323	037766	004737	047430		JSR	PC, SWTST	;SEE IF SW 14 OR 8 IS SET
8324	037772	000515			BR	TST40	;GO TO NEXT TEST
8325							;RETURN HERE IF SW 14 IS SET OR
8326							;SW 8 WITH SWR <7:0> APPLY
8327	037774			8\$:			
8328							
8329	037774	004737	045424		JSR	PC, SUBCLR	
8330	040000	104024			ERROR	24	;CERR AFTER SCRL
8331							
8332	040002	013765	001352	000020	76\$:	MOV	TOCYL, RKDC(R5)
8333							;CYL #
8334	040010	012737	000017	003322	MOV	#SEEK, HCS1	
8335	040016	004737	043434		JSR	PC, DOCMD	;DO SEEK CMD & GET CONTR READY
8336	040022	104131			ERROR	131	;NO RDY AFTER SEEK CMD.
8337							


```

8338 040024 013737 001426 003360      MOV      T50000,TEMP1
8339 040032 004737 044140      JSR      PC,FATT2      ;FIND ATTN
8340 040036 104132      ERROR   132           ;NO ATTN AFTER SEEK CMD
8341 040040 032737 100000 003322      BIT      #CERR,HCS1
8342 040046 001401      BEQ     78$
8343 040050 104210      ERROR   210           ;CERR AFTER SEEK CMD.
8344
8345 040052 004737 045424      78$:    JSR      PC,SUBCLR
8346 040056 104024      ERROR   24           ;CERR AFTER SCLR
8347
8348 040060 023737 001352 012072      CMP     TOCYL,FC      ;LAST CYL DONE?
8349 040066 001403      BEQ     77$           ;BR IF YES
8350 040070 005337 001352      DEC     TOCYL         ;ELSE DO ANOTHER
8351 040074 000742      BR
8352
8353 040076 004737 045424      77$:    JSR      PC,SUBCLR
8354 040102 104024      ERROR   24           ;CERR AFTER SCLR
8355
8356 040104 005037 001176      CLR     $ESCAPE
8357 040110 005737 001410      TST     LPFLG
8358 040114 001402      BEQ     79$
8359 040116 000177 140766      JMP     @SLPERR      ;SW 9 WAS SET.
8360 040122 000177 140760      JMP     @SLPADR      ;SW 14 OR 8 WAS SET
8361
8362 040126      10$:
8363 040126 005237 001410      INC     LPFLG
8364 040132 032777 001000 141000      BIT     #SW9,@SWR   ;LOOP ON ERROR?
8365 040140 001315      BNE     8$           ;YES, RECONDITION DRIVE
8366 040142 000137 036544      JMP     2$           ;RETURN TO MAINLINE
8367
8368 040146      12$:
8369 040152 005237 001410      INC     LPFLG
8370 040152 032777 001000 140760      BIT     #SW9,@SWR   ;LOOP ON ERROR?
8371 040160 001305      BNE     8$           ;YES, RECONDITION DRIVE
8372 040162 000137 037032      JMP     3$           ;RETURN TO MAINLINE
8373
8374 040166      14$:
8375 040172 005237 001410      INC     LPFLG
8376 040172 032777 001000 140740      BIT     #SW9,@SWR   ;LOOP ON ERROR?
8377 040200 001275      BNE     8$           ;YES, RECONDITION DRIVE
8378 040202 000137 037314      JMP     4$           ;RETURN TO MAINLINE
8379
8378 040206      16$:
8379 040206 005237 001410      INC     LPFLG
8380 040212 032777 001000 140720      BIT     #SW9,@SWR   ;LOOP ON ERROR?
8381 040220 001265      BNE     8$           ;YES, RECONDITION DRIVE
8382 040222 000137 037602      JMP     5$           ;RETURN TO MAINLINE
8383
8384      ;*****
8385      ;#TEST 40      SEEK TO ALL CYLS FROM LAST CYL & READ HEADERS
8386      ;*****
8387      TST40:  SCOPE
8388      MOV     #1,STIMES  ;DO 1 ITERATION
8389      MOV     #STACK,SP ;RESTORE STK PTR
8390
8391 040242 004737 045424      JSR      PC,SUBCLR
8392 040246 104024      ERROR   24           ;CERR AFTER SCLR
8393

```

E13

CZR6HED UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00MACY11 30A(1052) 25-JAN-78 12:08 PAGE 160
T40 SEEK TO ALL CYLS FROM LAST CYL & READ HEADERS

SEQ 0160

8394	040250	013765	012054	000020	MOV	LC,RKDC(R5)	;QUICK SEEK TO LAST CYL
8395							
8396	040256	012737	000017	003322	MOV	#SEEK,HCS1	
8397	040264	004737	043434		JSR	PC,DOCMD	;DO SEEK CMD & GET CONTR READY
8398	040270	104131			ERROR	131	;NO RDY AFTER SEEK CMD.
8399							
8400	040272	013737	001426	003360	MOV	T50000,TEMP1	
8401	040300	004737	044140		JSR	PC,FATT2	;FIND ATTN
8402	040304	104132			ERROR	132	;NO ATTN AFTER SEEK CMD
8403	040306	032737	100000	003322	BIT	#CERR,HCS1	
8404	040314	001401			BEQ	645	
8405	040316	104210			ERROR	210	;CERR AFTER SEEK CMD.
8406							
8407	040320	004737	045424		JSR	PC,SUBCLR	
8408	040324	104024			ERROR	24	;CERR AFTER SCLR
8409							
8410							
8411	040326	013737	012054	001350	MOV	LC,FRCYL	;SETUP FROM CYL
8412	040334	013737	012052	001352	MOV	LCM1,TOCYL	;SETUP TO CYL
8413							
8414	040342						
8415	040342	104415			SCOP1		
8416	040344	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
8417							
8418	040350	004737	045424		JSR	PC,SUBCLR	
8419	040354	104024			ERROR	24	;CERR AFTER SCLR
8420							
8421	040356	012737	042064	001176	MOV	#105,SESCAPE	
8422	040364	013737	001350	003364	MOV	FRCYL,TEMP3	;SETUP
8423	040372	013737	001352	003366	MOV	TOCYL,TEMP4	;CYL DIFF
8424	040400	163737	003366	003364	SUB	TEMP4,TEMP3	;FOR
8425	040406	013737	003364	001360	MOV	TEMP3,CALDIF	;ERROR PRINTOUT
8426							
8427	040414	013765	001352	000020	MOV	TOCYL,RKDC(R5)	;GO TO CYL #
8428							
8429	040422	012737	000017	003322	MOV	#SEEK,HCS1	
8430	040430	004737	043434		JSR	PC,DOCMD	;DO SEEK CMD & GET CONTR READY
8431	040434	104131			ERROR	131	;NO RDY AFTER SEEK CMD
8432	040436	012737	030140	003412	MOV	#<D.PIP!D.SPIN!D.VV!D.DRA>,E.A0	;EXPECTED A0
8433	040444	005037	003414		CLR	E.B0	
8434	040450	012737	005720	003416	MOV	#<D.REV!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	
8435	040456	012737	000001	003420	MOV	#1,E.B1	
8436							
8437	040464	004737	044252		JSR	PC,CHKMSG	;CHECK MSGS A0,B0,A1,B1
8438	040470	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
8439	040472	104203			ERROR	203	;MSG A0 ERROR DURING SEEK CMD
8440	040474	104204			ERROR	204	;MSG B0 ERROR
8441	040476	104205			ERROR	205	;MSG A1 ERROR
8442	040500	104206			ERROR	206	;MSG B1 ERROR
8443							
8444	040502	012737	042104	001176	MOV	#125,SESCAPE	
8445	040510	013737	001426	003360	MOV	T50000,TEMP1	;SETUP TIMEOUT
8446							
8447	040516	004737	044140		JSR	PC,FATT2	;FIND ATTN
8448	040522	104132			ERROR	132	;NO ATTN AFTER SEEK CMD
8449	040524	032737	100000	003322	BIT	#CERR,HCS1	

```

8450 040532 001401          BEQ      65$
8451 040534 104210          ERROR    210          ;CERR AFTER SEEK CMD
8452 040536          65$:
8453
8454 040536 012737 050340 003412  MOV     #<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
8455 040544 005037 003414          CLR     E.B0          ;EXPECTED MSG B0
8456 040550 012737 001720 003416  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
8457 040556 012737 000001 003420  MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
8458 040564 005037 003422          CLR     E.A2          ;EXPECTED MSG A2
8459 040570 012737 000002 003424  MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
8460 040576 012737 000003 003430  MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
8461
8462 040604 004737 044252          JSR     PC,CHKMSG     ;CHECK MSGS A0,B0,A1,B1
8463 040610 000003          .WORD  T.A2!T.B2!0    ;& MSGS SPECIFIED HERE
8464 040612 104133          ERROR   133          ;MSG A0 ERROR AFTER SEEK CMD
8465 040614 104134          ERROR   134          ;MSG B0 ERROR
8466 040616 104135          ERROR   135          ;MSG A1 ERROR
8467 040620 104136          ERROR   136          ;MSG B1 ERROR
8468 040622 005737 001362          TST     CYLDIF
8469 040626 001401          BEQ     66$
8470 040630 104137          ERROR   137          ;CYL DIFF NOT CLEARED AFTER SEEK CMD
8471
8472 040632          66$:
8473
8474 040632 012765 100000 000000  MOV     #CCLR,RKCS1(R5)
8475 040640 013765 001222 000010  MOV     SUNIT,RKCS2(R5) ;DRIVE#
8476 040646 012737 000005 003322  MOV     #CLEAR,HCS1
8477 040654 004737 043434          JSR     PC,DOCMD     ;DO DRIVE CLEAR CMD & GET CONTR RDY
8478 040660 104151          ERROR   151          ;NO RDY AFTER DRIVE CLEAR CMD
8479 040662 004737 044012          JSR     PC,TSTATN    ;TEST FOR ATTN
8480 040666 000401          BR      67$
8481 040670 104154          ERROR   154          ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8482 040672          67$:
8483
8484 040672 012737 010340 003412  MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
8485 040700 005037 003414          CLR     E.B0          ;EXPECTED MSG B0
8486 040704 012737 001720 003416  MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
8487 040712 012737 000001 003420  MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
8488 040720 005037 003422          CLR     E.A2          ;EXPECTED MSG A2
8489 040724 012737 000002 003424  MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
8490 040732 012737 000003 003430  MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
8491
8492 040740 004737 044252          JSR     PC,CHKMSG     ;CHECK MSGS A0,B0,A1,B1
8493 040744 000003          .WORD  T.A2!T.B2!0    ;& MSGS SPECIFIED HERE
8494 040746 104273          ERROR   273          ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8495 040750 104265          ERROR   265          ;MSG B0 ERROR
8496 040752 104274          ERROR   274          ;MSG A1 ERROR
8497 040754 104266          ERROR   266          ;MSG B1 ERROR
8498
8499 040756 023737 001364 001352          CMP     CYLADD,TOCYL
8500 040764 001401          BEQ     3$
8501 040766 104207          ERROR   207          ;CYL ADDR IN RKMR3 NOT=RKDC
8502
8503 040770          3$:
8504 040770 104415          SCOP1
8505 040772 012706 001100          MOV     #STACK,SP    ;RESTORE STK PTR

```

8506									
8507	040776	004737	045424		JSR	PC, SUBCLR			
8508	041002	104024			ERROR	24		;CERR AFTER SCLR	
8509									
8510	041004	005037	001176		CLR	\$ESCAPE			
8511	041010	013765	001352	000020	MOV	TOCYL, RKDC(R5)		;CYL #	
8512									
8513									
8514	041016	012700	001674		MOV	#RHTAB, RO			
8515	041022	012737	000025	003322	MOV	#<RDHEAD>, HCS1			
8516	041030	004737	043472		JSR	PC, DATCMD		;DO DATA XFER CMD & GET CONTR RDY	
8517	041034	104171			ERROR	171		;NO RDY AFTER READ HEADER CMD	
8518	041036	032737	100000	003322	BIT	#CERR, HCS1			
8519	041044	001405			BEQ	69\$			
8520	041046	104174			ERROR	174		;CERR AFTER READ HEADER CMD	
8521	041050	104401	056235		TYPE	MSG18		;ABORT BALANCE OF TESTS	
8522	041054	000137	042662		JMP	\$EOP		;ABORT DRIVE	
8523									
8524	041060	016520	000024	69\$:	MOV	RKDB(R5), (RO)+		;1'ST WORD FROM SILO TO RHTAB	
8525	041064	016520	000024		MOV	RKDB(R5), (RO)+		;2'ND WORD	
8526	041070	016520	000024		MOV	RKDB(R5), (RO)+		;3'RD WORD	
8527									
8528									
8529	041074	032765	100000	000010	BIT	#DLT, RKCS2(R5)			
8530	041102	001407			BEQ	70\$			
8531	041104	004737	045074		JSR	PC, GSTAT			
8532	041110	104173			ERROR	173		;DLT AFTER READ HEADER CMD	
8533	041112	104401	056235		TYPE	MSG18		;ABORTING BALANCE OF TESTS	
8534	041116	000137	042662		JMP	\$EOP		;ABORT DRIVE	
8535	041122			70\$:					
8536									
8537	041122	023737	001674	001352	CMP	RHTAB, TOCYL		;CHECK WORD 0 (CYL#) ONLY	
8538	041130	001401			BEQ	68\$;BR IF SAME	
8539	041132	104310			ERROR	310		;READ CYL WORD HEADER ERROR	
8540	041134			68\$:					
8541									
8542									
8543	041134	104415			SCOP1				
8544	041136	012706	001100		MOV	#STACK, SP		;RESTORE STK PTR	
8545									
8546	041142	004737	045424		JSR	PC, SUBCLR			
8547	041146	104024			ERROR	24		;CERR AFTER SCLR	
8548									
8549	041150	012737	042124	001176	MOV	#14\$, \$ESCAPE			
8550	041156	013765	001350	000020	MOV	FRCYL, RKDC(R5)		;RETURN TO CYL #	
8551	041164	013737	001350	001354	MOV	FRCYL, CCYL		;CURRENT CYL FOR TRUERROR ROUTINE	
8552									
8553	041172	012737	000017	003322	MOV	#SEEK, HCS1			
8554	041200	004737	043434		JSR	PC, DOCMD		;DO SEEK CMD & GET CONTR READY	
8555	041204	104131			ERROR	131		;NO RDY AFTER SEEK CMD	
8556	041206	012737	030140	003412	MOV	#<D.PIP!D.SPIN!D.VV!D.DRA>, E.A0		;EXPECTED A0	
8557	041214	005037	003414		CLR	E.B0			
8558	041220	012737	003720	003416	MOV	#<D.FWD!D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>, E.A1			
8559	041226	012737	000001	003420	MOV	#1, E.B1			
8560									
8561	041234	004737	044252		JSR	PC, CHKMSG		;CHECK MSGS A0, B0, A1, B1	

H13

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 163
T40 SEEK TO ALL CYLS FROM LAST CYL & READ HEADERS

SEQ 0163

8562	041240	000003				.WORD	T.A2!T.B2!0		& MSGS SPECIFIED HERE
8563	041242	104203				ERROR	203		;MSG A0 ERROR DURING SEEK CMD
8564	041244	104204				ERROR	204		;MSG B0 ERROR
8565	041246	104205				ERROR	205		;MSG A1 ERROR
8566	041250	104206				ERROR	206		;MSG B1 ERROR
8567									
8568									
8569	041252	012737	042144	001176	4S:	MOV	#16\$,SESCAPE		
8570	041260	013737	001426	003360		MOV	T50000,TEMP1		;SETUP TIMEOUT
8571									
8572	041266	004737	044140			JSR	PC,FATT2		;FIND ATTN
8573	041272	104132				ERROR	132		;NO ATTN AFTER SEEK CMD
8574	041274	032737	100000	003322		BIT	#CERR,HCS1		
8575	041302	001401				BEG	71\$		
8576	041304	104210				ERROR	210		;CERR AFTER SEEK CMD
8577	041306				71\$:				
8578									
8579	041306	012737	050340	003412		MOV	#<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
8580	041314	005037	003414			CLR	E.B0		;EXPECTED MSG B0
8581	041320	012737	001720	003416		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1		;EXPECTED A1
8582	041326	012737	000001	003420		MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
8583	041334	005037	003422			CLR	E.A2		;EXPECTED MSG A2
8584	041340	012737	000002	003424		MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
8585	041346	012737	000003	003430		MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
8586									
8587	041354	004737	044252			JSR	PC,CHKMSG		;CHECK MSGS A0,B0,A1,B1
8588	041360	000003				.WORD	T.A2!T.B2!0		& MSGS SPECIFIED HERE
8589	041362	104133				ERROR	133		;MSG A0 ERROR AFTER SEEK CMD
8590	041364	104134				ERROR	134		;MSG B0 ERROR
8591	041366	104135				ERROR	135		;MSG A1 ERROR
8592	041370	104136				ERROR	136		;MSG B1 ERROR
8593	041372	005737	001362			TST	CYLDIF		
8594	041376	001401				BEG	72\$		
8595	041400	104137				ERROR	137		;CYL DIFF NOT CLEARED AFTER SEEK CMD
8596									
8597	041402				72\$:				
8598									
8599	041402	012765	100000	000000		MOV	#CLR,RKCS1(R5)		
8600	041410	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)		;DRIVE#
8601	041416	012737	000005	003322		MOV	#CLEAR,HCS1		
8602	041424	004737	043434			JSR	PC,DOCMD		;DO DRIVE CLEAR CMD & GET CONTR RDY
8603	041430	104151				ERROR	151		;NO RDY AFTER DRIVE CLEAR CMD
8604	041432	004737	044012			JSR	PC,TSTATN		;TEST FOR ATTN
8605	041436	000401				BR	73\$		
8606	041440	104154				ERROR	154		;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
8607	041442				73\$:				
8608									
8609	041442	012737	010340	003412		MOV	#<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
8610	041450	005037	003414			CLR	E.B0		;EXPECTED MSG B0
8611	041454	012737	001720	003416		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1		;EXPECTED A1
8612	041462	012737	000001	003420		MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
8613	041470	005037	003422			CLR	E.A2		;EXPECTED MSG A2
8614	041474	012737	000002	003424		MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
8615	041502	012737	000003	003430		MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
8616									
8617	041510	004737	044252			JSR	PC,CHKMSG		;CHECK MSGS A0,B0,A1,B1

Address	OpCode	Operand 1	Operand 2	Operand 3	Instruction	Comments
8618	041514	000003			.WORD	T.A2!T.B2!0
8619	041516	104273			ERROR	273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD
8620	041520	104265			ERROR	265 ;MSG B0 ERROR
8621	041522	104274			ERROR	274 ;MSG A1 ERROR
8622	041524	104266			ERROR	266 ;MSG B1 ERROR
8623						
8624	041526	023737	001364	001350	CMP	CYLADD,FRCYL
8625	041534	001401			BEQ	5\$
8626	041536	104243			ERROR	243 ;CYL ADDR IN RKMR3 NOT=RKDC
8627						
8628	041540				5\$:	
8629	041540	104415			SCOP1	
8630	041542	012706	001100		MOV	#STACK,SP ;RESTORE STK PTR
8631						
8632	041546	004737	045424		JSR	PC,SUBCLR
8633	041552	104024			ERROR	24 ;CERR AFTER SCLR
8634						
8635	041554	005037	001176		CLR	\$ESCAPE
8636	041560	013765	001350	000020	MOV	FRCYL,RKDC(R5) ;CYL #
8637						
8638						
8639	041566	012700	001674		MOV	#RHTAB,RO
8640	041572	012737	000025	003322	MOV	#<RDHEAD>,HCS1
8641	041600	004737	043472		JSR	PC,DATCMD ;DO DATA XFER CMD & GET CONTR RDY
8642	041604	104171			ERROR	171 ;NO RDY AFTER READ HEADER CMD
8643	041606	032737	100000	003322	BIT	#CERR,HCS1
8644	041614	001405			BEQ	75\$
8645	041616	104174			ERROR	174 ;CERR AFTER READ HEADER CMD
8646	041620	104401	056235		TYPE	MSG18 ;ABORT BALANCE OF TESTS
8647	041624	000137	042662		JMP	\$EOP ;ABORT DRIVE
8648						
8649	041630	016520	000024		MOV	RKDB(R5),(RO)+ ;1'ST WORD FROM SILO TO RHTAB
8650	041634	016520	000024		MOV	RKDB(R5),(RO)+ ;2'ND WORD
8651	041640	016520	000024		MOV	RKDB(R5),(RO)+ ;3'RD WORD
8652						
8653						
8654	041644	032765	100000	000010	BIT	#DLT,RKCS2(R5)
8655	041652	001407			BEQ	76\$
8656	041654	004737	045074		JSR	PC,GSTAT
8657	041660	104173			ERROR	173 ;DLT AFTER READ HEADER CMD
8658	041662	104401	056235		TYPE	MSG18 ;ABORTING BALANCE OF TESTS
8659	041666	000137	042662		JMP	\$EOP ;ABORT DRIVE
8660	041672				76\$:	
8661						
8662	041672	023737	001674	001350	CMP	RHTAB,FRCYL ;CHECK WORD 0 (CYL#) ONLY
8663	041700	001401			BEQ	74\$;BR IF SAME
8664	041702	104311			ERROR	311 ;READ CYL WORD HEADER ERROR
8665	041704				74\$:	
8666						
8667						
8668	041704	023737	001352	012072	CMP	TOCYL,FC ;ALL CYL DONE?
8669	041712	001404			BEQ	6\$;BR IF YES
8670	041714	005337	001352		DEC	TOCYL ;ELSE DO ANOTHER
8671	041720	000137	040342		JMP	1\$
8672	041724				6\$:	
8673	041724	004737	047430		JSR	PC,SWTST ;SEE IF SW 14 OR 8 IS SET

8674	041730	000515			BR	TST41			::GO TO NEXT TEST
8675									::RETURN HERE IF SW 14 IS SET OR
8676									::SW 8 WITH SWR <7:0> APPLY
8677	041732				8\$:				
8678									
8679	041732	004737	045424			JSR	PC, SUBCLR		
8680	041736	104024				ERROR	24		;CERR AFTER SCRL
8681									
8682	041740	013765	001352	000020	77\$:	MOV	TOCYL, RKDC(R5)		;CYL#
8683									
8684	041746	012737	000017	003322		MOV	#SEEK, HCS1		
8685	041754	004737	043434			JSR	PC, DOCMD		;DO SEEK CMD & GET CONTR READY
8686	041760	104131				ERROR	131		;NO RDY AFTER SEEK CMD.
8687									
8688	041762	013737	001426	003360		MOV	T50000, TEMP1		
8689	041770	004737	044140			JSR	PC, FATT2		;FIND ATTN
8690	041774	104132				ERROR	132		;NO ATTN AFTER SEEK CMD
8691	041776	032737	100000	003322		BIT	#CERR, HCS1		
8692	042004	001401				BEQ	79\$		
8693	042006	104210				ERROR	210		;CERR AFTER SEEK CMD.
8694									
8695	042010	004737	045424		79\$:	JSR	PC, SUBCLR		
8696	042014	104024				ERROR	24		;CERR AFTER SCLR
8697									
8698	042016	023737	001352	012054		CMP	TOCYL, LC		;LAST CYL DONE?
8699	042024	001403				BEQ	78\$;BR IF YES
8700	042026	005237	001352			INC	TOCYL		;ELSE DO ANOTHER
8701	042032	000742				BR	77\$		
8702									
8703	042034	004737	045424		78\$:	JSR	PC, SUBCLR		
8704	042040	104024				ERROR	24		;CERR AFTER SCLR
8705									
8706	042042	005037	001176			CLR	\$ESCAPE		
8707	042046	005737	001410			TST	LPFLG		
8708	042052	001402				BEQ	80\$		
8709	042054	000177	137030			JMP	\$SLPERR		;SW 9 WAS SET.
8710	042060	000177	137022		80\$:	JMP	\$SLPADR		;SW 14 OR 8 WAS SET
8711									
8712	042064				10\$:				
8713	042064	005237	001410			INC	LPFLG		
8714	042070	032777	001000	137042		BIT	#SW9, \$SWR		;LOOP ON ERROR?
8715	042076	001315				BNE	8\$;YES, RECONDITION DRIVE
8716	042100	000137	040502			JMP	2\$;RETURN TO MAINLINE
8717	042104								
8718	042104	005237	001410		12\$:	INC	LPFLG		
8719	042110	032777	001000	137022		BIT	#SW9, \$SWR		;LOOP ON ERROR?
8720	042116	001305				BNE	8\$;YES, RECONDITION DRIVE
8721	042120	000137	040770			JMP	3\$;RETURN TO MAINLINE
8722	042124								
8723	042124	005237	001410		14\$:	INC	LPFLG		
8724	042130	032777	001000	137002		BIT	#SW9, \$SWR		;LOOP ON ERROR?
8725	042136	001275				BNE	8\$;YES, RECONDITION DRIVE
8726	042140	000137	041252			JMP	4\$;RETURN TO MAINLINE
8727	042144								
8728	042144	005237	001410		16\$:	INC	LPFLG		
8729	042150	032777	001000	136762		BIT	#SW9, \$SWR		;LOOP ON ERROR?

8730 042156 001265
8731 042160 000137 041540

BNE 8\$;YES, RECONDITION DRIVE
JMP 5\$;RETURN TO MAINLINE

8732
8733 042164

CYLINV:

8734
8735
8736
8737
8738
8739
8740
8741
8742
8743
8744
8745
8746

*TEST 41 SEEK TO ALL INVALID CYLINDERS
*
* THIS TEST VERIFIES THAT 'INV ADDR' & 'SEEK INCOMPLETE' IS
* PRODUCED & THAT HEADS DO NOT MOVE OR UNLOAD IF AN ILLEGAL
* CYL IS SPECIFIED IN A SEEK.
*
* INVALID CYLS ARE 633 THRU 777 (8) FOR THE RK06
* & 1457 THRU 1777 FOR THE RK07.

8747 042164 000004
8748 042166 012737 000001 001174
8749 042174 012706 001100

TST41: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR

8751 042200 004737 045424
8752 042204 104024

JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR

8754 042206 012737 000017 003322
8755 042214 004737 043434
8756 042220 104131

MOV #SEEK,HCS1
JSR PC,DOCMD ;DO SEEK CMD & GET CONTR READY
ERROR 131 ;NO RDY AFTER SEEK CMD.

8758 042222 013737 001426 003360
8759 042230 004737 044140

MOV T50000,TEMP1
JSR PC,FATT2 ;FIND ATTN
ERROR 132 ;NO ATTN AFTER SEEK CMD

8761 042236 032737 100000 003322
8762 042244 001401
8763 042246 104210

BIT #CERR,HCS1
BEQ 64\$
ERROR 210 ;CERR AFTER SEEK CMD.

8765 042250 004737 045424
8766 042254 104024

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

8768 042256 005000
8769 042260 005037 001350
8770 042264 013737 012056 001352

CLR R0
CLR FRCYL ;FROM CYL 0
MOV LCP1,TOCYL ;1'ST INV CYL

8772 042272
8773 042272 104415
8774 042274 012706 001100

1\$: SCOP1
MOV #STACK,SP ;RESTORE STK PTR

8776 042300 004737 045424
8777 042304 104024

JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR

8779 042306 013737 001352 001360
8780 042314 013765 001352 000020
8781 042322 012737 000017 003322

MOV TOCYL,CALDIF
MOV TOCYL,RKDC(R5)
MOV #SEEK,HCS1
JSR PC,DOCMD ;DO SEEK CMD & GET CONTR RDY
ERROR 131 ;NO RDY AFTER SEEK CMD

8782 042330 004737 043434
8783 042334 104131
8784 042336 004737 044012

JSR PC,TSTATN
ERROR 24\$;NO ATTN AFTER SEEK TO INV CYL

8785 042342 104245

8786										
8787	042344	032737	000040	003352		BIT	#D.IDAE,HMR3			
8788	042352	001001				BNE	2\$			
8789	042354	104246				ERROR	246		; IDAE NOT SET AFTER SEEK TO INVALID ADDR	
8790	042356	032737	000200	003352	2\$:	BIT	#D.FLT,HMR3			
8791	042364	001001				BNE	4\$			
8792	042366	104247				ERROR	247		; FLT NOT SET AFTER SEEK TO INV ADDR	
8793	042370	032737	020000	003350	4\$:	BIT	#D.PIP,HMR2			
8794	042376	001401				BEQ	5\$			
8795	042400	104250				ERROR	250		; PIP SET AFTER SEEK TO INV ADDR	
8796	042402	032737	040000	003350	5\$:	BIT	#D.DSC,HMR2			
8797	042410	001001				BNE	6\$			
8798	042412	104251				ERROR	251		; DSC NOT SET AFTER SEEK TO INV ADDR	
8799										
8800	042414	005237	001462		6\$:	INC	BYPCERR		; BYPASS CHECKING FOR CERR IN GSTAT1	
8801	042420	012737	050340	003412		MOV	#<D.DSC!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		; EXPECTED A0	
8802	042426	012737	002240	003414		MOV	#<D.SKI!D.FLT!D.IDAE>,E.B0			
8803	042434	012737	001720	003416		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1			
8804	042442	012737	000001	003420		MOV	#1,E.B1			
8805										
8806	042450	004737	044252			JSR	PC,CHKMSG		; CHECK MSGS A0,B0,A1,B1	
8807	042454	000003				.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE	
8808	042456	104252				ERROR	252		; MSG A0 ERROR AFTER SEEK TO INV CYL	
8809	042460	104253				ERROR	253		; MSG B0 ERROR	
8810	042462	104254				ERROR	254		; MSG A1 ERROR	
8811	042464	104255				ERROR	255		; MSG B1 ERROR	
8812	042466	023737	001362	001352		CMP	CYLDIF,TOCYL			
8813	042474	001401				BEQ	7\$			
8814	042476	104256				ERROR	256		; CYL DIFF IN RKMR2 NOT=CYL DIF	
8815	042500	023737	001364	001352	7\$:	CMP	CYLADD,TOCYL			
8816	042506	001401				BEQ	8\$			
8817	042510	104257				ERROR	257		; CYL ADDR IN RKMR3 NOT=RKDC	
8818										
8819	042512				8\$:					
8820										
8821	042512	012765	100000	000000		MOV	#CCLR,RKCS1(R5)			
8822	042520	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)			
8823	042526	012737	000013	003322		MOV	#RECAL,HCS1			
8824	042534	004737	043434			JSR	PC,DOCMD		; DO RECAL CMD & GET CONTR RDY	
8825	042540	104124				ERROR	124		; RDY NOT FOUND AFTER RECAL CMD	
8826										
8827	042542	012765	100000	000000		MOV	#CCLR,RKCS1(R5)			
8828	042550	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)		; DRIVE#	
8829	042556	012737	000005	003322		MOV	#CLEAR,HCS1			
8830	042564	004737	043434			JSR	PC,DOCMD		; DO DRIVE CLEAR CMD & GET CONTR RDY	
8831	042570	104151				ERROR	151		; NO RDY AFTER DRIVE CLEAR CMD	
8832	042572	004737	044012			JSR	PC,TSTATN		; TEST FOR ATTN	
8833	042576	000401				BR	66\$			
8834	042600	104154				ERROR	154		; ATTN NOT CLEARED AFTER DRIVE CLEAR CMD	
8835	042602				66\$:					
8836										
8837										
8838	042602	004737	045074			JSR	PC,GSTAT			
8839	042606	032737	000040	003352		BIT	#D.IDAE,HMR3		; SEE IF IDAE IS CLEARED	
8840	042614	001401				BEQ	65\$; BR IF YES	
8841	042616	104155				ERROR	155		; IDAE NOT CLEARED AFTER RECAL CMD	


```

8855 .SBTTL END OF PASS ROUTINE
8856
8857
8858
8859
8860
8861
8862
8863 042662
8864
8865 042662 000004 SCOPE
8866 042664 005037 001176 CLR $ESCAPE
8867 042670 012737 000001 001174 MOV #1,$TIMES
8868 042676 012706 001100 MOV #STACK,SP
8869 042702 005237 001220 INC $DEVCT ; INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
8870 042706 023737 003442 001220 CMP DRIVS,$DEVCT ; ARE ALL DRIVES PRESENT TESTED?
8871 042714 001403 BEQ $EOP1+2 ; BR IF YES
8872 042716 000137 011520 JMP NUDRV ; IF NOT , TEST NEXT DRIVE PRESENT
8873 042722 000004 $EOP1: SCOPE
8874 042724 005037 001102 CLR $TSTNM ; ZERO THE TEST NUMBER
8875 042730 005037 001174 CLR $TIMES ; ZERO THE NUMBER OF ITERATIONS
8876 042734 005237 001216 INC $PASS ; INCREMENT THE PASS NUMBER
8877 042740 042737 100000 001216 BIC #100000,$PASS ; DON'T ALLOW A NEG. NUMBER
8878 042746 005327 DEC (PC)+ ; LOOP?
8879 042750 000001 $EOPCT: .WORD 1
8880 042752 003022 BGT $DOAGN ; YES
8881 042754 012737 MOV (PC)+,a(PC)+ ; RESTORE COUNTER
8882 042756 000001 $ENDCT: .WORD 1
8883 042760 042750 $EOPCT
8884 042762 104401 043027 TYPE $SENDMG ; TYPE "END PASS #"
8885 042766 013746 001216 MOV $PASS,-(SP) ; SAVE $PASS FOR TYPEOUT
8886 042772 104405 TYPDS ; GO TYPE--DECIMAL ASCII WITH SIGN
8887 042774 104401 043024 TYPE $ENULL ; TYPE A NULL CHARACTER
8888 043000 013700 000042 $GET42: MOV #42,RO ; GET MONITOR ADDRESS
8889 043004 001405 BEQ $DOAGN ; BRANCH IF NO MONITOR
8890 043006 000005 RESET ; CLEAR THE WORLD
8891 043010 004710 $ENDAD: JSR PC,(RO) ; GO TO MONITOR
8892 043012 000240 NOP ; SAVE ROOM
8893 043014 000240 NOP ; FOR
8894 043016 000240 NOP ; ACT11
8895 043020 $DOAGN:
8896 043020 000137 JMP a(PC)+ ; RETURN
8897 043022 043044 $RTNAD: .WORD ST5XY
8898 043024 377 377 000 $ENULL: .BYTE -1,-1,0 ; NULL CHARACTER STRING
8899 043027 015 042412 042116 $SENDMG: .ASCIZ <15><12>/END PASS #/
8900 043034 050040 051501 020123
8901 043042 000043
8902 043044 122737 000001 001230 ST5XY: CMPB #APTENV,$ENV ; RUN UNDER APT ?
8903 043052 001007 BNE 2$ ; BRANCH IF NOT
8904 043054 022737 000002 001216 CMP #2,$PASS ; TWO PASSES DONE ?
8905 043062 101003 BHI 2$ ; BRANCH IF NOT
8906 043064 005237 001102 1$: INC $TSTNM ; CHANGE THE TEST NUMBER
8907 043070 000775 BR 1$ ; LOOP WAIT FOR LOADING THE NEXT PRG
8908 043072 000137 007730 2$: JMP ST5 ; EXIT

```

```

8909      .SBTTL  SUBROUTINES
8910
8911      ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
8912      ;
8913
8914      043076  012700  003432  CLRFLG: MOV      #DDUMP,R0
8915      043102  012701  177757      MOV      #-17.,R1
8916      043106  005020      1$:      CLR      (R0)+
8917      043110  005201      INC      R1
8918      043112  001375      BNE     1$
8919      043114  000207      RTS     PC
8920
8921
8922      ;TYPE PROGRAM ID IF FTITLE=0
8923      ;
8924
8925      043116  005737  001344  TITLE:  TST      FTITLE
8926      043122  001024      BNE     1$
8927      043124  005237  001344      INC      FTITLE
8928      043130  104401  054516      TYPE    MSG1      ;PROGRAM ID
8929
8930      .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
8931      043134  005737  000042      TST     @#42      ;ARE WE RUNNING UNDER XXDP/ACT?
8932      043140  001012      BNE     64$      ;BRANCH IF YES
8933      043142  123727  001230  000001  CMPB    $ENV,#1    ;ARE WE RUNNING UNDER APT?
8934      043150  001406      BEQ     64$      ;BRANCH IF YES
8935      043152  023727  001140  000176  CMP     SWR,#SWREG ;SOFTWARE SWITCH REG SELECTED?
8936      043160  001005      BNE     65$      ;BRANCH IF NO
8937      043162  104406      GTSWR   ;GET SOFT-SWR SETTINGS
8938      043164  000403      BR      65$
8939      043166  112737  000001  001134  64$:   MOVB    #1,$AUTOB ;SET AUTO-MODE INDICATOR
8940      043174  000207      65$:   RTS     PC
8941
8942
8943      ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
8944      ;DRIVS, DRIVO-DRIV? REGISTERS APPROPRIATELY
8945      ;
8946
8947      043176  104411  GDRVS:  RDLIN
8948      043200  012600      MOV     (SP)+,R0 ;GET STARTING ADDR OF ASCII STRING
8949      043202  012701  177770      MOV     #-8.,R1 ;SET UP COUNT
8950      043206  112002      1$:    MOVB   (R0)+,R2 ;GET ASCII CHAR
8951      043210  042702  177400      BIC    #177400,R2 ;MASK HI BYTE
8952      043214  012703  003444      MOV     #DRIVO,R3 ;DRIVE FLAG ADDR
8953      043220  012704  000060      MOV     #60,R4
8954
8955      043224  020402      2$:    CMP     R4,R2 ;WAS TYPED CHAR 0 THRU 7?
8956      043226  001415      BEQ    3$      ;BRANCH IF YES
8957      043230  005723      TST    (R3)+ ;NO, INCREMENT DR FLAG ADDR
8958      043232  005204      INC    R4
8959      043234  020427  000070      CMP    R4,#70
8960      043240  001371      BNE    2$      ;S/B 0-7 OR TERMINATOR
8961      043242  005702      TST    R2
8962      043244  001022      BNE    4$
8963      043246  020127  177770      CMP    R1,#-8.
8964      043252  001426      BEQ    6$      ;DEFAULT ALL DRIVES

```

```

8965 043254 005037 003472 7$: CLR      SIZFLG      ;BYPASS TEST 1 (SIZING)
8966 043260 000207          RTS      PC          ;FOUND TERMINATOR, EXIT
8967
8968 043262 005213          3$: INC      DR3        ;SET UP FLAG FOR THE DRIVE
8969 043264 005237 003442  INC      DRVS       ;INCREMENT TOTAL # DRIVES TO BE TESTED
8970 043270 112002          MOV      (R0)+,R2   ;GET NEXT ASCII CHAR.
8971 043272 042702 177400  BIC      #177400,R2 ;MASK
8972 043276 022702 000054  CMP      #54,R2    ;IS IT A COMMA?
8973 043302 001407          BEQ      5$         ;YES, GO TO NEXT WORD.
8974 043304 005702          TST      R2        ;NO, IS IT A TERMINATOR?
8975 043306 001001          BNE      4$        ;IF NOT, SOMETHING WRONG.
8976 043310 000761          BR       7$        ;FOUND TERMINATOR, EXIT
8977
8978 043312 104401 056744 4$: TYPE   EM1       ;ONLY 0-7 ALLOWED.
8979 043316 000137 007134  JMP      PRGSR     ;START ALL OVER
8980
8981 043322 005201          5$: INC      R1        ;S/B NO MORE THAN 8 DIFF
8982 043324 001330          BNE      1$        ;DRIVES TYPED IN.
8983 043326 000771          BR       4$        ;IF MORE, HAVE ERROR.
8984
8985 043330 005237 003472 6$: INC      SIZFLG   ;DO TEST 1 (SIZING)
8986 043334 000207          RTS      PC        ;EXIT.
8987
8988
8989
8990
8991
8992
8993 043336 104412          ;ROUTINE TO INPUT RKBAS OR DEFAULT.
8994 043340 012600          GBA:  RDOCT
8995 043342 005700          MOV      (SP)+,RO  ;GET LOW ORDER FROM STACK
8996 043344 001403          TST      RO
8997 043346 010037 001264  BEQ      1$        ;BRANCH IF DEFAULT.
8998 043352 000207          MOV      RO,$BASE
8999 043354 012737 177440 001264 1$: MOV      #177440,$BASE ;DEFAULT VALUE
9000 043362 000207          RTS      PC
9001
9002
9003
9004
9005 043364 104412          ;ROUTINE TO INPUT RKVEC OR DEFAULT
9006 043366 012600          GINT: RDOCT
9007 043370 005700          MOV      (SP)+,RO  ;GET LOW ORDER FROM STACK
9008 043372 001405          TST      RO
9009 043374 010037 001314  BEQ      1$        ;BRANCH IF DEFAULT
9010 043400 004737 043416 2$: MOV      RO,RKVEC
9011 043404 000207          JSR      PC,SETINT
9012 043406 012737 000210 001314 1$: MOV      #210,RKVEC ;DEFAULT VALUE
9013 043414 000771          BR       2$
9014
9015
9016
9017
9018
9019 043416 013700 001314  ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
9020 043422 012720 050156  SETINT: MOV      RKVEC,RO
          MOV      #INTER,(RO)+ ;INTER ADDR TO RKVEC

```

```

9021 043426 013710 001316          MOV    RKPRI, (R0)      ;PRS TO RKVEC+2
9022 043432 000207          RTS    PC
9023
9024
9025
9026
9027
9028 043434 053737 001170 003322  DOCMD:  BIS    STMP4, HCS1      ;ADD CDT IF RK07
9029 043442 013765 003322 000000      MOV    HCS1, RKCS1(R5) ;DO COMMAND
9030 043450 013737 001414 003360      MOV    T10, TEMP1
9031 043456 004737 043530          JSR    PC, FRDY        ;FIND CONTR READY
9032 043462 000207          RTS    PC              ;SET HERE IF NOT RDY
9033 043464 062716 000002          ADD    #2, (SP)        ;ELSE SKIP OVER ERROR
9034 043470 000207          RTS    PC
9035
9036
9037
9038
9039 043472 053737 001170 003322  DATCMD: BIS    STMP4, HCS1      ;ADD CDT IF RK07
9040 043500 013765 003322 000000      MOV    HCS1, RKCS1(R5) ;DO CMD
9041 043506 013737 001426 003360      MOV    T50000, TEMP1
9042 043514 004737 043530          JSR    PC, FRDY        ;FIND CONTR RDY
9043 043520 000207          RTS    PC
9044 043522 062716 000002          ADD    #2, (SP)
9045 043526 000207          RTS    PC
9046
9047
9048
9049
9050
9051
9052
9053
9054 043530 032765 000200 000000  FRDY:   BIT    #RDY, RKCS1(R5)
9055 043536 001010          BNE    IS
9056 043540 005337 003360          DEC    TEMP1
9057 043544 001371          BNE    FRDY
9058 043546 004737 043664          JSR    PC, HOLD        ;STORE ALL RK611 REGS IN HOLDING REGS.
9059 043552 004737 045012          JSR    PC, CKCERR      ;CHECK FOR SPECIAL CERR
9060 043556 000207          RTS    PC              ;NO RDY, EXIT
9061 043560 062716 000002          IS:    ADD    #2, (SP)   ;SKIP OVER ERROR
9062 043564 004737 043664          JSR    PC, HOLD
9063 043570 004737 045012          JSR    PC, CKCERR      ;CHECK FOR SPECIAL CERR
9064 043574 000207          RTS    PC
9065
9066
9067
9068
9069 043576 032765 000200 000000  FRDY1:  BIT    #RDY, RKCS1(R5)
9070 043604 001014          BNE    IS
9071 043606 005337 003360          DEC    TEMP1
9072 043612 001371          BNE    FRDY1
9073 043614 016537 000034 003350      MOV    RKMR2(R5), HMR2
9074 043622 016537 000036 003352      MOV    RKMR3(R5), HMR3
9075 043630 004737 045012          JSR    PC, CKCERR      ;CHECK FOR SPECIAL CERR CONDITIONS
9076 043634 000207          RTS    PC              ;NO RDY, EXIT

```

```

9077 043636 062716 000002          1S:  ADD      #2,(SP)          ;SKIP OVER ERROR
9078 043642 016537 000034 003350  MOV      RKMR2(R5),HMR2
9079 043650 016537 000036 003352  MOV      RKMR3(R5),HMR3
9080 043656 004737 045012          JSR      PC,CKCERR          ;CHECK FOR SPECIAL CERR CONDITIONS
9081 043662 000207          RTS      PC
9082
9083          ;STORE ALL RK611 REGISTERS IN HOLDING REGS
9084
9085
9086 043664 016537 000000 003322  HOLD:  MOV      RKCS1(R5),HCS1
9087 043672 016537 000010 003324  MOV      RKCS2(R5),HCS2
9088 043700 016537 000002 003326  MOV      RKWC(R5),HWC
9089 043706 016537 000004 003330  MOV      RKBA(R5),HBA
9090 043714 016537 000006 003332  MOV      RKDA(R5),HDA
9091 043722 016537 000012 003334  MOV      RKDS(R5),HDS
9092 043730 016537 000014 003336  MOV      RKER(R5),HER
9093 043736 016537 000016 003340  MOV      RKASOF(R5),HASOF
9094 043744 016537 000020 003342  MOV      RKDC(R5),HDC
9095 043752 016537 000026 003346  MOV      RKMR1(R5),HMR1
9096 043760 016537 000034 003350  MOV      RKMR2(R5),HMR2
9097 043766 016537 000036 003352  MOV      RKMR3(R5),HMR3
9098 043774 016537 000030 003354  MOV      RKECPS(R5),HPOS
9099 044002 016537 000032 003356  MOV      RKECPT(R5),HPAT
9100 044010 000207          RTS      PC
9101
9102
9103          ;ROUTINE TO CHECK FOR CORRECT ATTN
9104          ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
9105          ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
9106
9107 044012 010446          ;STATN: MOV      R4,-(SP)          ;SAV R4
9108 044014 013704 001222          MOV      $UNIT,R4
9109 044020 136437 003312 003341  BITB     ATTN(R4),HASOF+1
9110 044026 001404          BEQ      1S
9111 044030 012604          MOV      (SP)+,R4          ;BRANCH IF ATTN NOT PRESENT
9112 044032 062716 000002          ADD      #2,(SP)          ;RESTOR R4
9113 044036 000207          RTS      PC              ;INCR RET ADDR TO JUMP OVER ERROR.
9114 044040 012604          1S:  MOV      (SP)+,R4          ;RESTOR R4
9115 044042 000207          RTS      PC
9116
9117
9118          ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
9119          ;ENTER WITH TIME IN SECONDS IN TEMP2
9120          ;RETURN IF NO ATTN (ERROR CONDITION)
9121          ;RETURN +2 IF ATTN FOUND
9122          ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
9123
9124
9125 044044 010446          FATT1: MOV      R4,-(SP)          ;SAV R4
9126 044046 012737 177777 003360  3S:  MOV      #-1,TEMP1
9127 044054 013704 001222          MOV      $UNIT,R4
9128 044060 136465 003312 000017  1S:  BITB     ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
9129 044066 001014          BNE     2S
9130 044070 005337 003360          DEC     TEMP1
9131 044074 001371          BNE     1S
9132 044076 005337 003362          DEC     TEMP2

```

```

9133 044102 001361          BNE      3$
9134 044104 005065 000026    CLR      RKMR1(R5)      ;SELECT WORD 0
9135 044110 004737 045074    JSR      PC,GSTAT      ;GET LATEST STATUS
9136 044114 012604          MOV      (SP)+,R4      ;RESTOR R4
9137 044116 000207          RTS      PC
9138 044120 005065 000026    2$:     CLR      RKMR1(R5)
9139 044124 004737 045074    JSR      PC,GSTAT      ;GET STATUS AFTER ATTN SEEN
9140 044130 012604          MOV      (SP)+,R4      ;RESTOR R4
9141 044132 062716 000002    ADD      #2,(SP)      ;SKIP OVER ERROR
9142 044136 000207          RTS      PC
9143
9144
9145      ;ROUTINE TO FIND ATTN WITHIN 1 SEC
9146      ;ENTER WITH COUNT IN TEMP1
9147      ;RETURN IF NO ATTN (ERROR)
9148      ;RETURN +2 IF ATTN FOUND
9149      ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
9150
9151
9152 044140 010446          FATT2:  MOV      R4,-(SP)      ;SAV R4
9153 044142 013704 001222    2$:     MOV      $UNIT,R4
9154 044146 136465 003312 000017  BITB     ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
9155 044154 001011          BNE      1$
9156 044156 005337 003360    DEC      TEMP1
9157 044162 001367          BNE      2$
9158 044164 005065 000026    CLR      RKMR1(R5)      ;SELECT WORD 0
9159 044170 004737 045074    JSR      PC,GSTAT      ;GET LATEST STATUS.
9160 044174 012604          MOV      (SP)+,R4      ;RESTOR R4
9161 044176 000207          RTS      PC
9162 044200 005065 000026    1$:     CLR      RKMR1(R5)
9163 044204 004737 045074    JSR      PC,GSTAT
9164 044210 012604          MOV      (SP)+,R4      ;RESTOR R4
9165 044212 062716 000002    ADD      #2,(SP)      ;SKIP OVER ERROR
9166 044216 000207          RTS      PC
9167
9168      ;ENTER WITH A COUNT IN TEMP1
9169      ;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
9170      ;WHEN COUNT IS 0...BASED ON AN 11/05.
9171
9172 044220 005737 003360    DLY:    TST      TEMP1      ;5.6 US
9173 044224 001403          BEQ      1$            ;2.5 US
9174 044226 005337 003360    DEC      TEMP1        ;6.8 US
9175 044232 000772          BR       DLY          ;2.5 US
9176 044234 000207          1$:     RTS      PC      ;3.8 US
9177
9178      ;THIS ROUTINE TYPES BYPASSED DRIVE#. ENTER WITH DRIVE# IN RO
9179
9180
9181 044236 104401 056130    BYP:    TYPE     MSG14      ;BYPASS DRIVE
9182 044242 010046          MOV      RO,-(SP)      ;SAVE RO FOR TYPEOUT
9183
9184          TYPOS          ;TYPE DR#
9185 044244 104403          .BYTE   1            ;GO TYPE--OCTAL ASCII
9186 044246 001          .BYTE   0            ;TYPE 1 DIGIT(S)
9187 044247 000          .BYTE   0            ;SUPPRESS LEADING ZEROS
9188 044250 000207          RTS      PC

```



```

; THIS ROUTINE READS ALL MSG A&B WORDS & CHECKS THEM AS REQ'D
9189
9190
9191 044252 017637 000000 001466 CHKMSG: MOV 3(SP),CHKFLG ;PASS MSGS TO BE TESTED
9192 044260 062716 000002 ADD #2,(SP) ;BUMP RETURN ADDR TO 1ST ERROR
9193 044264 004737 045130 JSR PC,GSTAT1 ;GET ALL ACTUAL DRIVE & CONTR STATUS
9194
9195 044270 053737 001222 003412 BIS $UNIT,E.A0 ;SET UNIT #
9196 044276 053737 001222 003416 BIS $UNIT,E.A1
9197 044304 053737 001222 003422 BIS $UNIT,E.A2
9198 044312 053737 001222 003426 BIS $UNIT,E.A3
9199 044320 053737 012070 003412 BIS E.DDT,E.A0 ;ADD EXP. DRV TYPE
9200
9201 044326 013746 003360 MOV TEMP1,-(SP) ;SAVE TEMP 1
9202
9203 044332 013737 003412 003360 MOV E.A0,TEMP1
9204 044340 004737 047324 JSR PC,S&PAR ;GET PARITY FOR MSG A0
9205 044344 013737 003360 003412 MOV TEMP1,E.A0
9206
9207 044352 013737 003416 003360 MOV E.A1,TEMP1
9208 044360 004737 047324 JSR PC,S&PAR ;GET PARITY FOR MSG A1
9209 044364 013737 003360 003416 MOV TEMP1,E.A1
9210
9211 044372 013737 003422 003360 MOV E.A2,TEMP1
9212 044400 004737 047324 JSR PC,S&PAR ;GET PARITY FOR MSG A2
9213 044404 013737 003360 003422 MOV TEMP1,E.A2
9214
9215 044412 013737 003414 003360 MOV E.B0,TEMP1
9216 044420 004737 047324 JSR PC,S&PAR ;GET PARITY FOR MSG B0
9217 044424 013737 003360 003414 MOV TEMP1,E.B0
9218
9219 044432 013737 003420 003360 MOV E.B1,TEMP1
9220 044440 004737 047324 JSR PC,S&PAR ;GET PARITY FOR MSG B1
9221 044444 013737 003360 003420 MOV TEMP1,E.B1
9222
9223 044452 013737 003424 003360 MOV E.B2,TEMP1
9224 044460 004737 047324 JSR PC,S&PAR ;GET PARITY FOR MSG B2
9225 044464 013737 003360 003424 MOV TEMP1,E.B2
9226
9227 044472 013737 003430 003360 MOV E.B3,TEMP1
9228 044500 004737 047324 JSR PC,S&PAR ;GET PARITY FOR MSG B3
9229 044504 013737 003360 003430 MOV TEMP1,E.B3
9230
9231 044512 012637 003360 MOV (SP)+,TEMP1 ;RESTORE TEMP 1
9232 044516 013737 001176 001172 MOV $ESCAPE,$TMP5 ;SAVE ESCAPE
9233
9234 044524 023737 003372 003412 CMP H.A0,E.A0 ;TEST MSG A0
9235 044532 001411 BEQ 2$ ;BR IF OK
9236 044534 012737 044546 001176 MOV #1,$ESCAPE ;ELSE SETUP ESCAPE
9237 044542 011646 MOV (SP),-(SP) ;COPY RET ADDR
9238 044544 000207 RTS PC ;& RETURN TO MAINLINE ERROR
9239
9240 044546 032777 001000 134364 1$: BIT #SW9,$SWR ;RET HERE FROM MAINLINE ERROR
9241 044554 001107 BNE 2$ ;& BR IF LOOP ON ERROR
9242 044556 062716 000002 2$: ADD #2,(SP) ;BUMP RET ADDR TO NEXT ERROR
9243
9244 044562 023737 003374 003414 CMP H.B0,E.B0 ;TEST MSG B0

```

```

9245 044570 001411          BEQ      5$          ;BR IF OK
9246 044572 012737 044604 001176  MOV     #4$, $ESCAPE ;ELSE SETUP ESCAPE
9247 044600 011646          MOV     (SP), -(SP)  ;COPY RET ADDR
9248 044602 000207          RTS      PC          ;& RETURN TO MAINLINE ERROR
9249
9250 044604 032777 001000 134326 4$:  BIT     #SW9, @SWR   ;RETURN HERE FROM MAINLINE ERROR
9251 044612 001070          BNE     20$         ;& BR IF LOOP ON ERROR
9252 044614 062716 000002 5$:  ADD     #2, (SP)    ;BUMP RET ADDR TO NEXT ERROR
9253
9254 044620 023737 003376 003416  CMP     H.A1, E.A1  ;TEST MSG A1
9255 044626 001411          BEQ     8$          ;BR IF OK
9256 044630 012737 044642 001176  MOV     #7$, $ESCAPE
9257 044636 011646          MOV     (SP), -(SP)
9258 044640 000207          RTS      PC
9259
9260 044642 032777 001000 134270 7$:  BIT     #SW9, @SWR
9261 044650 001051          BNE     20$
9262 044652 062716 000002 8$:  ADD     #2, (SP)
9263
9264 044656 023737 003400 003420  CMP     H.B1, E.B1  ;TEST MSG B1
9265 044664 001411          BEQ     11$         ;BR IF OK
9266 044666 012737 044700 001176  MOV     #10$, $ESCAPE
9267 044674 011646          MOV     (SP), -(SP)
9268 044676 000207          RTS      PC
9269
9270 044700 032777 001000 134232 10$: BIT     #SW9, @SWR
9271 044706 001032          BNE     20$
9272 044710 062716 000002 11$: ADD     #2, (SP)
9273
9274 044714 032737 000001 001466 12$: BIT     #T.A2, CHKFLG ;TEST MSG A2?
9275 044722 001402          BEQ     13$         ;BR IF NO
9276 044724 004737 046064          JSR     PC, RCYLD   ;PUT INFO IN CYLDIF, DO NOT CHECK
9277
9278 044730 032737 000002 001466 13$: BIT     #T.B2, CHKFLG ;TEST MSG B2?
9279 044736 001402          BEQ     14$         ;BR IF NO
9280 044740 004737 046136          JSR     PC, RCYLA  ;PUT INFO IN CYLADD, DO NOT CHECK
9281
9282 044744 032737 000004 001466 14$: BIT     #T.B3, CHKFLG ;TEST MSG B3?
9283 044752 001404          BEQ     15$
9284 044754 004737 046174          JSR     PC, RSEC   ;PUT INFO IN SECTOR, DO NOT CHECK
9285 044760 004737 046232          JSR     PC, RHEAD  ;PUT INFO IN HEADA, DO NOT CHECK
9286
9287 044764 013737 001172 001176 15$: MOV     $TMP5, $ESCAPE ;RESTORE ESCAPE
9288 044772 000207          RTS      PC
9289
9290 044774 012706 001100 20$: MOV     #STACK, SP  ;RESET STACK PTR
9291 045000 013737 001172 001176  MOV     $TMP5, $ESCAPE ;RESTORE ESCAPE
9292 045006 000177 134076          JMP     @SLPERR
9293
9294
9295
9296
9297
9298 045012 005737 001462          CKCERR: TST      BYPCERR
9299 045016 001025          BNE     4$
9300 045020 032737 100000 003322  BIT     #CERR, HCS1

```

```

; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
; IE: IF NED, CTO OR MDS SET, MSG A&B ARE INVALID

```

```

9301 045026 001001      BNE      1$          ;BR IF CERR
9302 045030 000207      RTS      PC
9303
9304 045032 032737 004000 003322 1$:  BIT      #CTO,HCS1
9305 045040 001402          BEQ      2$          ;BR IF NOT CTO
9306 045042 104326          ERROR   326         ;CTO ERROR, MSG A&B INVALID
9307 045044 000207      RTS      PC
9308
9309 045046 032737 010000 003324 2$:  BIT      #NED,HCS2
9310 045054 001401          BEQ      3$          ;BR IF NOT NED
9311 045056 104327          ERROR   327         ;NED ERROR, MSG A&B INVALID
9312
9313 045060 032737 001000 003324 3$:  BIT      #MDS,HCS2
9314 045066 001401          BEQ      4$          ;MDS ERROR, MSG A&B INVALID
9315 045070 104330          ERROR   330
9316
9317 045072 000207      4$:  RTS      PC
9318
9319
9320
9321      ; THIS ROUTINE DOES THE SELECT DRIVE CMD TO GET STATUS
9322      ; IT THEN WAITS FOR CONTROLLER READY.
9323      ; IF RDY NOT RECEIVED BY A TIMEOUT, AN ERROR IS FLAGGED
9324 045074 013746 003360      GSTAT:  MOV      TEMP1,-(SP)      ;SAVE TEMP1
9325 045100 013765 001222 000010  MOV      $UNIT,RKCS2(R5)      ;CURRENT DRIVE #
9326 045106 012737 000001 003322  MOV      #SELDRV,HCS1
9327 045114 004737 043434      JSR      PC,DOCMD           ;DR SELDRV (STATUS) CMD & GET CONTR RDY
9328 045120 104117          ERROR   117           ;RDY NOT SET BY END OF SELECT DRIVE CMD
9329 045122 012637 003360      MOV      (SP)+,TEMP1        ;RESTOR TEMP1
9330 045126 000207      RTS      PC
9331
9332
9333      ; THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
9334      ; & ALL CONTROLLER REGISTERS
9335
9336 045130 013746 003360      GSTAT1: MOV      TEMP1,-(SP)      ;SAVE TEMP 1
9337 045134 004737 043664      JSR      PC,HOLD           ;GET ALL CONTR REGS
9338 045140 012765 100000 000000  MOV      #CCLR,RKCS1(R5)      ;CLEAR CONTR
9339 045146 013765 001222 000010  MOV      $UNIT,RKCS2(R5)      ;CURRENT DRIVE #
9340 045154 012765 000003 000026  MOV      #3,RKMR1(R5)        ;SELECT WORD 3
9341 045162 004737 045360      JSR      PC,GSTAT2
9342 045166 104117          ERROR   117           ;RDY NOT SET BY END OF SELECT DRV CMD
9343 045170 013737 003350 003406  MOV      HMR2,H.A3          ;STORE MSG A3
9344 045176 013737 003352 003410  MOV      HMR3,H.B3          ;STORE MSG B3
9345
9346 045204 012765 100000 000000  MOV      #CCLR,RKCS1(R5)
9347 045212 013765 001222 000010  MOV      $UNIT,RKCS2(R5)
9348 045220 012765 000002 000026  MOV      #2,RKMR1(R5)        ;SELECT WORD 2
9349 045226 004737 045360      JSR      PC,GSTAT2
9350 045232 104117          ERROR   117           ;RDY NOT SET BY END OF SELECT DRV CMD
9351 045234 013737 003350 003402  MOV      HMR2,H.A2          ;STORE MSG A2
9352 045242 013737 003352 003404  MOV      HMR3,H.B2          ;STORE MSG B2
9353
9354 045250 012765 100000 000000  MOV      #CCLR,RKCS1(R5)
9355 045256 013765 001222 000010  MOV      $UNIT,RKCS2(R5)
9356 045264 012765 000001 000026  MOV      #1,RKMR1(R5)        ;SELECT WORD 1

```

```

9357 045272 004737 045360 JSR PC,GSTAT2
9358 045276 104117 ERROR 117 ;RDY NOT SET BY END OF SELECT DRV CMD
9359 045300 013737 003350 003376 MOV HMR2,H.A1 ;STORE MSG A1
9360 045306 013737 003352 003400 MOV HMR3,H.B1 ;STORE MSG B1
9361
9362 045314 012765 100000 000000 MOV #CCLR,RKCS1(R5)
9363 045322 013765 001222 000010 MOV $UNIT,RKCS2(R5)
9364 045330 004737 045360 JSR PC,GSTAT2
9365 045334 104117 ERROR 117 ;RDY NOT SET BY END OF SEL DRV CMD
9366 045336 013737 003350 003372 MOV HMR2,H.A0 ;STORE MSG A0
9367 045344 013737 003352 003374 MOV HMR3,H.B0 ;STORE MSG B0
9368
9369 045352 012637 003360 MOV (SP)+,TEMP1 ;RESTORE TEMP1
9370 045356 000207 RTS PC
9371
9372
9373 045360 012737 000001 003322 GSTAT2: MOV #SELDRV,HCS1
9374 045366 053737 001170 003322 BIS $TMP4,HCS1 ;ADD CDT IF RK07
9375 045374 013765 003322 000000 MOV HCS1,RKCS1(R5) ;GET STATUS
9376 045402 013737 001414 003360 MOV T10,TEMP1
9377 045410 004737 043576 JSR PC,FRDY1 ;FIND CONTR RDY & STORE DRIVE REGS ONLY
9378 045414 000207 RTS PC ;RET HERE IF NOT RDY
9379 045416 062716 000002 ADD #2,(SP) ;RET HERE IF OK
9380 045422 000207 RTS PC
9381
9382 ; THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
9383 ; IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
9384 ; THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
9385 ; RETURN IF CERR SET
9386 ; RETURN +2 IF CERR CLEAR
9387
9388 045424 012765 000040 000010 SUBCLR: MOV #SCLR,RKCS2(R5) ;SUBSYS CLEAR
9389 045432 013737 001414 003360 MOV T10,TEMP1
9390 045440 004737 043530 JSR PC,FRDY ;FIND RDY
9391 045444 104120 ERROR 120 ;RDY NOT SET BY END OF SCLR
9392 045446 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;CURRENT DRIVE #
9393 045454 005065 000026 CLR RKMR1(R5) ;SELECT WORD 0
9394 045460 004737 045074 JSR PC,GSTAT ;GET STATUS
9395 045464 032737 100000 003322 BIT #CERR,HCS1 ;CHECK FOR CONT ERROR
9396 045472 001401 BEQ 1$
9397 045474 000207 RTS PC
9398 045476 062716 000002 1$: ADD #2,(SP) ;SKIP OVER ERROR
9399 045502 000207 RTS PC
9400
9401
9402 ; READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
9403
9404 045504 012765 000003 000026 RDSEC: MOV #3,RKMR1(R5) ;WORD 3
9405 045512 004737 045074 JSR PC,GSTAT
9406 045516 013737 003352 001406 MOV HMR3,SECTOR
9407 045524 042737 177017 001406 BIC #C<M.SECT>,SECTOR
9408 045532 006237 001406 ASR SECTOR ;RIGHT JUSTIFY
9409 045536 006237 001406 ASR SECTOR ;SECTOR
9410 045542 006237 001406 ASR SECTOR ;INFO
9411 045546 006237 001406 ASR SECTOR
9412 045552 000207 RTS PC

```

```

9413
9414
9415
9416
9417
9418
9419 045554 013746 003360
9420 045560 013737 001424 003360
9421 045566 004737 045504
9422 045572 005737 001406
9423 045576 001005
9424 045600 004737 045504
9425 045604 005737 001406
9426 045610 001406
9427 045612 005337 003360
9428 045616 001363
9429 045620 012637 003360
9430 045624 000207
9431 045626 012637 003360
9432 045632 062716 000002
9433 045636 000207
9434
9435
9436
9437
9438
9439
9440 045640 013746 003360
9441 045644 013737 001420 003360
9442 045652 004737 045504
9443 045656 023737 001402 001406
9444 045664 001406
9445 045666 004737 045504
9446 045672 023737 001402 001406
9447 045700 001006
9448 045702 005337 003360
9449 045706 001361
9450 045710 012637 003360
9451 045714 000207
9452 045716 012637 003360
9453 045722 062716 000002
9454 045726 000207
9455
9456
9457
9458 045730 012765 000002 000026
9459 045736 004737 045074
9460 045742 013737 003350 001362
9461 045750 043737 012066 001362
9462 045756 006237 001362
9463 045762 006237 001362
9464 045766 006237 001362
9465 045772 006237 001362
9466 045776 023737 001362 012064
9467 046004 001002
9468 046006 005037 001362

```

```

;FIND SECTOR 0 IN 22 SECTOR FORMAT.
;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
FS022: MOV     TEMP1, -(SP)      ;SAVE TEMP1
        MOV     T5000, TEMP1    ;SETUP TIMEOUT
1$:     JSR     PC, RDSEC        ;READ SECTOR
        TST     SECTOR          ;LOOK FOR SECTOR 0
        BNE     2$
        JSR     PC, RDSEC
        TST     SECTOR
        BEQ     3$              ;BR IF SAME TWICE
2$:     DEC     TEMP1
        BNE     1$              ;TRY AGAIN IF TIMEOUT NOT UP
        MOV     (SP)+, TEMP1     ;ELSE RESTORE TEMP1
        RTS     PC              ;EXIT
3$:     MOV     (SP)+, TEMP1
        ADD     #2, (SP)        ;SKIP OVER ERROR
        RTS     PC

;FIND NEXT SECTOR IN 22 SECTOR FORMAT
;ERROR FLAGGED IF NOT FOUND BY TIMEOUT
FNS22: MOV     TEMP1, -(SP)      ;SAVE TEMP 1
        MOV     T500, TEMP1     ;SETUP TIMEOUT
1$:     JSR     PC, RDSEC        ;READ SECTOR
        CMP     PSEC, SECTOR
        BEQ     3$              ;BR IF SAME
        JSR     PC, RDSEC        ;ELSE TRY READ DIFFERENT TWICE
        CMP     PSEC, SECTOR
        BNE     2$              ;BR IF DIFFERENT TWICE
3$:     DEC     TEMP1
        BNE     1$              ;ELSE TRY AGAIN IF TIME LEFT
        MOV     (SP)+, TEMP1     ;RESTORE TEMP 1
        RTS     PC
2$:     MOV     (SP)+, TEMP1     ;RESTORE TEMP 1
        ADD     #2, (SP)        ;SKIP OVER ERROR
        RTS     PC

;READ THE CYL DIFF/OFFSET IN RKM2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
RDCYLD: MOV     #2, RKM1(R5)     ;WORD 2
        JSR     PC, GSTAT
        MOV     HMR2, CYLDIF
        BIC     MASK1, CYLDIF
        ASR     CYLDIF          ;RIGHT JUSTIFY
        ASR     CYLDIF          ;CYL DIFF/OFFSET
        ASR     CYLDIF          ;INFO
        CMP     CYLDIF, MASK     ;CHK TO SEE IF RET IN COMPL. FORM
        BNE     1$              ;BR IF NOT
        CLR     CYLDIF          ;CLR IF YES

```

```

9469 046012 000207      1S:      RTS      PC
9470
9471      ; READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
9472
9473 046014 012765 000002 000026 RDCYLA: MOV      #2,RKMR1(R5)      ;WORD 2
9474 046022 004737 045074      JSR      PC,GSTAT
9475 046026 013737 003352 001364      MOV      HMR3,CYLADD
9476 046034 043737 012066 001364      BIC      MASK1,CYLADD
9477 046042 006237 001364      ASR      CYLADD      ;RIGHT JUSTIFY
9478 046046 006237 001364      ASR      CYLADD      ;CYL ADDR
9479 046052 006237 001364      ASR      CYLADD      ;INFO
9480 046056 006237 001364      ASR      CYLADD
9481 046062 000207      RTS      PC
9482
9483      ; READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
9484
9485 046064 013737 003402 001362 RCYLD: MOV      H.A2,CYLDIF
9486 046072 043737 012066 001362      BIC      MASK1,CYLDIF
9487 046100 006237 001362      ASR      CYLDIF      ;RIGHT JUSTIFY
9488 046104 006237 001362      ASR      CYLDIF
9489 046110 006237 001362      ASR      CYLDIF
9490 046114 006237 001362      ASR      CYLDIF
9491 046120 023737 001362 012064      CMP      CYLDIF,MASK      ;CHK TO SEE IF RET IN COMPL. FORM
9492 046126 001002      BNE      1S      ;BR IF NO
9493 046130 005037 001362      CLR      CYLDIF      ;ELSE CLEAR
9494 046134 000207      1S:      RTS      PC
9495
9496      ; READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
9497
9498 046136 013737 003404 001364 RCYLA: MOV      H.B2,CYLADD
9499 046144 043737 012066 001364      BIC      MASK1,CYLADD
9500 046152 006237 001364      ASR      CYLADD      ;RIGHT JUSTIFY
9501 046156 006237 001364      ASR      CYLADD
9502 046162 006237 001364      ASR      CYLADD
9503 046166 006237 001364      ASR      CYLADD
9504 046172 000207      RTS      PC
9505
9506      ; READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
9507
9508 046174 013737 003410 001406 RSEC: MOV      H.B3,SECTOR
9509 046202 042737 177017 001406      BIC      #1C<M.SECT>,SECTOR      ;CLEAR UNWANTED INFO
9510 046210 006237 001406      ASR      SECTOR      ;RIGHT JUSTIFY
9511 046214 006237 001406      ASR      SECTOR
9512 046220 006237 001406      ASR      SECTOR
9513 046224 006237 001406      ASR      SECTOR
9514 046230 000207      RTS      PC
9515
9516      ; READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
9517
9518 046232 013737 003410 001432 RHEAD: MOV      H.B3,HEADA
9519 046240 042737 170777 001432      BIC      #1C<M.HEAD>,HEADA      ;CLEAR UNWANTED INFO
9520 046246 006237 001432      ASR      HEADA      ;RIGHT JUSTIFY IT
9521 046252 000337 001432      SWAB     HEADA
9522 046256 000207      RTS      PC
9523
9524      ;FIND LIMIT DETECT ON SEEK IN RKMR3 BEFORE TIMEOUT

```

```

9525      ;RETURN IF NOT FOUND: ERROR
9526      ;RETURN+2 IF FOUND: SKIP OVER ERROR
9527
9528 046260 005037 001460      FLIM: CLR LIMERR      ;LIMIT DETECT ERROR FLAG
9529 046264 012737 177777 003360 MOV #-1,TEMP1      ;SETUP TIMEOUT
9530 046272 012765 000001 000026 MOV #1,RKMR1(R5)   ;WORD 1
9531 046300 004737 045074      1$: JSR PC,GSTAT
9532 046304 032737 020000 003352 BIT #D.LIMD,HMR3
9533 046312 001006      BNE 2$      ;EXIT IF SET
9534 046314 005337 003360      DEC TEMP1
9535 046320 001367      BNE 1$
9536 046322 005237 001460      INC LIMERR      ;SET LIMIT DETECT FLAG
9537 046326 000207      RTS PC
9538 046330 062716 000002      2$: ADD #2,(SP)      ;SKIP OVER ERROR
9539 046334 000207      RTS PC
9540
9541      ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE TIMEOUT
9542      ;ENTER WITH TIME IN SECONDS IN TEMP2
9543      ;RETURN IF NOT FOUND
9544      ;RETURN+2 IF FOUND - SKIP OVER ERROR
9545
9546 046336 012737 177777 003360 FHDHM: MOV #-1,TEMP1      ;ALL 1'S
9547 046344 012765 000001 000026 MOV #1,RKMR1(R5)   ;WORD 1
9548 046352 004737 045074      1$: JSR PC,GSTAT
9549 046356 032737 000040 003350 BIT #D.HDHM,HMR2
9550 046364 001007      BNE 2$
9551 046366 005337 003360      DEC TEMP1
9552 046372 001367      BNE 1$
9553 046374 005337 003362      DEC TEMP2
9554 046400 001356      BNE FHDHM
9555 046402 000207      RTS PC
9556 046404 062716 000002      2$: ADD #2,(SP)      ;SKIP OVER ERROR
9557 046410 000207      RTS PC
9558
9559      ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE TIMEOUT
9560      ;RETURN IF NOT FOUND
9561      ;RETURN+2 IF FOUND: SKIP OVER ERROR
9562
9563 046412 012737 177777 003360 FLOAD: MOV #-1,TEMP1      ;SETUP TIMEOUT
9564 046420 012765 000001 000026 MOV #1,RKMR1(R5)   ;WORD 1
9565 046426 004737 045074      1$: JSR PC,GSTAT
9566 046432 032737 010000 003350 BIT #D.LOAD,HMR2
9567 046440 001004      BNE 2$
9568 046442 005337 003360      DEC TEMP1
9569 046446 001367      BNE 1$
9570 046450 000207      RTS PC
9571 046452 062716 000002      2$: ADD #2,(SP)      ;SKIP OVER ERROR
9572 046456 000207      RTS PC
9573
9574      ;ROUTINE TO FIND SPOK BEFORE TIMEOUT
9575      ;ENTER WITH APPROX TIME IN TEMP2
9576      ;RETURN IF NOT CLEARED
9577      ;RETURN +2 IF CLEARED TO SKIP OVER ERROR
9578
9579 046460 012737 177777 003360 FSPOK: MOV #-1,TEMP1      ;ALL 1'S
9580 046466 012765 000001 000026 MOV #1,RKMR1(R5)   ;WORD 1

```

```

9581 046474 004737 045074
9582 046500 032737 001000 003350 1$: JSR PC,GSTAT
9583 046506 001407 BIT #D.SPOK,HMR2 ;SEE IF SPOK CLEARED
9584 046510 005337 003360 BEQ 2$
9585 046514 001367 DEC TEMP1
9586 046516 005337 003362 BNE 1$
9587 046522 001356 DEC TEMP2
9588 046524 000207 BNE FSPCK
9589 046526 062716 000002 RTS PC
9590 046532 000207 2$: ADD #2,(SP) ;SKIP OVER ERROR
9591
9592 ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
9593 ;ENTER WITH CYL # IN 'CALADD'
9594 ;ENTER WITH HEAD # IN 'HEAD'
9595 ;ENTER WITH FORMAT IN 'FORMAT'
9596
9597 046534 010046 FHDTAB: MOV RO,-(SP) ;SAV RO
9598 046536 010146 MOV R1,-(SP) ;SAV R1
9599 046540 012700 001470 MOV #HDTAB,RO ;HEADER WORD TABLE ADDR
9600 046544 005001 CLR R1 ;SECTOR COUNTER
9601 046546 013737 001430 001434 MOV HEAD,HD1
9602 046554 006337 001434 ASL HD1
9603 046560 006337 001434 ASL HD1
9604 046564 006337 001434 ASL HD1
9605 046570 006337 001434 ASL HD1
9606 046574 006337 001434 ASL HD1 ;SETUP HEAD # FOR WORD 2 OF HEADER
9607 046600 013737 001436 001440 MOV FORMAT,FMT1
9608 046606 000337 001440 SWAB FMT1
9609 046612 006337 001440 ASL FMT1 ;SETUP FORMAT FOR WORD 2 OF HEADER
9610
9611 046616 013720 001366 1$: MOV CALADD,(RO)+ ;HEADER WORD 1-CYL ADDR
9612 046622 010110 MOV R1,(RO) ;HEADER WORD 2-SECTOR NO
9613 046624 053710 001434 BIS HD1,(RO) ;
9614 046630 053710 001440 BIS FMT1,(RO) ;
9615 046634 005737 001464 TST BYPFMT ;
9616 046640 001403 BEQ 2$ ;BR IF TRUE FORMAT
9617 046642 052710 140000 BIS #<BIT14!BIT15>,(RO) ;SET GOOD SECTOR FLAGS
9618 046646 000402 BR 3$
9619 046650 004737 046730 2$: JSR PC,SECFLG ;GET SECTOR FLAGS
9620
9621 046654 013737 001366 003360 3$: MOV CALADD,TEMP1
9622 046662 011037 003362 MOV (RO),TEMP2
9623 046666 043737 001366 003362 BIC CALADD,TEMP2
9624 046674 042037 003360 BIC (RO)+,TEMP1
9625 046700 053737 003360 003362 BIS TEMP1,TEMP2
9626 046706 013720 003362 MOV TEMP2,(RO)+ ;HEADER WORD 3-HEADER CHECK
9627
9628 046712 005201 INC R1 ;SECTOR CTR
9629 046714 020127 000026 CMP R1,#22. ;ALL 22 SECTORS DONE? (66 WORDS)
9630 046720 001336 BNE 1$ ;BR IF NO
9631
9632 046722 012601 MOV (SP)+,R1 ;RESTOR R1
9633 046724 012600 MOV (SP)+,RO ;RESTOR RO
9634 046726 000207 RTS PC
9635
9636 ;

```



```

9637
9638
9639
9640 046730 010246
9641 046732 005737 001436
9642 046736 001016
9643 046740 012702 002314
9644 046744 004737 047000
9645 046750 052710 100000
9646
9647 046754 012702 054362
9648 046760 004737 047000
9649 046764 052710 040000
9650
9651 046770 012602
9652 046772 000207
9653
9654
9655 046774 012602
9656 046776 000207
9657
9658
9659
9660
9661
9662
9663
9664 047000 010346
9665
9666 047002 021227 177777
9667 047006 001002
9668 047010 012603
9669 047012 000207
9670
9671 047014 022237 001366
9672 047020 001403
9673 047022 062702 000002
9674 047026 000765
9675
9676 047030 013703 001430
9677 047034 000303
9678 047036 050103
9679 047040 022203
9680 047042 001401
9681 047044 000756
9682
9683 047046 012603
9684 047050 062716 000004
9685 047054 000207
9686
9687
9688
9689
9690 047056 010046
9691 047060 010146
9692 047062 004737 045504
    
```

```

; THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS TEST
; & SETS BITS 14 & 15 APPROPRIATELY.
    
```

```

SECFLG: MOV R2, -(SP) ;SAVE R2
        TST FORMAT
        BNE 1$ ;BR IF 20 SECTOR FORMAT
        MOV #BSE22H+8., R2
        JSR PC, FLGTST ;GET HARDWARE DETECTED FLAG
        BIS #BIT15, (R0) ;RETURN HERE IF GOOD SECTOR
        MOV #BSE22S+8., R2 ;ELSE RETURN HERE
        JSR PC, FLGTST ;GET SOFTWARE DETECTED FLAG
        BIS #BIT14, (R0) ;RETURN HERE IF GOOD SECTOR
        MOV (SP)+, R2 ;ELSE RETURN HERE
        RTS PC
    
```

```

1$: MOV (SP)+, R2 ;RESTORE R2
    RTS PC
    
```

```

; THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
; ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC) IN TEMP1
; RETURN IF NO COMPARE
; RETURN +4 IF COMPARE
    
```

```

FLGTST: MOV R3, -(SP) ;SAVE R3
1$: CMP (R2), #-1 ;SEE IF ALL 1'S
    BNE 2$ ;BR IF NO
    MOV (SP)+, R3 ;RESTORE R3
    RTS PC
2$: CMP (R2)+, CALADD ;SEE IF = CYL #, & ADV PTR TO TRACK/SECTOR WORD
    BEQ 3$
    ADD #2, R2 ;GO TO NEXT CYL WORD IN TABLE
    BR 1$
3$: MOV HEAD, R3 ;GET HEAD # FROM FHDTAB ROUTINE
    SWAB R3
    BIS R1, R3 ;ADD SECTOR # FROM FHDTAB ROUTINE
    CMP (R2)+, R3 ;SECTOR/HEAD COMPARE? & INCR TO NEXT CYL WORD
    BEQ 4$ ;BR IF YES
    BR 1$ ;TRY NEXT CYL
4$: MOV (SP)+, R3 ;RESTORE R3
    ADD #4, (SP) ;INCREMENT RET ADDR
    RTS PC
    
```

```

; THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
; WITH AND RE-WRITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
    
```

```

SORT: MOV R0, -(SP) ;SAVE R0
      MOV R1, -(SP) ;SAVE R1
      JSR PC, RDSEC
    
```

```

9693 047066 062737 000001 001406      ADD      #1,SECTOR
9694 047074 004737 047164          JSR      PC,MULT6      ;MULT SECTOR BY 6
9695
9696 047100 012700 000204          MOV      #132,RO
9697 047104 163700 001406          SUB      SECTOR,RO      ;RO-SECTOR TO RO = INDEX
9698 047110 010037 001406          MOV      RO,SECTOR
9699 047114 062737 001674 001406      ADD      #RHTAB,SECTOR ;SAVE INDEX
9700
9701 047122 062700 001674          ADD      #RHTAB,RO      ;INDEX TO BOT HALF OF RHTAB
9702 047126 012701 002100          MOV      #SRTTAB,R1    ;INDEX TO TOP HALF OF SRTTAB
9703
9704 047132 012021          1$:     MOV      (RO)+,(R1)+ ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
9705 047134 020027 002100          CMP      RO,#RHTAB+132.
9706 047140 001374          BNE      1$
9707
9708 047142 012700 001674          2$:     MOV      #RHTAB,RO      ;PUT TOP OF RHTAB TO BOT OF SRTTAB
9709 047146 012021          MOV      (RO)+,(R1)+
9710 047150 020037 001406          CMP      RO,SECTOR
9711 047154 001374          BNE      2$
9712
9713 047156 012601          MOV      (SP)+,R1      ;RESTOR R1
9714 047160 012600          MOV      (SP)+,RO      ;RESTOR RO
9715 047162 000207          RTS      PC
9716
9717
9718          ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
9719
9720 047164 006337 001406          MULT6:  ASL      SECTOR      ;2 X SECTOR
9721 047170 013746 001406          MOV      SECTOR,-(SP)
9722 047174 006337 001406          ASL      SECTOR      ;4 X SECTOR
9723 047200 062637 001406          ADD      (SP)+,SECTOR ;(4 X S)+(2 X S) = 6 X SECTOR
9724 047204 000207          RTS      PC
9725
9726
9727          ;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
9728
9729 047206 005037 001376          CLKON:  CLR      TIMUP
9730 047212 005737 003466          TST      PCLKF
9731 047216 001004          BNE      1$           ;BRANCH IF P-CLOCK PRESENT
9732 047220 012777 000100 132100      MOV      #100,ALKS    ;L-CLOCK, ENABLE INT
9733 047226 000207          RTS      PC
9734 047230 012777 177777 132064 1$:     MOV      #-1,APKSB    ;P-CLOCK, ALL 1'S

```

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 185
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0185

9735	047236	012777	000135	132054	MOV	#135,APKS	;ENABLE INT, CT UP, REP INT
9736	047244	000207			RTS	PC	;LINE FREQ & RUN
9737							
9738							
9739							
9740	047246	005037	001376				
9741	047252	005337	001372				
9742	047256	001010					
9743	047260	013737	001370	001372	MOV	HZ,COUNT	

;KW11-L & KW11-P INTERRUPT HANDLER
 CLOCK: CLR TIMUP
 DEC COUNT
 BNE 1\$
 MOV HZ,COUNT

9744	047266	005337	001374
9745	047272	001002	
9746	047274	005237	001376
9747	047300	000002	
9748			
9749			
9750			
9751	047302	005737	003466
9752	047306	001003	
9753	047310	005077	132012
9754	047314	000207	
9755	047316	005077	131776
9756	047322	000207	

```

DEC      SEC
BNE      IS
INC      TIMUP      ;SORRY, TIME IS UP
RTI

;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
CLKOF:  TST      PCLKF
        BNE      IS      ;BRACH IF P-CLOCK PRESENT
        CLR      @LKS    ;L-CLOCK, CLEAR INTERRUPT
        RTS      PC
IS:     CLR      @PKS    ;P-CLOCK, CLEAR INTERRUPT
        RTS      PC

```

```

9757
9758
9759
9760
9761
9762
9763
9764
9765
9766
9767
9768 047324 010046
9769 047326 010146
9770 047330 012700 000021
9771 047334 005001
9772 047336 000241
9773
9774 047340 006137 003360
9775 047344 103001
9776 047346 005201
9777 047350 005300
9778 047352 001372
9779
9780 047354 032701 000001
9781 047360 001003
9782 047362 052737 100000 003360
9783 047370 012601
9784 047372 012600
9785 047374 000207
9786
9787
9788
9789
9790
9791
9792 047376 032777 001000 131534
9793 047404 001406
9794 047406 105737 001103
9795 047412 001403
9796 047414 013716 001110
9797 047420 000002
9798
9799 047422 011637 001110
9800 047426 000002
9801
9802
9803
9804
9805
9806
9807
9808
9809
9810
9811 047430 005037 001176
9812 047434 005037 001410

```

; THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MSGS
 ; ENTER WITH THE EXPECTED WORD IN TEMP1
 ; TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
 ; R1 IS INCREMENTED. AT THE END OF 17 ROTATES (TEMP1 BACK TO ORIG),
 ; R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
 ; THE PARITY BIT IS NOT SET IN B
 ; IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
 ; SET IN TEMP1

```

SBPAR:  MOV    RO,-(SP)      ;SAVE RO
        MOV    R1,-(SP)    ;SAVE R1
        MOV    #17.,RO     ;SHIFT COUNTER
        CLR    R1          ;COUNT # OF 1'S IN TEMP1
        CLC                    ;CLEAR CARRY
1$:     ROL    TEMP1
        BCC    2$         ;BR IF CARRY CLEAR
        INC    R1         ;COUNT # OF 1'S
        DEC    RO         ;SHIFT COUNTER
        BNE    1$
3$:     BIT    #BIT0,R1    ;BR IF ODD # IN RO
        BNE    3$
        BIS    #M.PAR,TEMP1 ;SET PARITY BIT
        MOV    (SP)+,R1    ;RESTORE R1
        MOV    (SP)+,RO    ;RESTORE RO
        RTS    PC

```

; ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
 ; WHEN \$LPERR SET BY OTHER THAN SCOPE ROUTINE
 ; IE: MY LOOP MACRO

```

SCOPE$: BIT    #SW9,$SWR    ;LOOP ON ERROR?
        BEQ    1$         ;BR IF NO
        TSTB   $ERFLG     ;HAD ERROR?
        BEQ    1$         ;BR IF NO
        MOV    $LPERR,(SP)
        RTI
1$:     MOV    (SP),$LPERR  ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
        RTI

```

; CHECK FOR SW14 (LOOP ON TEST) OR SW8 (LOOP ON SPECIFIC TEST)
 ; RETURN IF NEITHER SET
 ; RETURN +2 IF EITHER SET

; THIS SUBROUTINE IS USED AT THE END OF ANY TEST THAT REQUIRES
 ; RECONDITIONING OF THE DRIVE BEFORE LOOPING ON AN ERROR OR TEST

```

SWTST:  CLR    $ESCAPE
        CLR    LPFLG

```

```

9813 047440 032777 040000 131472      BIT      #SW14,@SWR      ;LOOP ON TEST?
9814 047446 001403                BEQ      3$          ;BR IF NO
9815 047450 062716 000002      1$:      ADD      #2,(SP)
9816 047454 000207      2$:      RTS      PC
9817
9818 047456 032777 000400 131454 3$:      BIT      #SW8,@SWR      ;LOOP ON SPECIFIC TEST?
9819 047464 001773                BEQ      2$          ;BR IF NO
9820 047466 127737 131446 001102      CMPB    @SWR,$STSTM  ;RIGHT TEST? SWR <7:0>
9821 047474 001765                BEQ      1$          ;BR IF YES
9822 047476 000207      RTS      PC
9823
9824
9825
9826
9827
9828
9829
9830 047500 022626      STOP:    CMP      (SP)+,(SP)+ ;RESTORE STACK FROM INTERRUPT
9831
9832 047502 004737 045424      JSR      PC,SUBCLR
9833 047506 104024      ERROR   24          ;CERR AFTER
9834
9835 047510 005737 003304      TST     UNLD
9836 047514 001431                BEQ      3$          ;SEE IF HEADS UNLOADED
9837 047516 005737 000042      TST     42          ;BR IF NO
9838 047522 001403                BEQ      1$          ;SEE IF MANUAL OR AUTO MODE
9839 047524 104401 056640      TYPE    ,MSG74     ;BR IF MANUAL MODE
9840 047530 000402                BR       2$          ;PGM ABORT PENDING
9841 047532 104401 056667      1$:     TYPE    ,MSG75 ;HALT PENDING
9842 047536
9843
9844 047536 004737 045424      JSR      PC,SUBCLR
9845 047542 104024      ERROR   24          ;CERR AFTER SCLR
9846
9847 047544 012737 000011 003322      MOV     #SRTSPL,HCS1
9848 047552 004737 043434      JSR      PC,DOCMD  ;DO START SPINDLE CMD & GET CONTR RDY
9849 047556 104121      ERROR   121        ;RDY NOT FOUND AFTER ST SPIN CMD.
9850
9851 047560 013737 001420 003362      MOV     T500,TEMP2 ;SETUP TIMEOUT
9852 047566 004737 044044      JSR      PC,FATT1  ;FIND ATTN
9853 047572 104067      ERROR   67        ;NO ATTN AFTER ST SPIN CMD.
9854
9855 047574 005037 003304      CLR     UNLD
9856
9857 047600 005737 003306      3$:     TST     BADHDR   ;SEE IF HEADERS VALID
9858 047604 001460                BEQ      4$          ;BR IF YES
9859 047606 005237 003310      INC     HPEND
9860
9861 047612 012765 100000 000000      MOV     #CCLR,RKCS1(R5)
9862 047620 013765 001222 000010      MOV     $UNIT,RKCS2(R5)
9863 047626 012737 000013 003322      MOV     #RECAL,HCS1
9864 047634 004737 043434      JSR      PC,DOCMD  ;DO RECAL CMD & GET CONTR RDY
9865 047640 104124      ERROR   124        ;RDY NOT SET AFTER RECAL CMD
9866
9867 047642 012765 000001 000026      MOV     #1,RKMR1(R5) ;SELECT WORD 1
9868 047650 004737 045074      JSR      PC,GSTAT

```

```

; THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
; IT IS USED TO ALLOW THE OPERATOR TO HALT THE CPU WHILE INSURING
; THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
; THE CPU.

```

```

9869 047654 032737 020000 003350      BIT      #D,RTZ,HMR2
9870 047662 001001                BNE      64$
9871 047664 104244                ERROR   244      ;RTZ NOT SET DURING RECAL CMD
9872 047666 013737 001414 003362 64$:    MOV      T10,TEMP2  ;SETUP TIMEOUT
9873 047674 004737 044044                JSR     PC,FATT1   ;FIND ATTN
9874 047700 104055                ERROR   55        ;NO ATTN AFTER RECAL CMD
9875
9876 047702 012765 100000 000000      MOV     #CCLR,RKCS1(R5)
9877 047710 013765 001222 000010      MOV     $UNIT,RKCS2(R5) ;DRIVE#
9878 047716 012737 000005 003322      MOV     #CLEAR,HCS1
9879 047724 004737 043434                JSR     PC,DOCMD   ;DO DRIVE CLEAR CMD & GET CONTR RDY
9880 047730 104151                ERROR   151      ;NO RDY AFTER DRIVE CLEAR CMD
9881 047732 004737 044012                JSR     PC,TSTATN  ;TEST FOR ATTN
9882 047736 000401                BR      66$
9883 047740 104154                ERROR   154      ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
9884 047742
9885
9886
9887 047742 000137 031342                JMP     FORM       ;WRITE VALID FORMATS
9888
9889 047746 005737 000042                4$:    TST     42      ;SEE IF MANUAL OR AUTO MODE
9890 047752 001410                BEQ     5$         ;BR IF MANUAL MODE
9891 047754 104401 056711                TYPE   MSG76      ;PGM ABORTED
9892 047760 005037 042750                CLR     $EOPCT    ;SET UP EOP TO EXIT TO MONITOR
9893 047764 005037 001176                CLR     $ESCAPE
9894 047770 000137 042722                JMP     $EOP1     ;ABORT PROGRAM
9895
9896 047774 104401 056727                5$:    TYPE   ,MSG77  ;CPU HALTED
9897 050000 000000                HALT
9898 050002 000137 007730                JMP     ST5       ;START OVER IF CONTINUE PRESSED
9899
9900
9901      ;CHECK IF HALT PENDING
9902      ;RET IF YES
9903      ;RET+4 IF NO
9904
9905 050006 005037 003306                HPEN:  CLR     BADHDR ;CLR VALID HALT FLAG
9906 050012 005737 003310                TST     HPEND     ;SEE IF HALT PENDING
9907 050016 001002                BNE     1$        ;BR IF YES
9908 050020 062716 000004                ADD     #4,(SP)   ;ELSE BUMP RET ADDR
9909 050024 000207                1$:    RTS     PC    ;& RET
9910
9911      .SBTTL  UNEXPECTED TIMEOUT HANDLER
9912
9913      ;
9914      ;THIS ROUTINE IS ENTERED IF THERE IS
9915      ;A. NON EXISTANT MEMORY (NO SSYN)
9916      ;B. BOUNDARY ERROR
9917      ;C. STACK OVERFLOW
9918      ;
9919
9920 050026 011600                BADTMO: MOV     (SP),RO ;SAVE PC WHERE TIMEOUT OCCURRED.
9921 050030 005740                TST     -(RO)    ;GET PC BEFORE UPDATE
9922 050032 032777 020000 131100      BIT     #SW13,$SWR ;INHIBIT ERROR TYP0UT?
9923 050040 001005                BNE     1$        ;YES, DON'T TYPE
9924 050042 104401 057105                TYPE   ,EM3      ;ABORT TESTS,UNEXP T.O. @ PC=

```

```

9925 050046 010046          MOV      RO,-(SP)          ;;SAVE RO FOR TYPEOUT
9926                                ;;TYPE PC
9927 050050 104403          TYPOS          ;;GO TYPE--OCTAL ASCII
9928 050052      006          .BYTE      6          ;;TYPE 6 DIGIT(S)
9929 050053      000          .BYTE      0          ;;SUPPRESS LEADING ZEROS
9930 050054 032777 001000 131056 1$:  BIT      #SW9,@SWR      ;;LOOP ON ERROR?
9931 050062 001403          BEQ      2$          ;;NO BRANCH
9932 050064 022626          CMP      (SP)+,(SP)+    ;;YES, RESTORE STACK
9933 050066 000177 131014          JMP      @SLPADR       ;;GO TO STARTING ADDR OF TEST
9934                                ;;THAT GAVE BAD TIMEOUT
9935 050072 032777 040000 131040 2$:  BIT      #SW14,@SWR     ;;LOOP ON TEST?
9936 050100 001401          BEQ      3$          ;;NO BRANCH
9937 050102 000002          RTI                    ;;YES
9938
9939 050104 000000          3$:  HALT                ;;UNEXPECTED TIME OUT OCCURRED
9940                                ;;AS INDICATED. YOU CAN LOOP ON
9941                                ;;ERROR, LOOP ON TEST OR INHIBIT
9942                                ;;ERROR TYPEOUT BY SETTING THOSE
9943                                ;;SWITCHES.
9944
9945 050106 022626          CMP      (SP)+,(SP)+    ;;RESTORE STACK
9946 050110 000137 042722          JMP      $EOP1         ;;ABORT TESTS
9947
9948                                .SBTTL MEMORY CHECK ENABLE TRAP
9949
9950 050114 012737 050130 001176 MEMERR: MOV      #1$,$ESCAPE    ;;LOAD ESCAPE
9951 050122 011637 001334          MOV      (SP),TRAPPC  ;;STORE PC
9952 050126 104236          ERROR    236          ;;UNEXP MEM PARITY TRAP
9953
9954 050130 005037 001176 130776 1$:  CLR      $ESCAPE
9955 050134 032777 001000          BIT      #SW9,@SWR     ;;CHECK IF LOOP ON ERROR
9956 050142 001001          BNE     2$          ;;YES, FORCE STACK AND TRY AGAIN
9957 050144 000002          RTI                    ;;ELSE RETURN
9958
9959 050146 012706 001100 2$:  MOV      #STACK,SP    ;;INIT STACK
9960 050152 000177 130732          JMP      @SLPERR       ;;LOOP ON ERROR
9961
9962                                .SBTTL RK06 INTERRUPT HANDLER
9963
9964 050156 011600          INTER: MOV      (SP),RO  ;;SAVE PC WHERE INT OCCURRED.
9965 050160 005740          TST      -(RO)        ;;GET PC BEFORE UPDATE.
9966 050162 104401 055573          TYPE    MSG6         ;;INT AT PC=
9967 050166 010046          MOV      RO,-(SP)     ;;SAVE RO FOR TYPEOUT
9968                                ;;TYPE PC
9969 050170 104403          TYPOS          ;;GO TYPE--OCTAL ASCII
9970 050172      006          .BYTE      6          ;;TYPE 6 DIGIT(S)
9971 050173      000          .BYTE      0          ;;SUPPRESS LEADING ZEROS
9972 050174 000002          RTI
9973
9974                                .SBTTL POWER DOWN AND UP ROUTINES
9975                                ;POWER DOWN ROUTINE
9976
9977
9978 050176 012737 050210 000024 $PWRDN: MOV      #SPWRUP,PWRVEC ;;SET UP VECTOR
9979 050204 000000          HALT
9980 050206 000776          BR      .-2          ;;HANG UP.

```



```

9981
9982
9983
9984 050210 005037 050262
9985 050214 005237 050262
9986 050220 001375
9987 050222 012737 050176 000024
9988 050230 012737 000340 000026
9989 050236 012737 000340 000036
9990 050244 012706 001100
9991 050250 104401 055761
9992 050254 000005
9993 050256 000137 012170
9994
9995 050262 000000
9996

;POWER UP ROUTINE
$PWRUP: CLR $PWRCT ;WAIT LOOP FOR TTY
1$: INC $PWRCT ;WAIT FOR THE INCR
BNE 1$ ;OF WORD
MOV #PWRDN,PWRVEC ;SET POWER DOWN VECTOR
MOV #PR7,PWRVEC+2 ;PRIORITY 7
MOV #PR7,TRAPVEC+2 ;LOCKOUT ALL INTERRUPTS FOR TRAPS
MOV #STACK,SP ;INITIALIZE STACK
TYPE ,MSG11 ;REPORT POWER FAIL
RESET
JMP PFSRT

$PWRCT: 0 ;WAIT COUNT FOR TTY

```

.SBTTL SCOPE HANDLER ROUTINE

9997
9998
9999
10000
10001
10002
10003
10004
10005
10006
10007
10008
10009
10010
10011
10012
10013
10014
10015
10016
10017
10018
10019
10020
10021
10022
10023
10024
10025
10026
10027
10028
10029
10030
10031
10032
10033
10034
10035
10036
10037
10038
10039
10040
10041
10042
10043
10044
10045
10046
10047
10048
10049
10050
10051
10052

050264
050264 104407
050266 032777 040000 130644
050274 001114

050276 000416

050300 013746 000004
050304 012737 050324 000004
050312 005737 177060
050316 012637 000004
050322 000463
050324 022626
050326 012637 000004
050332 000423
050334
050334 032777 000400 130576
050342 001404
050344 127737 130570 001102
050352 001465
050354 105737 001103
050360 001421
050362 123737 001115 001103
050370 101015
050372 032777 001000 130540
050400 001404
050402 013737 001110 001106
050410 000446
050412 105037 001103
050416 005037 001174
050422 000415
050424 032777 004000 130506
050432 001011
050434 005737 001216
050440 001406
050442 005237 001104
050446 023737 001174 001104
050454 002024
050456 012737 000001 001104
050464 013737 050542 001174
050472 105237 001102
050476 113737 001102 001214

```
*****  
: THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT  
: AND LOAD THE TEST NUMBER($STSTM) INTO THE DISPLAY REG.(DISPLAY<7:0>)  
: AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>  
: THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:  
: *SW14=1 LOOP ON TEST  
: *SW11=1 INHIBIT ITERATIONS  
: *SW09=1 LOOP ON ERROR  
: *SW08=1 LOOP ON TEST IN SWR<7:0>  
: *CALL SCOPE ;;SCOPE=IOT  
: *  
$SCOPE:  
CKSWR  
1$: BIT #BIT14,$SWR ;;TEST FOR CHANGE IN SOFT-SWR  
BNE $OVER ;;LOOP ON PRESENT TEST?  
: YES IF SW14=1  
: *****START OF CODE FOR THE XOR TESTER*****  
$XTSTR: BR 6$  
: IF RUNNING ON THE "XOR" TESTER CHANGE  
: THIS INSTRUCTION TO A "NOP" (NOP=240)  
: SAVE THE CONTENTS OF THE ERROR VECTOR  
: SET FOR TIMEOUT  
: TIME OUT ON XOR?  
: RESTORE THE ERROR VECTOR  
: GO TO THE NEXT TEST  
5$: CMP (SP)+,(SP)+ CLEAR THE STACK AFTER A TIME OUT  
MOV (SP)+,$ERRVEC RESTORE THE ERROR VECTOR  
BR 7$ LOOP ON THE PRESENT TEST  
6$: *****END OF CODE FOR THE XOR TESTER*****  
BIT #BIT08,$SWR LOOP ON SPEC. TEST?  
BEQ 2$ BR IF NO  
CMPB $SWR,$STSTM ON THE RIGHT TEST? SWR<7:0>  
BEQ $OVER BR IF YES  
2$: TSTB $ERFLG HAS AN ERROR OCCURRED?  
BEQ 3$ BR IF NO  
CMPB $ERMAX,$ERFLG MAX. ERRORS FOR THIS TEST OCCURRED?  
BHI 3$ BR IF NO  
BIT #BIT09,$SWR LOOP ON ERROR?  
BEQ 4$ BR IF NO  
7$: MOV $LPERR,$LPADR SET LOOP ADDRESS TO LAST SCOPE  
BR $OVER  
4$: CLRB $ERFLG ZERO THE ERROR FLAG  
CLR $TIMES CLEAR THE NUMBER OF ITERATIONS TO MAKE  
BR 1$ ESCAPE TO THE NEXT TEST  
3$: BIT #BIT11,$SWR INHIBIT ITERATIONS?  
BNE 1$ BR IF YES  
TST $PASS IF FIRST PASS OF PROGRAM  
BEQ 1$ INHIBIT ITERATIONS  
INC $ICNT INCREMENT ITERATION COUNT  
CMP $TIMES,$ICNT CHECK THE NUMBER OF ITERATIONS MADE  
BGE $OVER BR IF MORE ITERATION REQUIRED  
1$: MOV #1,$ICNT REINITIALIZE THE ITERATION COUNTER  
MOV $MXCNT,$TIMES SET NUMBER OF ITERATIONS TO DO  
$SVLAD: INCB $STSTM COUNT TEST NUMBERS  
MOVB $STSTM,$TESTN SET TEST NUMBER IN APT MAILBOX
```

```

10053 050504 011637 001106      MOV      (SP), $LPADR      ;; SAVE SCOPE LOOP ADDRESS
10054 050510 011637 001110      MOV      (SP), $LPERR     ;; SAVE ERROR LOOP ADDRESS
10055 050514 005037 001176      CLR      $ESCAPE         ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
10056 050520 112737 000001 001115  #1, $ERMAX                ONLY ALLOW ONE(1) ERROR ON NEXT TEST
10057 050526 013777 001102 130406 $OVER:  MOV      $TSTNM, @DISPLAY  ;; DISPLAY TEST NUMBER
10058 050534 013716 001106      MOV      $LPADR, (SP)    ;; FUDGE RETURN ADDRESS
10059 050540 000002                RTI                      ;; FIXES PS
10060 050542 003720      $MXCNT: 2000.           ;; MAX. NUMBER OF ITERATIONS
10061      .SBTTL  ERROR HANDLER ROUTINE
10062
10063      ;*****
10064      ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
10065      ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
10066      ;*AND GO TO TYPERR ON ERROR
10067      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10068      ;*SW15=1      HALT ON ERROR
10069      ;*SW13=1      INHIBIT ERROR TYPEOUTS
10070      ;*SW10=1     BELL ON ERROR
10071      ;*SW09=1     LOOP ON ERROR
10072      ;*CALL
10073      ;*      ERROR      N      ;; ERROR=EMT AND N=ERROR ITEM NUMBER
10074
10075 050544      $ERROR:
10076 050544 104407                CKSWR                    ;; TEST FOR CHANGE IN SOFT-SWR
10077 050546 105237 001103      7$:  INCB      $ERFLG      ;; SET THE ERROR FLAG
10078 050552 001775                BEQ      7$              ;; DON'T LET THE FLAG GO TO ZERO
10079 050554 013777 001102 130360  MOV      $TSTNM, @DISPLAY  ;; DISPLAY TEST NUMBER AND ERROR FLAG
10080 050562 032777 002000 130350  BIT      #BIT10, @SWR     ;; BELL ON ERROR?
10081 050570 001402                BEQ      1$              ;; NO - SKIP
10082 050572 104401 001200                TYPE      $BELL         ;; RING BELL
10083 050576 005237 001112      1$:  INC      $ERTTL      ;; COUNT THE NUMBER OF ERRORS
10084 050602 011637 001116      MOV      (SP), $ERRPC    ;; GET ADDRESS OF ERROR INSTRUCTION
10085 050606 162737 000002 001116  SUB      @2, $ERRPC
10086 050614 117737 130276 001114  MOVVB   @2, $ERRPC, $ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
10087 050622 032777 020000 130310  BIT      #BIT13, @SWR     ;; SKIP TYPEOUT IF SET
10088 050630 001004                BNE      20$            ;; SKIP TYPEOUTS
10089 050632 004737 067404      JSR      PC, TYPERR     ;; GO TO USER ERROR ROUTINE
10090 050636 104401 001205                TYPE      , $CRLF
10091 050642
10092 050642 122737 000001 001230      20$:  CMPB     #APTENV, $ENV  ;; RUNNING IN APT MODE
10093 050650 001007                BNE      2$              ;; NO SKIP APT ERROR REPORT
10094 050652 113737 001114 050664  MOVVB   $ITEMB, 21$      ;; SET ITEM NUMBER AS ERROR NUMBER
10095 050660 004737 051470      JSR      PC, $ATY4     ;; REPORT FATAL ERROR TO APT
10096 050664 000                .BYTE  0
10097 050665 000                .BYTE  0
10098 050666 000777                BR      22$              ;; APT ERROR LOOP
10099 050670 005777 130244      2$:  TST      @SWR          ;; HALT ON ERROR
10100 050674 100002                BPL      3$              ;; SKIP IF CONTINUE
10101 050676 000000                HALT     ON ERROR!
10102 050700 104407                CKSWR                    ;; TEST FOR CHANGE IN SOFT-SWR
10103 050702 032777 001000 130230  3$:  BIT      #BIT09, @SWR  ;; LOOP ON ERROR SWITCH SET?
10104 050710 001402                BEQ      4$              ;; BR IF NO
10105 050712 013716 001110      MOV      $LPERR, (SP)   ;; FUDGE RETURN FOR LOOPING
10106 050716 005737 001176      4$:  TST      $ESCAPE     ;; CHECK FOR AN ESCAPE ADDRESS
10107 050722 001402                BEQ      5$              ;; BR IF NONE
10108 050724 013716 001176      MOV      $ESCAPE, (SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE

```

```

10109 050730
10110 050730 022737 043010 000042 5$: CMP #SENDAD,2#42 ;;ACT-11 AUTO-ACCEPT?
:0111 050736 001001 BNE 6$ ;;BRANCH IF NO
10112 050740 000000 HALT ;;YES
10113 050742
10114 050742 000002 6$: RTI ;;RETURN
10115 .SBTTL TYPE ROUTINE
10116
10117 *****
10118 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10119 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10120 *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10121 *NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
10122 *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
10123 *
10124 *CALL:
10125 *1) USING A TRAP INSTRUCTION
10126 * TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
10127 *OR
10128 * TYPE
10129 * MESADR
10130 *
10131
10132 050744 105737 001157 $TYPE: TSTB #TPFLG ;; IS THERE A TERMINAL?
10133 050750 100002 BPL 1$ ;; BR IF YES
10134 050752 000000 HALT ;; HALT HERE IF NO TERMINAL
10135 050754 000430 BR 3$ ;; LEAVE
10136 050756 010046 1$: MOV RO, -(SP) ;; SAVE RO
10137 050760 017600 000002 MOV @2(SP),RO ;; GET ADDRESS OF ASCIZ STRING
10138 050764 122737 000001 001230 CMPB #APTENV,$ENV ;; RUNNING IN APT MODE
10139 050772 001011 BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
10140 050774 132737 000100 001231 BITB #APTPOOL,$ENVM ;; SPOOL MESSAGE TO APT
10141 051002 001405 BEQ 62$ ;; NO GO CHECK FOR CONSOLE
10142 051004 010037 051014 MOV RO,61$ ;; SETUP MESSAGE ADDRESS FOR APT
10143 051010 004737 051460 JSR PC,$ATY3 ;; SPOOL MESSAGE TO APT
10144 051014 000000 .WORD 0 ;; MESSAGE ADDRESS
10145 051016 132737 000040 001231 61$: BITB #APTCSUP,$ENVM ;; APT CONSOLE SUPPRESSED
10146 051024 001003 BNE 60$ ;; YES, SKIP TYPE OUT
10147 051026 112046 2$: MOVB (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
10148 051030 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
10149 051032 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
10150 051034 012600 60$: MOV (SP)+,RO ;; RESTORE RO
10151 051036 062716 000002 3$: ADD #2,(SP) ;; ADJUST RETURN PC
10152 051042 000002 RTI ;; RETURN
10153 051044 122716 000011 4$: CMPB #HT,(SP) ;; BRANCH IF <HT>
10154 051050 001430 BEQ 8$
10155 051052 122716 000200 CMPB #CRLF,(SP) ;; BRANCH IF NOT <CRLF>
10156 051056 001006 BNE 5$
10157 051060 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
10158 051062 104401 TYPE ;; TYPE A CR AND LF
10159 051064 001205 $CRLF
10160 051066 105037 051222 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
10161 051072 000755 BR 2$ ;; GET NEXT CHARACTER
10162 051074 004737 051156 5$: JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
10163 051100 123726 001156 6$: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
10164 051104 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.

```

```

10165 051106 013746 001154      MOV      $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED
10166                                     ;; AND THE NULL CHAR.
10167 051112 105366 000001      7$: DECB  1(SP)          ;; DOES A NULL NEED TO BE TYPED?
10168 051116 002770                BLT   6$                ;; BR IF NO--GO POP THE NULL OFF OF STACK
10169 051120 004737 051156      JSR   PC,$TYPEC        ;; GO TYPE A NULL
10170 051124 105337 051222      DECB  $CHARCNT         ;; DO NOT COUNT AS A COUNT
10171 051130 000770                BR    7$                ;; LOOP
10172
10173                                     ;HORIZONTAL TAB PROCESSOR
10174
10175 051132 112716 000040      8$:  MOVB  #' ,(SP)      ;; REPLACE TAB WITH SPACE
10176 051136 004737 051156      9$:  JSR   PC,$TYPEC        ;; TYPE A SPACE
10177 051142 132737 000007 051222  BITB  #',$CHARCNT        ;; BRANCH IF NOT AT
10178 051150 001372                BNE  9$                ;; TAB STOP
10179 051152 005726                TST  (SP)+             ;; POP SPACE OFF STACK
10180 051154 000724                BR   2$                ;; GET NEXT CHARACTER
10181 051156 105777 127766      $TYPEC: TSTB  2$STPS        ;; WAIT UNTIL PRINTER IS READY
10182 051162 100375                BPL  $TYPEC
10183 051164 116677 000002 127760  MOVB  2(SP),2$TPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.
10184 051172 122766 000015 000002  CMPB  #CR,2(SP)        ;; IS CHARACTER A CARRIAGE RETURN?
10185 051200 001003                BNE  1$                ;; BRANCH IF NO
10186 051202 105037 051222      CLRB  $CHARCNT         ;; YES--CLEAR CHARACTER COUNT
10187 051206 000406                BR   $TYPEX            ;; EXIT
10188 051210 122766 000012 000002  1$:  CMPB  #LF,2(SP)     ;; IS CHARACTER A LINE FEED?
10189 051216 001402                BEQ  $TYPEX            ;; BRANCH IF YES
10190 051220 105227                INCB (PC)+             ;; COUNT THE CHARACTER
10191 051222 000000                $CHARCNT: .WORD 0      ;; CHARACTER COUNT STORAGE
10192 051224 000207                $TYPEX: RTS   PC
10193
10194                                     .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10195
10196                                     ;*****
10197                                     ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
10198                                     ;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
10199                                     ;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
10200                                     ;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
10201                                     ;REPLACED WITH SPACES.
10202                                     ;CALL:
10203                                     ;*
10204                                     ;*   MOV      NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
10205                                     ;*   TYPDS                    ;; GO TO THE ROUTINE
10206
10206 051226                $TYPDS:
10207 051226 010046                MOV   R0,-(SP)         ;; PUSH R0 ON STACK
10208 051230 010146                MOV   R1,-(SP)         ;; PUSH R1 ON STACK
10209 051232 010246                MOV   R2,-(SP)         ;; PUSH R2 ON STACK
10210 051234 010346                MOV   R3,-(SP)         ;; PUSH R3 ON STACK
10211 051236 010546                MOV   R5,-(SP)         ;; PUSH R5 ON STACK
10212 051240 012746 020200      MOV   #20200,-(SP)     ;; SET BLANK SWITCH AND SIGN
10213 051244 016605 000020      MOV   20(SP),R5        ;; GET THE INPUT NUMBER
10214 051250 100004                BPL  1$                ;; BR IF INPUT IS POS.
10215 051252 005405                NEG  R5                ;; MAKE THE BINARY NUMBER POS.
10216 051254 112766 000055 000001  MOVB  #'-,1(SP)        ;; MAKE THE ASCII NUMBER NEG.
10217 051262 005000                CLR  R0                ;; ZERO THE CONSTANTS INDEX
10218 051264 012703 051442      MOV   #SDBLK,R3        ;; SETUP THE OUTPUT POINTER
10219 051270 112723 000040      MOVB  #' ,(R3)+        ;; SET THE FIRST CHARACTER TO A BLANK
10220 051274 005002                2$:  CLR  R2                ;; CLEAR THE BCD NUMBER

```

10221	051276	016001	051432		MOV	\$DTBL(R0),R1	;; GET THE CONSTANT
10222	051302	160105		3\$:	SUB	R1,R5	;; FORM THIS BCD DIGIT
10223	051304	002402			BLT	4\$;; BR IF DONE
10224	051306	005202			INC	R2	;; INCREASE THE BCD DIGIT BY 1
10225	051310	000774			BR	3\$	
10226	051312	060105		4\$:	ADD	R1,R5	;; ADD BACK THE CONSTANT
10227	051314	005702			TST	R2	;; CHECK IF BCD DIGIT=0
10228	051316	001002			BNE	5\$;; FALL THROUGH IF 0
10229	051320	105716			TSTB	(SP)	;; STILL DOING LEADING 0'S?
10230	051322	100407			BMI	7\$;; BR IF YES
10231	051324	106316		5\$:	ASLB	(SP)	;; MSD?
10232	051326	103003			BCC	6\$;; BR IF NO
10233	051330	116663	000001 177777		MOVB	1(SP),-1(R3)	;; YES--SET THE SIGN
10234	051336	052702	000060	6\$:	BIS	#'0,R2	;; MAKE THE BCD DIGIT ASCII
10235	051342	052702	000040	7\$:	BIS	#',R2	;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
10236	051346	110223			MOVB	R2,(R3)+	;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
10237	051350	005720			TST	(R0)+	;; JUST INCREMENTING
10238	051352	020027	000010		CMP	R0,#10	;; CHECK THE TABLE INDEX
10239	051356	002746			BLT	2\$;; GO DO THE NEXT DIGIT
10240	051360	003002			BGT	8\$;; GO TO EXIT
10241	051362	010502			MOV	R5,R2	;; GET THE LSD
10242	051364	000764			BR	6\$;; GO CHANGE TO ASCII
10243	051366	105726		8\$:	TSTB	(SP)+	;; WAS THE LSD THE FIRST NON-ZERO?
10244	051370	100003			BPL	9\$;; BR IF NO
10245	051372	116663	177777 177776	9\$:	MOVB	-1(SP),-2(R3)	;; YES--SET THE SIGN FOR TYPING
10246	051400	105013			CLRB	(R3)	;; SET THE TERMINATOR
10247	051402	012605			MOV	(SP)+,R5	;; POP STACK INTO R5
10248	051404	012603			MOV	(SP)+,R3	;; POP STACK INTO R3
10249	051406	012602			MOV	(SP)+,R2	;; POP STACK INTO R2
10250	051410	012601			MOV	(SP)+,R1	;; POP STACK INTO R1
10251	051412	012600			MOV	(SP)+,R0	;; POP STACK INTO R0
10252	051414	104401	051442		TYPE	\$DBLK	;; NOW TYPE THE NUMBER
10253	051420	016666	000002 000004		MOV	2(SP),4(SP)	;; ADJUST THE STACK
10254	051426	012616			MOV	(SP)+,(SP)	
10255	051430	000002			RTI		;; RETURN TO USER
10256	051432	023420		\$DTBL:	10000.		
10257	051434	001750			1000.		
10258	051436	000144			100.		
10259	051440	000012			10.		
10260	051442	000004		\$DBLK:	.BLKW 4		
10261				.SBTTL	APT COMMUNICATIONS ROUTINE		
10262				*****			
10263				*****			
10264	051452	112737	000001 051716	\$ATY1:	MOVB	#1,\$FFLG	;; TO REPORT FATAL ERROR
10265	051460	112737	000001 051714	\$ATY3:	MOVB	#1,\$MFLG	;; TO TYPE A MESSAGE
10266	051466	000403			BR	\$ATYC	
10267	051470	112737	000001 051716	\$ATY4:	MOVB	#1,\$FFLG	;; TO ONLY REPORT FATAL ERROR
10268	051476			\$ATYC:			
10269	051476	010046			MOV	R0,-(SP)	;; PUSH R0 ON STACK
10270	051500	010146			MOV	R1,-(SP)	;; PUSH R1 ON STACK
10271	051502	105737	051714		TSTB	\$MFLG	;; SHOULD TYPE A MESSAGE?
10272	051506	001450			BEQ	5\$;; IF NOT: BR
10273	051510	122737	000001 001230		CMPB	#APTENV,\$ENV	;; OPERATING UNDER APT?
10274	051516	001031			BNE	3\$;; IF NOT: BR
10275	051520	132737	000100 001231		BITB	#APTPOOL,\$ENVM	;; SHOULD SPOOL MESSAGES?
10276	051526	001425			BEQ	3\$;; IF NOT: BR

```

10277 051530 017600 000004          MOV      @4(SP),RO      ;;GET MESSAGE ADDR.
10278 051534 062766 000002 000004  ADD      #2,4(SP)      ;;BUMP RETURN ADDR.
10279 051542 005737 001210          TST      $MSGTYPE     ;;SEE IF DONE W/ LAST XMISSION?
10280 051546 001375 001224          BNE      1$           ;;IF NOT: WAIT
10281 051550 010037 001224          MOV      RO,$MSGAD     ;;PUT ADDR IN MAILBOX
10282 051554 105720 001224          TSTB    (RO)+         ;;FIND END OF MESSAGE
10283 051556 001376 001224          BNE      2$           ;;SUB START OF MESSAGE
10284 051560 163700 001224          SUB      $MSGAD,RO     ;;GET MESSAGE LNTH IN WORDS
10285 051564 006200 001226          ASR      RO           ;;PUT LENGTH IN MAILBOX
10286 051566 010037 000004 001210  MOV      RO,$MSGLGT    ;;TELL APT TO TAKE MSG.
10287 051572 012737 000004          MOV      #4,$MSGTYPE
10288 051600 000413 000004 051626 3$:     BR       5$           ;;PUT MSG ADDR IN JSR LINKAGE
10289 051602 017637 000002 000004  ADD      #2,4(SP)      ;;BUMP RETURN ADDRESS
10290 051610 062766 000002          MOV      177776,-(SP) ;;PUSH 177776 ON STACK
10291 051616 013746 177776          JSR      PC,$TYPE     ;;CALL TYPE MACRO
10292 051622 004737 050744          PC      0
10293 051626 000000          .WORD   0
10294 051630          5$:
10295 051630 105737 051716          10$:    TSTB    $FFLG     ;;SHOULD REPORT FATAL ERROR?
10296 051634 001416 001230          BEQ     12$          ;;IF NOT: BR
10297 051636 005737 001230          TST     $ENV        ;;RUNNING UNDER APT?
10298 051642 001413 001210          BEQ     12$          ;;IF NOT: BR
10299 051644 005737 001210          TST     $MSGTYPE    ;;FINISHED LAST MESSAGE?
10300 051650 001375 000004 001212  BNE     11$          ;;IF NOT: WAIT
10301 051652 017637 000002 000004  MOV     @4(SP),$FATAL ;;GET ERROR #
10302 051660 062766 000002          ADD     #2,4(SP)     ;;BUMP RETURN ADDR.
10303 051666 005237 001210          INC     $MSGTYPE    ;;TELL APT TO TAKE ERROR
10304 051672 105037 051716          12$:    CLRB    $FFLG     ;;CLEAR FATAL FLAG
10305 051676 105037 051715          CLRB    $LFLG      ;;CLEAR LOG FLAG
10306 051702 105037 051714          CLRB    $MFLG      ;;CLEAR MESSAGE FLAG
10307 051706 012601          MOV     (SP)+,R1    ;;POP STACK INTO R1
10308 051710 012600          MOV     (SP)+,RO    ;;POP STACK INTO RO
10309 051712 000207          RTS      PC         ;;RETURN
10310 051714 000          $MFLG: .BYTE 0      ;;MESSG. FLAG
10311 051715 000          $LFLG: .BYTE 0      ;;LOG FLAG
10312 051716 000          $FFLG: .BYTE 0      ;;FATAL FLAG
10313 051720          .EVEN
10314 000200  APTSIZE=200
10315 000001  APTENV=001
10316 000100  APTSPool=100
10317 000040  APTCSUP=040
10318          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
10319
10320          ;;*****
10321          ;;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
10322          ;;OCTAL (ASCII) NUMBER AND TYPE IT.
10323          ;;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
10324          ;;CALL:
10325          ;;      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
10326          ;;      TYPOS      ;;CALL FOR TYPEOUT
10327          ;;      .BYTE  N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
10328          ;;      .BYTE  M      ;;M=1 OR 0
10329          ;;      ;;1=TYPE LEADING ZEROS
10330          ;;      ;;0=SUPPRESS LEADING ZEROS
10331          ;;
10332          ;;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST

```

```

10333      ;*STYPOS OR STYPOC
10334      ;*CALL:
10335      ;*      MOV      NUM,-(SP)      ;:NUMBER TO BE TYPED
10336      ;*      TYPON      ;:CALL FOR TYPEOUT
10337      ;*
10338      ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
10339      ;*CALL:
10340      ;*      MOV      NUM,-(SP)      ;:NUMBER TO BE TYPED
10341      ;*      TYPOC      ;:CALL FOR TYPEOUT
10342      ;*
10343      051720 017646 000000      STYPOS: MOV      2(SP),-(SP)      ;:PICKUP THE MODE
10344      051724 116637 000001      052143      MOVVB     1(SP),SOFILL      ;:LOAD ZERO FILL SWITCH
10345      051732 112637 052145      MOVVB     (SP)+,SOMODE+1      ;:NUMBER OF DIGITS TO TYPE
10346      051736 062716 000002      ADD      #2,(SP)      ;:ADJUST RETURN ADDRESS
10347      051742 000406      BR      STYPON
10348      051744 112737 000001      052143      STYPOC: MOVVB     #1,SOFILL      ;:SET THE ZERO FILL SWITCH
10349      051752 112737 000006      052145      MOVVB     #6,SOMODE+1      ;:SET FOR SIX(6) DIGITS
10350      051760 112737 000005      052142      STYPON: MOVVB     #5,SOCNT      ;:SET THE ITERATION COUNT
10351      051766 010346      MOV      R3,-(SP)      ;:SAVE R3
10352      051770 010446      MOV      R4,-(SP)      ;:SAVE R4
10353      051772 010546      MOV      R5,-(SP)      ;:SAVE R5
10354      051774 113704 052145      MOVVB     SOMODE+1,R4      ;:GET THE NUMBER OF DIGITS TO TYPE
10355      052000 005404      NEG      R4
10356      052002 062704 000006      ADD      #6,R4      ;:SUBTRACT IT FOR MAX. ALLOWED
10357      052006 110437 052144      MOVVB     R4,SOMODE      ;:SAVE IT FOR USE
10358      052012 113704 052143      MOVVB     SOFILL,R4      ;:GET THE ZERO FILL SWITCH
10359      052016 016605 000012      MOV      12(SP),R5      ;:PICKUP THE INPUT NUMBER
10360      052022 005003      CLR      R3      ;:CLEAR THE OUTPUT WORD
10361      052024 006105      1$:      ROL      R5      ;:ROTATE MSB INTO "C"
10362      052026 000404      BR      3$
10363      052030 006105      2$:      ROL      R5      ;:GO DO MSB
10364      052032 006105      ROL      R5      ;:FORM THIS DIGIT
10365      052034 006105      ROL      R5
10366      052036 010503      MOV      R5,R3
10367      052040 006103      3$:      ROL      R3      ;:GET LSB OF THIS DIGIT
10368      052042 105337 052144      DECB     SOMODE      ;:TYPE THIS DIGIT?
10369      052046 100016      BPL      7$      ;:BR IF NO
10370      052050 042703 177770      BIC      #177770,R3      ;:GET RID OF JUNK
10371      052054 001002      BNE      4$      ;:TEST FOR 0
10372      052056 005704      TST      R4      ;:SUPPRESS THIS 0?
10373      052060 001403      BEQ      5$      ;:BR IF YES
10374      052062 005204      4$:      INC      R4      ;:DON'T SUPPRESS ANYMORE 0'S
10375      052064 052703 000060      BIS      #'0,R3      ;:MAKE THIS DIGIT ASCII
10376      052070 052703 000040      5$:      BIS      #' ,R3      ;:MAKE ASCII IF NOT ALREADY
10377      052074 110337 052140      MOVVB     R3,8$      ;:SAVE FOR TYPING
10378      052100 104401 052140      TYPE     8$      ;:GO TYPE THIS DIGIT
10379      052104 105337 052142      7$:      DECB     SOCNT      ;:COUNT BY 1
10380      052110 003347      BGT      2$      ;:BR IF MORE TO DO
10381      052112 002402      BLT      6$      ;:BR IF DONE
10382      052114 005204      INC      R4      ;:INSURE LAST DIGIT ISN'T A BLANK
10383      052116 000744      BR      2$      ;:GO DO THE LAST DIGIT
10384      052120 012605      6$:      MOV      (SP)+,R5      ;:RESTORE R5
10385      052122 012604      MOV      (SP)+,R4      ;:RESTORE R4
10386      052124 012603      MOV      (SP)+,R3      ;:RESTORE R3
10387      052126 016666 000002 000004      MOV      2(SP),4(SP)      ;:SET THE STACK FOR RETURNING
10388      052134 012616      MOV      (SP)+,(SP)

```



```

10389 052136 000002
10390 052140 000
10391 052141 000
10392 052142 000
10393 052143 000
10394 052144 000000
10395
10396
10397
10398
10399 052146 000000
10400 052150 000000
10401 052152 000000
10402 052154 000001
10403 052155
10404 052156
10405
10406
10407
10408
10409
10410
10411
10412
10413
10414 052156 005037 052146
10415 052162 012737 052154 052150
10416 052170 013737 052150 052152
10417 052176 012737 052226 000060
10418 052204 012737 000200 000062
10419 052212 005777 126730
10420 052216 012777 000100 126720
10421 052224 000207
10422
10423
10424
10425
10426
10427
10428
10429
10430 052226 117746 126714
10431 052232 042716 177600
10432 052236 021627 000003
10433 052242 001007
10434 052244 104401 053354
10435 052250 004737 052156
10436 052254 005726
10437 052256 000137 047500
10438 052262 021627 000007
10439 052266 001004
10440 052270 022737 000176 001140
10441 052276 001500
10442
10443 052300
10444 052300 022737 000001 052146

```

```

RTI ;; RETURN
BS: .BYTE 0 ;; STORAGE FOR ASCII DIGIT
      .BYTE 0 ;; TERMINATOR FOR TYPE ROUTINE
SOCNT: .BYTE 0 ;; OCTAL DIGIT COUNTER
SOFILL: .BYTE 0 ;; ZERO FILL SWITCH
SOMODE: .WORD 0 ;; NUMBER OF DIGITS TO TYPE
.SBTTL TTY INPUT ROUTINE

;*****
.ENABL LSB
$TKCNT: .WORD 0 ;; NUMBER OF ITEMS IN QUEUE
$TKQIN: .WORD 0 ;; INPUT POINTER
$TKQOUT: .WORD 0 ;; OUTPUT POINTER
$TKQSRT: .BLKB 1 ;; TTY KEYBOARD QUEUE
$TKQEND=.
.EVEN

; *TK INITIALIZE ROUTINE
; *THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
; *SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
; *CALL:
; * JSR PC,$TKINT
; * RETURN
$TKINT: CLR $TKCNT ;; CLEAR COUNT OF ITEMS IN QUEUE
        MOV $TKQSRT,$TKQIN ;; MOVE THE STARTING ADDRESS OF THE
        MOV $TKQIN,$TKQOUT ;; QUEUE INTO THE INPUT & OUTPUT POINTERS.
        MOV $TKSRV,$TKVEC ;; INITIALIZE THE KEYBOARD VECTOR
        MOV #200,$TKVEC+2 ;; "BR" LEVEL 4
        TST $TKB ;; CLEAR DONE FLAG
        MOV #100,$TKS ;; ENABLE TTY KEYBOARD INTERRUPT
        RTS PC ;; RETURN TO CALLER

; *TK SERVICE ROUTINE
; *THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
; *BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
; *IT IN THE QUEUE.
; *IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
; *UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
$TKSRV: MOVB $TKB,-(SP) ;; PICKUP THE CHARACTER
        BIC #↑C177,(SP) ;; STRIP THE JUNK
        CMP (SP),#3 ;; IS IT A CONTROL C?
        BNE 1$ ;; BRANCH IF NO
        TYPE $CNTLC ;; TYPE A CONTROL-C (↑C)
        JSR PC,$TKINT ;; INIT THE KEYBOARD
        TST (SP)+ ;; CLEAN UP STACK
        JMP STOP ;; CONTROL C RESTART
1$: CMP (SP),#7 ;; IS IT A CONTROL G?
    BNE 2$ ;; BRANCH IF NO
    CMP #SWREG,SWR ;; IS SOFT-SWR SELECTED?
    BEQ 6$ ;; GO TO SWR CHANGE
2$: CMP #1,$TKCNT ;; IS THE QUEUE FULL?

```

```

10445 052306 001004          BNE      3$          ;; BRANCH IF NO
10446 052310 104401 001200  TYPE      $BELL      ;; RING THE TTY BELL
10447 052314 005726          TST      (SP)+       ;; CLEAN CHARACTER OFF OF STACK
10448 052316 000451          BR       5$          ;; EXIT
10449 052320 021627 000023  3$:     CMP      (SP),#23  ;; IS IT A CONTROL-S?
10450 052324 001021          BNE      32$        ;; BRANCH IF NO
10451 052326 005077 126612  CLR      @STKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
10452 052332 005726          TST      (SP)+       ;; CLEAN CHAR OFF STACK
10453 052334 105777 126604  31$:   TSTB     @STKS      ;; WAIT FOR A CHAR
10454 052340 100375          BPL      31$        ;; LOOP UNTIL ITS THERE
10455 052342 117746 126600  MOVB     @STKB,-(SP)  ;; GET THE CHARACTER
10456 052346 042716 177600  BIC      #1C177,(SP) ;; MAKE IT 7-BIT ASCII
10457 052352 022627 000021  CMP      (SP)+,#21  ;; IS IT A CONTROL-Q?
10458 052356 001366          BNE      31$        ;; BRANCH IF NO
10459 052360 012777 000100 126556  MOV      #100,@STKS  ;; REENABLE TTY KEYBOARD INTERRUPTS
10460 052366 000002          RTI                    ;; RETURN
10461 052370 005237 052146  32$:   INC      $TKCNT    ;; COUNT THIS CHARACTER
10462 052374 021627 000140  CMP      (SP),#140  ;; IS IT UPPER CASE?
10463 052400 002405          BLT      4$          ;; BRANCH IF YES
10464 052402 021627 000175  CMP      (SP),#175  ;; IS IT A SPECIAL CHAR?
10465 052406 003002          BGT      4$          ;; BRANCH IF YES
10466 052410 042716 000040  BIC      #40,(SP)   ;; MAKE IT UPPER CASE
10467 052414 112677 177530  4$:   MOVB     (SP)+,@STKQIN ;; AND PUT IT IN QUEUE
10468 052420 005237 052150  INC      $TKQIN     ;; UPDATE THE POINTER
10469 052424 023727 052150 052155  CMP      $TKQIN,#$TKQEND ;; GO OFF THE END?
10470 052432 001003          BNE      5$          ;; BRANCH IF NO
10471 052434 012737 052154 052150  MOV      #$TKQSRT,$TKQIN ;; RESET THE POINTER
10472 052442 000002          RTI                    ;; RETURN

```

```

10473
10474
10475
10476
10477
10478
10479
10480
10481
10482
10483
10484
10485
10486
10487
10488
10489
10490
10491
10492

```

```

*****
*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
$CKSWR: CMP      #SWREG,SWR      ;; IS THE SOFT-SWR SELECTED
        BNE      15$          ;; EXIT IF NOT
        TSTB     @STKS      ;; IS A CHAR WAITING?
        BPL      15$          ;; IF NOT, EXIT
        MOVB     @STKB,-(SP)  ;; YES
        BIC      #1C177,(SP)  ;; MAKE IT 7-BIT ASCII
        CMP      (SP),#7      ;; IS IT A CONTROL-G?
        BNE      2$          ;; IF NOT, PUT IT IN THE TTY QUEUE
        AND     EXIT

```

```

*****
*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.
6$:   CMPB     $AUTOB,#1      ;; ARE WE RUNNING IN AUTO-MODE?
        BEQ      2$          ;; BRANCH IF YES
        TST      (SP)+       ;; CLEAR CONTROL-G OFF STACK
        JSR      PC,$TKINT   ;; FLUSH THE TTY INPUT QUEUE
        CLR      @STKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
        MOVB     #1,$INTAG   ;; SET INTERRUPT MODE INDICATOR

```

```

10493 052500 123727 001134 000001 6$:   CMPB     $AUTOB,#1      ;; ARE WE RUNNING IN AUTO-MODE?
10494 052506 001674          BEQ      2$          ;; BRANCH IF YES
10495 052510 005726          TST      (SP)+       ;; CLEAR CONTROL-G OFF STACK
10496 052512 004737 052156  JSR      PC,$TKINT   ;; FLUSH THE TTY INPUT QUEUE
10497 052516 005077 126422  CLR      @STKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
10498 052522 112737 000001 001135  MOVB     #1,$INTAG   ;; SET INTERRUPT MODE INDICATOR
10499
10500 052530 104401 053366          TYPE      , $CNTLG    ;; ECHO THE CONTROL-G (↑G)

```

Address	Hex 1	Hex 2	Hex 3	Hex 4	Label	Op	Op Arg	Comment
10501	052534	104401	053373		\$GTSWR:	TYPE	\$MSWR	:: TYPE CURRENT CONTENTS
10502	052540	013746	000176			MOV	\$WREG, -(SP)	:: SAVE SWREG FOR TYPEOUT
10503	052544	104402				TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
10504	052546	104401	053404			TYPE	\$MNEW	:: PROMPT FOR NEW SWR
10505	052552	005046			19\$:	CLR	-(SP)	:: CLEAR COUNTER
10506	052554	005046				CLR	-(SP)	:: THE NEW SWR
10507	052556	105777	126362		7\$:	TSTB	\$STKS	:: CHAR THERE?
10508	052562	100375				BPL	7\$:: IF NOT TRY AGAIN
10509								
10510	052564	117746	126356			MOVB	\$TKB, -(SP)	:: PICK UP CHAR
10511	052570	042716	177600			BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
10512								
10513	052574	021627	000003			CMP	(SP), #3	:: IS IT A CONTROL-C?
10514	052600	001015				BNE	9\$:: BRANCH IF NOT
10515	052602	104401	053354			TYPE	\$CNTLC	:: YES, ECHO CONTROL-C (↑C)
10516	052606	062706	000006			ADD	#6, SP	:: CLEAN UP STACK
10517	052612	123727	001135	000001		CMPB	\$INTAG, #1	:: REENABLE TTY KEYBOARD INTERRUPTS?
10518	052620	001003				BNE	8\$:: BRANCH IF NO
10519	052622	012777	000100	126314		MOV	#100, \$STKS	:: ALLOW TTY KEYBOARD INTERRUPTS
10520	052630	000137	047500		8\$:	JMP	STOP	:: CONTROL-C RESTART
10521								
10522								
10523	052634	021627	000025		9\$:	CMP	(SP), #25	:: IS IT A CONTROL-U?
10524	052640	001005				BNE	10\$:: BRANCH IF NOT
10525	052642	104401	053361			TYPE	\$CNTLU	:: YES, ECHO CONTROL-U (↑U)
10526	052646	062706	000006		20\$:	ADD	#6, SP	:: IGNORE PREVIOUS INPUT
10527	052652	000737				BR	19\$:: LET'S TRY IT AGAIN
10528								
10529								
10530	052654	021627	000015		10\$:	CMP	(SP), #15	:: IS IT A <CR>?
10531	052660	001022				BNE	16\$:: BRANCH IF NO
10532	052662	005766	000004			TST	4(SP)	:: YES, IS IT THE FIRST CHAR?
10533	052666	001403				BEQ	11\$:: BRANCH IF YES
10534	052670	016677	000002	126242		MOV	2(SP), \$SWR	:: SAVE NEW SWR
10535	052676	062706	000006		11\$:	ADD	#6, SP	:: CLEAN UP STACK
10536	052702	104401	001205		14\$:	TYPE	\$CRLF	:: ECHO <CR> AND <LF>
10537	052706	123727	001135	000001		CMPB	\$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS?
10538	052714	001003				BNE	15\$:: BRANCH IF NOT
10539	052716	012777	000100	126220		MOV	#100, \$STKS	:: RE-ENABLE TTY KBD INTERRUPTS
10540	052724	000002			15\$:	RTI		:: RETURN
10541	052726	004737	051156		16\$:	JSR	PC, \$TYPEC	:: ECHO CHAR
10542	052732	021627	000060			CMP	(SP), #60	:: CHAR < 0?
10543	052736	002420				BLT	18\$:: BRANCH IF YES
10544	052740	021627	000067			CMP	(SP), #67	:: CHAR > 7?
10545	052744	003015				BGT	18\$:: BRANCH IF YES
10546	052746	042726	000060			BIC	#60, (SP)+	:: STRIP-OFF ASCII
10547	052752	005766	000002			TST	2(SP)	:: IS THIS THE FIRST CHAR
10548	052756	001403				BEQ	17\$:: BRANCH IF YES
10549	052760	006316				ASL	(SP)	:: NO, SHIFT PRESENT
10550	052762	006316				ASL	(SP)	:: CHAR OVER TO MAKE
10551	052764	006316				ASL	(SP)	:: ROOM FOR NEW ONE.
10552	052766	005266	000002		17\$:	INC	2(SP)	:: KEEP COUNT OF CHAR
10553	052772	056616	177776			BIS	-2(SP), (SP)	:: SET IN NEW CHAR
10554	052776	000667				BR	7\$:: GET THE NEXT ONE
10555	053000	104401	001204		18\$:	TYPE	\$QUES	:: TYPE ?<CR><LF>
10556	053004	000720				BR	20\$:: SIMULATE CONTROL-U

```

10557      .DSABL  LSB
10558
10559
10560      ;*****
10561      ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10562      ;*CALL:
10563      ;*      RDCHR          ;:GET A CHARACTER FROM THE QUEUE
10564      ;*      RETURN HERE   ;:CHARACTER IS ON THE STACK
10565      ;*                    ;:WITH PARITY BIT STRIPPED OFF
10566
10567
10568      053006 011646          SRDCHR: MOV      (SP),-(SP)      ;:PUSH DOWN THE PC AND
10569      053010 016666 000004 000002  MOV      4(SP),2(SP)    ;:THE PS
10570      053016 005066 000004          CLR      4(SP)          ;:GET READY FOR A CHARACTER
10571      053022 005046          CLR      -(SP)         ;:PUT NEW PS ON STACK
10572      053024 012746 053032  MOV      #64$,-(SP)    ;:PUT NEW PC ON STACK
10573      053030 000002          RTI                    ;:POP NEW PC AND PS
10574      053032
10575      053032 005737 052146 64$:  TST      $TKCNT          ;:WAIT ON A CHARACTER
10576      053036 001775          BEQ      1$
10577      053040 005337 052146          DEC      $TKCNT        ;:DECREMENT THE COUNTER
10578      053044 117766 177102 000004  MOVB    2$TKQOUT,4(SP) ;:GET ONE CHARACTER
10579      053052 005237 052152          INC      $TKQOUT      ;:UPDATE THE POINTER
10580      053056 023727 052152 052155  CMP     $TKQOUT,#$TKQEND ;:DID IT GO OFF OF THE END?
10581      053064 001003          BNE     2$           ;:BRANCH IF NO
10582      053066 012737 052154 052152  MOV     #$TKQSRRT,$TKQOUT ;:RESET THE POINTER
10583      053074 000002          RTI                    ;:RETURN
10584      ;*****
10585      ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10586      ;*CALL:
10587      ;*      RDLIN          ;:INPUT A STRING FROM THE TTY
10588      ;*      RETURN HERE   ;:ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10589      ;*                    ;:TERMINATOR WILL BE A BYTE OF ALL 0'S
10590
10591      053076 010346          SRDLIN: MOV     R3, -(SP) ;:SAVE R3
10592      053100 005046          CLR     -(SP)         ;:CLEAR THE RUBOUT KEY
10593      053102 012703 053332          MOV     #$TTYIN,R3   ;:GET ADDRESS
10594      053106 022703 053354          CMP     #$TTYIN+22,R3 ;:BUFFER FULL?
10595      053112 101456          BLOS   4$            ;:BR IF YES
10596      053114 104410          RDCHR  ;:GO READ ONE CHARACTER FROM THE TTY
10597      053116 112613          MOVB   (SP)+,(R3)    ;:GET CHARACTER
10598      053120 122713 000177 10$:  CMPB   #177,(R3)     ;:IS IT A RUBOUT
10599      053124 001022          BNE    5$           ;:BR IF NO
10600      053126 005716          TST    (SP)         ;:IS THIS THE FIRST RUBOUT?
10601      053130 001007          BNE    6$           ;:BR IF NO
10602      053132 112737 000134 053330  MOVB   #' \,9$      ;:TYPE A BACK SLASH
10603      053140 104401 053330          TYPE   9$
10604      053144 012716 177777          MOV    #-1,(SP)     ;:SET THE RUBOUT KEY
10605      053150 005303          DEC    R3           ;:BACKUP BY ONE
10606      053152 020327 053332          CMP    R3,$$TTYIN   ;:STACK EMPTY?
10607      053156 103434          BLO   4$           ;:BR IF YES
10608      053160 111337 053330          MOVB   (R3),9$      ;:SETUP TO TYPEOUT THE DELETED CHAR.
10609      053164 104401 053330          TYPE   9$
10610      053170 000746          BR     2$           ;:GO READ ANOTHER CHAR.
10611      053172 005716          TST    (SP)         ;:RUBOUT KEY SET?
10612      053174 001406          BEQ    7$           ;:BR IF NO

```

```

10613 053176 112737 000134 053330      MOVB    #' \,9$      ;;TYPE A BACK SLASH
10614 053204 104401 053330      TYPE    9$
10615 053210 005016          CLR     (SP)        ;;CLEAR THE RUBOUT KEY
10616 053212 122713 000025      7$:    CMPB    #25,(R3)  ;;IS CHARACTER A CTRL U?
10617 053216 001003          BNE     8$          ;;BR IF NO
10618 053220 104401 053361      TYPE    $CNTLU      ;;TYPE A CONTROL "U"
10619 053224 000726          BR     1$          ;;GO START OVER
10620 053226 122713 000022      8$:    CMPB    #22,(R3)  ;;IS CHARACTER A "↑R"?
10621 053232 001011          BNE     3$          ;;BRANCH IF NO
10622 053234 105013          CLRB   (R3)        ;;CLEAR THE CHARACTER
10623 053236 104401 001205      TYPE    $CRLF      ;;TYPE A "CR" & "LF"
10624 053242 104401 053332      TYPE    $TTYIN     ;;TYPE THE INPUT STRING
10625 053246 000717          BR     2$          ;;GO PICKUP ANOTHER CHACTER
10626 053250 104401 001204      4$:    TYPE    $QUES   ;;TYPE A '?'
10627 053254 000712          BR     1$          ;;CLEAR THE BUFFER AND LOOP
10628 053256 111337 053330      3$:    MOVB    (R3),9$   ;;ECHO THE CHARACTER
10629 053262 104401 053330      TYPE    9$
10630 053266 122723 000015      CMPB   #15,(R3)+   ;;CHECK FOR RETURN
10631 053272 001305          BNE     2$          ;;LOOP IF NOT RETURN
10632 053274 105063 177777      CLRB   -1(R3)     ;;CLEAR RETURN (THE 15)
10633 053300 104401 001206      TYPE    $LF        ;;TYPE A LINE FEED
10634 053304 005726          TST    (SP)+      ;;CLEAN RUBOUT KEY FROM THE STACK
10635 053306 012603          MOV    (SP)+,R3   ;;RESTORE R3
10636 053310 011646          MOV    (SP)-,(SP) ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10637 053312 016666 000004 000002      MOV    4(SP),2(SP) ;;FIRST ASCII CHARACTER ON IT
10638 053320 012766 053332 000004      MOV    #TTYIN,4(SP)
10639 053326 000002          RTI
10640 053330 000          9$:    .BYTE   0        ;;RETURN
10641 053331 000          .BYTE   0        ;;STORAGE FOR ASCII CHAR. TO TYPE
10642 053332 000022          $TTYIN: .BLKB   22   ;;TERMINATOR
10643 053354 041536 005015 000      $CNTLC: .ASCIZ  /↑C/<15><12> ;;RESERVE 22 BYTES FOR TTY INPUT
10644 053361 136 006525 000012      $CNTLU: .ASCIZ  /↑U/<15><12> ;;CONTROL "C"
10645 053366 043536 005015 000      $CNTLG: .ASCIZ  /↑G/<15><12> ;;CONTROL "U"
10646 053373 015 051412 051127      $MSWR:  .ASCIZ  <15><12>/SWR = / ;;CONTROL "G"
10647 053400 036440 000040
10648 053404 020040 042516 020127      $MNEW:  .ASCIZ  / NEW = /
10649 053412 020075 000
10650 053416
10651
10652
10653
10654
10655
10656
10657
10658
10659
10660
10661
10662
10663
10664
10665 053416 011646 000004 000002      $RDOCT: MOV    (SP)-,(SP) ;;PROVIDE SPACE FOR THE
10666 053420 016666          MOV    4(SP),2(SP) ;;INPUT NUMBER
10667 053426 010046          MOV    R0,-(SP)   ;;PUSH R0 ON STACK
10668 053430 010146          MOV    R1,-(SP)   ;;PUSH R1 ON STACK

```

.EVEN
.SBTTL READ AN OCTAL NUMBER FROM THE TTY

*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
*CHANGE IT TO BINARY.
*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
*CALL:
* RDOCT READ AN OCTAL NUMBER
* RETURN HERE LOW ORDER BITS ARE ON TOP OF THE STACK
* HIGH ORDER BITS ARE IN \$HIOCT

```

10669 053432 010246          MOV      R2,-(SP)          ;; PUSH R2 ON STACK
10670 053434 104411          RDLIN                     ;; READ AN ASCIZ LINE
10671 053436 012600          MOV      (SP)+,R0         ;; GET ADDRESS OF 1ST CHARACTER
10672 053440 010037 053544          MOV      R0,$5           ;; AND SAVE IT
10673 053444 005001          CLR      R1              ;; CLEAR DATA WORD
10674 053446 005002          CLR      R2
10675 053450 112046          2$:      MOVB      (R0)+,-(SP) ;; PICKUP THIS CHARACTER
10676 053452 001420          BEQ      3$              ;; IF ZERO GET OUT
10677 053454 122716 000060          CMPB      #'0,(SP)       ;; MAKE SURE THIS CHARACTER
10678 053460 003026          BGT      4$              ;; IS AN OCTAL DIGIT
10679 053462 122716 000067          CMPB      #'7,(SP)
10680 053466 002423          BLT      4$
10681 053470 006301          ASL      R1              ;; *2
10682 053472 006102          ROL      R2
10683 053474 006301          ASL      R1              ;; *4
10684 053476 006102          ROL      R2
10685 053500 006301          ASL      R1              ;; *8
10686 053502 006102          ROL      R2
10687 053504 042716 177770          BIC      #'C7,(SP)       ;; STRIP THE ASCII JUNK
10688 053510 062601          ADD      (SP)+,R1        ;; ADD IN THIS DIGIT
10689 053512 000756          BR       2$              ;; LOOP
10690 053514 005726          3$:      TST      (SP)+      ;; CLEAN TERMINATOR FROM STACK
10691 053516 010166 000012          MOV      R1,12(SP)       ;; SAVE THE RESULT
10692 053522 010237 053554          MOV      R2,$HIOCT
10693 053526 012602          MOV      (SP)+,R2        ;; POP STACK INTO R2
10694 053530 012601          MOV      (SP)+,R1        ;; POP STACK INTO R1
10695 053532 012600          MOV      (SP)+,R0        ;; POP STACK INTO R0
10696 053534 000002          RTI                      ;; RETURN
10697 053536 005726          4$:      TST      (SP)+      ;; CLEAN PARTIAL FROM STACK
10698 053540 105010          CLRB     (R0)            ;; SET A TERMINATOR
10699 053542 104401          TYPE     ;; TYPE UP THRU THE BAD CHAR.
10700 053544 000000          5$:      .WORD     0
10701 053546 104401 001204          TYPE     $QUES          ;; "?" "CR" & "LF"
10702 053552 000730          BR       1$              ;; TRY AGAIN
10703 053554 000000          $HIOCT: .WORD     0      ;; HIGH ORDER BITS GO HERE
10704          .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
10705
10706          ;; *****
10707          ;; *THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
10708          ;; *UNSIGNED OCTAL ASCIZ NUMBER.
10709          ;; *CALL
10710          ;; *      MOV      #PNTR,-(SP)          ;; * POINTER TO LOW WORD OF BINARY NUMBER
10711          ;; *      JSR      PC,2#$SDB20        ;; * CALL THE ROUTINE
10712          ;; *      RETURN                       ;; * THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
10713
10714
10715 053556 104413          $SDB20: SAVREG          ;; * SAVE ALL REGISTERS
10716 053560 016601 000002          MOV      2(SP),R1        ;; * PICKUP THE POINTER TO LOW WORD
10717 053564 012705 053675          MOV      #$OCTVL+13.,R5 ;; * POINTER TO DATA TABLE
10718 053570 012704 000014          MOV      #12.,R4         ;; * DO ELEVEN CHARACTERS
10719 053574 012703 177770          MOV      #'C7,R3         ;; * MASK
10720 053600 012100          MOV      (R1)+,R0        ;; * LOWER WORD
10721 053602 012101          MOV      (R1)+,R1        ;; * HIGH WORD
10722 053604 005002          CLR      R2              ;; * TERMINATOR
10723 053606 110245          1$:      MOVB      R2,-(R5)       ;; * PUT CHARACTER IN DATA TABLE
10724 053610 010002          MOV      R0,R2          ;; * GET THIS DIGIT

```

```

10725 053612 005304          DEC      R4          ;; COUNT THIS CHARACTER
10726 053614 003007          BGT     3$          ;; BR IF NOT THE LAST DIGIT
10727 053616 001405          BEQ     2$          ;; BR IF IT IS THE LAST DIGIT
10728 053620 005205          INC     R5          ;; ALL DIGITS DONE-ADJUST POINTER FOR FIRST
10729 053622 010566 000002  MOV     R5,2(SP)    ;; ASCII CHAR. & PUT IT ON THE STACK
10730 053626 104414          RESREG          ;; RESTORE ALL REGISTERS
10731 053630 000207          RTS     PC          ;; RETURN TO USER
10732 053632 006203          2$:    ASR     R3          ;; POSITION THE MASK FOR THE LAST DIGIT
10733 053634 006001          3$:    ROR     R1          ;; POSITION THE BINARY NUMBER FOR
10734 053636 006000          ROR     R0          ;; THE NEXT OCTAL DIGIT
10735 053640 006001          ROR     R1
10736 053642 006000          ROR     R0
10737 053644 006001          ROR     R1
10738 053646 006000          ROR     R0
10739 053650 040302          BIC     R3,R2      ;; MASK OUT ALL JUNK
10740 053652 062702 000060  ADD     #0,R2      ;; MAKE THIS CHAR. ASCII
10741 053656 000753          BR      1$          ;; GO PUT IT IN THE DATA TABLE
10742 053660 000016          $OCTVL: .BLKB 14.  ;; RESERVE DATA TABLE
10743          .SBTTL SINGLE LENGTH BINARY TO DECIMAL ASCII ROUTINE
10744
10745          ;; *****
10746          ;; THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
10747          ;; UNSIGNED DECIMAL ASCII NUMBER.
10748          ;; CALL
10749          ;; MOV     NUMBER -(SP)  ;; PUT BINARY NUMBER ON THE STACK
10750          ;; JSR     PC,@#$SDB2D  ;; CALL
10751          ;; RETURN  ;; ADDRESS OF THE 1ST ASCII CHAR.IS ON THE STACK
10752
10753          $SDB2D: MOV     2(SP),1$  ;; SAVE BINARY NUMBER
10754 053676 016637 000002 053726 $SDB2D: MOV     #1$ -(SP)  ;; SET POINTER
10755 053704 012746 053726  MOV     PC,@#$SDB2D  ;; CALL DOUBLE LENGTH CONVERT
10756 053710 004737 053732  ADD     #5,(SP)      ;; ONLY ALLOW FIVE CHARACTERS
10757 053714 062716 000005  MOV     (SP)+,2(SP)  ;; PICKUP POINTER
10758 053720 012666 000002  RTS     PC          ;; RETURN
10759 053724 000207 000000  1$:    .WORD 0,0
10760 053726 000000 000000  .SBTTL DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
10761
10762          ;; *****
10763          ;; THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
10764          ;; DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
10765          ;; POSITIVE.
10766          ;; CALL
10767          ;; MOV     #PNTR -(SP)  ;; POINTER TO LOW WORD OF BINARY NUMBER
10768          ;; JSR     PC,@#$SDB2D  ;;
10769          ;; RETURN  ;; THE FIRST ADDRESS OF ASCII
10770          ;; IS ON THE STACK
10771
10772          $SDB2D: SAVREG          ;; SAVE REGISTERS
10773          MOV     2(SP),R2      ;; PICKUP THE DATA POINTER
10774 053732 104413          MOV     #$DECVL,R0    ;; GET ADDRESS OF "$DECVL" STRING
10775 053734 016602 000002  MOV     R0,2(SP)      ;; PUT ADDRESS OF ASCII STRING ON STACK
10776 053740 012700 054112  MOV     (R2)+,R1      ;; PICKUP THE BINARY NUMBER
10777 053744 010066 000002  MOV     (R2)+,R2
10778 053750 012201          MOV     (R2)+,R2
10779 053752 012202          MOV     #10.,4$      ;; SET UP TO DO 10 CONVERSIONS
10780 053754 012737 000012 054030

```

```

10781 053762 012704 054042      MOV      #STNPWR,R4      ;; ADDRESS OF TEN POWER
10782 053766 012705 054044      MOV      #STNPWR+2,R5
10783 053772 005003      1$: CLR      R3          ;; CLEAR PARTIAL
10784 053774 161401      2$: SUB      (R4),R1     ;; SUBTRACT TEN POWER
10785 053776 005602      SBC      R2
10786 054000 161502      SUB      (R5),R2
10787 054002 002402      BLT      3$            ;; BR IF TEN POWER TO LARGE
10788 054004 005203      INC      R3            ;; ADD 1 TO PARTIAL
10789 054006 000772      BR       2$            ;; LOOP
-10790 054010 062401      3$: ADD      (R4)+,R1    ;; RESTORE SUBTRACTED VALUE
10791 054012 005502      ADC      R2
10792 054014 062402      ADD      (R4)+,R2
10793 054016 022525      CMP      (R5)+,(R5)+  ;; MOVE TO NEXT TEN POWER
10794 054020 052703 000060      BIS      #'0,R3        ;; CHANGE PARTIAL TO ASCII
10795 054024 110320      MOV      R3,(R0)+      ;; SAVE IT
10796 054026 005327      DEC      (PC)+         ;; DONE?
10797 054030 000000      4$: .WORD    0
10798 054032 001357      BNE      1$            ;; BR IF NO
10799 054034 105020      CLRB     (R0)+         ;; TERMINATOR
10800 054036 104414      RESREG   ;; RESTORE REGISTERS
10801 054040 000207      RTS      PC            ;; RETURN
10802 054042 145000      STNPWR: 145000         ;; 1.0E09
10803 054044 035632      35632
10804 054046 160400      160400         ;; 1.0E08
10805 054050 002765      2765
10806 054052 113200      113200         ;; 1.0E07
10807 054054 000230      230
10808 054056 041100      041100         ;; 1.0E06
10809 054060 000017      17
10810 054062 103240      103240         ;; 1.0E05
10811 054064 000001      1
10812 054066 023420      23420         ;; 1.0E04
10813 054070 000000      0
10814 054072 001750      1750           ;; 1.0E03
10815 054074 000000      0
10816 054076 000144      144           ;; 1.0E02
10817 054100 000000      0
10818 054102 000012      12           ;; 1.0E01
10819 054104 000000      0
10820 054106 000001      1           ;; 1.0E00
10821 054110 000000      0
10822 054112 000014      $DECVL: .BLKB 12     ;; RESERVE STORAGE FOR ASCIZ STRING
10823      .SBTTL TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
10824
10825      ;; *****
10826      ;; *THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
10827      ;; *LEADING NUMBERS.
10828      ;; *CALL
10829      ;; *      MOV      #NUMADR,-(SP)  ;; FIRST ADDRESS OF ASCIZ STRING
10830      ;; *      JSR      PC,@$$SUPRS
10831
10832
10833 054126 010046 000004      $$SUPRS: MOV      R0,-(SP)  ;; SAVE R0
10834 054130 016600      MOV      4(SP),R0      ;; PICKUP THE POINTER
10835 054134 105710 1$: TSTB     (R0)         ;; TERMINATE OR?
10836 054136 001403      BEQ      2$            ;; BR IF YES

```



```

10837 054140 122720 000060
10838 054144 001773
10839 054146 005300
10840 054150 010037 054156
10841 054154 104401
10842 054156 000000
10843 054160 012600
10844 054162 012616
10845 054164 000207
10846
10847
10848
10849
10850
10851
10852
10853
10854
10855
10856
10857
10858
10859
10860
10861
10862
10863 054166
10864 054166 010046
10865 054170 010146
10866 054172 010246
10867 054174 010346
10868 054176 010446
10869 054200 010546
10870 054202 016646 000022
10871 054206 016646 000022
10872 054212 016646 000022
10873 054216 016646 000022
10874 054222 000002
10875
10876
10877
10878
10879 054224
10880 054224 012666 000022
10881 054230 012666 000022
10882 054234 012666 000022
10883 054240 012666 000022
10884 054244 012605
10885 054246 012604
10886 054250 012603
10887 054252 012602
10888 054254 012601
10889 054256 012600
10890 054260 000002
10891
10892

```

```

          CMPB    #'0,(R0)+      ;; IS THIS AN ASCII "0" ?
          BEQ     1$              ;; BR IF YES
2$:      DEC     R0              ;; BACKUP BY "1"
          MOV     R0,3$          ;; SAVE FOR TYPING
          TYPE    GO TYPE       ;; GO TYPE
3$:      .WORD   0              ;; ASCIZ POINTER GOES HERE
          MOV     (SP)+,R0       ;; RESTORE R0
          MOV     (SP)+,(SP)    ;; RESTORE THE STACK
          RTS     PC            ;; RETURN
.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:
          MOV     R0,-(SP)       ;; PUSH R0 ON STACK
          MOV     R1,-(SP)       ;; PUSH R1 ON STACK
          MOV     R2,-(SP)       ;; PUSH R2 ON STACK
          MOV     R3,-(SP)       ;; PUSH R3 ON STACK
          MOV     R4,-(SP)       ;; PUSH R4 ON STACK
          MOV     R5,-(SP)       ;; PUSH R5 ON STACK
          MOV     22(SP),-(SP)   ;; SAVE PS OF MAIN FLOW
          MOV     22(SP),-(SP)   ;; SAVE PC OF MAIN FLOW
          MOV     22(SP),-(SP)   ;; SAVE PS OF CALL
          MOV     22(SP),-(SP)   ;; SAVE PC OF CALL
          RTI

; *RESTORE R0-R5
; *CALL:
; * RESREG
$RESREG:
          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
          RTI
.SBTTL   TRAP DECODER

```

10893
10894
10895
10896
10897
10898
10899 054262 010046
10900 054264 016600 000002
10901 054270 005740
10902 054272 111000
10903 054274 006300
10904 054276 016000 054316
10905 054302 000200
10906
10907
10908
10909
10910 054304 011646
10911 054306 016666 000004 000002
10912 054314 000002
10913
10914
10915
10916
10917
10918
10919
10920
10921 054316 054304
10922 054320 050744
10923 054322 051744
10924 054324 051720
10925 054326 051760
10926 054330 051226
10927
10928 054332 052534
10929
10930 054334 052444
10931 054336 053006
10932 054340 053076
10933 054342 053416
10934 054344 054166
10935 054346 054224
10936 054350 047376
10937

```

*****
;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.

```

```

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

```

```
;; THIS IS USE TO HANDLE THE "GETPRI" MACRO
```

```

$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.

```

```

; ROUTINE
; -----
$TRPAD: .WORD  $TRAP2           TRAP+1(104401)  TTY TYPEOUT ROUTINE
        $TYPE  ;; CALL=TYPE     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOC ;; CALL=TYPOC   TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPOS ;; CALL=TYPOS   TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPON ;; CALL=TYPON   TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
        $TYPDS ;; CALL=TYPDS
        $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
        $SCOPI  ;; CALL=SCOPI  TRAP+15(104415) INTERNAL LOOP ON ERROR

```

10938
10939
10940
10941 054352 000062
10942
10943
10944 054516 005015 047125 041111
10945 054524 051525 051040 030113
10946 054532 026466 045522 033460
10947 054540 042040 044522 042526
10948 054546 042040 040511 047107
10949 054554 051517 044524 026503
10950 054562 050055 051101 020124
10951 054570 061
10952 054571 015 041412 051132
10953 054576 044066 030105 005015
10954 054604 005015 025011 025052
10955 054612 041440 052501 044524
10956 054620 047117 025040 025052
10957 054626 005015
10958 054630 005015 044124 051511
10959 054636 050040 047522 051107
10960 054644 046501 051440 047510
10961 054652 046125 020104 042502
10962 054660 044040 046101 042524
10963 054666 020104 047117 054514
10964 054674 041040 020131 054524
10965 054702 044520 043516 041440
10966 054710 047117 051124 046117
10967 054716 041455
10968 054720 005015 052117 042510
10969 054726 053522 051511 026105
10970 054734 041440 051101 051124
10971 054742 042111 042507 043040
10972 054750 051117 040515 052124
10973 054756 047111 020107 047101
10974 054764 026104 051117 052040
10975 054772 042510 042040 044522
10976 055000 042526
10977 055002 005015 040515 020131
10978 055010 042502 046040 043105
10979 055016 020124 047111 040440
10980 055024 020116 047125 042504
10981 055032 042524 046522 047111
10982 055040 042105 051440 040524
10983 055046 042524 005015
10984 055052 005015 047111 052111
10985 055060 040511 046114 026131
10986 055066 042040 044522 042526
10987 055074 020123 047524 041040
10988 055102 020105 042524 052123
10989 055110 042105 051440 047510
10990 055116 046125 020104 040510
10991 055124 042526 006472 012
10992 055131 015 040412 020056
10993 055136 044040 040505 051504

.SBTTL SERVICE MSGS

.EVEN

BSE22S: .BLKW 50.

;22 SECTOR SOFTWARE INFO

MSG1: .ASCII <CR><LF>/UNIBUS RK06-RK07 DRIVE DIAGNOSTIC--PART 1/

.ASCII <CR><LF>/CZR6HED/<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/<CR><LF>

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

.ASCII <CR><LF>/A., HEADS MANUALLY LOADED/

10994	055144	046440	047101	040525	
10995	055152	046114	020131	047514	
10996	055160	042101	042105		
10997	055164	005015	027102	020040	.ASCII <CR><LF>/B. CORRECT PORT SELECTED/
10998	055172	047503	051122	041505	
10999	055200	020124	047520	052122	
11000	055206	051440	046105	041505	
11001	055214	042524	104		
11002	055217	015	041412	020056	.ASCII <CR><LF>/C. WRITE LOCK DISABLED/
11003	055224	053440	044522	042524	
11004	055232	046040	041517	020113	
11005	055240	044504	040523	046102	
11006	055246	042105			
11007	055250	005015	027104	020040	.ASCII <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
11008	055256	051104	053111	020105	
11009	055264	042522	042101	020131	
11010	055272	047111	044504	040503	
11011	055300	047524	020122	047117	
11012	055306	005015			
11013	055310	005015	051104	053111	.ASCII <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE BOTH/
11014	055316	051505	047040	052117	
11015	055324	052040	020117	042502	
11016	055332	052040	051505	042524	
11017	055340	020104	052515	052123	
11018	055346	044040	053101	020105	
11019	055354	047502	044124		
11020	055360	005015	047520	052122	.ASCIZ <CR><LF>/PORTS DESELECTED/<CR><LF>
11021	055366	020123	042504	042523	
11022	055374	042514	052103	042105	
11023	055402	005015	000		
11024	055405	015	041412	040510	MSG2: .ASCII <CR><LF>/CHANGE XXDP PACK/
11025	055412	043516	020105	054130	
11026	055420	050104	050040	041501	
11027	055426	113			
11028	055427	015	041412	042514	.ASCIZ <CR><LF>/CLEAR LOC 40,RESTART PROGRAM/
11029	055434	051101	046040	041517	
11030	055442	032040	026060	042522	
11031	055450	052123	051101	020124	
11032	055456	051120	043517	040522	
11033	055464	000115			
11034	055466	005015	051104	053111	MSG3: .ASCIZ <CR><LF>/DRIVE(S) TO BE TESTED: /
11035	055474	024105	024523	052040	
11036	055502	020117	042502	052040	
11037	055510	051505	042524	035104	
11038	055516	000040			
11039	055520	005015	052502	020123	MSG4: .ASCIZ <CR><LF>/BUS ADDR (177440): /
11040	055526	042101	051104	024040	
11041	055534	033461	032067	030064	
11042	055542	035051	000040		
11043	055546	005015	047503	052116	MSG5: .ASCIZ <CR><LF>/CONTR ADDR (210): /
11044	055554	020122	042101	051104	
11045	055562	024040	030462	024460	
11046	055570	020072	000		
11047	055573	015	044412	052116	MSG6: .ASCIZ <CR><LF>/INTR AT PC=/
11048	055600	020122	052101	050040	
11049	055606	036503	000		

11050	055611	015	042012	044522	MSG7:	.ASCIZ	<CR><LF>/DRIVE 0 WILL NOT BE TESTED/
11051	055616	042526	030040	053440			
11052	055624	046111	020114	047516			
11053	055632	020124	042502	052040			
11054	055640	051505	042524	000104			
11055	055646	005015	042524	052123	MSG8:	.ASCIZ	<CR><LF>/TEST 16 TAKES 2 TO 4 MIN./<CR><LF>
11056	055654	030440	020066	040524			
11057	055662	042513	020123	020062			
11058	055670	047524	032040	046440			
11059	055676	047111	006456	000012			
11060	055704	005015	054502	040520	MSG9:	.ASCIZ	<CR><LF>/BYPASSING TEST 16/<CR><LF>
11061	055712	051523	047111	020107			
11062	055720	042524	052123	030440			
11063	055726	006466	000012				
11064	055732	005015	053412	046111	MSG10:	.ASCIZ	<CR><LF><LF>/WILL TEST DRIVE(S):/
11065	055740	020114	042524	052123			
11066	055746	042040	044522	042526			
11067	055754	051450	035051	000			
11068	055761	015	005012	047520	MSG11:	.ASCIZ	<CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
11069	055766	042527	020122	050125			
11070	055774	051040	051505	040524			
11071	056002	052122	052040	020117			
11072	056010	042524	052123	030440			
11073	056016	005015	000				
11074	056021	015	050012	041501	MSG12:	.ASCIZ	<CR><LF>/PACK BEING FORMATTED/<CR><LF>
11075	056026	020113	042502	047111			
11076	056034	020107	047506	046522			
11077	056042	052101	042524	006504			
11078	056050	000012					
11079	056052	005015	047516	046040	MSG13:	.ASCII	<CR><LF>/NO L OR P CLOCKS/
11080	056060	047440	020122	020120			
11081	056066	046103	041517	051513			
11082	056074	005015	046101	020114		.ASCIZ	<CR><LF>/ALL TIMING TESTS BYPASSED/
11083	056102	044524	044515	043516			
11084	056110	052040	051505	051524			
11085	056116	041040	050131	051501			
11086	056124	042523	000104				
11087	056130	005015	054502	040520	MSG14:	.ASCIZ	<CR><LF>/BYPASSING DRIVE /
11088	056136	051523	047111	020107			
11089	056144	051104	053111	020105			
11090	056152	000					
11091	056153	015	005012	051104	MSG15:	.ASCIZ	<CR><LF><LF>/DRIVE /
11092	056160	053111	020105	000			
11093	056165	015	042012	044522	MSG16:	.ASCIZ	<CR><LF>/DRIVE SERIAL #/
11094	056172	042526	051440	051105			
11095	056200	040511	020114	000043			
11096	056206	005015	040503	052122	MSG17:	.ASCIZ	<CR><LF>/CARTRIDGE SERIAL NO./
11097	056214	044522	043504	020105			
11098	056222	042523	044522	046101			
11099	056230	047040	027117	000			
11100	056235	015	005012	041101	MSG18:	.ASCIZ	<CR><LF><LF>/ABORTING BAL OF TESTS/<CR><LF><LF>
11101	056242	051117	044524	043516			
11102	056250	041040	046101	047440			
11103	056256	020106	042524	052123			
11104	056264	006523	005012	000			
11105	056271	015	005012	046101	MSG19:	.ASCIZ	<CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>

11106	056276	020114	051104	053111	
11107	056304	051505	052040	051505	
11108	056312	042524	006504	005012	
11109	056320	000			
11110	056321	015	046412	042117	MSG20: .ASCII <CR><LF>/MODIFIED VERSION OF FORMAT PACK TEST FOR MODULE TESTING/
11111	056326	043111	042511	020104	
11112	056334	042526	051522	047511	
11113	056342	020116	043117	043040	
11114	056350	051117	040515	020124	
11115	056356	040520	045503	052040	
11116	056364	051505	020124	047506	
11117	056372	020122	047515	052504	
11118	056400	042514	052040	051505	
11119	056406	044524	043516		
11120	056412	005015	047524	051040	.ASCIZ <CR><LF>/TO RESTORE HEADERS ON CYL 0 & 1, ALL TRACKS/
11121	056420	051505	047524	042522	
11122	056426	044040	040505	042504	
11123	056434	051522	047440	020116	
11124	056442	054503	020114	020060	
11125	056450	020046	026061	040440	
11126	056456	046114	052040	040522	
11127	056464	045503	000123		
11128	056470	005015	054502	040520	MSG21: .ASCIZ <CR><LF>/BYPASSING TESTS 36,40,41 FOR MODULE TESTING/<CR><LF>
11129	056476	051523	047111	020107	
11130	056504	042524	052123	020123	
11131	056512	033063	032054	026060	
11132	056520	030464	043040	051117	
11133	056526	046440	042117	046125	
11134	056534	020105	042524	052123	
11135	056542	047111	006507	000012	
11136	056550	005015	047506	046522	MSG22: .ASCIZ <CR><LF>/FORMATTING FINISHED/<CR><LF>
11137	056556	052101	044524	043516	
11138	056564	043040	047111	051511	
11139	056572	042510	006504	000012	
11140	056600	005015	042524	052123	MSG23: .ASCIZ <CR><LF>/TEST 16 TAKES 10 TO 12 MIN./<CR><LF>
11141	056606	030440	020066	040524	
11142	056614	042513	020123	030061	
11143	056622	052040	020117	031061	
11144	056630	046440	047111	006456	
11145	056636	000012			
11146	056640	005015	043520	020115	MSG74: .ASCIZ <CR><LF>/PGM ABORT PENDING.../
11147	056646	041101	051117	020124	
11148	056654	042520	042116	047111	
11149	056662	027107	027056	000	
11150	056667	015	044012	046101	MSG75: .ASCIZ <CR><LF>/HALT PENDING.../
11151	056674	020124	042520	042116	
11152	056702	047111	027107	027056	
11153	056710	000			
11154	056711	015	050012	046507	MSG76: .ASCIZ <CR><LF>/PGM ABORTED/
11155	056716	040440	047502	052122	
11156	056724	042105	000		
11157	056727	015	041412	052520	MSG77: .ASCIZ <CR><LF>/CPU HALTED/
11158	056734	044040	046101	042524	
11159	056742	000104			
11160					
11161					

```

11162 .SBTTL ERR MSGS
11163
11164 056744 005015 051105 026122 EM1: .ASCIZ <CR><LF>/ERR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
11165 056752 047440 046116 020131
11166 056760 020060 044124 052522
11167 056766 033440 040440 046114
11168 056774 053517 042105 020054
11169 057002 051124 020131 043501
11170 057010 044501 006516 000012
11171 057016 042523 042514 052103 EM2: .ASCIZ /SELECTED DRIVE # IN RKCS2 CANNOT BE READ BACK IN RKMR2/
11172 057024 042105 042040 044522
11173 057032 042526 021440 044440
11174 057040 020116 045522 051503
11175 057046 020062 040503 047116
11176 057054 052117 041040 020105
11177 057062 042522 042101 041040
11178 057070 041501 020113 047111
11179 057076 051040 046513 031122
11180 057104 000
11181 057105 015 040412 047502 EM3: .ASCIZ <CR><LF>/ABORT TESTS...UNEXP TIME OUT AT PC=/
11182 057112 052122 052040 051505
11183 057120 051524 027056 052456
11184 057126 042516 050130 052040
11185 057134 046511 020105 052517
11186 057142 020124 052101 050040
11187 057150 036503 000
11188 057153 106 052101 046101 EM4: .ASCII /FATAL ERROR/<CR><LF>
11189 057160 042440 051122 051117
11190 057166 005015
11191 057170 041101 051117 044524 .ASCII /ABORTING BALANCE OF TESTS/<CR><LF>
11192 057176 043516 041040 046101
11193 057204 047101 042503 047440
11194 057212 020106 042524 052123
11195 057220 006523 012
11196 057223 110 040505 020104 .ASCIZ /HEAD POSITION CANNOT BE DETERMINED/
11197 057230 047520 044523 044524
11198 057236 047117 041440 047101
11199 057244 047516 020124 042502
11200 057252 042040 052105 051105
11201 057260 044515 042516 000104
11202 057266 042115 020123 042523 EM5: .ASCIZ /MDS SET IN RKCS2/
11203 057274 020124 047111 051040
11204 057302 041513 031123 000
11205 057307 125 042506 051440 EM6: .ASCIZ /UFE SET IN RKCS2/
11206 057314 052105 044440 020116
11207 057322 045522 051503 000062
11208 057330 051104 020101 047111 EM7: .ASCIZ /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
11209 057336 051040 042113 020123
11210 057344 020046 042516 020104
11211 057352 047111 051040 041513
11212 057360 031123 051040 051505
11213 057366 052105 020073 051127
11214 057374 047117 020107 047520
11215 057402 052122 051440 046105
11216 057410 041505 042524 037504
11217 057416 000

```

11218	057417	104	053122	052040	EM8:	.ASCIZ /DRV TYPE NOT SET IN RKMR2/
11219	057424	050131	020105	047516		
11220	057432	020124	042523	020124		
11221	057440	047111	051040	046513		
11222	057446	031122	000			
11223	057451	104	044522	042526	EM9:	.ASCIZ /DRIVE NOT PRESENT BUT TYPED BY OPERATOR/
11224	057456	047040	052117	050040		
11225	057464	042522	042523	052116		
11226	057472	041040	052125	052040		
11227	057500	050131	042105	041040		
11228	057506	020131	050117	051105		
11229	057514	052101	051117	000		
11230	057521	101	047502	052122	EM10:	.ASCIZ /ABORT TESTS...CANNOT REF CONTR REG/
11231	057526	052040	051505	051524		
11232	057534	027056	041456	047101		
11233	057542	047516	020124	042522		
11234	057550	020106	047503	052116		
11235	057556	020122	042522	000107		
11236	057564	051104	020101	047111	EM11:	.ASCIZ /DRA IN RKDS & NED IN RKCS2 BOTH SET/
11237	057572	051040	042113	020123		
11238	057600	020046	042516	020104		
11239	057606	047111	051040	041513		
11240	057614	031123	041040	052117		
11241	057622	020110	042523	000124		
11242	057630	047503	052116	020122	EM12:	.ASCIZ /CONTR NOT READY IN RKCS1/
11243	057636	047516	020124	042522		
11244	057644	042101	020131	047111		
11245	057652	051040	041513	030523		
11246	057660	000				
11247	057661	116	020117	052101	EM13:	.ASCIZ /NO ATTN IN RKASOF/
11248	057666	047124	044440	020116		
11249	057674	045522	051501	043117		
11250	057702	000				
11251	057703	127	047522	043516	EM14:	.ASCIZ /WRONG ATTN IN RKASOF/
11252	057710	040440	052124	020116		
11253	057716	047111	051040	040513		
11254	057724	047523	000106			
11255	057730	051104	054504	047040	EM15:	.ASCIZ /DRDY NOT CLEARED IN RKMR2/
11256	057736	052117	041440	042514		
11257	057744	051101	042105	044440		
11258	057752	020116	045522	051115		
11259	057760	000062				
11260	057762	051504	020103	047516	EM16:	.ASCIZ /DSC NOT SET IN RKMR2/
11261	057770	020124	042523	020124		
11262	057776	047111	051040	046513		
11263	060004	031122	000			
11264	060007	115	043523	040440	EM17:	.ASCIZ /MSG A0 ERROR/
11265	060014	020060	051105	047522		
11266	060022	000122				
11267	060024	051515	020107	030102	EM18:	.ASCIZ /MSG B0 ERROR/
11268	060032	042440	051122	051117		
11269	060040	000				
11270	060041	115	043523	040440	EM19:	.ASCIZ /MSG A1 ERROR/
11271	060046	020061	051105	047522		
11272	060054	000122				
11273	060056	051515	020107	030502	EM20:	.ASCIZ /MSG B1 ERROR/

IO1

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 215
ERR MSGS

SEQ 0215

11274	060064	042440	051122	051117		
11275	060072	000				
11276	060073	103	051105	020122	EM21:	.ASCIZ /CERR SET IN RKCS1/
11277	060100	042523	020124	047111		
11278	060106	051040	041513	030523		
11279	060114	000				
11280	060115	122	051514	044440	EM22:	.ASCIZ /RLS IN RKCS2 SET CERR IN RKCS1/
11281	060122	020116	045522	051503		
11282	060130	020062	042523	020124		
11283	060136	042503	051122	044440		
11284	060144	020116	045522	051503		
11285	060152	000061				
11286	060154	043125	020105	047111	EM23:	.ASCIZ /UFE IN RKCS2 SET (SACK) AFTER RLS IN RKCS2 SENT/
11287	060162	051040	041513	031123		
11288	060170	051440	052105	024040		
11289	060176	040523	045503	020051		
11290	060204	043101	042524	020122		
11291	060212	046122	020123	047111		
11292	060220	051040	041513	031123		
11293	060226	051440	047105	000124		
11294	060234	053126	047040	052117	EM24:	.ASCIZ /VV NOT SET IN RKMR2/
11295	060242	051440	052105	044440		
11296	060250	020116	045522	051115		
11297	060256	000062				
11298	060260	051104	020126	054524	EM25:	.ASCIZ /DRV TYPE SET IN RKMR2/
11299	060266	042520	051440	052105		
11300	060274	044440	020116	045522		
11301	060302	051115	000062			
11302	060306	042104	020124	042523	EM26:	.ASCIZ /DDT SET IN RKDS/
11303	060314	020124	047111	051040		
11304	060322	042113	000123			
11305	060326	052104	042531	051440	EM27:	.ASCIZ /DTYE SET IN RKER/
11306	060334	052105	044440	020116		
11307	060342	045522	051105	000		
11308	060347	104	054524	020105	EM28:	.ASCIZ /DTYE NOT SET IN RKER AFTER WRONG CDT IN RKCS1/
11309	060354	047516	020124	042523		
11310	060362	020124	047111	051040		
11311	060370	042513	020122	043101		
11312	060376	042524	020122	051127		
11313	060404	047117	020107	042103		
11314	060412	020124	047111	051040		
11315	060420	041513	030523	000		
11316	060425	104	054524	020105	EM29:	.ASCIZ /DTYE IN RKER DID NOT SET CERR IN RKCS1/
11317	060432	047111	051040	042513		
11318	060440	020122	044504	020104		
11319	060446	047516	020124	042523		
11320	060454	020124	042503	051122		
11321	060462	044440	020116	045522		
11322	060470	051503	000061			
11323	060474	026503	020104	040520	EM30:	.ASCIZ /C-D PAR ERR SET IN RKMR3/
11324	060502	020122	051105	020122		
11325	060510	042523	020124	047111		
11326	060516	051040	046513	031522		
11327	060524	000				
11328	060525	104	041455	050040	EM31:	.ASCIZ /D-C PAR SET IN RKCS1/
11329	060532	051101	051440	052105		

11330	060540	044440	020116	045522		
11331	060546	051503	000061			
11332	060552	046106	020124	047516	EM32:	.ASCIZ /FLT NOT SET IN RKMR3/
11333	060560	020124	042523	020124		
11334	060566	047111	051040	046513		
11335	060574	031522	000			
11336	060577	103	042055	050040	EM33:	.ASCIZ /C-D PAR ERR NOT SET IN RKMR3/
11337	060604	051101	042440	051122		
11338	060612	047040	052117	051440		
11339	060620	052105	044440	020116		
11340	060626	045522	051115	000063		
11341	060634	026504	020103	040520	EM34:	.ASCIZ /D-C PAR NOT SET IN RKCS1/
11342	060642	020122	047516	020124		
11343	060650	042523	020124	047111		
11344	060656	051040	041513	030523		
11345	060664	000				
11346	060665	104	041455	050040	EM35:	.ASCIZ /D-C PAR IN RKCS1 DID NOT SET CERR IN RKCS1/
11347	060672	051101	044440	020116		
11348	060700	045522	051503	020061		
11349	060706	044504	020104	047516		
11350	060714	020124	042523	020124		
11351	060722	042503	051122	044440		
11352	060730	020116	045522	051503		
11353	060736	000061				
11354	060740	054503	020114	042101	EM36:	.ASCIZ /CYL ADDR IN B2 NOT SAME AS RKDC/
11355	060746	051104	044440	020116		
11356	060754	031102	047040	052117		
11357	060762	051440	046501	020105		
11358	060770	051501	051040	042113		
11359	060776	000103				
11360	061000	054503	020114	044504	EM37:	.ASCIZ /CYL DIFF IN A2 NOT SAME AS RKDC/
11361	061006	043106	044440	020116		
11362	061014	031101	047040	052117		
11363	061022	051440	046501	020105		
11364	061030	051501	051040	042113		
11365	061036	000103				
11366	061040	054503	020114	044504	EM38:	.ASCIZ /CYL DIFF IN RKMR2 NOT SAME AS 'CYL DIFF'/
11367	061046	043106	044440	020116		
11368	061054	045522	051115	020062		
11369	061062	047516	020124	040523		
11370	061070	042515	040440	020123		
11371	061076	041447	046131	042040		
11372	061104	043111	023506	000		
11373	061111	103	046131	042040	EM39:	.ASCIZ /CYL DIFF & OFST IN A2 NOT =0/
11374	061116	043111	020106	020046		
11375	061124	043117	052123	044440		
11376	061132	020116	031101	047040		
11377	061140	052117	036440	000060		
11378	061146	054503	020114	042101	EM40:	.ASCIZ /CYL ADDR IN B2 NOT =0/
11379	061154	051104	044440	020116		
11380	061162	031102	047040	052117		
11381	061170	036440	000060			
11382	061174	054503	020114	042101	EM41:	.ASCIZ /CYL ADDR IN B2 DID NOT REMAIN =0/
11383	061202	051104	044440	020116		
11384	061210	031102	042040	042111		
11385	061216	047040	052117	051040		

K01

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 217
ERR MSGS

SEQ 0217

11386	061224	046505	044501	020116		
11387	061232	030075	000			
11388	061235	114	046511	052111	EM42:	.ASCIZ /LIMIT DET NOT SET IN B1/
11389	061242	042040	052105	047040		
11390	061250	052117	051440	052105		
11391	061256	044440	020116	030502		
11392	061264	000				
11393	061265	110	040505	020104	EM43:	.ASCIZ /HEAD ADDR IN B3 NOT =0/
11394	061272	042101	051104	044440		
11395	061300	020116	031502	047040		
11396	061306	052117	036440	000060		
11397	061314	042510	042101	042040	EM44:	.ASCIZ /HEAD DECODE IN B3 INCORRECT/
11398	061322	041505	042117	020105		
11399	061330	047111	041040	020063		
11400	061336	047111	047503	051122		
11401	061344	041505	000124			
11402	061350	051104	053111	020105	EM45:	.ASCII /DRIVE READY IN RKMR2 NOT SET BY 1 SEC FROM FWD/
11403	061356	042522	042101	020131		
11404	061364	047111	051040	046513		
11405	061372	031122	047040	052117		
11406	061400	051440	052105	041040		
11407	061406	020131	020061	042523		
11408	061414	020103	051106	046517		
11409	061422	043040	042127			
11410	061426	005015	047111	051040		.ASCIZ <CR><LF>/IN RTZ PORTION OF START SPIN CMD/
11411	061434	055124	050040	051117		
11412	061442	044524	047117	047440		
11413	061450	020106	052123	051101		
11414	061456	020124	050123	047111		
11415	061464	041440	042115	000		
11416	061471	115	043523	040440	EM46:	.ASCIZ /MSG A2 ERR/
11417	061476	020062	051105	000122		
11418	061504	051515	020107	031102	EM47:	.ASCIZ /MSG B2 ERR/
11419	061512	042440	051122	000		
11420	061517	115	043523	041040	EM48:	.ASCIZ /MSG B3 ERR/
11421	061524	020063	051105	000122		
11422	061532	053506	020104	047516	EM49:	.ASCIZ /FWD NOT SET IN RKMR2 IN RTZ PORTION OF START SPIN CMD/
11423	061540	020124	042523	020124		
11424	061546	047111	051040	046513		
11425	061554	031122	044440	020116		
11426	061562	052122	020132	047520		
11427	061570	052122	047511	020116		
11428	061576	043117	051440	040524		
11429	061604	052122	051440	044520		
11430	061612	020116	046503	000104		
11431	061620	053506	020104	047516	EM50:	.ASCIZ /FWD NOT SET IN RKMR2 FROM START SPIN CMD/
11432	061626	020124	042523	020124		
11433	061634	047111	051040	046513		
11434	061642	031122	043040	047522		
11435	061650	020115	052123	051101		
11436	061656	020124	050123	047111		
11437	061664	041440	042115	000		
11438	061671	106	042127	047040	EM51:	.ASCIZ /FWD NOT CLEARED IN RKMR2 BY 5 SEC OF MOTION FROM START SPIN CMD/
11439	061676	052117	041440	042514		
11440	061704	051101	042105	044440		
11441	061712	020116	045522	051115		

11442	061720	020062	054502	032440	
11443	061726	051440	041505	047440	
11444	061734	020106	047515	044524	
11445	061742	047117	043040	047522	
11446	061750	020115	052123	051101	
11447	061756	020124	050123	047111	
11448	061764	041440	042115	000	
11449	061771	062	020060	042523	EM52: .ASCIZ /20 SEC FORMAT NOT SET IN RKMR2/
11450	061776	020103	047506	046522	
11451	062004	052101	047040	052117	
11452	062012	051440	052105	044440	
11453	062020	020116	045522	051115	
11454	062026	000062			
11455	062030	042523	020103	020060	EM53: .ASCIZ /SEC 0 NOT FOUND BY 50 MS/
11456	062036	047516	020124	047506	
11457	062044	047125	020104	054502	
11458	062052	032440	020060	051515	
11459	062060	000			
11460	062061	104	043111	020106	EM54: .ASCIZ /DIFF SEC NOT FOUND BY 3 MS/
11461	062066	042523	020103	047516	
11462	062074	020124	047506	047125	
11463	062102	020104	054502	031440	
11464	062110	046440	000123		
11465	062114	052101	047124	047040	EM55: .ASCIZ /ATTN NOT CLEARED IN RKASOF/
11466	062122	052117	041440	042514	
11467	062130	051101	042105	044440	
11468	062136	020116	045522	051501	
11469	062144	043117	000		
11470	062147	125	042516	050130	EM56: .ASCIZ /UNEXP MEM PAR TRAP/
11471	062154	046440	046505	050040	
11472	062162	051101	052040	040522	
11473	062170	000120			
11474	062172	041527	020105	020100	EM57: .ASCIZ /WCE @ CYL 411, TRK 2, SEC 2/
11475	062200	054503	020114	030464	
11476	062206	026061	052040	045522	
11477	062214	031040	020054	042523	
11478	062222	020103	000062		
11479	062226	005015	050123	042505	EM58: .ASCIZ <CR><LF>/SPEED OK IN RKMR2 NOT =0 BY TIMEOUT/
11480	062234	020104	045517	044440	
11481	062242	020116	045522	051115	
11482	062250	020062	047516	020124	
11483	062256	030075	041040	020131	
11484	062264	044524	042515	052517	
11485	062272	000124			
11486	062274	044514	020115	042504	EM59: .ASCIZ /LIM DET NOT SET IN RKMR3/
11487	062302	020124	047516	020124	
11488	062310	042523	020124	047111	
11489	062316	051040	046513	031522	
11490	062324	000			
11491	062325	110	040505	051504	EM60: .ASCIZ /HEADS HOME NOT SET IN RKMR2/
11492	062332	044040	046517	020105	
11493	062340	047516	020124	042523	
11494	062346	020124	047111	051040	
11495	062354	046513	031122	000	
11496	062361	114	040517	020104	EM61: .ASCIZ /LOAD HEADS NOT SET IN RKMR2/
11497	062366	042510	042101	020123	

11498	062374	047516	020124	042523	
11499	062402	020124	047111	051040	
11500	062410	046513	031122	000	
11501	062415	104	052114	051440	EM63: .ASCIZ /DLT SET IN RKCS2/
11502	062422	052105	044440	020116	
11503	062430	045522	051503	000062	
11504	062436	051515	020107	031502	EM64: .ASCIZ /MSG B3 HEAD REG NOT =0/
11505	062444	044040	040505	020104	
11506	062452	042522	020107	047516	
11507	062460	020124	030075	000	
11508	062465	122	040505	020104	EM65: .ASCIZ /READ HEADER ERR/
11509	062472	042510	042101	051105	
11510	062500	042440	051122	000	
11511	062505	103	046131	040440	EM66: .ASCIZ /CYL ADDR IN RKMR3 INCORRECT/
11512	062512	042104	020122	047111	
11513	062520	051040	046513	031522	
11514	062526	044440	041516	051117	
11515	062534	042522	052103	000	
11516	062541	122	040505	044504	EM67: .ASCIZ /READING CYL 0 HEADERS ON CYL 1/
11517	062546	043516	041440	046131	
11518	062554	030040	044040	040505	
11519	062562	042504	051522	047440	
11520	062570	020116	054503	020114	
11521	062576	000061			
11522	062600	042522	042101	047111	EM68: .ASCIZ /READING CYL 1 HEADERS ON CYL 0/
11523	062606	020107	054503	020114	
11524	062614	020061	042510	042101	
11525	062622	051105	020123	047117	
11526	062630	041440	046131	030040	
11527	062636	000			
11528	062637	101	044514	047107	EM69: .ASCIZ /ALIGN CART USED/
11529	062644	041440	051101	020124	
11530	062652	051525	042105	000	
11531	062657	125	042516	050130	EM70: .ASCIZ /UNEXP ATTN/
11532	062664	040440	052124	000116	
11533	062672	051504	020103	042523	EM71: .ASCIZ /DSC SET IN RKMR2/
11534	062700	020124	047111	051040	
11535	062706	046513	031122	000	
11536	062713	106	051117	040515	EM72: .ASCIZ /FORMAT TEST BYPASSED/
11537	062720	020124	042524	052123	
11538	062726	041040	050131	051501	
11539	062734	042523	000104		
11540	062740	052103	020117	042523	EM73: .ASCIZ /CTO SET IN RKCS1/
11541	062746	020124	047111	051040	
11542	062754	041513	030523	000	
11543	062761	122	055124	047040	EM74: .ASCIZ /RTZ NOT SET IN RKMR2/
11544	062766	052117	051440	052105	
11545	062774	044440	020116	045522	
11546	063002	051115	000062		
11547	063006	042111	042501	047040	EM75: .ASCIZ /IDAE NOT SET IN RKMR3/
11548	063014	052117	051440	052105	
11549	063022	044440	020116	045522	
11550	063030	051115	000063		
11551	063034	044520	020120	042523	EM76: .ASCIZ /PIP SET IN RKMR2/
11552	063042	020124	047111	051040	
11553	063050	046513	031122	000	

11554	063055	106	052501	052114	EM77:	.ASCIZ /FAULT NOT =0 IN RKMR3/
11555	063062	047040	052117	036440		
11556	063070	020060	047111	051040		
11557	063076	046513	031522	000		
11558	063103	103	046131	042040	EM78:	.ASCIZ /CYL DIFF IN RKMR2 DID NOT REMAIN = 1 IN SEEK TO SELF/
11559	063110	043111	020106	047111		
11560	063116	051040	046513	031122		
11561	063124	042040	042111	047040		
11562	063132	052117	051040	046505		
11563	063140	044501	020116	020075		
11564	063146	020061	047111	051440		
11565	063154	042505	020113	047524		
11566	063162	051440	046105	000106		
11567	063170	042516	020104	042523	EM79:	.ASCIZ /NED SET IN RKCS2/
11568	063176	020124	047111	051040		
11569	063204	041513	031123	000		
11570	063211	125	046116	040517	EM80:	.ASCIZ /UNLOAD NOT SET IN RKMR2/
11571	063216	020104	047516	020124		
11572	063224	042523	020124	047111		
11573	063232	051040	046513	031122		
11574	063240	000				
11575	063241	123	044520	020116	EM81:	.ASCIZ /SPIN NOT SET IN RKMR2/
11576	063246	047516	020124	042523		
11577	063254	020124	047111	051040		
11578	063262	046513	031122	000		
11579	063267	122	055124	047040	EM82:	.ASCIZ /RTZ NOT SET IN RKMR2/
11580	063274	052117	051440	052105		
11581	063302	044440	020116	045522		
11582	063310	051115	000062			
11583	063314	042522	042101	044040	EM83:	.ASCIZ /READ HEADER ERR WORD 0 (CYL#)/
11584	063322	040505	042504	020122		
11585	063330	051105	020122	020040		
11586	063336	047527	042122	030040		
11587	063344	024040	054503	021514		
11588	063352	000051				
11589	063354	047506	046522	052101	EM84:	.ASCIZ /FORMAT IN RKMR3 NOT SET/
11590	063362	044440	020116	045522		
11591	063370	051115	020063	047516		
11592	063376	020124	042523	000124		
11593	063404	046111	020114	042101	EM85:	.ASCIZ /ILL ADDR IN RKMR3 NOT =0/
11594	063412	051104	044440	020116		
11595	063420	045522	051115	020063		
11596	063426	047516	020124	030075		
11597	063434	000				
11598	063435	127	044510	042514	EM86:	.ASCIZ /WHILE WAITING FOR CONTR RDY OR AFTER CONTR RDY REC'D/
11599	063442	053440	044501	044524		
11600	063450	043516	043040	051117		
11601	063456	041440	047117	051124		
11602	063464	051040	054504	047440		
11603	063472	020122	043101	042524		
11604	063500	020122	047503	052116		
11605	063506	020122	042122	020131		
11606	063514	042522	023503	000104		
11607	063522	040503	047116	052117	EM87:	.ASCIZ /CANNOT READ BSE INFO/
11608	063530	051040	040505	020104		
11609	063536	051502	020105	047111		

```

11610 063544 047506 000
11611 063547 116 020117 051104
11612 063554 053111 051505 043040
11613 063562 052517 042116 047440
11614 063570 020116 052502 006523
11615 063576 012
11616 063577 123 052105 050125
11617 063604 041440 051117 042522
11618 063612 052103 054514 023040
11619 063620 050040 042522 051523
11620 063626 023440 047503 052116
11621 063634 006447 000012
11622 063640 047516 042040 044522
11623 063646 042526 020123 047506
11624 063654 047125 020104 047111
11625 063662 042040 053105 041511
11626 063670 020105 040515 020120
11627 063676 022050 042504 046526
11628 063704 006451 012
11629 063707 123 052105 050125
11630 063714 041440 051117 042522
11631 063722 052103 054514 023040
11632 063730 051040 051505 040524
11633 063736 052122 005015 000
11634 063743 104 052104 047040
11635 063750 052117 051440 052105
11636 063756 044440 020116 045522
11637 063764 051504 000
11638
11639
11640
11641 063767 124 051505 020124
11642 063774 047516 020056 050040
11643 064002 000103
11644 064004 045522 051115 004461
11645 064012 045522 051115 004462
11646 064020 045522 051115 004463
11647 064026 045522 051105 051011
11648 064034 042113 004523 045522
11649 064042 051503 004461 045522
11650 064050 051503 000062
11651 064054 045522 041527 051011
11652 064062 041113 004501 045522
11653 064070 040504 051011 040513
11654 064076 047523 004506 045522
11655 064104 041504 051011 042513
11656 064112 050103 004523 045522
11657 064120 041505 052120 000
11658 064125 106 047522 020115
11659 064132 054503 020114 052040
11660 064140 020117 054503 020114
11661 064146 041440 046131 042040
11662 064154 043111 000106
11663 064160 044127 047105 042040
11664 064166 044522 042526 052440
11665 064174 046116 040517 042504

```

```

EM88: .ASCII /NO DRIVES FOUND ON BUS/<CR><LF>
      .ASCIZ /SETUP CORRECTLY & PRESS 'CONT'/<CR><LF>
EM89: .ASCII /NO DRIVES FOUND IN DEVICE MAP ($DEVN)/<CR><LF>
      .ASCIZ /SETUP CORRECTLY & RESTART/<CR><LF>
EM90: .ASCIZ /DDT NOT SET IN RKDS/
      .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/
DH6: .ASCIZ /FROM CYL TO CYL CYL DIFF/
DH8: .ASCIZ /WHEN DRIVE UNLOADED/

```

11666	064202	000104				
11667	064204	043101	042524	020122	DH9:	.ASCIZ /AFTER START SPIN CMD REC'D BY DRIVE/
11668	064212	052123	051101	020124		
11669	064220	050123	047111	041440		
11670	064226	042115	051040	041505		
11671	064234	042047	041040	020131		
11672	064242	051104	053111	000105		
11673	064250	052101	042440	042116	DH10:	.ASCIZ /AT END OF HEAD LOADING/
11674	064256	047440	020106	042510		
11675	064264	042101	046040	040517		
11676	064272	044504	043516	000		
11677	064277	101	052106	051105	DH11:	.ASCIZ /AFTER START SPIN CMD & FWD SET/
11678	064304	051440	040524	052122		
11679	064312	051440	044520	020116		
11680	064320	046503	020104	020046		
11681	064326	053506	020104	042523		
11682	064334	000124				
11683	064336	052101	044440	047116	DH12:	.ASCIZ /AT INNER LIM FROM START SPIN CMD/
11684	064344	051105	046040	046511		
11685	064352	043040	047522	020115		
11686	064360	052123	051101	020124		
11687	064366	050123	047111	041440		
11688	064374	042115	000			
11689	064377	106	047522	020115	DH13:	.ASCIZ /FROM OUTER LIM TO CYL 0 DURING LOADING/
11690	064404	052517	042524	020122		
11691	064412	044514	020115	047524		
11692	064420	041440	046131	030040		
11693	064426	042040	051125	047111		
11694	064434	020107	047514	042101		
11695	064442	047111	000107			
11696	064446	043101	042524	020122	DH14:	.ASCIZ /AFTER SEEK WITH BAD PAR/
11697	064454	042523	043505	053440		
11698	064462	052111	020110	040502		
11699	064470	020104	040520	000122		
11700	064476	043101	042524	020122	DH16:	.ASCIZ /AFTER LOADING HEAD REG & SEEK CMD/
11701	064504	047514	042101	047111		
11702	064512	020107	042510	042101		
11703	064520	051040	043505	023040		
11704	064526	051440	042505	020113		
11705	064534	046503	000104			
11706	064540	043101	042524	020122	DH17:	.ASCIZ /AFTER RECAL CMD/
11707	064546	042522	040503	020114		
11708	064554	046503	000104			
11709	064560	043101	042524	020122	DH18:	.ASCIZ /AFTER UNLOAD CMD/
11710	064566	047125	047514	042101		
11711	064574	041440	042115	000		
11712	064601	101	052106	051105	DH19:	.ASCIZ /AFTER PACK CMD/
11713	064606	050040	041501	020113		
11714	064614	046503	000104			
11715	064620	043101	042524	020122	DH20:	.ASCIZ /AFTER SELECT DRIVE CMD/
11716	064626	042523	042514	052103		
11717	064634	042040	044522	042526		
11718	064642	041440	042115	000		
11719	064647	101	052106	051105	DH21:	.ASCIZ /AFTER SUBSYSTEM CLEAR/
11720	064654	051440	041125	054523		
11721	064662	052123	046505	041440		

CZR6HE0 UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00MACY11 30A(1052) 25-JAN-78 12:08 PAGE 224
DATA HEADERS

SEQ 0224

11778	065330	047527	042122	004443	DH40: .ASCIZ /WORD# HEADER WAS SHOULD BE/
11779	065336	042510	042101	051105	
11780	065344	053440	051501	020040	
11781	065352	044123	052517	042114	
11782	065360	041040	000105		
11783	065364	052504	044522	043516	DH41: .ASCIZ /DURING RECAL CMD/
11784	065372	051040	041505	046101	
11785	065400	041440	042115	000	
11786	065405	117	020116	042523	DH42: .ASCIZ /ON SEC 0,2,4,6,8 CYL 410 TRK 2/
11787	065412	020103	026060	026062	
11788	065420	026064	026066	020070	
11789	065426	041440	046131	032040	
11790	065434	030061	052040	045522	
11791	065442	031040	000		
11792	065445	106	051117	040515	DH44: .ASCIZ /FORMAT & ALL R-W TESTS WILL BE BYPASSED/
11793	065452	020124	020046	046101	
11794	065460	020114	026522	020127	
11795	065466	042524	052123	020123	
11796	065474	044527	046114	041040	
11797	065502	020105	054502	040520	
11798	065510	051523	042105	000	
11799	065515	102	041505	052501	DH45: .ASCIZ /BECAUSE OF LIMIT DETECT ERROR ON PREVIOUS TEST/
11800	065522	042523	047440	020106	
11801	065530	044514	044515	020124	
11802	065536	042504	042524	052103	
11803	065544	042440	051122	051117	
11804	065552	047440	020116	051120	
11805	065560	053105	047511	051525	
11806	065566	052040	051505	000124	
11807	065574	047503	046125	020104	DH46: .ASCIZ /COULD NOT READ BSE INFO ON PREV TEST/
11808	065602	047516	020124	042522	
11809	065610	042101	041040	042523	
11810	065616	044440	043116	020117	
11811	065624	047117	050040	042522	
11812	065632	020126	042524	052123	
11813	065640	000			
11814	065641	101	052106	051105	DH48: .ASCIZ /AFTER SEEK CMD TO INV CYL/
11815	065646	051440	042505	020113	
11816	065654	046503	020104	047524	
11817	065662	044440	053116	041440	
11818	065670	046131	000		
11819	065673	115	043523	040440	DH49: .ASCIZ /MSG A&B IN RKMR2 & RKMR3 RESP., ARE INVALID/
11820	065700	041046	044440	020116	
11821	065706	045522	051115	020062	
11822	065714	020046	045522	051115	
11823	065722	020063	042522	050123	
11824	065730	026056	040440	042522	
11825	065736	044440	053116	046101	
11826	065744	042111	000		
11827	065747	101	052106	051105	DH51: .ASCIZ /AFTER SEEK TO SELF CMD/
11828	065754	051440	042505	020113	
11829	065762	047524	051440	046105	
11830	065770	020106	046503	000104	
11831	065776	054105	020120	054503	DH52: .ASCIZ /EXP CYL# CYL HEADER WAS/
11832	066004	021514	041411	046131	
11833	066012	044040	040505	042504	

11834	066020	020122	040527	000123	
11835	066026	047117	051440	041505	DHS3: .ASCIZ /ON SEC 10,12,14,16,18,20 CYL 410 TRK 2/
11836	066034	030440	026060	031061	
11837	066042	030454	026064	033061	
11838	066050	030454	026070	030062	
11839	066056	041440	046131	032040	
11840	066064	030061	052040	045522	
11841	066072	031040	000		
11842					.SBTTL ERR OUTPUT DATA
11843					
11844		066076			.EVEN
11845	066076	001214	001116		DT1: \$TESTN,\$ERRPC
11846	066102	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11847	066110	003336	003334	003322	
11848	066116	003324			HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11849	066120	003326	003330	003332	
11850	066126	003340	003342	003354	
11851	066134	003356			
11852	066136	001214	001116	001350	DT4: \$TESTN,\$ERRPC,FRCYL,TOCYL,CALDIF
11853	066144	001352	001360		
11854	066150	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11855	066156	003336	003334	003322	
11856	066164	003324			HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11857	066166	003326	003330	003332	
11858	066174	003340	003342	003354	
11859	066202	003356			
11860	066204	001214	001116	001402	DT6: \$TESTN,\$ERRPC,PSEC,ESEC
11861	066212	001404			
11862	066214	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11863	066222	003336	003334	003322	
11864	066230	003324			HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11865	066232	003326	003330	003332	
11866	066240	003340	003342	003354	
11867	066246	003356			
11868	066250	001214	001116	001442	DT7: \$TESTN,\$ERRPC,WDCNT,HDWD,TEMP1
11869	066256	001454	003360		
11870	066262	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11871	066270	003336	003334	003322	
11872	066276	003324			HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11873	066300	003326	003330	003332	
11874	066306	003340	003342	003354	
11875	066314	003356			
11876	066316	001214	001116	001352	DT8: \$TESTN,\$ERRPC,TOCYL,FRCYL,CALDIF
11877	066324	001350	001360		
11878	066330	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11879	066336	003336	003334	003322	
11880	066344	003324			HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11881	066346	003326	003330	003332	
11882	066354	003340	003342	003354	
11883	066362	003356			
11884	066364	001214	001116	001352	DT9: \$TESTN,\$ERRPC,TOCYL,RHTAB
11885	066372	001674			
11886	066374	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11887	066402	003336	003334	003322	
11888	066410	003324			HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11889	066412	003326	003330	003332	

11890	066420	003340	003342	003354	
11891	066426	003356			
11892	066430	001214	001116	001350	DT10: \$TESTN,\$ERRPC,FRCYL,RHTAB
11893	066436	001674			
11894	066440	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11895	066446	003335	003334	003322	
11896	066454	003324			
11897	066456	003326	003330	003332	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11898	066464	003340	003342	003354	
11899	066472	003356			
11900	066474	001214	001334		DT11: \$TESTN,TRAPPC
11901	066500	001214	001116	003412	DT13: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
11902	066506	003414	003416	003420	
11903	066514	003372	003374	003376	
11904	066522	003400			
11905	066524	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11906	066532	003336	003334	003322	
11907	066540	003324			
11908	066542	003326	003330	003332	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11909	066550	003340	003342	003354	
11910	066556	003356			
11911					
11912	066560	001214	001116	003412	DT14: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
11913	066566	003414	003416	003420	
11914	066574	003422	003424		
11915	066600	003372	003374	003376	H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
11916	066606	003400	003402	003404	
11917	066614	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11918	066622	003336	003334	003322	
11919	066630	003324			
11920	066632	003326	003330	003332	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11921	066640	003340	003342	003354	
11922	066646	003356			
11923					
11924	066650	001214	001116	003412	DT15: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
11925	066656	003414	003416	003420	
11926	066664	003422	003424	003430	
11927	066672	003372	003374	003376	H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
11928	066700	003400	003402	003404	
11929	066706	003410			
11930	066710	003346	003350	003352	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
11931	066716	003336	003334	003322	
11932	066724	003324			
11933	066726	003326	003330	003332	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
11934	066734	003340	003342	003354	
11935	066742	003356			
11936					
11937					.SBTTL ERR DATA FORMATS :
11938					
11939	066744	000003			DF1: 3
11940	066746	002	000		.BYTE 2,0
11941	066750	064004			DH2
11942	066752	007	000		.BYTE 7,0
11943	066754	064054			DH3
11944	066756	007	000		.BYTE 7,0
11945					

11946	066760	000005		DF2:	5
11947	066762	000	000		.BYTE 0,0
11948	066764	065673			DH49
11949	066766	000	000		.BYTE 0,0
11950	066770	063767			DH1
11951	066772	002	000		.BYTE 2,0
11952	066774	064004			DH2
11953	066776	007	000		.BYTE 7,0
11954	067000	064054			DH3
11955	067002	007	000		.BYTE 7,0
11956					
11957	067004	000001		DF3:	1
11958	067006	002	000		.BYTE 2,0
11959	067010	000003		DF4:	3
11960	067012	002	000		.BYTE 2,0
11961	067014	064004			DH2
11962	067016	007	000		.BYTE 7,0
11963	067020	064054			DH3
11964	067022	007	000		.BYTE 7,0
11965					
11966	067024	000004		DF5:	4
11967	067026	000	000		.BYTE 0,0
11968	067030	063767			DH1
11969	067032	002	000		.BYTE 2,0
11970	067034	064004			DH2
11971	067036	007	000		.BYTE 7,0
11972	067040	064054			DH3
11973	067042	007	000		.BYTE 7,0
11974					
11975	067044	000005		DF6:	5
11976	067046	000	000		.BYTE 0,0
11977	067050	063767			DH1
11978	067052	002	000		.BYTE 2,0
11979	067054	064125			DH6
11980	067056	003	000		.BYTE 3,0
11981	067060	064004			DH2
11982	067062	007	000		.BYTE 7,0
11983	067064	064054			DH3
11984	067066	007	000		.BYTE 7,0
11985					
11986					
11987					
11988	067070	000004		DF10:	4
11989	067072	000	000		.BYTE 0,0
11990	067074	063767			DH1
11991	067076	002	000		.BYTE 2,0
11992	067100	064004			DH2
11993	067102	007	000		.BYTE 7,0
11994	067104	064054			DH3
11995	067106	007	000		.BYTE 7,0
11996					
11997	067110	000005		DF12:	5
11998	067112	000	000		.BYTE 0,0
11999	067114	063767			DH1
12000	067116	002	000		.BYTE 2,0
12001	067120	065174			DH36

CZR6HED UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 228
ERR DATA FORMATS

SEQ 0228

12002	067122	002	000		.BYTE 2,0
12003	067124	064004			DH2
12004	067126	007	000		.BYTE 7,0
12005	067130	064054			DH3
12006	067132	007	000		.BYTE 7,0
12007					
12008	067134	000004		DF14:	4
12009	067136	002	000		.BYTE 2,0
12010	067140	065330			DH40
12011	067142	003	000		.BYTE 3,0
12012	067144	064004			DH2
12013	067146	007	000		.BYTE 7,0
12014	067150	064054			DH3
12015	067152	007	000		.BYTE 7,0
12016					
12017					
12018	067154	000004		DF15:	4
12019	067156	000	000		.BYTE 0,0
12020	067160	063767			DH1
12021	067162	002	000		.BYTE 2,0
12022	067164	064004			DH2
12023	067166	007	000		.BYTE 7,0
12024	067170	064054			DH3
12025	067172	007	000		.BYTE 7,0
12026					
12027	067174	000004		DF16:	4
12028	067176	000	000		.BYTE 0,0
12029	067200	063767			DH1
12030	067202	002	000		.BYTE 2,0
12031	067204	064004			DH2
12032	067206	007	000		.BYTE 7,0
12033	067210	064054			DH3
12034	067212	007	000		.BYTE 7,0
12035					
12036	067214	000005		DF17:	5
12037	067216	000	000		.BYTE 0,0
12038	067220	065445			DH44
12039	067222	000	000		.BYTE 0,0
12040	067224	063767			DH1
12041	067226	002	000		.BYTE 2,0
12042	067230	064004			DH2
12043	067232	007	000		.BYTE 7,0
12044	067234	064054			DH3
12045	067236	007	000		.BYTE 7,0
12046	067240	000007		DF20:	7
12047	067242	000	000		.BYTE 0,0
12048	067244	063767			DH1
12049	067246	002	000		.BYTE 2,0
12050	067250	065007			DH28
12051	067252	000	000		.BYTE 0,0
12052	067254	065055			DH32
12053	067256	004	000		.BYTE 4,0
12054	067260	065017			DH29
12055	067262	004	000		.BYTE 4,0
12056	067264	064004			DH2
12057	067266	007	000		.BYTE 7,0

12058	067270	064054			DH3	
12059	067272	007	000		.BYTE	7,0
12060	067274	000004		DF21:	4	
12061	067276	002	000		.BYTE	2,0
12062	067300	065776			DH52	
12063	067302	002	000		.BYTE	2,0
12064	067304	064004			DH2	
12065	067306	007	000		.BYTE	7,0
12066	067310	064054			DH3	
12067	067312	007	000		.BYTE	7,0
12068	067314	000007		DF22:	7	
12069	067316	000	000		.BYTE	0,0
12070	067320	063767			DH1	
12071	067322	002	000		.BYTE	2,0
12072	067324	065007			DH28	
12073	067326	000	000		.BYTE	0,0
12074	067330	065055			DH32	
12075	067332	006	000		.BYTE	6,0
12076	067334	065017			DH29	
12077	067336	006	000		.BYTE	6,0
12078	067340	064004			DH2	
12079	067342	007	000		.BYTE	7,0
12080	067344	064054			DH3	
12081	067346	007	000		.BYTE	7,0
12082						
12083	067350	000007		DF23:	7	
12084	067352	000	000		.BYTE	0,0
12085	067354	063767			DH1	
12086	067356	002	000		.BYTE	2,0
12087	067360	065007			DH28	
12088	067362	000	000		.BYTE	0,0
12089	067364	065055			DH32	
12090	067366	007	000		.BYTE	7,0
12091	067370	065017			DH29	
12092	067372	007	000		.BYTE	7,0
12093	067374	064004			DH2	
12094	067376	007	000		.BYTE	7,0
12095	067400	064054			DH3	
12096	067402	007	000		.BYTE	7,0

```

12097
12098
12099
12100
12101
12102
12103
12104
12105
12106
12107 067404 104413
12108 067406 113700 001114
12109 067412 042700 177400
12110 067416 005300
12111 067420 006300
12112 067422 006300
12113 067424 006300
12114 067426 062700 003474
12115 067432 012037 067446
12116 067436 001404
12117 067440 104401 001205
12118 067444 104401
12119 067446 000000
12120 067450 012037 067464
12121 067454 001404
12122 067456 104401 001205
12123 067462 104401
12124 067464 000000
12125 067466 012001
12126 067470 001455
12127 067472 005004
12128 067474 012000
12129 067476 012002
12130 067500 001446
12131 067502 005104
12132 067504 104401 001205
12133 067510 112003
12134 067512 105720
12135 067514 005703
12136 067516 001407
12137 067520 013146
12138 067522 104402
12139 067524 005303
12140 067526 001403
12141 067530 104401 067660
12142 067534 000771
12143 067536 005302
12144 067540 003431
12145 067542 104401 001205
12146 067546 005760 000002
12147 067552 001404
12148 067554 005104
12149 067556 001002
12150 067560 104401 067660
12151 067564 012037 067572
12152 067570 104401

```

```

*****
;SBTTL TYPE ERR ROUTINE
;ENTRY JSR PC,TYP ERR
;RETURN RTS PC
;
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;ERR IS TO BE REPORTED. IT THEN USES THE "ERR TABLE" ($ERRTB)
;ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
;THE ERR.
*****
↑PERR: SAVREG
MOVSB $ITEMB,RO ;ENTER ERR NUMBER
BIC #177400,RO ;CLEAR SIGN EXTENSION
DEC RO ;FORM INDEX FOR ERR TABLE
ASL RO
ASL RO
ASL RO
1$: ADD $ERRTB,RO ;FORM ADDRESS OF ERR ENTRY
MOV (RO)+,2$ ;GET EM POINTER
BEQ 3$ ;BRANCH IF THERE ISN'T ONE
TYPE ,$CRLF ;TYPE CARRIAGE RETURN LINE FEED
TYPE ;TYPE ERR MSG (EM)
2$: .WORD 0 ;EM POINTER GOES HERE
3$: MOV (RO)+,4$ ;GET DH POINTER
BEQ 5$ ;BRANCH IF THERE ISN'T ONE
TYPE ,$CRLF ;TYPE CR-LF
TYPE ;TYPE DATA HEADER
4$: .WORD 0 ;DH POINTER GOES HERE
5$: MOV (RO)+,R1 ;GET DT POINTER
BEQ 20$ ;BRANCH IF THERE ARE NONE
CLR R4 ;SET INDENT SWITCH
MOV (RO)+,RO ;GET DF POINTER
MOV (RO)+,R2 ;STORE NUMBER OF DH'S
BEQ 17$ ;DH NUM IS 0-BRANCH
COM R4 ;NO INDENT
10$: TYPE ,$CRLF
MOVSB (RO)+,R3 ;GET & STORE NUMBER OF DATA WORDS
TSTB (RO)+ ;BUMP PAST FORMAT WORD
R3 ;TEST IF ANY DATA FOR THIS HEADER
BEQ 14$ ;NO - SKIP DATA PRINT
11$: MOV 2(R1)+,-(SP) ;PUT FIRST DATA WORD ON STACK
TYPOC ;TYPE IT
DEC R3 ;MORE DATA WORDS
BEQ 14$ ;NO-BRANCH
TYPE ,SPACE2 ;TYPE SEPARATORS
BR 11$ ;LOOP
14$: DEC R2 ;MORE DH'S?
BLE 20$ ;NO-BRANCH
TYPE , $CRLF
TST 2(RO) ;ONLY A DH IN THIS REQUEST?
BEQ 15$ ;YES-BRANCH BYPASS INDENT
COM R4 ;INDENT?
BNE 15$ ;NO-BRANCH
TYPE ,SPACE2 ;YES-TYPE SPACES
15$: MOV (RO)+,16$ ;GET NEXT DH POINTER
TYPE ;TYPE DH

```


12153	067572	000000			16\$:	.WORD	0		;DH POINTER GOES HERE
12154	067574	105710				TSTB	(R0)		;TYPE A DT?
12155	067576	001003				BNE	21\$;YES-BRANCH
12156	067600	062700	000002			ADD	#2,R0		;INCREMENT DF POINTER
12157	067604	000754				BR	14\$;SEE IF END OF DF BLOCK
12158	067606	104401	001205		21\$:	TYPE	\$CRLF		
12159	067612	005704				TST	R4		;INDENT?
12160	067614	001335				BNE	10\$;NO-BRANCH
12161	067616	104401	067660		17\$:	TYPE	SPACE2		;YES-TYPE SPACES
12162	067622	000732				BR	10\$;LOOP
12163	067624	104414			20\$:	RESREG			
12164									
12165	067626	032777	010000	111304		BIT	#SW12,2SWR		;ABORT DRV AFTER 20 ERRS?
12166	067634	001410				BEQ	25\$;BR IF NO
12167	067636	023727	001103	000024		CMP	\$ERFLG,#20.		;ELSE SEE IF 20 ERRS
12168	067644	001004				BNE	25\$;BR IF NO
12169	067646	012706	001100			MOV	#STACK,SP		;ELSE RESTORE STK
12170	067652	000137	042662			JMP	\$EOP		;AND DROP DRIVE
12171	067656	000207			25\$:	RTS	PC		
12172	067660	020040	000		SPACE2:	.ASCIZ/	/		;2 SPACES

12173
12174
12175
12176
12177
12178
12179
12180
12181
12182
12183
12184
12185
12186
12187
12188
12189
12190
12191
12192
12193
12194
12195
12196
12197
12198
12199
12200
12201
12202
12203
12204
12205
12206
12207
12208
12209
12210
12211
12212
12213
12214
12215
12216
12217
12218
12219
12220
12221
12222
12223
12224
12225
12226
12227
12228

067664
067744
000000
000001
000002
000003
000004
000005
000006
000007
177776
000014
000340
000020
000003
000006
177562
177560
177566
177564
067744 000413
067746 000417
067750 013737 177776 067724
067756 013737 000016 177776
067764 010737 067722
067770 000137 071122
067774 012706 067704
070000 010637 067720
070004 000414
070006 004037 071330
070012 013777 067742 177716
070020 113704 067726
070024 106004

```
; ODT-11 -- V005A
; DEC-11-UODPA-A-LA
; COPYRIGHT 1969,1970,1972
; DIGITAL EQUIPMENT CORPORATION
; MAYNARD, MASSACHUSETTS 01754
; .ENABL ABS,AMA
; .EVEN
; .+.60
R0 = %0 ; REGISTER
R1 = %1 ; NAMING
R2 = %2 ; CONVENTIONS
R3 = %3
R4 = %4
R5 = %5
SP = %6
PC = %7
ST = 177776 ; STATUS REGISTER
O.TVEC = 14 ; TRT VECTOR LOCATION
O.STM = 340 ; PRIORITY MASK - STATUS REGISTER
O.TBT = 20 ; T-BIT MASK - STATUS REGISTER
TRT = 000003 ; TRT INSTRUCTION
RTT = 000006 ; RTT INSTRUCTION
;
; R5 IS USUALLY CONSIDERED SAFE. THE CURRENT ADDRESS WORD
; RESIDES IN IT. AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
; OPERATIONS LEAVE IT RANDOMLY FILLED. OTHERWISE, IT SHOULD NOT
; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
O.RDB = 177562 ; R DATA BUFFER
O.RCSR = 177560 ; R C/SR
O.TDB = 177566 ; T DATA BUFFER
O.TCSR = 177564 ; T C/SR
;
; INITIALIZE ODT
; USE O.ODT FOR A NORMAL ENTRY
; USE O.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
; USE O.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
O.ODT: BR O.STRT ; NORMAL ENTRY
BR O.RST ; RESTART
O.ENTR: MOV ST,O.UST ; RE-ENTER -- SAVE STATUS
MOV O.TVEC+2,ST ; SET UP LOCAL STATUS
MOV PC,O.UPC ; FAKE THE PC
JMP O.BK1
;
O.STRT: MOV #O.URD,SP ; SET UP STACK
MOV SP,O.USR ; FAKE THE SAVED STACK
BR O.RST1 ; CLEAR BREAKPOINT TABLES
O.RST: JSR O,O.SVR ; SAVE REGISTERS
MOV O.UIN,O.ADR1 ; REMOVE THE BREAKPOINT
MOVB O.PRI,R4 ; GET ODT PRIORITY
RORB R4 ; SHIFT
```

```

12229 070026 106004          RORB    R4          ; INTO
12230 070030 106004          RORB    R4          ; POSITION
12231 070032 110437 177776  MOVB   R4,ST      ; STORE IN STATUS
12232 070036 000127          BR      (PC)+
12233 070040 000403          BR      0.45
12234 070042 012737 000002 071032  MOV    #RTI,0.RTIT ; SET TO RTI IF 11/20 OR /05
12235 070050 105037 071751 0.45:  CLRB   0.P        ; DISALLOW PROCEED
12236 070054 012737 000340 000016  MOV    #0.STM,0.TVEC+2 ; STATUS WORD TO TRT VECTOR + 2
12237 070062 012737 071112 000014  MOV    #0.BRK,0.TVEC ; PC TO TRT VECTOR
12238 070070 000447          BR      0.RALL     ; CLEAR BREAKPOINT TABLES
12239
12240          ; SPECIAL NAME HANDLER
12241          ; DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URD
12242
12243 070072 004537 071552 0. REGT: JSR    5.0.GET ; SPECIAL NAME, GET ONE MORE CHARACTER
12244 070076 012704 071775          MOV    #0.TL,R4    ; TABLE START ADDRESS
12245 070102 120024 0. RSP:  CMPB   R0,(R4)+  ; IS THIS THE CORRECT CHARACTER?
12246 070104 001413          BEQ    0.SP        ; JUMP IF YES
12247 070106 022704 072003          CMP    #0.TL+0.LG,R4 ; IS THE SEARCH DONE?
12248 070112 101373          BHI   0.RSP       ; BRANCH IF NOT
12249 070114 042700 177770          BIC   #177770,R0  ; MASK OFF OCTAL
12250 070120 010004          MOV    R0,R4
12251 070122 006304 0. SP1:  ASL    R4
12252 070124 062704 067704          ADD   #0.URD,R4   ; GENERATE ADDRESS
12253 070130 005202          INC   R2          ; SET FOUND FLAG
12254 070132 000444          BR    0.SCAN     ; GO FIND NEXT CHARACTER
12255 070134 162704 071766 0. SP:  SUB    #0.TL-7,R4 ; CORRECT CONSTANT
12256 070140 000770          BR    0.SP1
12257
12258          ; + HANDLER - OPEN INDEXED ON THE PC
12259
12260 070142 004737 071676 0. ORPC: JSR    PC,0.TCLS ; CURRENT ADDRESS IN R2
12261 070146 010502          MOV    R5,R2     ; COMPUTE
12262 070150 061202          ADD   @R2,R2    ; MOVE ONE BIT TO CARRY
12263 070152 006202          ASR   R2        ; ERR IF ODD NUMBER
12264 070154 103421          BCS   0.ERR     ; RESTORE WORD
12265 070156 006302          ASL   R2        ; AND INCREMENT BY TWO
12266 070160 005722          TST   (R2)+
12267 070162 010205          MOV   R2,R5    ; UPDATE CAD
12268 070164 000137 070436          JMP   0.OP2     ; GO FINISH UP
12269
12270          ; B HANDLER - SET AND REMOVE BREAKPOINTS
12271
12272 070170 005702 0. BKPT: TST    R2          ; IF NO NUMBER TYPED
12273 070172 001406          BEQ   0.RALL    ; REMOVE BREAKPOINT
12274 070174 006204          ASR   R4        ; CHECK IF ODD
12275 070176 103410          BCS   0.ERR     ; JUMP IF ODD
12276 070200 006304          ASL   R4        ; RESTORE ONE BIT
12277 070202 010437 067736          MOV   R4,0.ADR1 ; SET A BREAKPOINT
12278 070206 000412          BR    0.DCD
12279 070210 012737 072012 067736 0. RALL: MOV   #0.TRTC,0.ADR1 ; CLEAR BREAKPOINT
12280 070216 000406          BR    0.DCD
12281
12282          ; CMD DECODER - ODT11
12283
12284          ; REGISTERS R0-R4 MAY BE USED,

```

```

12285
12286
12287 070220 052705 000001
12288 070224 012700 000077
12289 070230 004537 071630
12290 070234 004537 071730
12291 070240 005004
12292 070242 005002
12293 070244 004537 071552
12294 070250 022700 000060
12295 070254 101013
12296 070256 022700 000067
12297 070262 103410
12298 070264 042700 177770
12299 070270 006304
12300 070272 006304
12301 070274 006304
12302 070276 060004
12303 070300 005202
12304 070302 000760
12305 070304 005001
12306 070306 120061 071761
12307 070312 001405
12308 070314 005201
12309 070316 020127 000014
12310 070322 103336
12311 070324 000770
12312 070326 006301
12313 070330 000171 070334
12314
12315 070334 070364
12316 070336 070416
12317 070340 070072
12318 070342 070726
12319 070344 070430
12320 070346 070142
12321 070350 070462
12322 070352 070472
12323 070354 070550
12324 070356 070544
12325 070360 070170
12326 070362 071034
12327 000030
12328
12329
12330
12331 070364 005702
12332 070366 001410
12333 070370 010405
12334 070372 006205
12335 070374 103711
12336 070376 006305
12337 070400 011500
12338 070402 004537 071466
12339 070406 000714
12340 070410 042705 000001

```

```

; REGISTER R5 WILL BE CONSIDERED SAFE
;
; .ERR: BIS #1,R5 ; CLOSE EVERYTHING
; MOV #?,R0 ; ? TO BE TYPED
; JSR 5,O.FTYP ; OUTPUT ?
; O.DCD: JSR 5,O.CRLS ; TYPE <CR><LF>*
; O.DCD1: CLR R4 ; R4 CONTAINS THE CONVERTED OCTAL
; CLR R2 ; R2 IS THE NUMBER FOUND FLAG
; O.SCAN: JSR 5,O.GET ; GET A CHAR, RETURN IN R0
; CMP #0,R0 ; COMPARE WITH ASCII 0
; BHI O.CLGL ; CHECK LEGALITY IF NON-NUMERIC
; CMP #7,R0 ; COMPARE WITH ASCII 7
; BLO O.CLGL ; CHECK LEGALITY IF NOT OCTAL
; BIC #177770,R0 ; CONVERT TO BCD
; ASL R4 ; MAKE ROOM
; ASL R4 ; IN
; ASL R4 ; R4
; ADD R0,R4 ; PACK THREE BITS IN R4
; INC R2 ; R2 HAS NUMERIC FLAG
; BR O.SCAN ; AND TRY AGAIN
; 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 20.LGDR(R1) ; GO TO PROPER ROUTINE
;
; .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 0 OFFSET
; O.WSCH W SEARCH WORD
; O.EFF E SEARCH EFFECTIVE ADDRESS
; O.BKPT B BREAKPOINTS
; O.PROC P PROCEED
;
; O.LGL = -.LGDR ;LGL MUST EQUAL 2X CHLGT ALWAYS
;
; PROCESS / - OPEN WORD
;
; .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

```

```

TYPE ERR ROUTINE
12341 070414 000766          BR      0.WRD1          ;GO BACK TO MAIN-LINE
12342
12343          ; PROCESS CARRIAGE RETURN
12344
12345 070416 004737 071676 0.CRET: JSR      PC,0.TCLS      ;CLOSE LOCATION
12346 070422 052705 000001      BIS      #1,R5          ;CLOSE EVERYTHING
12347 070426 000702          BR      0.DCD          ;RETURN TO DECODER
12348
12349          ; PROCESS <LF>, OPEN NEXT WORD
12350
12351 070430 004737 071676 0.OP1: JSR      PC,0.TCLS      ;CLOSE PRESENT CELL
12352 070434 005725          TST      (R5)+          ;GENERATE NEW ADDRESS
12353 070436 004537 071722 0.OP2: JSR      S,0.CRLF      ;<CR><LF>
12354 070442 010500          MOV      R5,R0          ;NUMBER TO TYPE
12355 070444 004537 071466      JSR      S,0.CADV       ;TYPE OUT ADDRESS
12356 070450 012700 000057      MOV      #1,R0          ;TYPE A /
12357 070454 004537 071630      JSR      S,0.FTYP       ;
12358 070460 000744          BR      0.WRD1          ;GO PROCESS IT
12359
12360          ; PROCESS †, OPEN PREVIOUS WORD
12361
12362 070462 004737 071676 0.BACK: JSR      PC,0.TCLS      ;GENERATE NEW ADDRESS
12363 070466 005745          TST      -(R5)          ;GO DO THE REST
12364 070470 000762          BR      0.OP2
12365
12366          ; PROCESS 0, COMPUTE OFFSET
12367
12368 070472 006205 000040 0.OFST: ASR      R5          ;GET LOW ORDER BIT
12369 070474 103737          BCS      0.ERR2         ;ERR IF CLOSED
12370 070476 006305          ASL      R5          ;RESTORE WORD
12371 070500 012700 000040      MOV      #7,R0          ;TYPE ONE BLANK
12372 070504 004537 071630      JSR      S,0.FTYP       ;AS A SEPARATOR
12373 070510 160504          SUB      R5,R4          ;COMPUTE
12374 070512 005304          DEC      R4
12375 070514 005304          DEC      R4          ; 16 BIT OFFSET
12376 070516 010400          MOV      R4,R0          ;TYPE A
12377 070520 010402          MOV      R4,R2          ;SAVE R4
12378 070522 004537 071466      JSR      S,0.CADV       ;NUMBER IN R0 - WORD MODE
12379 070526 010200          MOV      R2,R0
12380 070530 006200          ASR      R0          ;DIVIDE BY TWO
12381 070532 103402          BCS      0.OF1          ;BRANCH IF ODD
12382 070534 004537 071466      JSR      S,0.CADV       ;NUMBER IN R0 - BYTE MODE
12383 070540 000137 070240 0.OF1: JMP      0.DCD1          ;ALL DONE
12384
12385          ; SEARCHES - $MSK HAS THE MASK
12386          ; $MSK+2 HAS THE FWA
12387          ; $MSK+4 HAS THE LWA
12388
12389 070544 005201 067732 0.EFF: INC      R1          ;SET EFFECTIVE SEARCH
12390 070546 000401          BR      0.WDS
12391 070550 005001          0.WSCH: CLR     R1          ;SET WORD SEARCH
12392 070552 005702          0.WDS: TST     R2          ;CHECK FOR OBJECT FOUND
12393 070554 001621          0.ERR1: BEQ     0.ERR       ;ERR IF NO OBJECT
12394 070556 013702 067730      MOV      0.MSK+2,R2     ;SET ORIGIN
12395 070562 013705          MOV      0.MSK,R5       ;SET MASK
12396 070566 005105          COM      R5            ;AND COMPLEMENT IT

```

12397	070570	020237	067734	0.WDS2:	CMP	R2,0,MSK+4	;	IS THE SEARCH ALL DONE?
12398	070574	101217			BHI	0,DCD	;	YES
12399	070576	011200			MOV	2R2,RO	;	GET OBJECT
12400	070600	005701			TST	R1	;	NO
12401	070602	001027			BNE	0,EFF1	;	BRANCH IF EFFECTIVE SEARCH
12402	070604	010046			MOV	RO,-(SP)		
12403	070606	010403			MOV	R4,R3	;	EXCLUSIVE OR
12404	070610	040400			BIC	R4,RO	;	IS DONE
12405	070612	042603			BIC	(SP)+,R3		
12406	070614	050003			BIS	RO,R3		
12407	070616	040503			BIC	R5,R3	;	FANCY MANNER HERE
12408	070620	001016		0.WDS3:	BNE	0,WDS4	;	AND RESULT WITH MASK
12409	070622	010446			MOV	R4,-(SP)	;	RE-LOOP IF NO MATCH
12410	070624	004537	071722		JSR	5,0,CRLF	;	REGISTERS R2,R4, AND R5 ARE SAFE
12411	070630	010200			MOV	R2,RO	;	TYPE <CR,LF>
12412	070632	004537	071466		JSR	5,0,CADV	;	GET READY TO TYPE
12413	070636	012700	000057		MOV	#1,RO	;	TYPE ADDRESS
12414	070642	004537	071630		JSR	5,0,FTYP	;	SLASH TO RO
12415	070646	011200			MOV	2R2,RO	;	TYPE IT
12416	070650	004537	071466		JSR	5,0,CADV	;	GET CONTENTS
12417	070654	012604			MOV	(SP)+,R4	;	TYPE CONTENTS
12418	070656	005722		0.WDS4:	TST	(R2)+	;	RESTORE R4
12419	070660	000743			BR	0,WDS2	;	INCREMENT TO NEXT CELL AND
12420	070662	020004		0.EFF1:	CMP	RO,R4	;	RETURN
12421	070664	001755			BEQ	0,WDS3	;	IS (X)=K?
12422	070666	010003			MOV	RO,R3	;	TYPE IF EQUAL
12423	070670	060203			ADD	R2,R3	;	(X) TO R3
12424	070672	005203			INC	R3	;	(X)+X
12425	070674	005203			INC	R3	;	(X)+X+2
12426	070676	020304			CMP	R3,R4	;	IS (X)+X+2=K?
12427	070700	001747			BEQ	0,WDS3	;	BRANCH IF EQUAL
12428	070702	042700	177400		BIC	#177400,RO	;	WIPE OUT EXTRANEIOUS BITS
12429	070706	110000			MOVB	RO,RO	;	EXTEND SIGN
12430	070710	000257			CCC			
12431	070712	006300			ASL	RO	;	MULTIPLY BY TWO
12432	070714	005200			INC	RO	;	ADD TWO
12433	070716	005200			INC	RO		
12434	070720	060200			ADD	R2,RO	;	ADD PC
12435	070722	020004			CMP	RO,R4	;	IS THE RESULT A PROPER REL. BRANCH?
12436	070724	000735			BR	0,WDS3		
12437								
12438								
12439								
12440	070726	105037	071751	0.GO:	CLRB	0,P	;	DISALLOW PROCEED
12441	070732	006204			ASR	R4	;	CHECK LOW ORDER BIT
12442	070734	103617			BCS	0,ERR2	;	ERR IF ODD NUMBER
12443	070736	006304			ASL	R4	;	RESTORE WORD
12444	070740	010437	067722		MOV	R4,0,UPC	;	SET UP NEW PC
12445	070744	112737	000340	177776	MOVB	#0,STM,ST	;	SET HIGH PRIORITY
12446	070752	004537	071420		JSR	5,0,RSTT	;	RESTORE TELETYPE
12447	070756	103037	071750	0.TBIT:	CLRB	0,T	;	CLEAR BOTH
12448	070762	042737	000020	067724	BIC	#0,TBT,0,UST	;	T-BIT FLAGS
12449	070770	017737	176742	067742	MOV	20,ADR1,0,UIN	;	SAVE INSTRUCTION
12450	070776	013777	072012	176732	MOV	0,TRTC,20,ADR1	;	REPLACE WITH TRAP
12451	071004	012600			MOV	(SP)+,RO	;	RESTORE
12452	071006	012601			MOV	(SP)+,R1	;	RO

```

12453 071010 012602          MOV      (SP)+,R2          ; THRU
12454 071012 012603          MOV      (SP)+,R3
12455 071014 012604          MOV      (SP)+,R4
12456 071016 012605          MOV      (SP)+,R5          ; R5
12457 071020 012606          MOV      (SP)+,SP          ; AND SP
12458 071022 013746 067724    MOV      0.UST,-(SP)      ; AND STATUS
12459 071026 013746 067722    MOV      0.UFC,-(SP)      ; AND PC
12460 071032 000006          O.RTIT: RTT                ; CHANGED TO RTI FOR 11/20 AND /05
12461
12462          ; PROCESS P - PROCEED
12463          ; ONLY ALLOWED AFTER A BREAKPOINT
12464
12465 071034 105737 071751    O.PROC: TSTB 0.P          ; CHECK LEGALITY OF PROCEED
12466 071040 001645          BEQ      0.ERR1          ; NOT LEGAL
12467 071042 105037 071751    CLR      0.P            ; CLEAR PROCEED FLAG
12468 071046 005702          TST      R2             ; WAS COUNT SPECIFIED?
12469 071050 001402          BEQ      0.PR1          ; NO
12470 071052 010437 067740    MOV      R4,0.CT        ; YES, PUT AWAY COUNT
12471 071056 112737 000340 177776 O.PR1: MOV      #0,STM,ST  ; FORCE HIGH PRIORITY
12472 071064 004537 071420    JSR      5,0.RSTT       ; RESTORE TTY
12473 071070 112737 000340 177776 O.C1:  MOV      #0,STM,ST  ; SET HIGH PRIORITY
12474 071076 105237 071750    INCB     0.T            ; SET T-BIT FLAG
12475 071102 052737 000020 067724    BIS      #0,TBT,0.UST   ; SET T-BIT
12476 071110 000735          BR       0.G02
12477
12478          ; BREAKPOINT HANDLER
12479          ; A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
12480          ; VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
12481          ; AND GIVES CONTROL TO THE CMD DECODER
12482
12483 071112 012637 067722    O.BRK: MOV      (SP)+,0.UPC ; PRIORITY IS 7 UPON ENTRY
12484 071116 012637 067724    MOV      (SP)+,0.UST    ; SAVE STATUS AND PC
12485 071122 004037 071330    O.BK1: JSR      0,0.SVR   ; SAVE VARIOUS REGISTERS
12486 071126 105737 071750    TSTB    0.T            ; CHECK FOR T-BIT SET
12487 071132 001311          BNE      0.TBIT         ; JUMP IF SET
12488 071134 013777 067742 176574    MOV      0.UIN,30.ADR1  ; REMOVE BREAKPOINTS
12489 071142 105737 067726    TSTB    0.PR1         ; CHECK IF PRIORITY
12490 071146 100003          BPL      0.BK2         ; IS AS SAME AS USER PGM
12491 071150 113705 067724    MOV      0.UST,R5       ; PICK UP USER UST IF SO
12492 071154 000407          BR       0.BK3         ; AND DON'T COMPUTE THE PRIORITY

```

F03

CZR6HED UNIBUS RKG DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 238
TYPE ERR ROUTINE

SEQ 0238

12493 071156 113705 067726

C.BK2: MOV8 0.PRI,R5

; OTHERWISE PICK UP ACTUAL PRIORITY


```

12494 071162 000257          CCC          ;CLEAR CARRY
12495 071164 106005          RORB R5     ;SHIFT LOW ORDER BITS
12496 071166 106005          RORB R5     ;      INTO
12497 071170 106005          RORB R5     ;      HIGH ORDER
12498 071172 106005          RORB R5     ;      POSITION
12499 071174 110537 177776 0.BK3: MOV R5,ST   ;PUT THE STATUS AWAY WHERE IT BELONGS
12500 071200 013705 067722   MOV 0,UPC,R5 ;GET PC, IT POINTS TO THE TRT
12501 071204 005745          TST -(R5)   ;SUBTRACT TWO
12502 071206 010537 067722   MOV R5,0,UPC ;FROM THE USER'S PC
12503 071212 020537 067736   CMP R5,0,ADR1 ;COMPARE WITH LIST
12504 071216 001417          BEQ 0.B2    ;JUMP IF FOUND
12505 071220 004537 071366   JSR 5,0,SVTT ;SAVE TELETYPE STATUS
12506 071224 004537 071722   JSR 5,0,CRLF
12507 071230 012704 071754   MOV #0,BD,R4 ;ERR, NOTHING FOUND
12508 071234 012703 071755   MOV #0,BD+1,R3
12509 071240 004537 071614   JSR 5,0,TYPE ;OUTPUT "BE" FOR BAD ENTRY
12510 071244 010500          MOV R5,RO   ;
12511 071246 042737 000020 067724 BIC #0,TBT,0,UST ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
12512 071254 000420          BR 0.B3     ; AND CONTINUE
12513 071256 005337 067740 0.B2: DEC 0,CT   ;
12514 071262 003302          BGT 0.C1    ;JUMP IF REPEAT
12515 071264 012737 000001 067740 MOV #1,0,CT  ;RESET COUNT TO 1
12516 071272 105237 071751   INCB 0,P    ;ALLOW PROCEED
12517 071276 004537 071366   JSR 5,0,SVTT ;SAVE TELETYPE STATUS, R4 IS SAFE
12518 071302 012700 000102   MOV #1,B,RO ;TYPE "B"
12519 071306 004537 071630   JSR 5,0,FTYP ;GET ADDRESS OF BREAK
12520 071312 013700 067736   MOV 0,ADR1,RO ;TYPE ADDRESS
12521 071316 004537 071466 0.B3: JSR 5,0,CADV ;CLEAR CAD
12522 071322 005005          CLR R5     ;GO TO DECODER
12523 071324 000137 070234   JMP 0.DCD  ;
12524          ;
12525          ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
12526          ;
12527 071330 012637 071746 0.SVR: MOV (SP)+,0,XXX ;PICK REGISTER FROM STACK AND SAVE
12528 071334 010637 067720   MOV SP,0,USP ;SAVE USER STACK ADDRESS
12529 071340 012706 067720   MOV #0,USP,SP ;SET TO INTERNAL STACK
12530 071344 010546          MOV R5,-(SP) ;SAVE
12531 071346 010446          MOV R4,-(SP) ;REGISTERS
12532 071350 010346          MOV R3,-(SP) ;
12533 071352 010246          MOV R2,-(SP) ;1
12534 071354 010146          MOV R1,-(SP) ;THRU
12535 071356 013746 071746   MOV 0,XXX,-(SP) ;5
12536 071362 005746          TST -(SP)   ;PUT SAVED REGISTER ON STACK
12537 071364 000200          RTS RO     ;
12538          ;
12539          ; SAVE TELETYPE STATUS
12540          ;
12541 071366 113737 177560 071752 0.SVTT: MOV 0,RCSR,0,CSR1 ;SAVE R C/SR
12542 071374 113737 177564 071753   MOV 0,TCSR,0,CSR2 ;SAVE T C/SR
12543 071402 105037 177560   CLRB 0,RCSR ;CLEAR ENABLE AND MAINTENANCE
12544 071406 105037 177564   CLRB 0,TCSR ;BITS IN BOTH C/SR
12545 071412 004537 071722   JSR 5,0,CRLF ;TYPE <CR,LF>
12546 071416 000205          RTS R5     ;
12547          ;
12548          ; RESTORE TELETYPE STATUS
12549          ;

```

```

12550 071420 004537 071722      0.RSTT: JSR      5,0.CRLF          ;<CR,LF> BEFORE RESTORING
12551 071424 105737 177564      TSTB      0.TCSR          ;WAIT READY ON PRINTER
12552 071430 100375                BPL      -4
12553 071432 032737 004000 177560 BIT      #4000,0.RCSR      ;CHECK BUSY FLAG ON READER
12554 071440 001403                BEQ      0.RSE1          ;SKIP READY LOOP IF NOT BUSY
12555 071442 105737 177560      TSTB      0.RCSR          ;WAIT READY
12556 071446 100375                BPL      -4              ;ON READER
12557 071450 113737 071752 177560 0.RSE1: MOV      0.CSR1,0.RCSR    ;RESTORE
12558 071456 113737 071753 177564 MOV      0.CSR2,0.TCSR    ; THE STATUS REGISTERS
12559 071464 000205                RTS      R5
12560
12561      ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
12562      ; WORD IS IN R0
12563
12564 071466 010246                0.CADV: MOV      R2,-(SP)      ;SAVE R2
12565 071470 012704 072011      MOV      #0,BUF+6,R4      ;BUFFER START ADDRESS
12566 071474 012746 000060      MOV      #'0,-(SP)        ;CONSTANT ASCII 0
12567 071500 010002                0.SPC: MOV      R0,R2          ; GET
12568 071502 042702 177770      BIC      #177770,R2        ;OCTAL CHARACTER
12569 071506 061602                ADD      @SP,R2            ;CONVERT TO ASCII
12570 071510 110244                MOV      R2,-(R4)         ;STORE IN BUFFER
12571 071512 006200                ASR      R0                ;SHIFT THIS MESS
12572 071514 006200                ASR      R0                ; RIGHT
12573 071516 006200                ASR      R0                ; THREE WHOLE PLACES
12574 071520 020427 072004      CMP      R4,#0.BUF+1      ;DONE?
12575 071524 101365                BHI      0.SPC            ;NO
12576 071526 042700 177776      BIC      #177776,R0        ;GET LAST BIT
12577 071532 062600                ADD      (SP)+,R0         ;CONVERT TO ASCII
12578 071534 110044                MOV      R0,-(R4)         ;AND PUT IT AWAY
12579 071536 012703 072011      MOV      #0,BUF+6,R3      ;LWA
12580 071542 004537 071614      JSR      5,0.TYPE         ;TYPE WHOLE STRING OF CHARACTERS
12581 071546 012602                MOV      (SP)+,R2         ;RESTORE R2
12582 071550 000205                RTS      R5
12583
12584      ; GENERAL CHARACTER INPUT ROUTINE
12585      ; CHARACTER INPUT GOES TO R0
12586
12587 071552 105737 177560      0.GET: TSTB      0.RCSR          ;WAIT FOR
12588 071556 100375                BPL      -4              ; INPUT FROM KEYBOARD
12589 071560 113700 177562      MOV      0.RDB,R0         ;GET A CHARACTER
12590 071564 004537 071630      JSR      5,0.F1YP         ;ECHO CHARACTER
12591 071570 042700 177600      BIC      #177600,R0        ;STRIP OFF PARITY FROM CHARACTER
12592 071574 001766                BEQ      0.GET           ;IGNORE NULLS
12593 071576 122700 000040      CMP      #40,R0           ;CHECK FOR SPACES
12594 071602 001763                BEQ      0.GET           ;IGNORE NULLS
12595 071604 122700 000073      CMP      #'',R0          ;CHECK FOR SEMI-COLON
12596 071610 001760                BEQ      0.GET           ;IGNORE THEM IF FOUND
12597 071612 000205                RTS      R5
12598
12599      ; GENERAL CHARACTER OUTPUT ROUTINE
12600      ; ADDRESS OF FIRST BYTE IN R4,
12601      ; ADDRESS OF LAST BYTE IN R3,(R3)>(R4)
12602
12603 071614 020304                0.TYPE: CMP      R3,R4          ;CHECK FOR COMPLETION
12604 071616 103426                BLO      0.TYP1          ; EXIT WHEN DONE
12605 071620 112400                MOV      (R4)+,R0        ;GET A CHARACTER

```

```

12606 071622 004537 071630      JSR      5,0.FTYP      ;TYPE ONE CHARACTER
12607 071626 000772              BR        0.TYPE      ;LOOP UNTIL DONE
12608
12609      ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
12610
12611 071630 105737 177564      0.FTYP: TSTB      0.TCSR      ;CHECK STATUS
12612 071634 100375              BPL      -4           ;WAIT UNTIL READY
12613 071636 110037 177566      MOV      RO,0.TDB    ;TYPE ONE CHARACTER
12614 071642 120037 000045      CMP      RO,#45     ;IS CHAR TO BE FILLED?
12615 071646 001012              BNE      0.TYP1     ;NO
12616 071650 113746 000044      MOV      #44,-(SP)  ;YES, INIT THE COUNT
12617 071654 105737 177564      0.TYP2: TSTB      0.TCSR
12618 071660 100375              BPL      0.TYP2
12619 071662 105037 177566      CLRB     0.TDB      ;GENERATE NULL FILLER
12620 071666 105316              DECB    #SP
12621 071670 003371              BGT      0.TYP2
12622 071672 005726              TST     (SP)+
12623 071674 000205      0.TYP1: RTS      R5      ;POP STACK
12624
12625      ; CLOSE WORD OR BYTE AND EXIT,
12626      ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
12627
12628 071676 006205      0.TCLS: ASR      R5      ;GET LOW ORDER BIT
12629 071700 103405              BCS     0.TC        ;JUMP IF ALREADY CLOSED
12630 071702 006305              ASL     R5
12631 071704 005702              TST     R2
12632 071706 001401              BEQ     0.CLS1     ;IF NO NUMBER WAS TYPED THERE IS
12633 071710 010415              MOV     R4,R5     ;NO CHANGE TO THE OPEN CELL
12634 071712 000207      0.CLS1: RTS      PC        ;STORE WORD
12635 071714 005746      0.TC:   TST     -(SP)   ;POP EXTRA CELL FROM STACK
12636 071716 000137 070220      JMP     0.ERR     ;AND SCREAM BLOODY MURDER
12637
12638      ; 0.CRLF - TYPE <CR,LF>
12639      ; 0.CRLS - TYPE <CR,LF>*
12640
12641 071722 012703 071757      0.CRLF: MOV     #0.CR+1,R3 ;LWA <CR,LF>
12642 071726 000402              BR     0.CRS
12643 071730 012703 071760      0.CRLS: MOV     #0.CR+2,R3 ;LWA <CR,LF>*
12644 071734 012704 071756      0.CRS:  MOV     #0.CR,R4  ;FWA
12645 071740 004537 071614      JSR     5,0.TYPE   ;TYPE SOMETHING
12646 071744 000205              RTS    R5
12647
12648 071746 000000      0.XXX:  .WORD   0      ;TEMPORARY STORAGE
12649 071750 000          0.T:    .BYTE   0      ;T-BIT FLAG
12650 071751 000          0.P:    .BYTE   0      ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
12651
12652 071752 000          0.CSR1: .BYTE   0      ;SAVE CELL - R C/SR
12653 071753 000          0.CSR2: .BYTE   0      ;SAVE CELL - T C/SR
12654
12655
12656 071754 042502      0.BD:  .EVEN   "BE
12657
12658 071756 015          0.CR:  .BYTE   015     ; <CR>
12659 071757 012          .BYTE   012     ; <LF>
12660 071760 052          .BYTE   '*      ; *
12661

```

12662 071761 057
 12663 071762 015
 12664 071763 044
 12665 071764 107
 12666 071765 012
 12667 071766 137
 12668 071767 136
 12669 071770 117
 12670 071771 127
 12671 071772 105
 12672 071773 102
 12673 071774 120
 12674 000014
 12675
 12676 071775 123
 12677 071776 120
 12678 071777 115
 12679 072000 000
 12680 072001 000
 12681 072002 102
 12682 000006
 12683
 12684 072003
 12685 072011 072011
 12686 072011 040
 12687
 12688
 12689 072012 000003
 12690
 12691
 12692
 12693 067704 067704
 12694 067704 000000
 12695 067706 000000
 12696 067710 000000
 12697 067712 000000
 12698 067714 000000
 12699 067716 000000
 12700 067720 000000
 12701 067722 000000
 12702 067724 000000
 12703 067726 000007
 12704 067730 000000
 12705 067732 000000
 12706 067734 000000
 12707
 12708
 12709
 12710
 12711 067736 000000
 12712 067740 000000
 12713 067742 000000
 12714 000001

```

O.LGCH: .BYTE  '/'      /
          .BYTE  '015    / CARRIAGE RETURN
          .BYTE  '$      $
          .BYTE  'G      G
          .BYTE  '012    <LF>
          .BYTE  '+'      +
          .BYTE  'O      O
          .BYTE  'W      W
          .BYTE  'E      E
          .BYTE  'B      B
          .BYTE  'P      P
O.CLGT =  -O.LGCH      ; TABLE LENGTH
O.TL:   .BYTE  'S      : DO
          .BYTE  'P      : NOT
          .BYTE  'M      : CHANGE
          .BYTE  'O      : THE
          .BYTE  'O      : ORDER
          .BYTE  'B      : HERE
O.LG =  -O.TL
O.BUF:  =  ; +6      ; 6 CHAR. BUFFER WITH
          .BYTE  ;      ; TRAILING BLANK
          .EVEN:
O.TRTC: TRT      ; TRACE TRAP PROTOTYPE
; THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
O.URD:  =  O.ODT-40
          000000 : USER R0
          000000 : R1
          000000 : R2
          000000 : R3
          000000 : R4
          000000 : R5
O.USP:  000000 : USER SP
O.UPC:  000000 : USER PC
O.UST:  000000 : USER ST
O.PRI:  7      : ODT PRIORITY
O.MSK:  000000 : 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	2314	2355	2369#		
ACDW1 = 000000	2314	2357			
ACDW2 = 000000	2314	2358			
ACLO = 000010	1377#				
ACPUOP = 000000	2314	2329			
ACT11 = 003436	2533#	3826*			
ADDW0 = 000000	2314	2359			
ADDW1 = 000000	2314	2360			
ADDW10 = 000000	2314				
ADDW11 = 000000	2314				
ADDW12 = 000000	2314				
ADDW13 = 000000	2314				
ADDW14 = 000000	2314				
ADDW15 = 000000	2314				
ADDW2 = 000000	2314	2361			
ADDW3 = 000000	2314	2362			
ADDW4 = 000000	2314	2363			
ADDW5 = 000000	2314	2364			
ADDW6 = 000000	2314	2365			
ADDW7 = 000000	2314	2366			
ADDW8 = 000000	2314				
ADDW9 = 000000	2314				
ADEVCT = 000000	2314	2320			
ADEVN = 000000	2314	2356			
AENV = 000000	2314	2325			
AENVN = 000000	2314	2326			
AFATAL = 000000	2314	2317			
AMADR1 = 000000	2314	2342			
AMADR2 = 000000	2314	2346			
AMADR3 = 000000	2314	2349			
AMADR4 = 000000	2314	2352			
AMAMS1 = 000000	2314	2336			
AMAMS2 = 000000	2314	2344			
AMAMS3 = 000000	2314	2347			
AMAMS4 = 000000	2314	2350			
AMSGAD = 000000	2314	2322			
AMSGLG = 000000	2314	2323			
AMSGTY = 000000	2314	2316			
AMTYP1 = 000000	2314	2337			
AMTYP2 = 000000	2314	2345			
AMTYP3 = 000000	2314	2348			
AMTYP4 = 000000	2314	2351			
APASS = 000000	2314	2319			
APRIOR = 000000	2314				
APTC SU = 000040	10145	10317#			
APTENV = 000001	8902	10092	10138	10273	10315#
APTSIZ = 000200	3754	10314#			
APTSP0 = 000100	10140	10275	10316#		
ASWREG = 000000	2314	2327			
ATESTN = 000000	2314	2318			
ATTN = 003312	2465#	5320	9109	9128	9154
AUNIT = 000000	2314	2321			
AUSWR = 000000	2314	2328			
AVECT1 = 000000	2314	2353			
AVECT2 = 000000	2314	2354			
BADHDR = 003306	2456#	3834*	5704*	7349*	9857 9905*

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 248
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0247

DM39	065301	3217	3222	3493	3498	3533	3538	3553	3558	3589	11774#				
DM40	065330	11778#	12010												
DM41	065364	3272	3277	3282	3287	3292	3398	3574	11783#						
DM42	065405	3352	11786#												
DM44	065445	3362	11792#	12038											
DM45	065515	3177	11799#												
DM46	065574	3212	11807#												
DM48	065641	3403	3408	3413	3418	3423	3428	3433	3438	3443	3448	3453	11814#		
DM49	065673	11819#	11948												
DMS1	065747	3127	3132	11827#											
DMS2	065776	11831#	12062												
DMS3	066026	3337	11835#												
DM6	064125	11658#	11979												
DM8	064160	2644	2649	2654	2659	2664	2669	2807	11663#						
DM9	064204	2812	2817	2822	2827	2982	3569	3609	11667#						
DI	= 040000	1332#													
DISPLA	001142	2289#	3742*	3750*	10057*	10079*									
DISPRE	000174	1478#	3750												
DLT	= 100000	1351#	5793	5928	6057	6221	6374	6532	7454	7626	7843	7968	8179	8304	
		8529	8654												
DLY	= 044220	3997	4110	9172#	9175										
DMD	= 000040	1392#													
DOCMD	043434	4285	4416	4593	4647	4775	4839	4876	4909	4987	5024	5065	5086	5111	
		5147	5198	5235	5274	5338	5344	5390	5424	5514	5533	5558	5629	5665	
		5881	5937	6115	6168	6248	6429	6490	6557	6635	6686	6735	6910	6929	
		6954	7025	7061	7094	7201	7257	7361	7424	7520	7573	7659	7744	7791	
		7868	7916	7999	8080	8127	8204	8252	8335	8397	8430	8477	8554	8602	
		8685	8755	8782	8824	8830	9028#	9327	9848	9864	9879				
		8057	8059#												
		2551#	3856*	3865*	4616										
		1374#	4044	4147											
DOSEEK	036370	1381#													
DOTIM	003470	2535#	3942*	3949*	3966	3981*	4011*	4030	4134*	4200	8870	8969*			
DRA	= 000001	2540#	3838	3944	3983	4096	8952								
DRDY	= 000200	2541#													
DRIVS	003442	2542#													
DRIV0	003444	2543#													
DRIV1	003446	2544#													
DRIV2	003450	2545#													
DRIV3	003452	2546#													
DRIV4	003454	2547#													
DRIV5	003456	1379#													
DRIV6	003460	1358#													
DRIV7	003462	1338#	4004	4117											
DROT	= 000040	2389#	3838*	4205	4223*										
DRPAR	= 000010	1385#													
DRVMSK	= 000007	1190#	2288	3741											
DRVPT	001346	1367#													
DSC	= 040000	1360#	4036	4138	4486	4521									
DSWR	= 177570	2572	2578	2584	2590	2595	2601	2607	2613	2619	2625	2630	2635	2640	
DTE	= 010000	2676	2681	2687	2693	2698	2703	2708	2713	2718	2723	2728	2733	2738	
DTYPE	= 000040	2743	2748	2803	2853	2858	2883	2908	2933	2968	2973	2978	2983	2988	
DT1	066076	2993	2998	3003	3008	3013	3018	3023	3028	3063	3068	3073	3078	3098	
		3103	3108	3118	3123	3128	3133	3138	3163	3168	3173	3178	3183	3193	
		3198	3213	3218	3223	3258	3263	3298	3313	3318	3323	3328	3333	3338	
		3348	3353	3363	3373	3378	3399	3404	3469	3565	3570	3575	3590	3595	

H04

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 254
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0253

		5872	5930	5961	6010	6059	6223	6327	6376	6433	6478	6534	6818	6957
		7016	7158	7303	7456	7537	7628	7845	7970	8181	8306	8531	8656	8838
		9135	9139	9159	9163	9324*	9394	9405	9459	9474	9531	9548	9565	9581
		9868												
GSTAT1	045130	9193	9336*											
GSTAT2	045360	9341	9349	9357	9364	9373*								
GTSWR =	104406	8936	10928*											
HASOF	003340	2483*	9093*	9109	11849	11857	11865	11873	11881	11889	11897	11908	11920	11933
HBA	003330	2479*	9089*	11849	11857	11865	11873	11881	11889	11897	11908	11920	11933	
HCS1	003322	2476*	3993*	3994*	3995	4001	4106*	4107*	4108	4114	4284*	4415*	4449	4524
		4549	4568	4571	4592*	4646*	4774*	4838*	4875*	4908*	4986*	5023*	5064*	5085*
		5110*	5146*	5197*	5234*	5273*	5337*	5343*	5389*	5396	5423*	5513*	5532*	5557*
		5628*	5664*	5724*	5728	5779*	5782	5869*	5873	5880*	5914*	5917	5936*	6007*
		6011	6044*	6047	6114*	6140	6167*	6207*	6210	6247*	6254	6324*	6328	6361*
		6364	6428*	6457	6489*	6518*	6521	6556*	6563	6634*	6658	6685*	6734*	6741
		6815*	6819	6909*	6928*	6953*	7024*	7060*	7067	7093*	7155*	7159	7200*	7229
		7256*	7300*	7304	7360*	7367	7423*	7430	7440*	7443	7519*	7545	7572*	7612*
		7615	7658*	7665	7743*	7763	7790*	7829*	7832	7867*	7888	7915*	7954*	7957
		7998*	8005	8079*	8099	8126*	8165*	8168	8203*	8224	8251*	8290*	8293	8334*
		8341	8396*	8403	8429*	8449	8476*	8515*	8518	8553*	8574	8601*	8640*	8643
		8684*	8691	8754*	8761	8781*	8823*	8829*	9028*	9029	9039*	9040	9086*	9300
		9304	9326*	9373*	9374*	9375	9395	9847*	9863*	9878*	11846	11854	11862	11870
HCS2	003324	11878	11886	11894	11905	11917	11930							
		2477*	4040	4042	4046	4058	4143	4145	4149	4152	4435	4452	9087*	9309
		9313	11846	11854	11862	11870	11878	11886	11894	11905	11917	11930		
HDA	003332	2480*	9090*	11849	11857	11865	11873	11881	11889	11897	11908	11920	11933	
HDB	003344	2485*												
HDC	003342	2484*												
HDS	003334	2481*	9094*	11849	11857	11865	11873	11881	11889	11897	11908	11920	11933	
		11886	4044	4147	4475	4483	4510	4518	9091*	11846	11854	11862	11870	11878
HDTAB	001470	2447*	5714	5722*	5996	6071	6313	6388	7149	7295	9599			
HDWD	001454	2435*	5836*	5837	5973*	5974	6073*	6075	6234*	6236	6390*	6392	11868	
HD1	001434	2425*	9601*	9602*	9603*	9604*	9605*	9606*	9613	9613				
HEAD	001430	2423*	5705*	5716*	5717	5718*	5720	5761*	5762	5765*	5774*	5775	5776*	5826*
		5827	5828*	5833	5844*	5845	6001*	6318*	7144*	7151*	7152	7153*	7166*	7167
		7174*	7196*	7197	7198*	7292*	7293	7294*	7331*	7332	7338	7340*	7415*	7416
		9601	9676											
HEADA	001432	2424*	4348	5258	5321	9518*	9519*	9520*	9521*					
HER	003336	2482*	4036	4138	4486	4521	9092*	11846	11854	11862	11870	11878	11886	11894
HMR1	003346	11905	11917	11930										
HMR2	003350	2486*	9095*	11846	11854	11862	11870	11878	11886	11894	11905	11917	11930	
		2487*	4003	4116	4319	4373	4406	4419	4472	4480	4507	4515	4597	4677
		4691	4718	4744	4760	4844	4992	5203	5565	5883	5939	6434	6961	8793
		8796	9073*	9078*	9096*	9343	9351	9359	9366	9460	9549	9566	9582	9869
HMR3	003352	11846	11854	11862	11870	11878	11886	11894	11905	11917	11930			
		2488*	4546	4562	4565	5353	8787	8790	8839	9074*	9079*	9097*	9344	9352
		9360	9367	9406	9475	9532	11846	11854	11862	11870	11878	11886	11894	11905
		11917	11930											
HOLD	043664	9058	9062	9086*	9337									
HPAT	003356	2490*	9099*	11849	11857	11865	11873	11881	11889	11897	11908	11920	11933	
HPEN	050006	7389	7467	9905*										
HPEND	003310	2458*	3835*	7344	7348*	9859*	9906							
HPOS	003354	2489*	9098*	11849	11857	11865	11873	11881	11889	11897	11908	11920	11933	
HT	= 000011	1182*	10153	10194										
HVRC	= 000400	1363*												
HWC	003326	2478*	9088*	11849	11857	11865	11873	11881	11889	11897	11908	11920	11933	

O.CRLF	071722	12353	12410	12506	12545	12550	12641#			
O.CRLS	071730	12290	12643#							
O.CRS	071734	12642	12644#							
O.CSR1	071752	12541#	12557	12652#						
O.CSR2	071753	12542#	12558	12653#						
O.CT	067740	12470#	12513#	12515#	12712#					
O.C1	071070	12473#	12514							
O.DCD	070234	12278	12280	12290#	12347	12398	12523			
O.DCD1	070240	12291#	12339	12383						
O.EFF	070544	12324	12389#							
O.EFF1	070662	12401	12420#							
O.ENTR	067750	12217#								
O.ERR	070220	12264	12275	12287#	12310	12335	12393	12636		
O.ERR1	070554	12393#	12466							
O.ERR2	070374	12335#	12369	12442						
O.FTYP	071630	12289	12357	12372	12414	12519	12590	12606	12611#	
O.GET	071552	12243	12293	12587#	12592	12594	12596			
O.GO	070726	12318	12440#							
O.GO2	071004	12451#	12476							
O.LG =	000006	12247	12682#							
O.LGCH	071761	12306	12662#	12674						
O.LGDR	070334	12313	12315#	12327						
O.LGL =	000030	12327#								
O.LGL1	070306	12306#	12311							
O.LGL2	070326	12307	12312#							
O.MSK	067730	12394	12395	12397	12704#					
O.ODT	067744	1489	12215#	12693						
O.OFST	070472	12322	12368#							
O.OF1	070540	12381	12383#							
O.OP1	070430	12319	12351#							
O.OP2	070436	12268	12353#	12364						
O.ORPC	070142	12260#	12320							
O.P	071751	12235#	12440#	12465	12467#	12516*	12650#			
O.PRI	067726	12227	12489	12493	12703#					
O.PROC	071034	12326	12465#							
O.PRI	071056	12469	12471#							
O.RALL	070210	12238	12273	12279#						
O.RCSR =	177560	12205#	12541	12543#	12553	12555	12557*	12587		
O.RDB =	177562	12204#	12589							
O.REGT	070072	12243#	12317							
O.RSE1	071450	12554	12557#							
O.RSP	070102	12245#	12248							
O.RST	070006	12216	12225#							
O.RSTT	071420	12446	12472	12550#						
O.RST1	070036	12224	12232#							
O.RTIT	071032	12234#	12460#							
O.SCAN	070244	12254	12293#	12304						
O.SP	070134	12246	12255#							
O.SPC	071500	12567#	12575							
O.SP1	070122	12251#	12256							
O.STM =	000340	12194#	12236	12445	12471	12473				
O.STRT	067774	12215	12222#							
O.SVR	071330	12225	12485	12527#						
O.SVTT	071366	12505	12517	12541#						
O.T	071750	12447#	12474#	12486	12649#					
O.TBIT	070756	12447#	12487							

PURVEC=	000024	1278#	3729*	3730*	9978*	9987*	9988*							
RCYLA	046136	9280	9498#											
RCYLD	046064	9276	9485#											
RDCHR =	104410	10596	10931#											
RDCYLA	046014	5071	5117	5378	9473#									
RDCYLD	045730	5076	5126	5132	5137	9458#								
RDDATA=	000021	1317#	6815											
RDGATE=	100000	1402#												
RDHEAD=	000025	1319#	5779	5914	6044	6207	6361	6518	7440	7612	7829	7954	8165	8290
		8515	8640											
RDLIN =	104411	8947	10670	10932#										
RDOCT =	104412	8992	9005	10933#										
RDSEC	045504	9404#	9421	9424	9442	9445	9692							
RDY =	000200	1325#	9054	9069										
RECAL =	000013	1314#	4838	4986	5197	5337	6428	8823	9863					
RESREG=	104414	10730	10800	10935#	12163									
RESVEC=	000010	1273#												
RHEAD	046232	9285	9518#											
RHTAB	001674	2448#	5778	5830	5913	5969	6042	6064	6206	6230	6234	6359	6381	6517
		6540	6544	7439	7462	7611	7634	7828	7851	7953	7976	8164	8187	8289
		8312	8514	8537	8639	8662	9699	9701	9705	9708	11884	11892		
RKASOF=	000016	1296#	3899	9093	9128	9154								
RKBA =	000004	1291#	3895	5714#	5865#	5996#	6313#	6809#	6851#	6870#	7149#	7295#	9089	
RKCS1 =	000000	1289#	3892	3995#	4108#	4499#	4501#	4619#	4644#	4671#	4684#	4757#	4836#	4873#
		4984#	5021#	5069#	5083#	5115#	5144#	5195#	5232#	5335#	5341#	5357#	5421#	5530#
		5560#	5626#	6165#	6431#	6487#	6683#	6926#	6956#	7022#	7254#	7570#	7788#	7913#
		8124#	8249#	8474#	8599#	8821#	8827#	8843#	9029#	9040#	9054	9069	9086	9338#
		9346#	9354#	9362#	9375#	9861#	9876#							
RKCS2 =	000010	1293#	3893	3974#	3988#	3992#	4101#	4105#	4446#	4645#	4837#	4874#	4985#	5022#
		5084#	5145#	5196#	5233#	5336#	5342#	5422#	5531#	5627#	5713#	5793	5864#	5928
		6057	6166#	6221	6374	6488#	6532	6684#	6927#	7023#	7255#	7454	7571#	7626
		7789#	7843	7914#	7968	8125#	8179	8250#	8304	8475#	8529	8600#	8654	8822#
		8828#	9087	9325#	9334#	9347#	9355#	9363#	9388#	9392#	9862#	9877#		
RKDA =	000006	1292#	3896	5270#	5717#	5775#	5827#	6811#	6853#	6872#	7152#	7197#	7293#	7420#
		9090												
RKDB =	000024	1298#	3901	5788	5789	5790	5923	5924	5925	6053	6054	6055	6216	6217
		6218	6370	6371	6372	6527	6528	6529	7449	7450	7451	7621	7622	7623
		7838	7839	7840	7963	7964	7965	8174	8175	8176	8299	8300	8301	8524
		8525	8526	8649	8650	8651								
RKDC =	000020	1297#	3900	4907#	5055#	5070#	5101#	5116#	5512#	5556#	6005#	6040#	6111#	6204#
		6322#	6357#	6554#	6632#	6732#	6813#	6908#	6952#	7058#	7195#	7297#	7358#	7517#
		7608#	7656#	7741#	7825#	7864#	7950#	7996#	8077#	8161#	8200#	8286#	8332#	8394#
		8427#	8511#	8550#	8636#	8682#	8780#	9094						
RKDS =	000012	1294#	3897	9091										
RKECPS=	000030	1302#	3905	9098										
RKECPT=	000032	1303#	3906	9099										
RKER =	000014	1295#	3898	9092										
RKMR1 =	000026	1299#	3902	4339#	4371#	4404#	4560#	4675#	4689#	4716#	4742#	4758#	4842#	4866#
		4906#	4990#	5014#	5054#	5100#	5201#	5225#	5306#	5511#	5619#	5751#	5817#	5960#
		6432#	6477#	6907#	7015#	7536#	9095	9134#	9138#	9158#	9162#	9340#	9348#	9356#
		9393#	9404#	9458#	9473#	9530#	9547#	9564#	9580#	9867#				
RKMR2 =	000034	1300#	3903	9073	9078	9096								
RKMR3 =	000036	1301#	3904	9074	9079	9097								
RKPRI	001316	2371#	9021											
RKVEC	001314	2370#	3829#	9009#	9012#	9019								
RKWC =	000002	1290#	3894	5715#	5866#	5997#	6314#	6814#	7150#	7296#	9088			

N04

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 260
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0259

RLS = 000010	1339#	4446															
RSEC = 046174	9284	9508#															
RTT = 000006	12197#																
SAVREG= 104413	10715	10774	10934#	12107													
SBPAR = 047324	9204	9208	9212	9216	9220	9224	9228	9768#									
SCLR = 000040	1341#	3974	3988	4101	9388												
SCOP1 = 104415	3985	4098	4409	4440	4491	4554	4829	4899	4977	5047	5094	5187	5263				
	5383	5707	5766	5906	6034	6197	6351	6625	7184	7284	7510	7600	7729				
	7818	7857	7943	8065	8154	8193	8279	8415	8504	8543	8629	8773	10936#				
SCOP1\$ = 047376	9792#	10936															
SDC = ***** U	1169	1488	12173														
SEC = 001374	2404#	4673#	4687*	4714*	4740*	9744*											
SECFLG = 046730	9619	9640#															
SECNT = 001400	2406#	5461*	5483*	5904*	6033*	6350*											
SECTOR = 001406	2409#	5471	5480	9406*	9407*	9408*	9409*	9410*	9411*	9422	9425	9443	9446				
	9508#	9509*	9510*	9511*	9512*	9513*	9693*	9697	9698*	9699*	9710	9720*	9721				
	9722#	9723*															
SEEK = 000017	1316#	4908	5064	5110	5273	5389	5513	5557	6114	6247	6556	6634	6734				
	6909	6953	7060	7200	7360	7423	7519	7658	7743	7867	7998	8079	8203				
	8334	8396	8429	8553	8684	8754	8781										
	1309#	3993	4106	4499	4501	5880	5936	9326	9373								
SELDRV= 000001	3830	9010	9019#														
SETINT = 043416	2552#	3831*	3978	8965*	8985*												
SIZFLG = 003472	1356#																
SKI = 000002	6067	6384	9690#														
SORT = 047056	12141	12150	12161	12172#													
SPACE2 = 067660	1313#	4592	5664	7093	9847												
SRTSPL= 000011	2449#	6070	6387	9702													
SRTTAB = 002100	12191#	12217	12218*	12231*	12445*	12471*	12473*	12499*									
ST = 177776	1177#	3694	3721	3883	3934	3986	4094	4099	4196	4277	4313	4363	4402				
STACK = 001100	4410	4431	4441	4466	4492	4542	4555	4587	4823	4830	4900	4959	4978				
	5048	5095	5183	5188	5264	5374	5384	5455	5502	5697	5708	5767	5860				
	5907	5989	6035	6105	6198	6306	6352	6420	6616	6626	6800	6898	7122				
	7185	7285	7499	7511	7601	7723	7730	7819	7858	7944	8054	8066	8155				
	8194	8280	8388	8416	8505	8544	8630	8749	8774	8868	9290	9959	9990				
	12169																
START = 007120	1481	3690#															
START1 = 007536	3783#																
STKLMT= 177774	1188#																
STOP = 047500	7350	9830#	10437	10520													
ST2 = 007606	3791	3802#															
ST3 = 007652	3792	3820#															
ST4 = 007704	3804	3825	3828#														
ST5 = 007730	3811	3833#	3972	4034	8908	9898											
ST5XY = 043044	8897	8902#															
SUBCLR = 045424	4281	4290	4315	4367	4412	4433	4443	4468	4494	4544	4557	4589	4771				
	4780	4787	4832	4902	4980	5050	5097	5190	5266	5376	5386	5457	5504				
	5591	5661	5710	5769	5862	5909	5991	6037	6106	6200	6244	6258	6308				
	6354	6422	6512	6552	6567	6618	6628	6729	6745	6753	6802	6838	6900				
	6987	7055	7071	7079	7090	7136	7187	7287	7355	7371	7379	7434	7513				
	7603	7653	7669	7677	7732	7821	7860	7946	7993	8009	8017	8068	8157				
	8196	8282	8329	8345	8353	8391	8407	8418	8507	8546	8632	8679	8695				
	8703	8751	8765	8776	9388#	9832	9844										
SVAL = 100000	1386#																
SWR = 001140	2288#	3719	3741*	3743	3749*	3756*	4800	4805	5674	6289	6294	6598	6603				
	6766	6772	7103	7404	7410	7476	7481	7690	7696	8028	8033	8038	8043				

		8364	8369	8374	8379	8714	8719	8724	8729	8934	9240	9250	9260	9270
		9792	9813	9818	9820	9922	9930	9935	9955	10013	10027	10029	10035	10042
		10080	10087	10099	10103	10440	10479	10534*	12165					
SWREG	000176	1479#	3749	8934	10440	10479	10502							
SWTST	047430	4765	5651	6240	6548	6720	7047	7392	7470	7645	7987	8323	8673	9811#
SW0	= 000001	1241#												
SW00	= 000001	1231#	1241											
SW01	= 000002	1230#	1240											
SW02	= 000004	1229#	1239											
SW03	= 000010	1228#	1238											
SW04	= 000020	1227#	1237											
SW05	= 000040	1226#	1236											
SW06	= 000100	1225#	1235											
SW07	= 000200	1224#	1234											
SW08	= 000400	1223#	1233											
SW09	= 001000	1222#	1232											
SW1	= 000002	1240#												
SW10	= 002000	1221#												
SW11	= 004000	1220#												
SW12	= 010000	1219#	12165											
SW13	= 020000	1218#	9922											
SW14	= 040000	1217#	9813	9935										
SW15	= 100000	1216#												
SW2	= 000004	1239#												
SW3	= 000010	1238#												
SW4	= 000020	1237#												
SW5	= 000040	1236#												
SW6	= 000100	1235#												
SW7	= 000200	1234#												
SW8	= 000400	1233#	9818											
SW9	= 001000	1232#	4800	4805	5674	6289	6294	6598	6603	6766	6772	7103	7404	7410
		7476	7481	7690	7696	8028	8033	8038	8043	8364	8369	8374	8379	8714
		8719	8724	8729	9240	9250	9260	9270	9792	9930	9955			
TBITVE=	000014	1274#												
TEMP1	003360	2492#	3975#	3989#	3996#	3998#	4003#	4004#	4005	4102#	4109#	4111#	4116#	4117#
		4118	4502#	5319#	5320#	5321	5392#	5835#	5837	5972#	5974	6074#	6075	6135#
		6235#	6236	6251#	6391#	6392	6454#	6560#	6654#	6738#	7064#	7225#	7364#	7427#
		7541#	7662#	7759#	7884#	8002#	8095#	8220#	8338#	8400#	8445#	8570#	8688#	8758#
		9030#	9041#	9056#	9071#	9126#	9130#	9156#	9172	9174#	9201	9203#	9205	9207#
		9209	9211#	9213	9215#	9217	9219#	9221	9223#	9225	9227#	9229	9231#	9324
		9329#	9336	9369#	9376#	9389#	9419	9420#	9427#	9429#	9431#	9440	9441#	9448#
		9450#	9452#	9529#	9534#	9546#	9551#	9563#	9568#	9579#	9584#	9621#	9624#	9625
		9774#	9782#	11868										
TEMP2	003362	2493#	4293#	4620#	4783#	4847#	4995#	5206#	5358#	5593#	5598#	5668#	6804#	6841#
		6842	6868#	6989#	6994#	7097#	8844#	9132#	9553#	9586#	9622#	9623#	9625#	9626
		9851#	9872#											
TEMP3	003364	2494#	5057#	5059	5103#	5105	6805#	6844	6865#	6866	7736#	7738	8072#	8074
		8422#	8424#	8425										
TEMP4	003366	2495#	5058#	5059#	5061#	5062	5104#	5105#	5107#	5108	6808#	6809	6851	6869#
		6870	7737#	7738#	7739	8073#	8074#	8075	8423#	8424				
TEMP5	003370	2496#	4826#	4941#	6810#	6811	6852#	6853	6871#	6872				
TIMUP	001376	2405#	4679	4693	4720	4746	9729#	9740#	9746#					
TITLE	043116	3785	3823	8925#										
TKVEC =	000060	1281#	10417#	10418#										
TOCYL	001352	2392#	4973#	5055	5056	5070	5072	5122	5124	5155#	5156	5163#	5998#	6000
		6109#	6315#	6317	6425#	6621#	6632	6708	6712	6715#	6732	6748	6750#	7058

COS

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 262
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0261

		7074	7076*	7141*	7143	7179*	7343*	7358	7374	7376*	7421*	7462	7507*	7517
		7595	7608	7634	7641*	7656	7672	7674*	7726*	7737	7741	7813	7825	7851
		7982	7984*	7996	8012	8014*	8062*	8073	8077	8149	8161	8187	8318	8320*
		8332	8348	8350*	8412*	8423	8427	8499	8511	8537	8668	8670*	8682	8698
		8700*	8770*	8779	8780	8812	8815	8849*	8850	11852	11876	11884		
TPVEC =	000064	1282#												
TRAPPC	001334	2382#	9951*	11900										
TRAPVE=	000034	128J#	3727*	3728*	9989*									
TRT =	000003	12196#	12689											
TRTVEC=	000014	1275#												
TSTATN	044012	4287	4602	4649	4698	4725	4751	4777	4878	4911	5026	5067	5088	5113
		5149	5237	5276	5346	5426	5516	5535	5587	5631	6170	6492	6688	6912
		6931	6983	7027	7259	7575	7793	7918	8129	8254	8479	8604	8784	8832
		9107#	9881											
TST1	010120	3859	3881#											
TST10	012524	4366	4400#											
TST11	012622	4407	4420	4429#										
TST12	012730	4453	4464#											
TST13	013224	4525	4540#											
TST14	013370	4572	4585#											
TST15	014712	4766	4821#											
TST16	015524	4945	4957#											
TST17	016766	5162	5181#											
TST2	010302	3910	3932#											
TST20	020006	5372#												
TST21	020352	5447	5453#											
TST22	020476	5500#												
TST23	021556	5652	5695#											
TST24	022476	5846	5858#											
TST25	023256	5987#												
TST26	023732	6103#												
TST27	025026	6241	6304#											
TST3	011126	4092#												
TST30	025502	6418#												
TST31	026602	6549	6614#											
TST32	027502	6721	6798#											
TST33	030140	6857	6876	6896#										
TST34	031342	7048	7118#											
TST35	033412	7393	7471	7497#										
TST36	034470	7646	7721#											
TST37	036342	7988	8052#											
TST4	011530	4128	4134#											
TST40	040226	8324	8386#											
TST41	042164	8674	8747#											
TST5	012076	4244	4254	4275#										
TST6	012170	4311#												
TST7	012366	4349	4361#											
TYPDS =	104405	8886	10926#											
TYPE =	104401	3789	3805	3807	3809	3824	3866	3941	3951	3980	4014	4060	4202	4224
		4370	4391	4392	4963	4968	4970	5584	5731	5785	5797	5876	5920	5932
		6014	6050	6061	6213	6225	6331	6367	6378	6524	6536	6877	6881	6882
		6980	7125	7139	7162	7307	7446	7458	7501	7504	7618	7630	7835	7847
		7960	7972	8171	8183	8296	8308	8521	8533	8646	8658	8884	8887	8928
		8978	9181	9839	9841	9891	9896	9924	9966	9991	10082	10090	10158	10252
		10378	10434	10446	10500	10501	10504	10515	10525	10536	10555	10603	10609	10614
		10618	10623	10624	10626	10629	10633	10699	10701	10841	10922#	12117	12118	12122

CALIE	1657*	4835	4983	5194	9860										
CHECK	1570*	4331	4609	4632	4661	4705	4732	4858	4890	4924	5006	5038	5217	5249	5297
	5408	5438	5522	5547	5611	5643	5742	5808	5896	5952	6025	6121	6152	6182	6269
	6342	6442	6469	6504	6580	6641	6670	6700	6830	6918	6943	7007	7039	7207	7241
	7271	7318	7526	7557	7587	7750	7775	7805	7874	7900	7930	8086	8111	8141	8210
	8236	8266	8436	8461	8491	8560	8586	8616	8805						
CIDAE	1691*	5333	8819												
COMMEN	1284*														
CWD2	1584*	4338	5305	5750	5816	5959	6476								
DRCLR	1627*	4642	4872	5020	5081	5142	5231	5340	5419	5529	5624	6163	6486	6681	6925
	7020	7252	7568	7786	7911	8122	8247	8472	8597	8826	9875				
ENDCOM	1284*														
EOPGM	2247*	8864													
ERROR	1178*	3913	3970	3977	3991	4000	4023	4032	4051	4055	4066	4070	4104	4113	4131
	4158	4162	4166	4170	4282	4286	4288	4291	4295	4316	4334	4335	4336	4337	4343
	4346	4350	4368	4413	4417	4421	4434	4437	4444	4451	4454	4469	4474	4477	4482
	4485	4488	4495	4504	4509	4512	4517	4520	4523	4526	4545	4548	4551	4558	4564
	4567	4570	4573	4590	4594	4599	4604	4612	4613	4614	4615	4622	4635	4636	4637
	4638	4641	4648	4651	4664	4665	4666	4667	4681	4695	4700	4708	4709	4710	4711
	4722	4727	4735	4736	4737	4738	4748	4752	4762	4772	4776	4778	4781	4785	4788
	4833	4840	4846	4849	4861	4862	4863	4864	4870	4877	4880	4893	4894	4895	4896
	4903	4910	4912	4927	4928	4929	4930	4934	4938	4981	4988	4994	4997	5009	5010
	5011	5012	5018	5025	5028	5041	5042	5043	5044	5051	5066	5068	5074	5079	5087
	5090	5098	5112	5114	5120	5129	5135	5140	5148	5151	5191	5199	5205	5208	5220
	5221	5222	5223	5229	5236	5239	5252	5253	5254	5255	5260	5267	5275	5277	5301
	5302	5303	5304	5310	5313	5325	5339	5344	5348	5355	5360	5377	5381	5387	5391
	5395	5398	5411	5412	5413	5414	5417	5425	5428	5441	5442	5443	5444	5448	5458
	5464	5468	5482	5505	5515	5517	5525	5526	5527	5528	5534	5537	5550	5551	5552
	5553	5559	5563	5567	5578	5579	5580	5585	5588	5592	5595	5597	5600	5614	5615
	5616	5617	5623	5630	5633	5646	5647	5648	5649	5662	5666	5670	5700	5711	5726
	5730	5745	5746	5747	5748	5755	5758	5770	5781	5784	5796	5811	5812	5813	5814
	5821	5824	5839	5863	5871	5875	5882	5885	5899	5900	5901	5902	5910	5916	5919
	5931	5938	5941	5955	5956	5957	5958	5964	5967	5976	5992	6009	6013	6028	6029
	6030	6031	6038	6046	6049	6060	6077	6107	6116	6124	6125	6126	6127	6131	6139
	6142	6155	6156	6157	6158	6161	6169	6172	6185	6186	6187	6188	6193	6201	6209
	6212	6224	6232	6238	6245	6249	6253	6256	6259	6272	6273	6274	6275	6278	6309
	6326	6330	6345	6346	6347	6348	6355	6363	6366	6377	6394	6423	6430	6436	6445
	6446	6447	6448	6456	6459	6472	6473	6474	6475	6481	6484	6491	6494	6507	6508
	6509	6510	6513	6520	6523	6535	6542	6546	6553	6558	6562	6565	6568	6583	6584
	6585	6586	6589	6619	6629	6636	6644	6645	6646	6647	6651	6657	6660	6673	6674
	6675	6676	6679	6687	6690	6703	6704	6705	6706	6710	6730	6736	6740	6743	6746
	6754	6803	6817	6821	6833	6834	6835	6836	6839	6846	6848	6861	6901	6911	6913
	6921	6922	6923	6924	6930	6933	6946	6947	6948	6949	6955	6959	6963	6974	6975
	6976	6981	6984	6988	6991	6993	6996	7010	7011	7012	7013	7019	7026	7029	7042
	7043	7044	7045	7056	7062	7066	7069	7072	7080	7091	7095	7099	7129	7133	7137
	7157	7161	7188	7202	7210	7211	7212	7213	7217	7228	7231	7244	7245	7246	7247
	7250	7258	7261	7274	7275	7276	7277	7281	7288	7302	7306	7321	7322	7323	7324
	7356	7362	7366	7369	7372	7380	7425	7429	7432	7435	7442	7445	7457	7464	7514
	7521	7529	7530	7531	7532	7538	7544	7547	7560	7561	7562	7563	7566	7574	7577
	7590	7591	7592	7593	7597	7604	7614	7617	7629	7636	7654	7660	7664	7667	7670
	7678	7733	7745	7753	7754	7755	7756	7762	7765	7778	7779	7780	7781	7784	7792
	7795	7808	7809	7810	7811	7815	7822	7831	7834	7846	7853	7861	7869	7877	7878
	7879	7880	7887	7890	7903	7904	7905	7906	7909	7917	7920	7933	7934	7935	7936
	7940	7947	7956	7959	7971	7978	7994	8000	8004	8007	8010	8018	8069	8081	8089
	8090	8091	8092	8098	8101	8114	8115	8116	8117	8120	8128	8131	8144	8145	8146
	8147	8151	8158	8167	8170	8182	8189	8197	8205	8213	8214	8215	8216	8223	8226

CZR6HEO UNIBUS RK6 DR PRT1
CZR6HE.P11 25-JAN-78 12:00

MACY11 30A(1052) 25-JAN-78 12:08 PAGE 272
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0270

.STYPD 1131# 10194
.STYPE 1131# 10115
.STYPO 1131# 10318

. ABS. 072014 000

ERRORS DETECTED: 0

DSKZ:CZR6HE, DSKZ:CZR6HE.SEG/SOL/CRF/NL:TOC/DOC=DSKM:CZR6HE.P11
RUN-TIME: 36 32 4 SECONDS
RUN-TIME RATIO: 486/73=6.6
CORE USED: 32K (63 PAGES)

DOCUMENT PAGES: 270