

RK611/RK06

DRIVE DIAGNOSTIC PART 3
MD-11-DZR6J-D

EP-DZR6J-D-DL-A

APR 1977

COPYRIGHT © 1977

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains 336 frames of technical information, organized in a 14x24 grid. The frames are arranged in 14 columns and 24 rows. Each frame displays a small, high-contrast image of a technical diagram or data table, typical of a drive diagnostic manual. The content is too small to read in detail but appears to include various diagnostic charts, tables, and possibly flowcharts related to the drive system.

B01

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6J-D-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 3
DATE:	JANUARY 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAZIAN

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

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

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

COPYRIGHT (C) 1976,1977 BY DIGITAL EQUIPMENT CORPORATION

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97

TABLE OF CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
 - 2.1 HARDWARE
 - 2.2 PRELIMINARY TESTING AND 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
- 1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 3 OF THE DRIVE DIAGNOSTICS TO INSURE THAT THE DISK IS CAPABLE OF PROPERLY PERFORMING ALL OPERATOR INTERVENTION FUNCTIONS. ERROR DETECTION LOGIC IS CHECKED BY MANUAL & SOFTWARE ERROR FORCING.

AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF THIS PART, PRECEDED BY THE SUCCESSFUL RUN OF PARTS 1 & 2, IT CAN BE ASSERTED THAT THE RK06 DRIVE WILL WORK SUCCESSFULLY

IN THE STAND-ALONE MODE. SYSTEMS INTERACTION, & ERROR RATE ANALYSIS ARE LEFT TO OTHER PROGRAMS.

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 RK06 DRIVES

NOTES: 1. IF NEITHER KW11-L OR P CLOCK IS USED, ALL TIMING TESTS WILL BE BYPASSED. A MESSAGE 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 SUCCESSFUL FOLLOWED BY THE RK06 DRIVE DIAGNOSTICS - PARTS 1 & 2.

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

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

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
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
203
204
205
206
207
208
209

THIS PROGRAM SHOULD NOT BE CHAINED BY XXDP.

CHAIN MODE OPERATION (MONITOR)

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

DUMP MODE OPERATION (MANUAL)

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

3.3 ACT/APT

THIS PROGRAM IS ACT COMPATIBLE.
HOWEVER, IT SHOULD NOT BE RUN IN THE AUTO MODE.

AUTOMATIC MODE (MONITOR)

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

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

3.3.1 APT ETABLE DEFINITIONS

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

1. SOFTWARE ENVIRONMENT:
 - =1 IF APT SCRIPT MODE
 - =0 IF STANDALONE MODE
2. ENVIRONMENT MODE:
 - BIT 7 = 1 ETABLE DOES SIZING
 - = 0 PROGRAM DOES SIZING
 - BIT 6 = 1 SPOOL MESSAGES TO APT IF SCRIPT MODE
 - = 0 DON'T SPOOL TO APT
 - BIT 5 = 1 SUPPRESS CONSOLE OUTPUT
 - = 0 ALLOW CONSOLE OUTPUT
 - BITS 4-0 NOT USED

3. SWITCH 1 (SOFTWARE SWITCH REGISTER)
IF ENVIRONMENT MODE BIT 7 (SIZING BIT) IS SET TO 1,
THE SOFTWARE SWITCH REGISTER WILL BE USED, INSTEAD
OF THE HARDWARE CONSOLE SWITCH REGISTER. REGARDLESS
OF WHICH ONE IS USED, ALL BITS DEFINED IN SECTIONS
4.3 & 4.4 (SWITCH REGISTER OPTIONS) MAY BE USED
WHEN RUNNING IN STANDALONE MODE. IN APT SCRIPT MODE,
HOWEVER, BIT 14 (LOOP ON TEST) MUST ALWAYS BE SET

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
259
260
261
262
263
264
265

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

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 INCCRPORATED 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.

266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321

3.7 BAD SECTOR

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

3.8 EXECUTION TIME

TOTAL TIME: APPROX 5 MINUTES TO DO ALL THE TESTS
(BASED ON THE PDP 11/50)

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 BUS ADDRESS	1264	177440
CONTROLLER INTERRUPT VECTOR	1334	210
CONTROLLER PRIORITY	1336	240
P-CLOCK STATUS REG	1340	172540
P-CLOCK SET BUFFER	1342	172542
P-CLOCK READ BUFFER	1344	172544
L-CLOCK STATUS REG	1346	177546
L-CLOCK INTERRUPT VECTOR	1350	100
P-CLOCK INTERRUPT VECTOR	1352	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.

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
371
372
373
374
375
376
377

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

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT

CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

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

4.3.1 SW<15>

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

4.3.2 SW<14>

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

4.3.3 SW<13>

THIS SWITCH INHIBITS ALL ERROR MESSAGES. NORMALLY USED WHEN LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9).

4.3.4 SW<12>

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

4.3.5 SW<11>

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

4.3.6 SW<10>

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

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
427
428
429
430
431
432

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

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.

4.4 'SOFTWARE' SWITCH REGISTER

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

SWR = NNNNNN NEW =

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

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

4.5 INPUT DIALOGUE

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

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

4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

THE DEFAULT RESPONSE IS CARRIAGE RETURN TO TEST ALL DRIVES

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
539
540
541
542
543
544
545

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. ALL OPERATOR RESPONSES ARE UNDERLINED.

UNIBUS RK06 DRIVE DIAGNOSTIC
PART 3
MAINDEC-11-DZR6J-D-PB

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)

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
595
596
597
598
599
600
601

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 DRIVE DIAGNOSTIC
PART 3
MAINDEC-11-DZR6J-D-PB

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

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:

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
651
652
653
654
655
656
657

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 CYLINDERS
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.

5.0 DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION

5.1 GENERAL

OPERATOR INTERVENTION TESTS

THESE TESTS CHECK OUT ALL THE DRIVE INTERLOCKS, FRONT PANEL
SWITCHES AND LIGHTS.

THE OPERATOR IS INSTRUCTED TO PERFORM A TEST
AND TYPE A SPACE WHEN FINISHED.

OPERATOR INTERVENTION TESTS CAN BE INDIVIDUALLY
BYPASSED BY TYPING A CONTROL-E (↑E). ONLY AT THE
BEGINNING OF EACH TEST, AS INSTRUCTED BY THE TYPEOUT.

IF THE PROGRAM DETERMINES IT WAS LOADED UNDER
ACT APT, OPERATOR INTERVENTION TESTS WILL BE
BYPASSED UNLESS THE 'LOAD & DUMP MODE' IS BEING
USED.

THEY WILL BE BYPASSED IN 'MONITOR MODE' AS OPERATOR
INTERVENTION MAY NOT BE FEASIBLE IN OVER-NITE
TESTING.

5.2 TEST DESCRIPTIONS

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

BASIC CONTROLLER TESTS, SIZING & SETUP

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

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

TEST 2 SIZE THE BUSS

THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE MANUAL MODE.

EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED. CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE DRIVE WILL BE TESTED. IF SET, THE PROGRAM WILL BYPASS TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF

MDS, LIFE OR NED BEING SET; OR BOTH NED & DRA RESET INDICATING THE OTHER PORT IS ACCESSED.

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. IT WILL THEN CHECK TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING. IF NOT, IT WILL BE AN ERROR. IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED ONLY IF THE ERROR WAS A RESULT OF MDS OR LIFE SET OR BOTH NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFOR TO VERIFY IT WAS NOT SPECIFIED.

TEST 4 FIND NEXT DRIVE TO BE TESTED

THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT ADDRESS IN 'DRVAD'. THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS THE DRIVE WHOSE ADDRESS IS IN 'DRVAD'.

TEST 5 PRINT DRIVE SERIAL NUMBER

THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11

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
763
764
765
766
767
768
769

IN DECIMAL & IS PERFORMED ON THE 1ST PASS ONLY

TEST 6 SET VV WITH PACK COMMAND

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

TEST 7 UNLOAD DRIVE TO BE TESTED

THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE.

OPERATOR INTERVENTION TESTS

TEST 10 INTERLOCKS TESTS

THIS TEST VERIFIES THAT THE DOOR & CARTRIDGE STATUS BITS
ARE OPERATING PROPERLY IN MESSAGE A0 & THAT THE REMOVAL
OF THE CARTRIDGE CLEARS VOLUME VALID.
IT FURTHER VERIFIES ALL REMAINING STATUS BITS OF
MESSAGE A & B, WORDS 0 & 1.
THIS TEST ALSO CHECKS THAT THE SPINDLE CANNOT BE STARTED
WHEN THE DOOR IS OPEN OR THE CARTRIDGE IS REMOVED.
IT ALSO VERIFIES THAT LOSS OF VOLUME VALID RESETS SACK WHICH
ASSERTS NON EXISTENT DRIVE IN RKCS2

TEST 11 UNIT SELECT PLUG TEST

THIS TEST VERIFIES THAT WHEN THE UNIT SELECT PLUG IS PULLED
OUT, THE QUAL LOGIC RESETS ATTN & VOLUME VALID, THAT
THE DRIVE DE-SELECTS & NON EXISTENT DRIVE ASSERTS.
FURTHER, THE OPERATOR IS ASKED TO INSERT ANY NUMBER OF
UNIT SELECT PLUGS. THE PROGRAM WILL RESPOND BY TYPING
THE PLUG CODE NUMBER AS SOON AS IT IS INSERTED.
THIS PORTION OF THE TEST IS TERMINATED AT ANY TIME BY A CONTROL-C

TEST 12 PORT SELECTION TESTS

THE OPERATOR IS ASKED TO SWITCH TO THE WRONG PORT
& THEN DESELECT BOTH PORTS.
IN BOTH CASES, NON EXISTENT DRIVE SHOULD ASSERT IN RKCS2

TEST 13 AC LOW DETECTION PART 1

A PRELIMINARY AC LOW TEST IS PERFORMED HERE WHILE HEADS ARE UNLOADED.
BATTERY RETRACT WILL BE TESTED LATER.
ACLO CANNOT BE TESTED DIRECTLY SINCE IT REQUIRES

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
819
820
821
822
823
824
825

SECTOR PULSES, I.E. HEADS LOADED.
THEREFORE THE INDICATOR CHECKED IS NON-EXISTENT
DRIVE (NED) ASSERTING IN RKCS2 AS A RESULT OF DCLO.
AFTER POWER UP, VOLUME VALID IS CHECKED TO BE CLEARED.

TEST 14 CHECK NXF LOGIC

THIS TEST VERIFIES NON-EXECUTABLE FUNCTION (NXF) IS DETECTED
AS A RESULT OF DOING A SEEK WITH VOLUME VALID RESET.

TEST 15 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL

THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
THIS AREA CONTAINS BAD SECTOR INFOR WHICH IS WRITTEN BY THE
FACTORY DURING MANF. ALL BAD SECTOR INFOR (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 MESSAGE WILL BE TYPED INDICATING THAT ALL
FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
THIS IS DONE SO AS NOT TO DESTROY BSE INFOR OR AN ALIGNMENT PACK BY WRIT
THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.

TEST 16 WRITE LOCK TEST

THIS TEST VERIFIES THAT DATA WRITTEN ON A SECTOR CANNOT BE
ALTERED ONCE THE WRITE LOCK SWITCH HAS BEEN ENABLED.
IT ALSO CHECKS THAT WRITE PROTECT TAKES EFFECT ONLY AT
SECTOR BOUNDARIES WHEN DOING CONTINUOUS WRITING ON CYL 0, HEAD 0

TEST 17 AC LOW DETECTION PART 2

THIS TEST VERIFIES THAT WHEN AC POWER IS LOST, THAT WRITING CEASES
AT SECTOR BOUNDARIES & THAT THE BATTERY RETRACT IS FUNCTIONAL.
THERE IS APPROX 4 MS BETWEEN AC LOW ASSERTING AND NED ASSERTING
WHEN THE INTERFACE SHUTS DOWN.

TEST 20 END OF PROGRAM

THIS IS NOT A TEST BUT A LINKAGE TO PERFORM ALL THE
ABOVE TESTS FOR THE NEXT DRIVE PRESENT.
THE NEXT TEST IS ENTERED ONLY AFTER ALL DRIVES PRESENT
HAVE BEEN TESTED.
DO NOT LOOP ON THIS 'TEST'.

TEST 21 MULTIPLE DRIVE DETECTION TEST

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
875
876
877
878
879
880
881

THIS TEST IS PERFORMED ONLY ONCE AT THE END OF PASS 1 AND IS BYPASSED IF ONLY ONE DRIVE IS PRESENT.

THE MULTIPLE DRIVE DETECTION LOGIC IS TESTED BY THE FOLLOWING METHOD:

- A. HEADS MUST BE LOADED (SECTOR PULSES REQ'D)
- B. THE OPERATOR IS INSTRUCTED TO INSERT THE SAME UNIT SELECT PLUG NUMBER (1 PAIR AT A TIME) ON ANY 2 DRIVES TO BE TESTED
- C. THE OPERATOR THEN DEPRESSES THE SPACE BAR TO CONTINUE THE TEST OR A CONTROL-C TO EXIT THE TEST

THE PROGRAM VERIFIES THAT MULTIPLE DRIVE SELECT & DRIVE UNSAFE BOTH SET & THAT THE DRIVE UNLOADS

THE OPERATOR IS ASKED TO VERIFY THAT HEADS UNLOAD FROM BOTH DRIVES

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

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

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION

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

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS ADVISED TO LOOK UP THE PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION OF WHAT IS BEING CHECKED IN THE TEST IS GIVEN AT THE BEGINNING OF EVERY TEST. ALL THE NUMBERS GIVEN WITH ERROR MESSAGES ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

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

6.2 ERROR PRINTOUT EXAMPLE:

DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE DOOR OPEN
VERIFY SPINDLE DOES NOT START & HEADS DO NOT LOAD

DEPRESS SPACE BAR WHEN FINISHED

SPINDLE ON SET IN RKMR2
AFTER MANUALLY LOADING HEADS WITH DOOR OPEN

TEST NO.	PC					
000010	015700					
RKMR2	RKMR3	RKER	RKDS	RKCS1	RKCS2	RKASOF
050343	100000	000000	140301	040200	000103	004000

MESSAGE A0 ERROR
AFTER MANUALLY LOADING HEADS WITH DOOR OPEN

TEST NO.	PC					
000010	015736					
	EXPECT					
A0	B0	A1	B1	A2	B2	B3
100143	100000	000543	000001			
	ACTUAL					
050343	100000	001723	000001			
RKCS1	RKCS2	RKASOF	RKER	RKDS	RKDC	
040200	000103	004000	000000	140301	000000	

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

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

[END OF DOCUMENT]

%

923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967

167400
000001

```
*** PGM REV 033 ***
.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 UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
*COPYRIGHT (C) 1976
*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	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
12	ABORT DRIVE AFTER 20 ERRORS
11	INHIBIT ITERATIONS
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SWR<7:0>

```
.SBTTL SUMMARY OF STARTING LOCATIONS
```

200	DEFAULT PARAMETERS
220	INPUT PARAMETERS
240	ODT11

```

968 .SBTTL BASIC DEFINITIONS
969
970
971 001100 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
972 STACK= 1100
973 .EQUIV EMT,ERROR ;:BASIC DEFINITION OF ERROR CALL
974 .EQUIV IOT,SCOPE ;:BASIC DEFINITION OF SCOPE CALL
975
976 ;*MISCELLANEOUS DEFINITIONS
977 AT= 11 ;:CODE FOR HORIZONTAL TAB
978 LF= 12 ;:CODE FOR LINE FEED
979 CR= 15 ;:CODE FOR CARRIAGE RETURN
980 CRLF= 200 ;:CODE FOR CARRIAGE RETURN-LINE FEED
981 PS= 177776 ;:PROCESSOR STATUS WORD
982 .EQUIV PS,PSW
983 STKLM= 177774 ;:STACK LIMIT REGISTER
984 PIRQ= 177772 ;:PROGRAM INTERRUPT REQUEST REGISTER
985 DSWR= 177570 ;:HARDWARE SWITCH REGISTER
986 DDISP= 177570 ;:HARDWARE DISPLAY REGISTER
987
988 ;*GENERAL PURPOSE REGISTER DEFINITIONS
989 R0= %0 ;:GENERAL REGISTER
990 R1= %1 ;:GENERAL REGISTER
991 R2= %2 ;:GENERAL REGISTER
992 R3= %3 ;:GENERAL REGISTER
993 R4= %4 ;:GENERAL REGISTER
994 R5= %5 ;:GENERAL REGISTER
995 R6= %6 ;:GENERAL REGISTER
996 R7= %7 ;:GENERAL REGISTER
997 SP= %6 ;:STACK POINTER
998 PC= %7 ;:PROGRAM COUNTER
999
1000 ;*PRIORITY LEVEL DEFINITIONS
1001 PRO= 0 ;:PRIORITY LEVEL 0
1002 PR1= 40 ;:PRIORITY LEVEL 1
1003 PR2= 100 ;:PRIORITY LEVEL 2
1004 PR3= 140 ;:PRIORITY LEVEL 3
1005 PR4= 200 ;:PRIORITY LEVEL 4
1006 PR5= 240 ;:PRIORITY LEVEL 5
1007 PR6= 300 ;:PRIORITY LEVEL 6
1008 PR7= 340 ;:PRIORITY LEVEL 7
1009
1010 ;*"SWITCH REGISTER" SWITCH DEFINITIONS
1011 SW15= 100000
1012 SW14= 40000
1013 SW13= 20000
1014 SW12= 10000
1015 SW11= 4000
1016 SW10= 2000
1017 SW09= 1000
1018 SW08= 400
1019 SW07= 200
1020 SW06= 100
1021 SW05= 40
1022 SW04= 20
1023 SW03= 10
SW02= 4

```

1024 000002
 1025 000001
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038 100000
 1039 040000
 1040 020000
 1041 010000
 1042 004000
 1043 002000
 1044 001000
 1045 000400
 1046 000200
 1047 000100
 1048 000040
 1049 000020
 1050 000010
 1051 000004
 1052 000002
 1053 000001
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066 000004
 1067 000010
 1068 000014
 1069 000014
 1070 000014
 1071 000020
 1072 000024
 1073 000030
 1074 000034
 1075 000060
 1076 000064
 1077 000240
 1078
 1079

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
 TRITVEC= 14 ;: "I" 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

```

1080
1081
1082
1083      000000      RKCS1= 0      ;CONTROL AND STATUS REGISTER 1
1084      000002      RKWC= 2      ;WORD COUNT REGISTER
1085      000004      RKBA= 4      ;BUS ADDRESS REGISTER
1086      000006      RKDA= 6      ;DESIRED TRACK SECTOR REGISTER
1087      000010      RKCS2= 10     ;CONTROL AND STATUS REGISTER 2
1088      000012      RKDS= 12     ;DRIVE STATUS REGISTER
1089      000014      RKER= 14     ;ERROR REGISTER
1090      000016      RKASOF= 16    ;ATTENTION SUMMARY AND OFFSET REGISTER
1091      000020      RKDC= 20     ;DESIRED CYLINDER REGISTER
1092      000024      RKDB= 24     ;DATA BUFFER
1093      000026      RKMR1= 26    ;MAINTENANCE REGISTER 1
1094      000034      RKMR2= 34    ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1095      000036      RKMR3= 36    ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1096      000030      RKECPS= 30   ;ECC POSITION INFORMATION
1097      000032      RKECPT= 32   ;ECC PATTERN INFORMATION
1098
1099      .SBTTL CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1100
1101      ; DRIVE COMMANDS
1102
1103      000001      SELDRV= 1     ;SELECT DRIVE (GET STATUS)
1104      000003      PACK= 3      ;PACK ACKNOWLEDGE
1105      000005      CLEAR= 5     ;DRIVE CLEAR
1106      000007      UNLOAD= 7    ;UNLOAD
1107      000011      SRTSPL= 11   ;START SPINDLE
1108      000013      RECAL= 13    ;RECALIBRATE
1109      000015      OFFSET= 15   ;OFFSET
1110      000017      SEEK= 17     ;SEEK
1111      000021      RCDATA= 21   ;READ DATA
1112      000023      WRDATA= 23   ;WRITE DATA
1113      000025      ROHEAD= 25   ;READ HEADER
1114      000027      WRHEAD= 27   ;WRITE HEADER AND DATA
1115      000031      WRTCHK= 31   ;WRITE CHECK
1116
1117      000001      GO= BIT0      ;GO BIT
1118      000100      IE= BIT6      ;INTERRUPT ENABLE
1119      000200      RDY= BIT7     ;CONTROLLER READY
1120      000400      BA16= BIT8    ;BUS ADDRESS BIT 16
1121      001000      BA17= BIT9    ;BUS ADDRESS BIT 17
1122      002000      CDT= BIT10    ;CONTROLLER DRIVE TYPE (0=RK06)
1123      004000      CTO= BIT11    ;CONTROLLER TIMEOUT
1124      010000      CFMT= BIT12   ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1125      020000      DCPAR= BIT13  ;SERCON PARITY ERROR DETECTED BY CONTROLLER
1126      040000      DI= BIT14    ;DRIVE INTERRUPT
1127      100000      CERR= BIT15  ;CONTROLLER ERROR
1128      100000      CCLR= BIT15  ;CONTROLLER CLEAR
1129
1130      .SBTTL CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1131
1132      000007      DRVMSK= 7     ;MASK FOR DRIVE SELECTION CODE
1133      000010      RLS= BIT3     ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1134      000020      BAI= BIT4     ;BUS ADDRESS INCREMENT IN BIT
1135      000040      SCLR= BITS    ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES

```

K02

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 22
CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)

SEQ 0022

1136	000100	IR=	BIT6	; INPUT READY
1137	000200	OR=	BIT7	; OUTPUT READY
1138	000400	UFE=	BIT8	; UNIT FIELD ERROR
1139	001000	MDS=	BIT9	; MULTIPLE DRIVE SELECT
1140	002000	PGE=	BIT10	; PROGRAMMING ERROR
1141	004000	NEM=	BIT11	; NON-EXISTENT MEMORY
1142	010000	NED=	BIT12	; NON-EXISTENT DRIVE
1143	020000	UPE=	BIT13	; UNIBUS PARITY ERROR
1144	040000	WCE=	BIT14	; WRITE CHECK ERROR
1145	100000	DLT=	BIT15	; DATA LATE ERROR
1146				
1147		.SBTTL	ERROR REGISTER BIT DEFINITION (RKER:14)	
1148				
1149	000001	ILF=	BIT0	; ILLEGAL FUNCTION CODE
1150	000002	SKI=	BIT1	; SEEK INCOMPLETE
1151	000004	NXF=	BIT2	; NON-EXECUTABLE FUNCTION
1152	000010	DRPAR=	BIT3	; DRIVE DETECTED SERCON PARITY ERROR
1153	000020	FMTE=	BIT4	; FORMAT ERROR
1154	000040	DTYPE=	BIT5	; DRIVE TYPE ERROR
1155	000100	ECH=	BIT6	; ECC HARD
1156	000200	BSE=	BIT7	; BAD SECTOR ERROR
1157	000400	HVRC=	BIT8	; HEADER VRC ERROR
1158	001000	COE=	BIT9	; CYLINDER ADDRESS OVERFLOW ERROR
1159	002000	IDAE=	BIT10	; INVALID DISK ADDRESS ERROR: HEAD/CYL
1160	004000	WLE=	BIT11	; WRITE LOCK ERROR
1161	010000	DTE=	BIT12	; DRIVE TIMING ERROR
1162	020000	OPI=	BIT13	; OPERATION (SEARCH) INCOMPLETE
1163	040000	UNS=	BIT14	; DRIVE UNSAFE
1164	100000	DCK=	BIT15	; DATA CHECK
1165				
1166		.SBTTL	STATUS REGISTER BIT DEFINITION (RKDS:12)	
1167				
1168	000001	DRA=	BIT0	; DRIVE AVAILABLE (CONTROLLER IS SET IF THIS BIT IS RESET)
1169				
1170	000004	OFST=	BIT2	; DRIVE OFFSET
1171	000010	ACLO=	BIT3	; AC LOW
1172	000020	DCLO=	BIT4	; DC LOW
1173	000040	DROT=	BIT5	; DRIVE OFF TRACK
1174	000100	VV=	BIT6	; VOLUME VALID
1175	000200	DRDY=	BIT7	; DRIVE READY
1176	000400	DDT=	BIT8	; DRIVE TYPE (0=RK06)
1177	004000	WRL=	BIT11	; WRITE LOCK
1178	020000	PIP=	BIT13	; POSITIONING IN PROGRESS
1179	040000	DSC=	BIT14	; DRIVE STATUS CHANGE
1180	100000	SVAL=	BIT15	; STATUS VALID
1181				
1182		.SBTTL	MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)	
1183				
1184	000017	MESMSK=	17	; MESSAGE MASK
1185	000020	PAT=	BIT4	; FORCE EVEN PARITY ON SERCON MESSAGE LINES
1186	000040	DMD=	BIT5	; DIAGNOSTIC MODE
1187	000100	MSP=	BIT6	; MAINTENANCE SECTOR PULSE
1188	000200	MIND=	BIT7	; MAINTENANCE INDEX
1189	000400	MCLK=	BIT8	; MAINTENANCE CLOCK
1190	001000	MERD=	BIT9	; MAINTENANCE ENCODED READ DATA
1191	002000	MEWD=	BIT10	; MAINTENANCE ENCODED WRITE DATA

L02

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 23
MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)

SEQ 0023

1192	004000	PCA= BIT11	;PRECOMPENSATION ADVANCE
1193	010000	PCD= BIT12	;PRECOMPENSATION DELAY
1194	020000	ECCW= BIT13	;ECC WORD IS BEING READ OR WRITTEN
1195	040000	WRTGAT= BIT14	;WRITE GATE
1196	100000	RDGATE= BIT15	;READ GATE
1197			
1198		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1199			
1200	000040	D.DRA= BITS	;DRIVE AVAILABLE
1201	000100	D.VV= BIT6	;VOLUME VALID
1202	000200	D.DRDY= BIT7	;DRIVE READY
1203	000400	D.DDT= BIT8	;DRIVE TYPE (0=RK06)
1204	001000	D.FORM= BIT9	;DRIVE FORMAT
1205	002000	D.OFF= BIT10	;OFFSET ON
1206	004000	D.WRL= BIT11	;WRITE LOCK
1207	010000	D.SPIN= BIT12	;SPINDLE ON
1208	020000	D.PIP= BIT13	;POSITIONING IN PROGRESS
1209	040000	D.DSC= BIT14	;DRIVE STATUS CHANGE
1210			
1211		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1212			
1213	000020	D.SSP= BIT4	;SERVO SIG PRESENT
1214	000040	D.HOHM= BITS	;HEADS HOME
1215	000100	D.BRHM= BIT6	;BRUSHES HOME
1216	000200	D.DOOR= BIT7	;DOOR INTERLOCKED
1217	000400	D.CART= BIT8	;CARTRIDGE INTERLOCK
1218	001000	D.SPOK= BIT9	;SPEED OK
1219	002000	D.FWD= BIT10	;FORWARD
1220	004000	D.REV= BIT11	;REVERSE
1221	010000	D.LOAD= BIT12	;HEADS LOADING
1222	020000	D.RTZ= BIT13	;RETURN TO ZERO
1223	040000	D.UNLD= BIT14	;HEADS UNLOADING
1224			
1225		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1226			
1227	000040	D.IDAE= BITS	;INVALID DISK ADDRESS ERROR:HEAD/CYL
1228	000100	D.ACLO= BIT6	;AC LOW
1229	000200	D.FLT= BIT7	;DRIVE FAULT
1230	000400	D.NXF= BIT8	;NON-EXECUTABLE FUNCTION CODE
1231	001000	D.PAR= BIT9	;DRIVE DETECTED SERCON PARITY ERROR
1232	002000	D.SKI= BIT10	;SEEK INCOMPLETE
1233	004000	D.WLE= BIT11	;WRITE LOCK ERROR
1234	010000	D.SPLS= BIT12	;SPEED LOSS
1235	020000	D.DROT= BIT13	;DRIVE OFF TRACK
1236	040000	D.UNS= BIT14	;R/W UNSAFE
1237			
1238		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1239			
1240	000020	D.SECT= BIT4	;SECTOR ERROR
1241	000040	D.WCUR= BITS	;WRITE CURRENT AND NO WRITE GATE
1242	000100	D.WGAT= BIT6	;WRITE GATE AND NO TRANSISTIONS
1243	000200	D.HDFL= BIT7	;HEAD FAULT
1244	000400	D.MHD= BIT8	;MULTIPLE HEAD SELECT
1245	001000	D.XERR= BIT9	;INDEX ERROR
1246	002000	D.TIB= BIT10	;TRIBIT ERROR
1247	004000	D.PLO= BIT11	;PLO ERROR

M02

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 24
DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)

SEQ 0024

1248 C10000
1249 020000
1250 C40000
1251
1252
1253
1254 000007
1255 017760
1256 017760
1257 077770
1258
1259
1260
1261 000003
1262 017760
1263 040000
1264 000760
1265 007000
1266 100000

D.NMOV= BIT12 ;SEEK AND NO MOTION
D.LIMD= BIT13 ;LIMIT DETECT ON SEEK
D.SUNS= BIT14 ;SERVO UNSAFE

.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)

M.DPV= 7 ;DRIVE CODE, ALL BYTES
M.CDIF= 17760 ;CYLINDER DIFF, BYTE 10
M.OFST= 17760 ;OFFSET VALUE, BYTE 10
M.SER= 77770 ;DRIVE SERIAL #, BYTE 11

.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)

M.ID= 3 ;BYTE 10, ALL BYTES
M.CADD= 17760 ;CYLINDER ADDRESS, BYTE 10
M.ALGN= BIT14 ;ALIGN SIGN, BYTE 10
M.SECT= 760 ;SECTOR COUNT, BYTE 11
M.HEAD= 7000 ;HEAD DECODE, BYTE 11
M.PAR= BIT15 ;PARITY, MESS A/B. ALL BYTES

```

1267
1268
1269
1270          000000
1271
1272
1273
1274          000174
1275 000174 000000
1276 000176 000000
1277
1278 000200 000137 010060
1279          000220
1280 000220 000137 010050
1281
1282          000240
1283 000240 000137 055342
1284
1285
1286
1287
1288
1289          000244
1290          000046
1291 000046 024030
1292          000052
1293 000052 120000
1294          000244
1295          001000
1296
1297
1298
1299
1300
1301          001000
1302          000024
1303 000024 000200
1304          000044
1305 000044 001000
1306          001000
1307
1308
1309
1310
1311 001000
1312 001000 000000
1313 001002 001210
1314 001004 000454
1315 001006 001130
1316 001010 001130
1317 001012 000052
1318
1319
1320
1321
1322

```

```

.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 2#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
      .=220
      JMP PARSRT          ;INPUT ALL PARAMETERS & START TESTING
      .=240
      JMP 0.0DT          ;ENTER 00T11

.SBTTL ACT11 HOOKS
; *****
; HOOKS REQUIRED BY ACT11
      $SVPC=.          ;SAVE PC
      .=46
      $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
      .=52
      .WORD 120000      ;;2)SET LOC.52 TO 120000
      .=$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.
$MBAOR: .WORD $MAIL      ;; ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT:  .WORD 300.       ;; RUN TIM OF LONGEST TEST
$PASTM: .WORD 600.       ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 600.       ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
      .WORD SETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

      .LIST MD
;

```

1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378

```

;USE LOOP X TO OMIT SUBCLR
;MACRO LOOP A
;   SCOP1
;   MOV #STACK,SP ;RESTORE STK PTR
;IF B A
;   JSR PC,SUBCLR
;   ERROR 24 ;CERR AFTER SCLR
;ENDC
;ENDM LOOP

;THIS MACRO FILLS EXPECTED MSG A0, B0, A1, B1, A2, B2 & B3 WITH STANDARD
;BITS SEE A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
;NOTE: A CAN BE ANY BIT COMBINATION DESIRED.
;MACRO F.EAB A
;   MOV #<A!D.SPIN!D.DROY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
;   CLR E.B0 ;EXPECTED MSG B0
;   MOV #<D.SPOK!D.CART!D.DOOR!D.BRM!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
;ENDM F.EAB

;THIS MACRO ASSUMES DRIVE MSG A0, B0, A1, B1 WILL ALWAYS BE TESTED
;USE A,C,D,E FOR MSG A0, B0, A1, B1 ERROR NUMBERS RESP.
;USE G=T.A2 TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
;   H=T.B2 TO READ MSG B2 & PUT INFO INTO 'CYLADD'
;   I=T.B3 TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEAD'
;USE F=<ERROR DESCRIPTION>
;MACRO CHECK A,C,D,E,F,G,H,I
;   JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
;   .WORD G!H!I ;& MSGS SPECIFIED HERE
;   ERROR A ;MSG A0 ERROR F
;   ERROR C ;MSG B0 ERROR
;   ERROR D ;MSG A1 ERROR
;   ERROR E ;MSG B1 ERROR
;ENDM CHECK

;A=CYL DIFF/OFFSET ERROR #
;B=CYL ADDR ERROR #
;C=<ERROR DESCRIPTION>
;MACRO CWD2 A,B,C,'D','E'
;   TST CYLDIF ;SEE IF MSG A2=0

```

```

1379          BEQ      D           ;BR IF YES
1380          ERROR   A           ;MSG A2 NOT CLEARED C
1381 D:        TST     CYLADD      ;SEE IF MSG B2=0
1382          BEQ      E           ;BR IF YES
1383          ERROR   B           ;MSG B2 NOT CLEARED C
1384 E:
1385 .ENDM      CMD2
1386
1387 .MACRO     DRCLR  ?A
1388
1389          MOV      #CCLR,RKCS1(R5)
1390          MOV      $UNIT,RKCS2(R5) ;DRIVE#
1391          MOV      #CLEAR,RKCS1(R5) ;DRIVE CLEAR CMD
1392          MOV      T10,TEMP1      ;SETUP TIMEOUT
1393          JSR      PC,FRDY        ;FIND RDY
1394          ERROR   151            ;NO RDY AFTER DRIVE CLEAR CMD
1395          JSR      PC,TSTATN      ;TEST FOR ATTN
1396          BR      A
1397          ERROR   154            ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1398 A:
1399 .ENDM      DRCLR
1400
1401 .MACRO     CALIB  ?A
1402
1403          MOV      #CCLR,RKCS1(R5)
1404          MOV      $UNIT,RKCS2(R5)
1405          MOV      #RECAL,RKCS1(R5) ;RECAL CMD
1406          ;RECAL CMD
1407          ;RESET CYL DIFF/OFFSET & CYL ADDR REG
1408          ;IN RKMR2 & RKMR3 RESP.
1409          MOV      T10,TEMP1      ;SETUP TIMEOUT
1410          JSR      PC,FRDY        ;FIND RDY
1411          ERROR   124            ;RDY NOT SET AFTER RECAL CMD
1412
1413          MOV      #1,RKMR1(R5)   ;SELECT WORD 1
1414          JSR      PC,GSTAT
1415          BIT      #D.RTZ,HMR2
1416          BNE     A
1417          ERROR   214            ;RTZ NOT SET DURING RECAL CMD
1418 A:        MOV      T10,TEMP2      ;SETUP TIMEOUT
1419          JSR      PC,FATT1       ;FIND ATTN
1420          ERROR   55            ;NO ATTN AFTER RECAL CMD
1421          DRCLR
1422 .ENDM      CALIB
1423
1424 ;
1425 ; A=WRHEAD/<CFMT!WRHEAD>
1426 ; USE WRHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1
1427 ;
1428 .MACRO     WRHDR  A,C,?D
1429
1430          MOV      #<A>,RKCS1(R5) ;WRITE HEADER CMD
1431          MOV      T5000,TEMP1    ;SETUP TIMEOUT
1432          JSR      PC,FRDY        ;FIND RDY
1433
1434

```

```

1435          ERROR 200          ;NO RDY AFTER WRITE HEADER CMD
1436          JSR   PC_GSTAT      ;GET FRESH STATUS
1437          BIT   #CERR,HCS1
1438          BEQ   0
1439          ERROR 201          ;CERR AFTER WRITE HEADER CMD
1440
D:
1441          .IF B C
1442          F.EAB 0
1443          CHECK 215,216,217,220,<AFTER WRITE HEADER CMD>,0,0,0
1444          .ENDC
1445
1446          .ENDM WRHDR
1447
1448          ;
1449          ; A=RDHEAD/<CFMT!RDHEAD>
1450          ; USE RDHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1
1451          ;
1452          .MACRO RDHDR A,C,?D,?E
1453
1454          MOV   #RHTAB RD
1455          MOV   #<A>,RKCS1(R5) ;READ HEADER CMD
1456          MOV   T5000,TEMP1 ;SETUP TIMEOUT
1457          JSR   PC_FRDY      ;FIND RDY
1458          ERROR 171          ;NO RDY AFTER READ HEADER CMD
1459          BIT   #CERR,HCS1
1460          BEQ   0
1461          ERROR 174          ;CERR AFTER READ HEADER CMD
1462          TYPE  MSG26        ;ABORTING DATA TESTS TO DO TIMING TESTS
1463          JMP   ↑TIMING
1464
D:
1465          MOV   RKDB(R5),(R0)+ ;1'ST WORD FROM SILO TO RHTAB
1466          MOV   RKDB(R5),(R0)+ ;2'ND WORD
1467          MOV   RKDB(R5),(R0)+ ;3'RD WORD
1468
1469
1470          BIT   #DLT,RKCS2(R5)
1471          BEQ   E
1472          JSR   PC_GSTAT
1473          ERROR 173          ;DLT AFTER READ HEADER CMD
1474
E:
1475          .IF B C
1476          NOP
1477          .ENDC
1478
1479          .ENDM RDHDR
1480
1481          .MACRO HDCHK3 ?A
1482
1483          RDHDR RDHEAD,X
1484          CMP   RHTAB,↑CYL ;CHECK WORD 0 ONLY, CYL#
1485          BEQ   A ;BR IF SAME
1486          ERROR 51 ;WRONG CYL# ON HEADER
1487
A:
1488
1489          .ENDM HDCHK3
1490

```

E03

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 29
APT PARAMETER BLOCK

SEQ 0029

1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546

; A=Y0CYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR

```
.MACRO HDTBL A,B,C
      MOV A,CALADD ;SETUP
      MOV #B,HEAD ;TO FILL
      MOV #C,FORMAT ;HEADER
      JSR PC,FHDTAB ;TABLE
.ENDM HDTBL
```

; QUICK SEEK. ENTER WITH CYL# IN RKDC

```
.MACRO QKSEEK ?A
      MOV #SEEK,RKCS1(R5) ;SEEK CMD
      MOV T50,TEMP1 ;SETUP TIMEOUT
      JSR PC,FRDY ;FIND RDY
      ERROR 131 ;NO RDY AFTER SEEK CMD
      MOV T5000,TEMP1 ;SETUP TIMEOUT
      JSR PC,FATT2 ;FIND ATTN
      ERROR 132 ;NO ATTN AFTER SEEK CMD
      BIT #CERR,HCS1
      BEQ A
      ERROR 210 ;CERR AFTER SEEK CMD
```

A:

```
.ENDM QKSEEK
```

; A=WRDATA/<CFMT!WRDATA>
; C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
; D=ADDR TO JMP TO BYPASS TEST
; E: IF BLANK WILL CHECK A0, B0, A1 & B1 AT THE END OF WRITING
; E: IF NON BLANK WILL OMIT CHECKING A0 THRU B1

```
.MACRO WDATA A,C,D,E,?F,?G,?H,?I
      MOV #<A>,RKCS1(R5) ;WRITE DATA CMD
      MOV T5000,TEMP1 ;SETUP TIMEOUT
      JSR PC,FRDY ;FIND RDY
      ERROR 11 ;NO RDY AFTER WRITE DATA CMD
      JSR PC,GSTAT ;GET FRESH STATUS
      BIT #CERR,HCS1
      BEQ I ;BR IF NO ERRORS
      BIT #BSE,HER ;SEE IF BAD SECTOR FLAG
      BEQ G ;BR IF NO
```

```

1547          JSR    PC,TRUERR      ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
1548          BR     H               ;RETURN HERE IF NO
1549
1550          INC    SECTOR          ;RETURN HERE IF YES
1551          CMP    SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
1552          BNE   F               ;BR IF NO
1553          ERROR  40              ;ABORTING TEST DETECTED 10 BAD SECTORS
1554          JMP    D               ;BYPASS TEST
1555
1556 F:         MOV    #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
1557          JMP    C
1558
1559 G:         ERROR  12              ;CERR WITH WRITE DATA CMD
1560          F.EAB  0
1561          CHECK  52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1562          TYPE   MSG72           ;ABORTING BALANCE OF TESTS
1563          JMP    $EOP
1564 H:         ERROR  47              ;BAD SECTOR NOT LISTED IN TABLE
1565 I:
1566 .IF      B      E
1567          F.EAB  0
1568          CHECK  52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1569 .ENDC
1570 .ENDM    WDATA
1571
1572
1573 ;
1574 ;A=RDDATA/<CFMT!RDDATA>
1575 ;USE RDATA <A>,X TO OMIT CKWD12
1576 ;
1577 ;
1578 .MACRO   RDATA  A,C,'D','E','F','G','H
1579
1580          MOV    #<A>,RKCS1(R5)  ;READ DATA CMD
1581          MOV    T5000,TEMP1     ;SETUP TIMEOUT
1582          JSR    PC,FRDY         ;FIND RDY
1583          ERROR  13              ;NO RDY AFTER READ DATA CMD
1584          JSR    PC,GSTAT        ;GET FRESH STATUS
1585          BIT    #CERR,HCS1
1586          BEQ   G
1587          BIT    #BSE,HER        ;SEE IF BAD SECTOR
1588          BEQ   E
1589          ERROR  50              ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1590          BR     H
1591 D:        TYPE   MSG72           ;ABORTING BALANCE OF TESTS
1592          JMP    $EOP
1593
1594 E:        BIT    #DCK,HER        ;SEE IF DATA CHECK ERROR
1595          BEQ   F
1596          ERROR  21              ;DATA CHECK ERROR AFTER READ CMD (ECC)
1597          BR     H
1598
1599 F:        ERROR  14              ;CERR AFTER READ DATA CMD.
1600
1601 H:        F.EAB  0
1602          CHECK  54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0

```



```

1603          BR      D
1604
1605 G:
1606 .IF      B      C
1607         F.EAB  O
1608         CHECK  54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,O
1609 .ENDC
1610 .ENDM    RDATA
1611
1612
1613 ;
1614 ;A=WRCHK/<CFMT!WRCHK>
1615 ;C=16 FOR STD ERROR MSG
1616 ;C=134/135/136/137 FOR ERROR MSG USED IN 'SBOUND' ROUTINE
1617 ;USE WRCHK      <A>,C X TO OMIT CKWD12
1618 ;
1619
1620 .MACRO  WRCHK  A,C,D,?E,?F
1621
1622         MOV      #<A>,RKCS1(R5) ;WRITE CHECK CMD
1623         MOV      T5000,TEMP1 ;SETUP TIMEOUT
1624         JSR      PC,FRDY ;FIND RDY
1625         ERROR   15 ;NO RDY AFTER WRITE CHECK CMD
1626         JSR      PC,GSTAT ;GET FRESH STATUS
1627         BIT      #CERR,HCS1
1628         BEQ     F
1629         BIT      #WCE,HCS2 ;SEE IF WRITE CHECK ERROR
1630         BEQ     F
1631         MOV      RKDB(R5),WD1 ;ACTUAL WORD FOR PRINTOUT
1632         ERROR   C ;WCE AFTER WRITE CMD
1633         BR      F
1634
1635 E:      ERROR   22 ;CERR AFTER WRITE CHECK CMD
1636         F.EAB  O
1637         CHECK  57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,O
1638         TYPE   MSG72 ;ABORTING BALANCE OF TESTS
1639         JMP     ENDRV
1640
1641 F:
1642 .IF      B      D
1643         F.EAB  O
1644         CHECK  57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,O
1645 .ENDC
1646 .ENDM    WRCHK
1647
1648 ;
1649 ;MACRO TO TEST THAT WRITE CHECK OCCURRED AT SECTOR BOUNDRY
1650 ;A&B=134,135 FOR WRITE PROTECT SW TEST
1651 ;A&B=136,137 FOR AC LOW TEST PART 2
1652 ;C=JUMP ADDR TO REPEAT TEST
1653 ;
1654 .MACRO  SBOUND  A,B,C,?D,?E,?F,?G,?H,?I,?J,?K
1655
1656         JSR      PC,SUBCLR
1657         ERROR   24 ;CERR AFTER SCLR
1658

```

```

1659          TST      STMP2          ;SEE IF TRK/SECTOR 0
1660          BEQ      K              ;REPEAT,NO NEW DATA XFER TOOK PLACE
1661          CMP      STMP2,#1023    ;SEE IF TRK 2,SECTOR 19
1662          BEQ      K              ;REPEAT,NO OLD DATA TO CHECK AGAINST
1663          BIT      #BIT0,STMP1
1664          BNE      D              ;BR IF WRITING 1'S WHEN WLE OCCURRED
1665          MOV      #DATA1,RKBA(R5);WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
1666          BR      E
1667
1668          K:      JMP      C
1669
1670          D:      MOV      #DATA0,RKBA(R5);WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
1671          E:      BIS      #BA1,RKCS2(R5)
1672          MOV      #-256,RKWC(R5)
1673          MOV      STMP3,RKDA(R5) ;REFRESH RKDA
1674          MOV      @RKBA(R5),WD2 ;EXPECTED WORD FOR TRUERR TYPEOUT
1675          MOV      TOCYL,RKDC(R5)
1676          WRCHK   WRTCHK,A,X
1677          NOP
1678          NOP
1679
1680          CMP      STMP2,#400      ;SEE IF WRL AT TRK 1, SECTOR 0
1681          BNE      F              ;BR IF NO
1682          MOV      #21.,RKDA(R5) ;ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
1683          BR      H
1684          F:      CMP      STMP2,#1000 ;SEE IF WRL AT TRK 2,SECTOR 0
1685          BNE      G              ;BR IF NO
1686          MOV      #425,RKDA(R5) ;ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
1687          BR      H
1688
1689          G:      DEC      STMP2          ;GET SECTOR BEFORE WRL
1690          MOV      STMP2,RKDA(R5)
1691          H:      MOV      RKDA(R5),STMP3 ;FOR ERROR PRINTOUT
1692          BIT      #BIT0,STMP1
1693          BNE      I              ;BR IF WRITING 1'S WHEN WLE OCCURRED
1694          MOV      #DATA0,RKBA(R5) ;WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
1695          BR      J
1696
1697          I:      MOV      #DATA1,RKBA(R5);WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
1698          J:      BIS      #BA1,RKCS2(R5)
1699          MOV      #-256,RKWC(R5)
1700          MOV      @RKBA(R5),WD2 ;EXPECTED WORD FOR TRUERR TYPEOUT
1701          MOV      TOCYL,RKDC(R5)
1702          WRCHK   WRTCHK,B,X
1703
1704          .ENDM   SBOUND
1705
1706          ;
1707          ;QUICK START SPINDLE
1708          ;
1709          .MACRO  QKSRT   ?A
1710
1711          JSR      PC,SUBCLR
1712          ERROR   24          ;CERR AFTER SCLR
1713
1714          BIT      #D.SPIN,HMR2 ;SEE IF SPINDLE ALREADY ON

```

```

1715          BNE      A           ;BR IF YES
1716          TYPE    ,MSG29       ;PLEASE WAIT, HEADS BEING LOADED
1717
1718          MOV      #SRTSPL,RKCS1(R5) ;START SPINDLE CMD
1719          MOV      T10,TEMP1
1720          JSR      PC,FRDY       ;FIND CONTR RDY
1721          ERROR    143          ;CONTR RDY NOT SET AFTER CMD
1722
1723          MOV      T100,TEMP2
1724          JSR      PC,FATT1      ;FIND ATTN
1725          ERROR    144          ;NO ATTN AFTER CMD
1726
1727          CLR      UNLD
1728
1729          A:
1730          .ENDM    QKSRT
1731
1732          ;ROUTINE TO ISSUE PACK COMMAND
1733          ;
1734          .MACRO   QKPACK  ?A
1735          MOV      #CCLR,RKCS1(R5)
1736          MOV      $UNIT,RKCS2(R5) ;DRIVE #
1737          MOV      #PACK,RKCS1(R5) ;PACK CMD
1738          MOV      T10,TEMP1
1739          JSR      PC,FRDY       ;FIND CONTR RDY
1740          ERROR    116          ;CONTR NOT RDY
1741
1742          BIT      #D.VV,HMR2
1743          BNE      A
1744          ERROR    27           ;VOLUME VALID NOT SET AFTER PACK CMD
1745
1746          A:
1747          .ENDM    QKPACK
1748
1749          ;QUICK UNLOAD
1750          ;
1751          .MACRO   QKUNLD
1752
1753          JSR      PC,SUBCLR
1754          ERROR    24           ;CERR AFTER SCLR
1755
1756          MOV      #UNLOAD,RKCS1(R5) ;UNLOAD CMD
1757          MOV      T10,TEMP1
1758          JSR      PC,FRDY       ;FIND CONTR RDY
1759          ERROR    17           ;NO RDY AFTER UNLD CMD
1760          JSR      PC,TSTATN
1761          ERROR    20           ;NO ATTN AFTER UNLOAD CMD
1762          .ENDM    QKUNLD
1763
1764
1765          .NLIST  MD
1766
1767
1768

```

1769
1770
1771
1772
1773
1774
1775 001100
.1776 001100
1777 001100 000000
1778 001102 000
1779 001103 000
1780 001104 000000
1781 001106 000000
1782 001110 000000
1783 001112 000000
1784 001114 000
1785 001115 001
1786 001116 000000
1787 001120 000000
1788 001122 000000
1789 001124 000000
1790 001126 000000
1791 001130 000000
1792 001132 000000
1793 001134 000
1794 001135 000
1795 001136 000000
1796 001140 177570
1797 001142 177570
1798 001144 177560
1799 001146 177562
1800 001150 177564
1801 001152 177566
1802 001154 000
1803 001155 002
1804 001156 012
1805 001157 000
1806 001160 000000
1807 001162 000000
1808 001164 000000
1809 001166 000000
1810 001170 000000
1811 001172 000000
1812 001174 000000
1813 001176 000000
1814 001200 177607 000377
1815 001204 077
1816 001205 015
1817 001206 000012
1818
1819
1820
1821
1822
1823 001210
1824 001210 000000

.SBTTL COMMON TAGS

```

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

```

```

.SBTTL COMMON TAGS
      .=1100
SCMTAG: .WORD 0 ;; START OF COMMON TAGS
$STNM:  .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
$GDADR: .WORD 0 ;; CONTAINS 'GOOD' DATA
$BDADR: .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 0 ;; ADDRESS OF SWITCH REGISTER
$DISP: .WORD 0 ;; 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"
$STPFLG: .BYTE 0 ;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
$TMPO:  .WORD 0 ;; USER DEFINED
$TMP1:  .WORD 0 ;; USER DEFINED
$TMP2:  .WORD 0 ;; USER DEFINED
$TMP3:  .WORD 0 ;; USER DEFINED
$TMP4:  .WORD 0 ;; USER DEFINED
$TMP5:  .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: .WORD ;; APT MAILBOX
$MSGTY: .WORD AMSGTY ;; MESSAGE TYPE CODE

```

K03

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
D2RK6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 35
APT MAILBOX-ETABLE

SEQ 0035

1825	001212	000000	\$FATAL: .WORD	AFATAL	:: FATAL ERROR NUMBER
1826	001214	000000	\$TESTN: .WORD	ATESTN	:: TEST NUMBER
1827	001216	000000	\$PASS: .WORD	APASS	:: PASS COUNT
1828	001220	000000	\$DEVCT: .WORD	ADEVCT	:: DEVICE COUNT
1829	001222	000000	\$UNIT: .WORD	AUNIT	:: I/O UNIT NUMBER
1830	001224	000000	\$MSGAD: .WORD	AMSGAD	:: MESSAGE ADDRESS
1831	001226	000000	\$MSGLG: .WORD	AMSGLG	:: MESSAGE LENGTH
1832	001230		\$ETABLE:		:: APT ENVIRONMENT TABLE
1833	001230	000	\$ENV: .BYTE	AENV	:: ENVIRONMENT BYTE
1834	001231	000	\$ENVM: .BYTE	AENVM	:: ENVIRONMENT MODE BITS
1835	001232	000000	\$SWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
1836	001234	000000	\$USWR: .WORD	AUSWR	:: USER SWITCHES
1837	001236	000000	\$CPUOP: .WORD	ACPUOP	:: CPU TYPE, OPTIONS
1838			*		BITS 15-11=CPU TYPE
1839			*		11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1840			*		11/70=06, PDQ=07, Q=10
1841			*		BIT 10=REAL TIME CLOCK
1842			*		BIT 9=FLOATING POINT PROCESSOR
1843			*		BIT 8=MEMORY MANAGEMENT
1844	001240	000	\$MAMS1: .BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
1845	001241	000	\$MTYP1: .BYTE	AMTYP1	:: MEM. TYPE, BLK#1
1846			*		MEM. TYPE BYTE -- (HIGH BYTE)
1847			*		900 NSEC CORE=001
1848			*		300 NSEC BIPOLAR=002
1849			*		500 NSEC MOS=003
1850	001242	000000	\$MADR1: .WORD	AMADR1	:: HIGH ADDRESS, BLK#1
1851			*		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1852	001244	000	\$MAMS2: .BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
1853	001245	000	\$MTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
1854	001246	000000	\$MADR2: .WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
1855	001250	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
1856	001251	000	\$MTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
1857	001252	000000	\$MADR3: .WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
1858	001254	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
1859	001255	000	\$MTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
1860	001256	000000	\$MADR4: .WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
1861	001260	000000	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
1862	001262	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
1863	001264	177440	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
1864	001266	000000	\$DEVN: .WORD	ADEVN	:: DEVICE MAP
1865	001270	000000	\$CDW1: .WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
1866	001272	000000	\$CDW2: .WORD	ACDW2	:: CONTROLLER DESCRIPTION WORD#2
1867	001274	000000	\$DDW0: .WORD	ADDW0	:: DEVICE DESCRIPTOR WORD#0
1868	001276	000000	\$DDW1: .WORD	ADDW1	:: DEVICE DESCRIPTOR WORD#1
1869	001300	000000	\$DDW2: .WORD	ADDW2	:: DEVICE DESCRIPTOR WORD#2
1870	001302	000000	\$DDW3: .WORD	ADDW3	:: DEVICE DESCRIPTOR WORD#3
1871	001304	000000	\$DDW4: .WORD	ADDW4	:: DEVICE DESCRIPTOR WORD#4
1872	001306	000000	\$DDW5: .WORD	ADDW5	:: DEVICE DESCRIPTOR WORD#5
1873	001310	000000	\$DDW6: .WORD	ADDW6	:: DEVICE DESCRIPTOR WORD#6
1874	001312	000000	\$DDW7: .WORD	ADDW7	:: DEVICE DESCRIPTOR WORD#7
1875	001314	000000	\$DDW8: .WORD	ADDW8	:: DEVICE DESCRIPTOR WORD#8
1876	001316	000000	\$DDW9: .WORD	ADDW9	:: DEVICE DESCRIPTOR WORD#9
1877	001320	000000	\$DDW10: .WORD	ADDW10	:: DEVICE DESCRIPTOR WORD#10
1878	001322	000000	\$DDW11: .WORD	ADDW11	:: DEVICE DESCRIPTOR WORD#11
1879	001324	000000	\$DDW12: .WORD	ADDW12	:: DEVICE DESCRIPTOR WORD#12
1880	001326	000000	\$DDW13: .WORD	ADDW13	:: DEVICE DESCRIPTOR WORD#13

1881	001330	000000	\$DDW14: .WORD	ADDW14	:::DEVICE DESCRIPTOR WORD#14
1882	001332	000000	\$DDW15: .WORD	ADDW15	:::DEVICE DESCRIPTOR WORD#15
1883					
1884					
1885	001334		SETEND:		
1886					
1887		177440	ABASE=	177440	; DEFAULT BUSS ADDRESS
1888	001334	000210	RKVEC:	210	; DEFAULT CONTROLLER INTERRUPT VECTOR
1889	001336	000240	RKPRI:	PRS	; PRIORITY
1890	001340	172540	PKS:	172540	; P-CLOCK STATUS REG
1891	001342	172542	PKSB:	172542	; P-CLOCK SET BUFFER
1892	001344	172544	PKRB:	172544	; P-CLOCK READ BUFFER
1893	001346	177546	LKS:	177546	; L-CLOCK STATUS REG.
1894					
1895	001350	000100	LCVEC:	100	; L-CLOCK INTERRUPT VECTOR
1896	001352	000104	PCVEC:	104	; P-CLOCK INTERRUPT VECTOR.
1897					
1898		000114	MEMVEC=	114	; MEMORY PARITY VECTOR
1899		172100	MEMBAS=	172100	; MEMORY PARITY OPTION CSR START ADDR
1900					
1901	001354	000000	TRAPPC:	0	; PC FOR MEMORY PARITY ERROR TRAP
1902					
1903	001356	000000	PARAM:	0	; 1 FOR 220 START, NO DEFAULT
1904	001360	000000	FTITLE:	0	; FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
1905					
1906	001362	000000	DRVPTR:	0	; CONTAINS THE POINTER TO THE DRIVE FLAG
1907					; (DRIVO-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
1908					
1909		000040	SPBAR=	40	; SPACE BAR
1910	001364	000000	FRCYL:	0	; FROM CYLINDER
1911	001366	000000	TOCYL:	0	; TO CYLINDER
1912	001370	000000	CCYL:	0	; CURRENT CYL, USED IN N SQUARE TEST
1913	001372	000000	PCYL:	0	; PREV CYL, USED IN N SQUARE TEST
1914	001374	000000	CALDIF:	0	; CALC CYL DIFF USED IN N SQUARE TEST
1915	001376	000000	CYLDIF:	0	; CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
1916	001400	000000	CYLFJD:	0	; CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
1917	001402	000000	CALMOD:	0	; CYL ADDR USED IN FHD0AB ROUTINE
1918					
1919	001404	000074	HZ:	60.	; 60 FOR 60 CPS
1920					; 50 FOR 50 CPS
1921	001406	000000	COUNT:	0	; LOADED TO 50 OR 60 TO COUNT TO 1 SEC
1922					; OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
1923	001410	000000	SEC:	0	; SECOND COUNTER
1924	001412	000000	TIMUP:	0	; FLAG TO INDICATE TIME IS UP
1925	001414	000000	SECT:	0	; SECTOR COUNT
1926	001416	000000	PSEC:	0	; PREVIOUS SECTOR
1927	001420	000000	ESEC:	0	; EXPECTED SECTOR
1928	001422	000000	SECTOR:	0	; SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
1929					
1930	001424	000001	T1:	1	; TIMEOUT CONSTANTS
1931	001426	000012	T10:	10.	
1932	001430	000062	T50:	50.	
1933	001432	000764	T500:	500.	
1934	001434	000144	T100:	100.	
1935	001436	011610	T5000:	5000.	
1936	001440	141520	T50000:	50000.	

1937					
1938	001442	000077	CYL:	63.	;CYLINDER NUMBERS USED IN
1939	001444	000177		127.	;CURRENT CROSSOVER TEST
1940	001446	000277		191.	
1941	001450	000377		255.	
1942	001452	000477		319.	
1943	001454	000577		383.	
1944					
1945	001456	000000	TIM1:	0	;USED IN TIMING TESTS
1946	001460	000000	TIM2:	0	
1947	001462	000000	TIM3:	0	
1948	001464	000000	TIM4:	0	
1949					
1950	001466	000000	LPCNT:	0	;LOOP CTR USED IN CALCLK
1951	001470	000000	LPTIM:	0	;LOOP TIME IN USEC
1952					
1953	001472	000000	SUM:	0	;LO ORDER FOR TIMING TESTS
1954	001474	000000		0	;HI ORDER FOR TIMING TESTS
1955	001476	000000	SUM1:	0	;LO ORDER FOR TIMING TESTS
1956	001500	000000		0	;HI ORDER FOR TIMING TESTS
1957					
1958	001502	000000	WD1:	0	;ACTUAL HEADER/DATA WORD
1959	001504	000000	WD2:	0	;EXPECTED DATA WORD
1960					
1961	001506	000000	OFFERR:	0	;SET WHEN WRITE CHECK ERROR ON OFFSET
1962					
1963					
1964	001510	000000	HEAD:	0	;HEAD NUMBER
1965	001512	000000	HEADR:	0	;HEAD # FROM H.B3, RT, JUSTIFIED
1966	001514	000000	HD1:	0	;SHIFTED HEAD# F'3 FORMATTER ROUTINE
1967	001516	000000	FORMAT:	0	;FORMAT TYPE
1968	001520	000000	FMT1:	0	;SHIFTED FORMAT FOR FORMATTER ROUTINE
1969	001522	000000	WDCNT:	0	;WORD COUNT
1970					
1971	001524	000000	DATA0:	0	;ALL 0'S
1972	001526	052525	DATA01:	52525	;0101 PATT
1973	001530	177777	DATA1:	177777	;ALL 1'S
1974	001532	133467	DPAT1:	133467	
1975	001534	070627	DPAT2:	70627	
1976					
1977	001536	000000	WORD:	0	;HEADER/DATA WORD
1978	001540	000000	HOWD:	0	;HEADER WORD FROM RKDB
1979					
1980	001542	000000	BSERR:	0	;CANNOT READ BSE INFO WHEN SET
1981	001544	000000	LIMERR:	0	;LIMIT DETECT ERROR FLAG
1982	001546	000000	MDSERR:	0	;MULT DRIVE SEL ERROR FLAG
1983	001550	000000	BYPCERR:	0	;SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
1984	001552	000000	CHKFLG:	0	;WORDS TO BE TESTED
1985					
1986	001554	000102	HDTAB:	.BLKW 66.	;CALCULATED HEADER WORD TABLE
1987	001760	000102	RHTAB:	.BLKW 66.	;FILLED AFTER READ HEADER CMD
1988	002164	000102	SRTTAB:	.BLKW 66.	;ABOVE RHTAB SORTED STARTING FORM
1989					;SECTOR 0 BY SORT ROUTINE
1990	002370	000400	BSE22H:	.BLKW 256.	;22 SECTOR HARDWARE BSE INFO.
1991	003370	000400	BSE22S:	.BLKW 256.	;22 SECTOR SOFTWARE BSE INFO.
1992	004370	000400	RDTab:	.BLKW 256.	;FILLED AFTER READ DATA CMD

1993										
1994	005370	000000				UNLD:	0			;SET TO 0 IF HEADS ARE LOADED
1995										;SET TO 1 IF HEADS UNLOADED
1996	005372	000000				BADHDR:	0			;SET TO 0 IF FORMATTING OK
1997										;SET TO 1 IF FORMATTING ALTERED
1998	005374	000000				HPEND:	0			;SET TO 0 IF HALT NOT PENDING
1999										;SET TO 1 IF HALT PENDING

;THE ABOVE 3 FLAGS ARE USED
;BY 'STOP' ROUTINE TO BRING
;THE CPU TO A VALID HALT.

2005										
2006	005376	001	002	004	ATTN:	.BYTE 1,2,4,10,20,40,100,200				;ATN 0-7 RESP.
2007	005401	010	020	040						
2008	005404	100	200							
2009						.EVEN				

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

2014										
2015	005406	000000			HCS1:	0				;HOLD RKCS1
2016	005410	000000			HCS2:	0				;HOLD RKCS2
2017	005412	000000			HWC:	0				;HOLD RKWC
2018	005414	000000			HBA:	0				;ETC.
2019	005416	000000			HDA:	0				
2020	005420	000000			HDS:	0				
2021	005422	000000			HER:	0				
2022	005424	000000			HASOF:	0				
2023	005426	000000			HDC:	0				
2024	005430	000000			HDB:	0				
2025	005432	000000			HMR1:	0				
2026	005434	000000			HMR2:	0				
2027	005436	000000			HMR3:	0				
2028	005440	000000			HPOS:	0				
2029	005442	000000			HPAT:	0				

;TEMPORARY STORAGE.

2030										
2031	005444	000000			TEMP1:	0				
2032	005446	000000			TEMP2:	0				
2033	005450	000000			TEMP3:	0				
2034	005452	000000			TEMP4:	0				
2035	005454	000000			TEMP5:	0				

;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).

2038										
2039	005456	000000			H.A0:	0				
2040	005460	000000			H.B0:	0				
2041	005462	000000			H.A1:	0				
2042	005464	000000			H.B1:	0				
2043	005466	000000			H.A2:	0				
2044	005470	000000			H.B2:	0				
2045	005472	000000			H.A3:	0				
2046	005474	000000			H.B3:	0				

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

2047
2048

2049
 2050 005476 000000
 2051 005500 000000
 2052 005502 000000
 2053 005504 000000
 2054 005506 000000
 2055 005510 000000
 2056 005512 000000
 2057 005514 000000
 2058
 2059
 2060
 2061 000001
 2062 000002
 2063 000004
 2064
 2065
 2066
 2067
 2068
 2069 005516 000000
 2070 005520 000000
 2071 005522 000000
 2072 005524 000000
 2073 005526 000000
 2074
 2075
 2076
 2077
 2078 005530 000000
 2079 005532 000000
 2080 005534 000000
 2081 005536 000000
 2082 005540 000000
 2083 005542 000000
 2084 005544 000000
 2085 005546 000000
 2086
 2087 005550 000000
 2088 005552 000000
 2089 005554 000000
 2090 005556 000000

·
 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 IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.

·
 †.A2=BIT0 ;TEST MSG A2 IF SET
 T.B2=BIT1
 T.B3=BIT2

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

·
 DDUMP: 0 ;FLAG - SET WHEN IN DDP DUMP MODE
 DDPCH: 0 ;FLAG - SET WHEN IN DDP CHAIN MODE
 ACT11: 0 ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
 PPTP: 0 ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
 DRIVS: 0 ;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.

DRIV0: 0 ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
 DRIV1: 0 ;FOR DRIVE 1
 DRIV2: 0 ;FOR DRIVE 2
 DRIV3: 0 ;FOR DRIVE 3
 DRIV4: 0 ;FOR DRIVE 4
 DRIV5: 0 ;FOR DRIVE 5
 DRIV6: 0 ;FOR DRIVE 6
 DRIV7: 0 ;FOR DRIVE 7

LCLKF: 0 ;L-CLOCK FLAG PRESENT FLAG
 PCLKF: 0 ;P-CLOCK FLAG PRESENT FLAG
 DOTIM: 0 ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
 SIZFLG: 0 ;SET IF DEFAULT DO SIZING IN TEST 1

2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105 005560
2106
2107
2108 005560 044315
2109 005562 050646
2110 005564 053764
2111 005566 054406
2112
2113
2114 005570 044534
2115 005572 050646
2116 005574 053764
2117 005576 054406
2118
2119
2120 005600 044555
2121 005602 050646
2122 005604 053764
2123 005606 054406
2124
2125
2126 005610 044576
2127 005612 050646
2128 005614 053764
2129 005616 054406
2130
2131 005620 044665
2132 005622 050646
2133 005624 053764
2134 005626 054406
2135
2136
2137 005630 044741
2138 005632 050646
2139 005634 053764
2140 005636 054406
2141
2142
2143 005640 045015
2144 005642 050646
2145 005644 053764
2146 005646 054406

.SBTTL ERROR POINTER TABLE

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

```

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

\$ERRTB:

```

;ERROR 1
      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 UFE
      DH1
      DT1
      DF1

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

;ERROR 5
      EM8      ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
      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
    
```


2259			;ERR 35		
2260	006120	045330		EM17	;MSG AD ERROR
2261	006122	051052		DH5	;AFTER AC SW OFF
2262	006124	054142		DT13	
2263	006126	054646		DF21	
2264			;ERR 36		
2265	006130	045372		EM19	;MSG A1 ERROR
2266	006132	051052		DH5	
2267	006134	054142		DT13	
2268	006136	054646		DF21	
2269			;ERR 37		
2270	006140	045413		EM20	;MSG A1 ERROR
2271	006142	051052		DH5	;AFTER OFFSET CMD
2272	006144	054142		DT13	
2273	006146	054646		DF21	
2274			;ERR 40		
2275	006150	050035		EM71	;DETECTED 10 BAD SECTORS
2276	006152	051733		DH27	;AFTER WRITE DATA CMD
2277	006154	053764		DT1	
2278	006156	054512		DF10	
2279			;ERR 41		
2280	006160	045224		EM14	;WRONG ATTN
2281	006162	051443		DH18	;AFTER UNLOAD CMD
2282					
2283	006164	053764		DT1	
2284	006166	054512		DF10	
2285			;ERR 42		
2286	006170	045251		EM15	;DRDY NOT CLEARED
2287	006172	051443		DH18	
2288					
2289	006174	053764		DT1	
2290	006176	054512		DF10	
2291			;ERR 43		
2292	006200	045303		EM16	;DSC NOT SET
2293	006202	051443		DH18	
2294	006204	053764		DT1	
2295	006206	054512		DF10	
2296			;ERR 44		
2297	006210	045456		EM22	;DOOR NOT CLEARED
2298	006212	051100		DH8	;AFTER DRIVE UNLOADED & DOOR OPENED
2299	006214	053764		DT1	
2300	006216	054512		DF10	
2301			;ERR 45		
2302	006220	045330		EM17	;MSG AD ERROR
2303	006222	051100		DH8	
2304	006224	054142		DT13	
2305	006226	054646		DF21	
2306			;ERR 46		
2307	006230	045351		EM18	;MSG B0 ERROR
2308	006232	051100		DH8	
2309	006234	054142		DT13	
2310	006236	054646		DF21	
2311			;ERR 47		
2312	006240	050105		EM72	;BSE ERROR IN WRITE CMD NOT ON BSE TABLE
2313	006242	051733		DH27	;AFTER WRITE DATA CMD
2314	006244	053764		DT1	

2315	006246	054512	DF10	
2316				
2317				
2318	006250	050164	EM73	; DETECTED BSE IN READ BUT NOT IN WRITE CMD.
2319	006252	050646	DH1	
2320	006254	053764	DT1	
2321	006256	054406	DF1	
2322				
2323	006260	050603	EM93	; WRONG CYL# IN HEADER WORD
2324	006262	051660	DH25	; AFTER SEEK CMD
2325	006264	054076	DT9	
2326	006266	054622	DF20	
2327				
2328	006270	045330	EM17	; MSG A0 ERROR
2329	006272	051733	DH27	; AFTER WRITE DATA CMD
2330	006274	054142	DT13	
2331	006276	054646	DF21	
2332				
2333	006300	045372	EM19	; MSG A1 ERROR
2334	006302	051733	DH27	
2335	006304	054142	DT13	
2336	006306	054646	DF21	
2337				
2338	006310	045330	EM17	; MSG A0 ERROR
2339	006312	051703	DH26	; AFTER READ DATA CMD
2340	006314	054142	DT13	
2341	006316	054646	DF21	
2342				
2343	006320	045202	EM13	; NO ATTN
2344	006322	051417	DH17	; AFTER RECAL CMD
2345	006324	053764	DT1	
2346	006326	054512	DF10	
2347				
2348	006330	045372	EM19	; MSG A1 ERROR
2349	006332	051703	DH26	
2350	006334	054142	DT13	
2351	006336	054646	DF21	
2352				
2353	006340	045330	EM17	; MSG A0 ERROR
2354	006342	052142	DH32	; AFTER WRITE CHECK CMD
2355	006344	054142	DT13	
2356	006346	054646	DF21	
2357				
2358	006350	045372	EM19	; MSG A1 ERROR
2359	006352	052142	DH32	
2360	006354	054142	DT13	
2361	006356	054646	DF21	
2362				
2363	006360	047645	EM69	; NO DRIVES PRESENT
2364	006362	050646	DH1	
2365	006364	053764	DT1	
2366	006366	054406	DF1	
2367				
2368	006370	050330	EM75	; FOUND 10 BAD CYL
2369	006372	051733	DH27	; AFTER WRITE DATA CMD
2370	006374	053764	DT1	

H04

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 45
ERROR POINTER TABLE

SEQ 0045

2371	006376	054512		
2372			;ERR 63	DF10
2373	006400	045372		EM19 ;MSG A1 ERROR
2374	006402	051100		DH8 ;AFTER DRIVE UNLOADED & DOOR OPENED
2375	006404	054142		DT13
2376	006406	054646		DF21 ;
2377			;ERR 64	
2378	006410	045413		EM20 ;MSG B1 ERROR
2379	006412	051100		DH8
2380	006414	054142		DT13
2381	006416	054646		DF21
2382			;ERR 65	
2383	006420	045517		EM23 ;SPIN SET
2384	006422	051143		DH11 ;AFTER LOADING HEADS WITH DOOR OPEN
2385	006424	053764		DT1
2386	006426	054512		DF10
2387			;ERR 66	
2388	006430	045330		EM17 ;MSG A0 ERROR
2389	006432	051143		DH11
2390	006434	054142		DT13
2391	006436	054646		DF21
2392			;ERR 67	
2393	006440	045351		EM18 ;MSG B0 ERROR
2394	006442	051143		DH11
2395	006444	054142		DT13
2396	006446	054646		DF21
2397			;ERR 70	
2398	006450	045372		EM19 ;MSG A1 ERROR
2399	006452	051143		DH11
2400	006454	054142		DT13
2401	006456	054646		DF21
2402			;ERR 71	
2403	006460	045413		EM20 ;MSG B1 ERROR
2404	006462	051143		DH11
2405	006464	054142		DT13
2406	006466	054646		DF21
2407			;ERR 72	
2408	006470	045605		EM25 ;CARTRIDGE NOT CLEARED
2409	006472	051217		DH12 ;AFTER DISK PACK REMOVED
2410	006474	053764		DT1
2411	006476	054512		DF10
2412			;ERR 73	
2413	006500	000000		0
2414	006502	000000		0
2415	006504	000000		0
2416	006506	000000		0
2417			;ERR 74	
2418	006510	045330		EM17 ;MSG A0 ERROR
2419	006512	051004		DH4 ;AFTER PACK RE-INSERTED & HDS LOADED
2420	006514	054142		DT13
2421	006516	054646		DF21
2422			;ERR 75	
2423	006520	045351		EM18 ;MSG B0 ERROR
2424	006522	051004		DH4
2425	006524	054142		DT13
2426	006526	054646		DF21

2427				
2428	006530	045372		
2429	006532	051004		
2430	006534	054142		
2431	006536	054646		
2432				
2433	006540	045413		
2434	006542	051004		
2435	006544	054142		
2436	006546	054646		
2437				
2438	006550	045517		
2439	006552	051247		
2440	006554	053764		
2441	006556	054512		
2442				
2443	006560	000000		
2444	006562	000000		
2445	006564	000000		
2446	006566	000000		
2447				
2448	006570	000000		
2449	006572	000000		
2450	006574	000000		
2451	006576	000000		
2452				
2453	006600	000000		
2454	006602	000000		
2455	006604	000000		
2456	006606	000000		
2457				
2458	006610	000000		
2459	006612	000000		
2460	006614	000000		
2461	006616	000000		
2462				
2463	006620	047043		
2464	006622	053374		
2465	006624	053764		
2466	006626	054512		
2467				
2468	006630	045742		
2469	006632	051333		
2470	006634	053764		
2471	006636	054512		
2472				
2473	006640	045774		
2474	006642	053050		
2475	006644	053764		
2476	006646	054512		
2477				
2478	006650	046632		
2479	006652	052655		
2480	006654	053764		
2481	006656	054512		
2482				

```

;ERR 76
EM19 ;MSG A1 ERROR
DH4
DT13
DF21
;ERR 77
EM20 ;MSG B1 ERROR
DH4
DT13
DF21
;ERR 100
EM23 ;SPIN SET
DH13 ;AFTER LOADING HEADS WITH CART. OUT
DT1
DF10
;ERR 101
0
0
0
0
;ERR 102
0
0
0
0
;ERR 103
0
0
0
0
;ERR 104
0
0
0
0
;ERR 105
EM52 ;UNS NOT SET
DH64 ;AFTER MDS FOUND
DT1
DF10
;ERR 106
EM28 ;VV SET
DH15 ;WITHOUT PACK CMD
DT1
DF10
;ERR 107
EM29 ;DSC NOT SET
DH59 ;AFTER EVEN PARITY ISSUED
DT1
DF10
;ERR 110
EM48 ;WRL NOT CLEARED
DH48 ;AFTER WRITE LOCK SWITCH DISABLED
DT1
DF10
;ERR 111

```


2483	006660	046021	EM30	;ATTN NOT CLEARED
2484	006662	051360	DH16	;AFTER UNIT SELECT PLUG REMOVED
2485	006664	053764	DT1	
2486	006666	054512	DF10	
2487			;ERR 112	
2488	006670	045715	EM27	;NED NOT SET
2489	006672	051360	DH16	
2490	006674	053764	DT1	
2491	006676	054512	DF10	
2492			;ERR 113	
2493	006700	045202	EM13	;ATTN NOT SET
2494	006702	053050	DH59	;AFTER EVEN PARITY ISSUED
2495	006704	053764	DT1	
2496	006706	054512	DF10	
2497			;ERROR 114	
2498	006710	047315	EM58	;PARITY NOT SET
2499	006712	053050	DH59	
2500	006714	053764	DT1	
2501	006716	054512	DF10	
2502			;ERR 115	
2503	006720	046672	EM49	;WRL NOT SET
2504	006722	052716	DH49	;AFTER WRITE LOCK SW ENABLED
2505	006724	053764	DT1	
2506	006726	054512	DF10	
2507			;ERROR 116	
2508	006730	045144	EM12	;CONT NOT RDY
2509	006732	051470	DH19	;AFTER PACK CMD
2510	006734	053764	DT1	
2511	006736	054512	DF10	
2512			;ERROR 117	
2513	006740	045144	EM12	;CONT NOT RDY
2514	006742	051513	DH20	;AFTER SEL DR CMD
2515	006744	053764	DT1	
2516	006746	054512	DF10	
2517			;ERROR 120	
2518	006750	045144	EM12	
2519	006752	051546	DH21	;AFTER SUBSYS CLEAR
2520	006754	053764	DT1	
2521	006756	054512	DF10	
2522			;ERR 121	
2523	006760	046726	EM50	;WLE NOT SET
2524	006762	052756	DH50	;AFTER WRITING WITH WRITE LOCK SET
2525	006764	053764	DT1	
2526	006766	054512	DF10	
2527			;ERR 122	
2528	006770	045715	EM27	;NED NOT SET
2529	006772	051626	DH23	;AFTER WRONG PORT SELECTED
2530	006774	053764	DT1	
2531	006776	054512	DF10	
2532			;ERR 123	
2533	007000	045330	EM17	;MSG AD ERROR
2534	007002	052756	DH50	;AFTER WRITING WITH WRITE LOCK ENABLED
2535	007004	054142	DT13	
2536	007006	054646	DF21	
2537			;ERROR 124	
2538	007010	045144	EM12	

K04

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR&JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 48
ERROR POINTER TABLE

SEQ 0048

2539	007012	051417	DH17	;AFTER RECAL CMD
2540	007014	053764	DT1	
2541	007016	054512	DF10	
2542			;ERR 125	
2543	007020	045351	E418	;MSG B0 ERROR
2544	007022	052756	DH50	;AFTER WITING WITH WRL ENABLED
2545	007024	054142	DT13	
2546	007026	054646	DF21	
2547			;ERR 126	
2548	007030	045372	EM19	;MSG A1 ERROR
2549	007032	052756	DH50	
2550	007034	054142	DT13	
2551	007036	054646	DF21	
2552			;ERR 127	
2553	007040	045413	EM20	;MSG B1 ERROR
2554	007042	052756	DH50	
2555	007044	054142	DT13	
2556	007046	054646	DF21	
2557			;ERR 130	
2558	007050	046052	EM31	;NED NOT CLEARED
2559	007052	052020	DH29	;AFTER CORRECT PORT SELECTED
2560	007054	053764	DT1	
2561	007056	054512	DF10	
2562			;ERROR 131	
2563	007060	045144	EM12	;NO RDY
2564	007062	051660	DH25	;AFTER SEEK CMD
2565	007064	053764	DT1	
2566	007066	054512	DF10	
2567			;ERROR 132	
2568	007070	045202	EM13	;NO ATTN
2569	007072	051660	DH25	
2570	007074	053764	DT1	
2571	007076	054512	DF10	
2572			;ERR 133	
2573	007100	045715	EM27	;NED NOT SET
2574	007102	051764	DH28	;AFTER BOTH PORTS DESELECTED
2575	007104	053764	DT1	
2576	007106	054512	DF10	
2577			;ERR 134	
2578	007110	046770	EM51	;WRITE LOCK NOT SET SECTOR BOUNDRY
2579	007112	052756	DH50	;AFTER WRITING WITH WRL ENABLED
2580	007114	054024	DT3	
2581	007116	054422	DF3	
2582			;ERR 135	
2583	007120	046770	EM51	
2584	007122	053124	DH60	;AFTER WRITE LOCK ENABLED WHILE WRITING
2585	007124	054024	DT3	
2586	007126	054446	DF4	
2587			;ERR 136	
2588	007130	046770	EM51	
2589	007132	053323	DH63	;AFTER WRITE LOCK ENABLED FROM AC OFF
2590	007134	054024	DT3	
2591	007136	054422	DF3	
2592			;ERR 137	
2593	007140	046770	EM51	
2594	007142	053323	DH63	

2595	007144	054024	DT3	
2596	007146	054446	DF4	
2597				
2598	007150	046103	EM32	; SPINDLE ON NOT SET
2599	007152	052106	DH31	; AFTER DRIVE MANUALLY LOADED
2600	007154	053764	DT1	
2601	007156	054512	DF10	
2602				
2603	007160	046137	EM33	; DRIVE NOT READY
2604	007162	052315	DH38	; AFTER AC POWERED UP
2605	007164	053764	DT1	
2606	007166	054512	DF10	
2607				
2608	007170	046170	EM34	
2609	007172	052106	DH31	
2610	007174	053764	DT1	
2611	007176	054512	DF10	
2612				
2613	007200	045144	EM12	; CONT NOT READY
2614	007202	052174	DH34	; AFTER ST SPIN. CMD
2615	007204	053764	DT1	
2616	007206	054512	DF10	
2617				
2618	007210	045202	EM13	; NO ATTN
2619	007212	052174	DH34	
2620	007214	053764	DT1	
2621	007216	054512	DF10	
2622				
2623	007220	046231	EM35	; HEADS NOT HOME
2624	007222	052230	DH35	; AFTER MANUAL UNLOAD
2625	007224	053764	DT1	
2626	007226	054512	DF10	
2627				
2628	007230	047267	EM57	; CERR NOT SET
2629	007232	052261	DH37	; AFTER TIMEOUT TO POWER DOWN
2630	007234	053764	DT1	
2631	007236	054512	DF10	
2632				
2633	007240	046765	EM37	; AC LOW NOT SET
2634	007242	052261	DH37	
2635	007244	053764	DT1	
2636	007246	054512	DF10	
2637				
2638	007250	046315	EM42	; NED NOT SET
2639	007252	052261	DH37	
2640	007254	053764	DT1	
2641	007256	054512	DF10	
2642				
2643	007260	045144	EM12	; NO RDY
2644	007262	051574	DH22	; AFTER CLEAR CMD
2645	007264	053764	DT1	
2646	007266	054512	DF10	
2647				
2648	007270	046342	EM43	; AC LO NOT CLEARED
2649	007272	052315	DH38	; AFTER AC POWERED UP
2650	007274	053764	DT1	

2651	007276	054512	DF10	
2652			;ERR 153	
2653	007300	046374	EM44	;VV NOT CLEARED
2654	007302	052315	DH38	
2655	007304	053764	DT1	
2656	007306	054512	DF10	
2657			;ERROR 154	
2658	007310	047170	EM55	;ATTN NOT CLEARED
2659	007312	051574	DH22	
2660	007314	053764	DT1	
2661	007316	054512	DF10	
2662			;ERR 155	
2663	007320	046436	EM45	;VV SET AFTER HDS LOADED
2664	007322	051333	DH15	;WITHOUT 'PACK' CMD
2665	007324	053764	DT1	
2666	007326	054512	DF10	
2667			;ERR 156	
2668	007330	046513	EM46	;NXF=0
2669	007332	052551	DH45	;AFTER SEEK WITH VV=0
2670	007334	053764	DT1	
2671	007336	054512	DF10	
2672			;ERR 157	
2673	007340	046572	EM47	;CYL ADDR CHANGED FROM 0
2674	007342	052551	DH45	
2675	007344	054222	DT14	
2676	007346	054702	DF22	
2677			;ERR 160	
2678	007350	045330	EM17	;MSG A0 ERROR
2679	007352	052551	DH45	
2680	007354	054142	DT13	
2681	007356	054646	DF21	
2682			;ERR 161	
2683	007360	045351	EM18	;MSG B0 ERROR
2684	007362	052551	DH45	
2685	007364	054142	DT13	
2686	007366	054646	DF21	
2687			;ERR 162	
2688	007370	045372	EM19	;MSG A1 ERROR
2689	007372	052551	DH45	
2690	007374	054142	DT13	
2691	007376	054646	DF21	
2692			;ERR 163	
2693	007400	045413	EM20	;MSG B1 ERROR
2694	007402	052551	DH45	
2695	007404	054142	DT13	
2696	007406	054646	DF21	
2697			;ERR 164	
2698	007410	046513	EM46	;NXF NOT SET
2699	007412	052610	DH46	;AFTER WRITE DATA WITH VV=0
2700	007414	053764	DT1	
2701	007416	054512	DF10	
2702			;ERR 165	
2703	007420	047603	EM68	;CANNOT READ BSE INFO
2704	007422	052421	DH42	;ON SECTORS 0, 2, 4, 6, 8
2705	007424	053764	DT1	
2706	007426	054576	DF17	

2707			;ERR 166	
2708	007430	000000		0
2709	007432	000000		0
2710	007434	000000		0
2711	007436	000000		0
2712			;ERR 167	
2713	007440	047603		EM68
2714	007442	053700		DH74
2715	007444	053764		DT1
2716	007446	054576		DF17
2717			;ERR 170	
2718	007450	000000		0
2719	007452	000000		0
2720	007454	000000		0
2721	007456	000000		0
2722			;ERROR 171	
2723	007460	045144		EM12
2724	007462	052054		DH30
2725	007464	053764		DT1
2726	007466	054512		DF10
2727			;ERROR 172	
2728	007470	047366		EM61
2729	007472	052551		DH45
2730	007474	053764		DT1
2731	007476	054512		DF10
2732			;ERROR 173	
2733	007500	047414		EM63
2734	007502	052054		DH30
2735	007504	053764		DT1
2736	007506	054556		DF15
2737			;ERROR 174	
2738	007510	045434		EM21
2739	007512	052054		DH30
2740	007514	053764		DT1
2741	007516	054556		DF15
2742			;ERR 175	
2743	007520	047070		EM53
2744	007522	053374		DH64
2745	007524	053764		DT1
2746	007526	054512		DF10
2747			;ERR 176	
2748	007530	047120		EM54
2749	007532	053430		DH65
2750	007534	053764		DT1
2751	007536	054512		DF10
2752			;ERROR 177	
2753	007540	045653		EM26
2754	007542	051360		DH16
2755	007544	053764		DT1
2756	007546	054512		DF10
2757			;ERROR 200	
2758	007550	045144		EM12
2759	007552	052341		DH39
2760	007554	053764		DT1
2761	007556	054556		DF15
2762			;ERROR 201	

;ON SEC 10, 12....20

;NO RDY
;AFTER READ HEADER CMD

;NXF DID NOT SET FAULT
;AFTER SEEK WITH VV=0

;DLT SET

;CERR SET

;UNLD NOT SET
;AFTER MDS FOUND

;CANNOT FIND MDS
;AFTER SEARCHING ALL DRIVES

;VV NOT CLEARED
;AFTER UNIT SEL PLUG REMOVED

;NO RDY
;AFTER WRITE HEADER CMD

2763	007560	045434	EM21	;CERR SET
2764	007562	052341	DH39	
2765	007564	053764	DT1	
2766	007566	054556	DF15	
2767			;ERROR 202	
2768	007570	047223	EM56	;UNEXP MEMORY PARITY ERROR
2769	007572	053473	DH65	;TEST #,PRAP PC
2770	007574	054072	DT6	
2771	007576	054472	DF5	
2772			;ERROR 203	
2773	007600	045351	EM18	;MSG BO ERROR
2774	007602	051052	DH5	;AFTER AC SWITCHED OFF
2775	007604	054142	DT13	
2776	007606	054646	DF21	
2777			;ERR 204	
2778	007610	047267	EM57	;CERR NOT SET
2779	007612	053514	DH67	;AFTER TIMEOUT TO ENABLE WRL
2780	007614	053764	DT1	
2781	007616	054512	DF10	
2782			;ERR 205	
2783	007620	046726	EM50	;WRL NOT SET
2784	007622	053514	DH67	
2785	007624	053764	DT1	
2786	007626	054512	DF10	
2787			;ERROR 206	
2788	007630	047435	EM64	;WCE AT CYL 411,TRK 2, SEC 21
2789	007632	050646	DH1	
2790	007634	053764	DT1	
2791	007636	054476	DF7	
2792			;ERROR 207	
2793	007640	047267	EM57	;CERR NOT SET
2794	007642	052756	DH50	;AFTER WRITING WITH WRL ENABLED
2795	007644	053764	DT1	
2796	007646	054512	DF10	
2797			;ERROR 210	
2798	007650	045434	EM21	;CERR SET
2799	007652	051660	DH25	
2800	007654	053764	DT1	
2801	007656	054512	DF10	
2802			;ERR 211	
2803	007660	047345	EM59	;CTO SET
2804	007662	047744	EM70	;WHILE WAITING FOR OR REC'D CONTR RDY MSG A & B BAD
2805	007664	053764	DT1	
2806	007666	054532	DF12	
2807			;ERR 212	
2808	007670	047562	EM67	;NED SET
2809	007672	047744	EM70	
2810	007674	053764	DT1	
2811	007676	054532	DF12	
2812			;ERR 213	
2813	007700	044534	EM5	;MDS SET
2814	007702	047744	EM70	
2815	007704	053764	DT1	
2816	007706	054532	DF12	
2817			;ERR 214	
2818	007710	050303	EM74	;RTZ NOT SET

2819	007712	052374	DH41	;DURING RECD CMD
2820	007714	053764	DT1	
2821	007716	054512	DF10	
2822				
2823	007720	045330	EM17	;MSG AD ERROR
2824	007722	052341	DH39	;AFTER WRITE HEADER CMD.
2825	007724	054142	DT13	
2826	007726	054646	DF21	
2827				
2828	007730	045351	EM18	;B0 ERROR
2829	007732	052341	DH39	
2830	007734	054142	DT13	
2831	007736	054646	DF21	
2832				
2833	007740	045372	EM19	;A1 ERROR
2834	007742	052341	DH39	
2835	007744	054142	DT13	
2836	007746	054646	DF21	
2837				
2838	007750	045413	EM20	;B1 ERROR
2839	007752	052341	DH39	
2840	007754	054142	DT13	
2841	007756	054646	DF21	
2842				
2843	007760	000000		
2844	007762	000000		
2845	007764	000000		
2846	007766	000000		
2847				
2848	007770	000000		
2849	007772	000000		
2850	007774	000000		
2851	007776	000000		
2852				
2853	010000	000000		
2854	010002	000000		
2855	010004	000000		
2856	010006	000000		
2857				
2858	010010	000000		
2859	010012	000000		
2860	010014	000000		
2861	010016	000000		
2862				
2863	010020	000000		
2864	010022	000000		
2865	010024	000000		
2866	010026	000000		
2867				
2868	010030	045144	EM12	;NO RDY
2869	010032	051703	DH26	;AFTER READ DATA CMD
2870	010034	053764	DT1	
2871	010036	054512	DF10	
2872				
2873	010040	045434	EM21	;CERR SET
2874	010042	051703	DH26	

D05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 54
ERROR POINTER TABLE

SEQ 0054

2875 010044 053764
2876 010046 054556

DT1
DF15


```

2877
2878
2879
2880 010050 012737 000001 001356 PARSRT: MOV #1,PARAM ;SET FLAG FOR 220 START
2881 010056 000402 BR PRGSRT ;START PROGRAM
2882
2883 010060 005037 001356 START: CLR PARAM ;CLEAR FOR 200 START
2884 010064 000005 PRGSRT: RESET ;CLEAR ALL INT ENABLE & INIT
2885 010066 012706 001100 MOV #STACK,SP ;SETUP STACK POINTER
2886 010072 012746 000000 MOV #PRO,-(SP) ;PSW LOADED TO BE
2887 010076 012746 010104 MOV #IS,-(SP) ;LSI-11 COMPATABLE
2888 010102 000002 RTI ;ENABLE ALL INTERRUPTS
2889
2890 010104 004737 033262 IS: JSR PC,STKINT ;SETUP KB VECTOR ADDR, PRIORITY 4
2891 ; & TURN ON KB INTERRUPT
2892
2893
2894 ;*** CPU PRIORITY LEVEL NOW AT 0 ***
2895 ;*** ANY DEVICE WHICH SETS ITS ***
2896 ;*** INTERRUPT ENABLE BIT WILL ***
2897 ;*** SERVICED. ***
2898
2899 ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'STS')
2900 ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
2901 ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
2902
2903
2904 ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
2905
2906 ;SYSMAC 'SETUP'
2907 .SBTTL INITIALIZE THE COMMON TAGS ;CLEAR THE COMMON TAGS ($CMTAG) AREA
2908 010110 012706 001100 MOV #SCMTAG,R6 ;FIRST LOCATION TO BE CLEARED
2909 010114 005026 CLR (R6)+ ;CLEAR MEMORY LOCATION
2910 010116 022706 001140 CMP #SWR,R6 ;;DONE?
2911 010122 001374 BNE -6 ;LOOP BACK IF NO
2912 010124 012706 001100 MOV #STACK,SP ;SETUP THE STACK POINTER
2913
2914 ;; INITIALIZE A FEW VECTORS
2915 010130 012737 031370 000020 MOV #SCOPE,#IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
2916 010136 012737 000340 000022 MOV #340,#IOTVEC+2 ;LEVEL 7
2917 010144 012737 031650 000030 MOV #ERROR,#EMTVEC ;EMT VECTOR FOR ERROR ROUTINE
2918 010152 012737 000340 000032 MOV #340,#EMTVEC+2 ;LEVEL 7
2919 010160 012737 035500 000034 MOV #TRAP,#TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
2920 010166 012737 000340 000036 MOV #340,#TRAPVEC+2 ;LEVEL 7
2921 010174 012737 031124 000024 MOV #SPWRON,#PWAVEC ;POWER FAILURE VECTOR
2922 010202 012737 000340 000026 MOV #340,#PWAVEC+2 ;LEVEL 7
2923 010210 013737 023776 023770 MOV $ENDCT,$EOPCT ;SETUP END-OF-PROGRAM COUNTER
2924 010216 005037 001174 CLR $TIMES ;INITIALIZE NUMBER OF ITERATIONS
2925 010222 005037 001176 CLR $ESCAPE ;CLEAR THE ESCAPE ON ERROR ADDRESS
2926 010226 012737 000001 001115 MOVB #1,$SERMAX ;ALLOW ONE ERROR PER TEST
2927 010234 012737 010234 001106 MOV #,$SLPADR ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2928 010242 012737 010242 001110 MOV #,$SLPERR ;SETUP THE ERROR LOOP ADDRESS
2929 ;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2930 ;; EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
2931 010250 013746 000004 MOV #ERRVEC,-(SP) ;SAVE ERROR VECTOR
2932 010254 012737 010310 000004 MOV #64,$ERRVEC ;SET UP ERROR VECTOR
2933 010262 012737 177570 001140 MOV #DSWR,$SWR ;SETUP FOR A HARDWARE SWICH REGISTER

```

F05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 56
INITIALIZE THE COMMON TAGS

SEQ 0056

```

2933 010270 012737 177570 001142      MOV    #DDISP,DISPLAY ;AND A HARDWARE DISPLAY REGISTER
2934 010276 022777 177777 170634      CMP    #-1,#SWR      ;TRY TO REFERENCE HARDWARE SWR
2935 010304 001012                BNE    66$           ;BRANCH IF NO TIMEOUT TRAP OCCURRED
2936                                ;AND THE HARDWARE SWR IS NOT = -1
2937 010306 000403                BR     65$           ;BRANCH IF NO TIMEOUT
2938 010310 012716 010316 64$:    MOV    #65$, (SP)    ;SET UP FOR TRAP RETURN
2939 010314 000002                RTI
2940 010316 012737 000176 001140 65$:    MOV    #SWREG,SWR    ;POINT TO SOFTWARE SWR
2941 010324 012737 000174 001142      MOV    #DISPREG,DISPLAY
2942 010332 012637 000004 66$:    MOV    (SP)+,#ERRVEC ;RESTORE ERROR VECTOR
2943
2944 010336 005037 001216                CLR    $PASS        ;CLEAR PASS COUNT
2945 010342 132737 000200 001231      BITB  #APTSIZE,$ENVM ;TEST USER SIZE UNDER APT
2946 010350 001403                BEQ    67$          ;YES,USE NON-APT SWITCH
2947 010352 012737 001232 001140      MOV    #SSWREG,SWR  ;NO,USE APT SWITCH REGISTER
2948 010360
2949
2950 010360 012737 010424 000004 MEMPAR: MOV    #1$,ERRVEC    ;TIMEOUT VECTOR
2951 010366 012737 000340 000006      MOV    #PR7,ERRVEC+2
2952
2953 010374 012701 172100                MOV    #MEMBAS,R1   ;ADDR OF MEM CSR
2954 010400 005011 3$:    CLR    (R1)         ;SEE IF CAN REFERENCE
2955 010402 012711 000001                MOV    #1,(R1)      ;SET ENABLE BIT IF YES
2956 010406 012737 031026 000114      MOV    #MEMERR,MEMVEC ;LD MEMORY CHK VECTOR IF DONT TIMEOUT
2957 010414 012737 000340 000116      MOV    #PR7,MEMVEC+2
2958 010422 000401                BR     2$
2959
2960 010424 022626 1$:    CMP    (SP)+,(SP)+  ;ADJ STACK
2961 010426 062701 000002 2$:    ADD    #2,R1        ;TRY NEXT CSR
2962 010432 020127 172140                CMP    R1,#MEMBAS+40 ;ALL TRIED?
2963 010436 001360                BNE    3$           ;BR IN NO
2964 010440 012737 000006 000004      MOV    #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
2965 010446 005037 000006
2966                                CLR    ERRVEC+2
2967 010452 004737 024064                JSR    PC,CLRFLG    ;CLEAR DDUMP THRU SIZFLG
2968 010456 005037 001220                CLR    $DEVCT
2969 010462 005037 001222                CLR    $UNIT
2970
2971                                ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
2972                                ;
2973                                ;
2974
2975 010466 005737 000042  START1: TST    42
2976 010472 001014                BNE    1$
2977 010474 004737 024104                JSR    PC,TITLE     ;BR IF AUTO
2978 010500 123727 000041 000013      CMPB  41,#13       ;MANUAL, TYPE PROG ID
2979 010506 001010                BNE    2$           ;13=LOADED BY XXDP
2980 010510 005237 005516                INC    DDUMP        ;SET RK06 DUMP MODE FLAG
2981 010514 104401 036462                TYPE  #MSG2         ;REPLACE DR0 PACK W/SCRATCH & DO<CR>
2982 010520 000137 010534                JMP    $T2
2983 010524 000137 010600 1$:    JMP    $T3
2984 010530 005237 005524 2$:    INC    PPTP        ;SET ACT/APT/PTP DUMP MODE FLAG
2985
2986                                ;
2987                                ;CHECK IF ALL PARAMETERS DEFAULTED. IF NOT, BEGIN INPUT DIALOGUE
2988                                ;WITH OPERATOR. THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE

```

G05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 57
INITIALIZE THE COMMON TAGS

SEQ 0057

```

2989 ;DRIVE NOS. SEPERATED BY COMMAS & TERMINATED BY <CR>
2990 ; EX: DRIVES TO BE TESTED: 1,2,4<CR>
2991 ;
2992 ;
2993 010534 005737 001356 ST2: TST PARAM
2994 010540 001002 BNE 1$ ;BR IF 220 START
2995 010542 000137 010632 JMP ST4 ;200 START, DEFAULT & SIZE THE BUSS
2996 010546 104401 036533 1$: TYPE MSG3 ;DRIVES TO BE TESTED
2997 010552 004737 024164 JSR PC,GDRVS ;GET DR NOS.
2998 010556 104401 036565 TYPE MSG4 ;BUSS ADDR
2999 010562 004737 024324 JSR PC,GBA ;GET BA
3000 010566 104401 036632 TYPE MSG5 ;CONT INT VECTOR
3001 010572 004737 024352 JSR PC,GINT ;GET INT VECTOR
3002 010576 000427 BR ST5
3003
3004
3005 ;AUTO MODE
3006 ;CHECK IF LOADED BY XXDP OR OTHER. SET FLAGS & NO INPUT DIALOGUE.
3007 ;DEFAULT ALL PARAMETERS. TEST ONLY THOSE DRIVES THAT ARE READY
3008 ;ON THE BUSS
3009 ;
3010
3011 010600 123727 000041 000013 ST3: CMPB 41,#13 ;13=LOADED BY XXDP
3012 010606 001007 BNE 1$
3013 010610 005237 005520 INC DDPCH ;SET RK06 CHAIN MODE FLAG
3014 010614 004737 024104 JSR PC,TITLE
3015 010620 104401 036747 TYPE MSG7 ;DRO NOT TSTD
3016 010624 000402 BR ST4
3017 010626 005237 005522 1$: INC ACT11 ;SET ACT AUTO FLAG.
3018
3019 010632 012737 177440 001264 ST4: MOV #177440,$BASE ;DEFAULT VALUE
3020 010640 012737 000210 001334 MOV #210,RKVEC ;DEFAULT VALUE
3021 010646 004737 024404 JSR PC,SETINT
3022 010652 005237 005556 INC SI2FLG ;DO "SIZE THE BUSS" TEST
3023
3024 010656 005037 005370 ST5: CLR UNLD ;INITIALIZE FLAGS
3025 010662 005037 005372 CLR BADHDR ;USED IN 'STOP' ROUTINE
3026 010666 005037 005374 CLR HPEND ;FOR VALID PROGRAM HALTS
3027 010672 005037 001176 CLR $ESCAPE
3028 010676 012737 005530 001362 MOV #DRIVO,DRVPTD ;SETUP
3029 010704 005037 001220 CLR $DEVCT ;NO. OF DRVS DONE
3030 010710 005037 001222 CLR $UNIT ;CURRENT DRV UNDER TEST
3031 010714 012737 010762 000004 MOV #1$,ERRVEC ;SETUP TIMEOUT ERROR VECTOR
3032 010722 005777 170420 TST @LK$ ;SEE IF L-CLOCK THERE
3033 010726 005237 005550 INC LCLKF ;PRESENT, SET FLAG.
3034 010732 013700 001350 MOV LCVEC,RO ;VECTOR ADDR
3035 010736 012737 011024 000004 MOV #2$,ERRVEC
3036 010744 005777 170370 TST @PK$ ;SEE IF P-CLOCK THERE
3037 010750 005237 005552 INC PCLKF ;PRESENT, SET FLAG
3038 010754 013700 001352 MOV PCVEC,RO ;VECTOR ADDR
3039 010760 000412 BR 3$
3040
3041 010762 022626 1$: CMP (SP)+,(SP)+ ;L-CLOCK NOT THERE, CLEAR STACK
3042 010764 012737 011030 000004 MOV #4$,ERRVEC
3043 010772 005777 170342 TST @PK$ ;SEE IF P-CLOCK THERE
3044 010776 005237 005552 INC PCLKF ;PRESENT, SET FLAG

```

H05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 58
INITIALIZE THE COMMON TAGS

SEQ 0058

3045	011002	013700	001352
3046	011006	005237	005554
3047	011012	012720	030124
3048	011016	012710	000300
3049	011022	000407	
3050			
3051	011024	022626	
3052	011026	000767	
3053			
3054	011030	022626	
3055	011032	005037	005554
3056	011036	104401	037156
3057			
3058			

35:	MOV	PCVEC,RO	;VECTOR ADDR
	INC	DOTIM	;INDICATES TIMING TESTS CAN BE DONE
	MOV	#CLOCK,(RO)+	;SERVICE ROUTINE FOR CLOCKS
	MOV	#PR6,(RO)	
	BR	TST1	::GO TO NEXT TEST
25:	CMP	(SP)+,(SP)+	;P-CLOCK NOT THERE, CLEAR STACK
	BR	35	
45:	CMP	(SP)+,(SP)+	;NEITHER CLOCK THERE, CLEAR STACK
	CLR	DOTIM	;TIMING TESTS CANNOT BE DONE.
	TYPE	,MSG13	;ALL TIMING TESTS BYPASSED

.SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

*TEST 1 REFERENCE ALL CONTROLLER REGISTERS

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

TST1: SCOPE

```
MOV #1,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
BR TST2 ;GO TO NEXT TEST
```

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

*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. IF SET, THE PROGRAM WILL BYPASS

3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071 011042 000004
3072 011044 012737 000001 001174
3073 011052 012706 001100
3074
3075 011056 012746 000000
3076 011062 012746 011070
3077 011066 000002
3078 011070
3079
3080 011070 012737 011206 000004
3081 011076 013705 001264
3082 011102 005765 000000
3083 011106 005765 000010
3084 011112 005765 000002
3085 011116 005765 000004
3086 011122 005765 000006
3087 011126 005765 000012
3088 011132 005765 000014
3089 011136 005765 000016
3090 011142 005765 000020
3091 011146 005765 000024
3092 011152 005765 000026
3093 011156 005765 000034
3094 011162 005765 000036
3095 011166 005765 000030
3096 011172 005765 000032
3097
3098 011176 012737 030740 000004
3099 011204 000404
3100
3101 011206 022626
3102 011210 104007
3103 011212 000137 023742
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114

```

3115
3116
3117
3118
3119
3120 011216 000004
3121 011220 012737 000001 001174
3122 011226 012706 001100
3123
3124 011232 005237 001550
3125
3126 011236 132737 000200 001231
3127 011244 001002
3128 011246 000137 011362
3129
3130 011252 104401 037062
3131 011256 007037 005526
3132 011262 005000
3133 011264 012701 005530
3134 011270 013702 001266
3135
3136 011274 032702 000001
3137 011300 001410
3138 011302 005237 005526
3139 011306 005211
3140 011310 104401 001205
3141 011314 010046
3142
3143 011316 104403
3144 011320 001
3145 011321 000
3146
3147 011322 005721
3148 011324 005200
3149 011326 022700 000010
3150 011332 001402
3151
3152 011334 006002
3153 011336 000756
3154
3155 011340 005737 005526
3156 011344 001402
3157 011346 000137 012302
3158
3159 011352 104034
3160 011354 000000
3161 011356 000137 010656
3162
3163 011362 012765 000040 000010
3164
3165
3166
3167
3168
3169 011370 013737 001426 005444
3170 011376 004737 024422

```

```

*****
TST2: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
INC BYPCERR ;DO NOT TEST CERR IN FRDY
BITB #BIT7,$ENVM ;SEE IF USE APT SELECTED DRIVES
BNE 14$ ;BR IF YES
JMP 12$ ;ELSE DO NORM SIZING OR VERIFY
14$: TYPE MSG10 ;WILL TEST DRIVES
CLR DRIVS ;# OF DRIVES PRESENT
CLR RD ;DRV ADDR
MOV #DRIV0,R1 ;DRV FLAG
MOV $DEV0,R2 ;APT DEVICE MAP
15$: BIT #BIT0,R2 ;SEE IF DRV IN DEVICE MAP
BEQ 16$ ;BR IF NO
INC DRIVS ;ELSE INCR DRIVE COUNT
INC (R1) ;& SET DRIVE PRESENT FLAG
TYPE $CRLF
MOV RD,-(SP) ;SAVE RD FOR TYPEOUT
;TYPE DRIVE #
;GO TYPE--OCTAL ASCII
;TYPE 1 DIGIT(S)
;SUPPRESS LEADING ZEROS
16$: TST (R1)+ ;ADV POINTER TO NEXT FLAG
INC RD ;INC DRIVE #
CMP #8,RD ;ALL 8 TESTED?
BEQ 17$ ;BR IF YES
ROR R2 ;ELSE GET NEXT BIT OFF DEVICE MAP
BR 15$ ;& TRY AGAIN
17$: TST DRIVS ;SEE IF MORE DRIVES PRESENT
BEQ 18$ ;BR IF NO
JMP NUORV ;ELSE EXIT TEST
18$: ERROR 34 ;NO DRIVES FOUND IN $DEV0
HALT ;SETUP CORRECTLY & PRESS 'CONTINUE'
JMP ST5 ;TO TRY AGAIN
12$: MOV #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR

```

K05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
 DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 61
 T2 SIZE THE BUSS

SEQ 0061

```

3171 011402 104120          ERROR 120          ;RDY NOT SET BY END OF SCLR
3172
3173
3174 011404 005737 005556    TST     SIZELG
3175 011410 001562          BEQ     TST3          ;: DO NOT SIZE, GOTO NEXT TEST
3176 011412 104401 037062    TYPE   ,MSG10       ;: WILL TEST DRIVES
3177 011416 005037 005526    CLR    DRVS         ;: # OF DRIVES PRESENT
3178 011422 005000          CLR    RD           ;: DRV ADDR
3179 011424 012701 005530    MOV    #DRIVO,R1   ;: DRV FLAG
3180 011430
3181 011430 104415          1$:     SCOP1
3182 011432 012706 001100    MOV    #STACK,SP   ;: RESTORE STK PTR
3183
3184 011436 012765 000040 000010  MOV    #SCLR,RKCS2(R5) ;: SUBSYSTEM CLEAR
3185 011444 013737 001426 005444  MOV    T10,TEMP1   ;: SET TIMEOUT
3186 011452 004737 024422    JSR    PC,FRDY     ;: FIND RDY
3187 011456 104120          ERROR 120          ;: RDY NOT SET BY END OF SCLR
3188 011460 010065 000010  MOV    RD,RKCS2(R5) ;: SELECT THE DRIVE ADDR
3189 011464 012765 000001 000000  MOV    #SELDRV,RKCS1(R5) ;: SELECT DRIVE CMD
3190 011472 013737 001426 005444  MOV    T10,TEMP1
3191 011500 004737 024422    JSR    PC,FRDY     ;: FIND RDY
3192 011504 104117          ERROR 117          ;: NO RDY AFTER SELECT DRIVE CMD.
3193 011506 032737 100000 005406  BIT    #CERR,HCS1
3194 011514 001046          BNE    2$
3195 011516 013737 005434 005444  MOV    HMR2,TEMP1
3196 011524 042737 177770 005444  BIC    #1C<DRVMSK>,TEMP1
3197 011532 020037 005444    CMP    RD,TEMP1   ;: S/B SAME
3198 011536 001016          BNE    3$
3199 011540 005700          TST    RD
3200 011542 001003          BNE    4$
3201 011544 005737 005520    TST    DDPCH       ;: SEE IF XXDP CHAIN MODE
3202 011550 001014          BNE    5$
3203 011552 005237 005526          4$:     INC    DRVS         ;: INC DRIVE COUNT
3204 011556 005211          INC    (R1)        ;: SET DRIVE PRESENT FLAG
3205 011560 104401 001205    INC    (R1)
3206 011564 010046          TYPE   ,SCLF
3207                                MOV    RD,-(SP)     ;: SAVE RD FOR TYPEOUT
3208                                ;: TYPE DR #
3209 011566 104403          TYPOS
3210 011570 001          .BYTE 1          ;: GO TYPE--OCTAL ASCII
3211 011571 000          .BYTE 0          ;: TYPE 1 DIGIT(S)
3212 011572 000403          BR    5$          ;: SUPPRESS LEADING ZEROS
3213 011574 004737 025130          3$:     JSR    PC,BYP     ;: TYPE BYPASS DR #
3214 011600 104001          ERROR 1          ;: WRITTEN DR # DOES NOT MATCH RKMR2 DR #
3215
3216 011602 005721          5$:     TST    (R1)+      ;: SHIFT PTR TO NEXT DR. FLAG
3217 011604 005200          INC    RD          ;: INC DR #
3218 011606 022700 000010  CMP    #8.,RD     ;:
3219 011612 001306          BNE    1$          ;: MORE LEFT.
3220 011614 005737 005526    TST    DRVS
3221 011620 001054          BNE    10$
3222 011622 104061          ERROR 61          ;: NO DRIVES SEEN
3223 011624 000000          HALT
3224 011626 000137 010656    JMP    ST5        ;: SETUP & PRESS 'CONTINUE'
3225
3226 011632 032737 001000 005410  2$:     BIT    #MDS,HCS2

```

```

3227 011640 001015      BNE      6$
3228 011642 032737 000400 005410  BIT      #UFE,HCS2
3229 011650 001015      BNE      7$
3230 011652 032737 000001 005420  BIT      #DRA,MDS
3231 011660 001015      BNE      8$
3232 011662 032737 010000 005410  BIT      #NED,HCS2
3233 011670 001424      BEQ      9$
3234 011672 000743      BR       5$
3235
3236 011674 004737 025130      6$:   JSR      PC,BYP      ;TYPE BYP DR #
3237 011700 104002      ERROR   2              ;MDS DETECTED
3238 011702 000737      BR       5$
3239
3240 011704 004737 025130      7$:   JSR      PC,BYP
3241 011710 104003      ERROR   3              ;UFE DETECTED
3242 011712 000733      BR       5$
3243
3244 011714 032737 010000 005410  8$:   BIT      #NED,HCS2
3245 011722 001713      BEQ      4$
3246 011724 104401 037267      TYPE    MSG15          ;DRV#
3247 011730 010046      MOV     RO,-(SP)       ;SAVE RO FOR TYPEOUT
3248
3249 011732 104403      TYPOS
3250 011734      .BYTE  1              ;TYPE DR#
3251 011735      .BYTE  0              ;GO TYPE--OCTAL ASCII
3252 011736 104010      ERROR   10             ;TYPE 1 DIGIT(S)
3253 011740 000720      BR       5$           ;SUPPRESS LEADING ZEROS
3254
3255 011742 004737 025130      9$:   JSR      PC,BYP
3256 011746 104004      ERROR   4              ;NO DRA & NO NED = OTHER PORT SELECTED
3257 011750 000714      BR       5$
3258 011752 000137 012302      10$:  JMP      NUDRV
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275 011756 000004      *TEST 3  VERIFY OPERATOR DRIVE SELECTIONS
3276 011760 012737 000001 001174      *
3277 011766 012706 001100      *
3278 011772 005000      *
3279 011774 012701 005530      *
3280 012000      *
3281 012000 104415      *
3282 012002 012706 001100      *

```

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. IT WILL THEN CHECK
TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING. IF NOT, IT WILL
BE AN ERROR. 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 INFOR TO
VERIFY IT WAS NOT SPECIFIED.

```

15:  SCOPE
      MOV     #1,STIMES      ;DO 1 ITERATION
      MOV     #STACK,SP     ;RESTORE STK PTR
      CLR     RO             ;DRIVE ADDR
      MOV     #DRIVO,R1     ;DRIVE FLAG

```


M05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 63
T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 0063

3283									
3284	012006	012765	000040	000010		MOV	#SCLR,RKCS2(R5)	;SUBSYSTEM CLEAR	
3285	012014	013737	001426	005444		MOV	T10,TEMP1	;SET TIME OUT	
3286	012022	004737	024422			JSR	PC,FRDY	;FIND RDY	
3287	012026	104120				ERROR	120	;NO RDY AFTER SCLR	
3288	012030	010065	000010			MOV	RO,RKCS2(R5)	;DRV ADDR	
3289	012034	012765	000001	000000		MOV	#SELDRV,RKCS1(R5)	;SELECT DRIVE CMD	
3290	012042	013737	001426	005444		MOV	T10,TEMP1		
3291	012050	004737	024422			JSR	PC,FRDY	;FIND RDY	
3292	012054	104117				ERROR	117	;NO RDY AFTER SELECT DRIVE CMD.	
3293	012056	032737	100000	005406		BIT	#CERR,HCS1		
3294	012064	001036				BNE	2\$		
3295	012066	013737	005434	005444		MOV	HMR2,TEMP1		
3296	012074	042737	177770	005444		BIC	#C<DRVMSK>,TEMP1		
3297	012102	020037	005444			CMP	RO,TEMP1	;S/B SAME	
3298	012106	001010				BNE	3\$		
3299	012110	005711			11\$:	TST	(R1)		
3300	012112	001417				BEQ	5\$		
3301	012114	005721			4\$:	TST	(R1)+	;SHIFT PTR TO NEXT DR FLAG	
3302	012116	005200				INC	RO	;INC DR#	
3303	012120	022700	000010			CMP	#8.,RO		
3304	012124	001325				BNE	1\$;MORE LEFT	
3305	012126	000467				BR	TST4	;GO TO NEXT TEST	
3306									
3307	012130	004737	025130		3\$:	JSR	PC,BYP	;TRY BYPASS DRIVE#	
3308	012134	104001				ERROR	1	;WRITTEN DR# DOES NOT MATCH RKMR2 DR#	
3309	012136	005711				TST	(R1)		
3310	012140	001765				BEQ	4\$;BRANCH IF NOT SPEC BY INPUT	
3311	012142	005337	005526		12\$:	DEC	DRIVS	;DECREMENT TOTAL DRIVS	
3312	012146	005011				CLR	(R1)	;CLEAR DRIVE FLAG	
3313	012150	000761				BR	4\$		
3314									
3315	012152	004737	025130		5\$:	JSR	PC,BYP		
3316	012156	104005				ERROR	5	;DR PRESENT BUT NOT SPECIFIED BY OPERATOR	
3317	012160	000755				BR	4\$		
3318									
3319	012162	032737	001000	005410	2\$:	BIT	#MDS,HCS2		
3320	012170	001027				BNE	6\$		
3321	012172	032737	000400	005410		BIT	#UFE,HCS2		
3322	012200	001027				BNE	7\$		
3323	012202	032737	000001	005420		BIT	#DRA,HDS		
3324	012210	001005				BNE	8\$		
3325	012212	032737	010000	005410		BIT	#NED,HCS2		
3326	012220	001423				BEQ	9\$		
3327	012222	000404				BR	10\$		
3328	012224	032737	010000	005410	8\$:	BIT	#NED,HCS2		
3329	012232	001726				BEQ	11\$		
3330	012234	005711			10\$:	TST	(R1)		
3331	012236	001726				BEQ	4\$		
3332									
3333	012240	004737	025130			JSR	PC,BYP	;TYPE BYPASS DRIVE#	
3334	012244	104006				ERROR	6		
3335	012246	000735				BR	12\$		
3336									
3337	012250	004737	025130		6\$:	JSR	PC,BYP	;TYPE BYPASS DRIVE#	
3338	012254	104002				ERROR	2	;MDS DETECTED	

N05

UNIBUS RKG DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28 JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 64
T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 0064

```

3339 012256 000762          BR      8$
3340
3341 012260 004737 025130  7$:   JSR      PC,BYP
3342 012264 104003          ERROR   3           ;UFE DETECTED
3343 012266 000756          BR      8$
3344
3345 012270 004737 025130  9$:   JSR      PC,BYP
3346 012274 104004          ERROR   4           ;DRA & NED RESET - OTHER PORT SELECTED
3347 012276 000752          BR      8$
3348
3349 012300 001237          BNE     1$           ;BRANCH IF MORE LEFT.
3350
3351
3352
3353           ; THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
3354           ; DRIVE PRESENT
3355           ; 'SUNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
3356           ; UNDER TEST
3357
3358
3359 012302 005037 001550  NUDRV: CLR      BYPCERR           ;ENTER HERE FROM LAST TEST
3360                                     ;TEST CERR IN 'FRDY'
3361
3362           ;*****
3363           ;TEST 4          FIND NEXT DRIVE TO BE TESTED
3364           ;
3365           ; THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
3366           ; ADDRESS IN 'SUNIT'.
3367           ; THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
3368           ; THE DRIVE WHOSE ADDRESS IS IN 'SUNIT'.
3369           ;*****
3370
3371 012306 000004          †ST4: SCOPE
3372 012310 012737 000001 001174  MOV     #1,$TIMES           ;:DO 1 ITERATION
3373 012316 012706 001100          MOV     #STACK,$SP         ;:RESTORE STK PTR
3374 012322 012737 000004 001214  MOV     #STN-1,$TESTN
3375 012330 012737 000004 001102  MOV     #STN-1,$STNM
3376
3377 012336 005737 005526          TST     DRVS              ;:ANY DRIVES PRESENT?
3378 012342 001004          BNE     4$              ;:YES BRANCH
3379 012344 104401 037647          TYPE   ,MSG27           ;:ALL DRIVES TESTED
3380 012350 000137 023742          JMP     $EOP            ;:NO, GO TO END
3381
3382 012354 013701 001362          4$:   MOV     DRVPTR,R1     ;:ADDR OF NEXT DRIVE FLAG
3383 012360 005737 001220          TST     $DEVCT          ;:IS FIRST DRIVE BEING CHECKED
3384 012364 001402          BEQ     2$              ;:YES BRANCH
3385 012366 005237 001222          1$:   INC     $SUNIT      ;:INCR DRIVE ADDR TO NEXT DRIVE
3386 012372 005721          2$:   TST     (R1)+        ;:IS DRIVE PRESENT?
3387 012374 001774          BEQ     1$              ;:NO, FIND NEXT DRIVE PRESENT
3388 012376 005737 005520          TST     DDPCH           ;:DDP CHAIN MODE?
3389 012402 001403          BEQ     3$              ;:NO BRANCH
3390 012404 005737 001222          TST     $SUNIT          ;:YES, IS IT DRIVE 0?
3391 012410 001766          BEQ     1$              ;:IF YES, DON'T TEST DR 0
3392 012412 010137 001362          3$:   MOV     R1,DRVPT     ;:STORE POINTER TO THE NEXT DR. FLAG
3393 012416 104401 037267          TYPE   ,MSG15          ;:DRIVE
3394 012422 013700 001222          MOV     $SUNIT,R0

```

3395 012426 010046
3396
3397 012430 104403
3398 012432 001
3399 012433 000
3400
3401 012434 104401 001205
3402
3403 012440
3404
3405
3406
3407
3408
3409
3410
3411 012440 000004
3412 012442 012737 000001 001174
3413 012450 012706 001100
3414
3415 012454 005737 001216
3416 012460 001046
3417 012462 004737 026332
3418 012466 104024
3419
3420 012470 104401 037301
3421 012474 012765 000003 000026
3422 012502 004737 025760
3423 012506 013701 005434
3424 012512 012704 034764
3425 012516 010446
3426 012520 012703 000003
3427 012524 006101
3428 012526 006101
3429 012530 006101 15:
3430 012532 006101
3431 012534 006101
3432 012536 006101
3433 012540 010100
3434 012542 042700 177760
3435 012546 052700 000060
3436 012552 110024
3437 012554 005303
3438 012556 001364
3439 012560 105014
3440
3441 012562 004737 035232
3442 012566 104401 001205
3443 012572 104401 001205
3444
3445
3446
3447
3448
3449
3450

```

MOV      RD,-(SP)          ;;SAVE RD FOR TYPEOUT
                        ;;DRIVE #
TYPOS    GO TYPE--OCTAL ASCII
.BYTE    1                  ;TYPE 1 DIGIT(S)
.BYTE    0                  ;SUPPRESS LEADING ZEROS

TYPE     ,SCLF

PFSRT:    ;ENTER HERE FOR POWER FAIL RESTART
;*****
;TEST 5    PRINT DRIVE SERIAL NUMBER
;
; THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
; IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
;*****
TSTS:    SCOPE
MOV      #1,STIMES          ;;DO 1 ITERATION
MOV      #STACK,SP         ;;RESTORE STK PTR

TST      $PASS
BNE      TST6              ;;GO TO NEXT IF NOT FIRST PASS
JSR      PC,SUBCLR         ;;DO SUBSYS CLEAR
ERROR    24                ;CERR AFTER SCLR

TYPE     MSG16             ;DRIVE SERIAL NO.
MOV      #3,RKMR:(R5)     ;SELECT BYTE 3
JSR      PC,GSTAT         ;GET STATUS
MOV      #R2,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
ROL      R1
ROL      R1
MOV      R1,RO            ;GET WORKING COPY
BIC      #177760,RO       ;CLEAR ALL BUT LOW 4 BITS
BIS      #60,RO           ;CONVERT TO ASCII DIGIT
MOV      R0,(R4)+         ;PUT ASCII DIGIT INTO CHAR BUFF
DEC      R3
BNE      15              ;BR IF ALL 3 CHARS NOT DONE
CLR      (R4)            ;ELSE INSERT NULL TERMINATOR

JSR      PC,$SUPRS        ;TYPE
TYPE     ,SCLF
TYPE     ,SCLF

;*****
;TEST 6    SET VV WITH PACK COMMAND
;
; IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
;*****

```

```

3451 012576 000004 TST6: SCOPE
3452 012600 012737 000001 001174 MOV #1,STIMES ;:DO 1 ITERATION
3453 012606 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
3454
3455 012612 004737 026332 JSR PC,SUBCLR
3456 012616 104024 ERROR 24 ;CERR AFTER SCLR
3457
3458 012620 032737 000100 005434 BIT #D.VV,HMR2
3459 012626 001024 BNE TST7 ;;GO TO NEXT TEST IF VV SET
3460
3461 012630 104415 SCOPI
3462 012632 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
3463
3464 012636 004737 026332 JSR PC,SUBCLR
3465 012642 104024 ERROR 24 ;CERR AFTER SCLR
3466
3467 012644 012765 000003 000000 MOV #PACK,RKCS1(R5) ;CMD TO SET VV
3468 012652 012737 000010 005444 MOV #10,TEMP1
3469 012660 004737 024422 JSR PC,FRDY ;FIND RDY
3470 012664 104116 ERROR 116 ;RDY NOT SET AFTER PACK CMD
3471
3472 012666 032737 000100 005434 BIT #D.VV,HMR2
3473 012674 001001 BNE TST7 ;:GO TO NEXT TEST IF VV NOW SET
3474 012676 104027 ERROR 27 ;PACK DID NOT SET VV
3475

```

```

*****
:TEST 7 UNLOAD DRIVE TO BE TESTED
:
: THIS TEST UNLOADS THE DRIVE TO BE TESTED NEXT,
: WAITS FOR ATTN & VERIFIES IT CAME FROM THE CORRECT DRIVE.
:
*****

```

```

3483 012700 000004 TST7: SCOPE
3484 012702 012737 000001 001174 MOV #1,STIMES ;:DO 1 ITERATION
3485 012710 012706 001100 MOV #STACK,SP
3486
3487 012714 005237 005370 INC UNLD ;USED TO CHECK FOR VALID HALT
3488
3489 012720 004737 026332 JSR PC,SUBCLR
3490 012724 104024 ERROR 24 ;CERR AFTER SCLR
3491
3492 012726 012765 000007 000000 MOV #UNLOAD,RKCS1(R5) ;UNLOAD CMD
3493 012734 013737 001426 005444 MOV #10,TEMP1
3494 012742 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
3495 012746 104017 ERROR 17 ;NO RDY AFTER UNLD CMD
3496 012750 004737 024704 JSR PC,TSTATN
3497 012754 104020 ERROR 20 ;NO ATTN AFTER UNLOAD CMD
3498
3499

```

.SBTTL OPERATOR INTERVENTION TESTS

```

3500
3501
3502 *****
3503 :TEST 10 INTERLOCKS TESTS
3504 :
3505 : THIS TEST VERIFIES THAT THE DOOR & CARTRIDGE STATUS BITS
3506 : ARE OPERATING PROPERLY IN MESSAGE AD & THAT THE REMOVAL

```

```

3507
3508
3509
3510
3511
3512
3513
3514
3515
3516 012756 000004
3517 012760 012737 000001 001174
3518 012766 012706 001100
3519
3520 012772 004737 026332
3521 012776 104024
3522
3523 013000 005237 001550
3524
3525 013004 104401 037004
3526 013010 104401 041656
3527 013014 004737 030306
3528 013020 000137 013634
3529
3530 013024 104401 041350
3531 013030 104401 040502
3532 013034 004737 030346
3533
3534 013040 104401 043323
3535 013044 104401 040005
3536 013050 104401 040502
3537 013054 004737 030346
3538 013060 012765 000001 000026
3539 013066 004737 025760
3540 013072 032737 000200 005434
3541 013100 001401
3542 013102 104044
3543
3544 013104 012737 000140 005476 1S:
3545 013112 005037 005500
3546 013116 012737 000540 005502
3547 013124 012737 000001 005504
3548
3549 013132 004737 025144
3550 013136 000000
3551 013140 104045
3552 013142 104046
3553 013144 104063
3554 013146 104064
3555
3556 013150 004737 026332
3557 013154 104024
3558
3559 013156 104401 040135
3560 013162 104401 040222
3561 013166 104401 040502
3562 013172 004737 030346

```

```

*****
TST10: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESET STACK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
INC BYPCERR ;DONT DO CKCERR ROUTINE
TYPE ,MSG8 ;INTERLOCKS TEST
TYPE ,MSG52 ;CONT-E TO EXIT OR SPACE TO CONTINUE
JSR PC,CCSP ;INPUT CONT-E OR SPACE
JMP BS ;RET HERE FOR CONT-E
TYPE ,MSG47 ;VERIFY DOOR CANNOT BE OPENED
TYPE ,MSG37 ;DEPRESS SPACE WHEN DONE
JSR PC,GETSP ;GET SPACE
TYPE ,MSG65 ;DEPRESS RUN/STOP SW TO 'STOP'
TYPE ,MSG30 ;OPEN DOOR & LEAVE IT OPEN
TYPE ,MSG37 ;DEPRESS SPACE BAR WHEN DONE
JSR PC,GETSP ;INPUT SPACE
MOV #1,RKMR1(RS) ;SELECT WORD 1
JSR PC,GSTAT
BIT #D.DOOR,HMR2
BEQ 1S
ERROR 44 ;DOOR STATUS BIT NOT CLEARED
1S: MOV #<D.DRA!D.VV>,E.A0 ;EXPECTED A0
CLR E.B0
MOV #<D.HDMM!D.BRHM!D.CART>,E.A1
MOV #1,E.B1
JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
; & MSGS SPECIFIED HERE
; MSG A0 ERROR AFTER DRIVE UNLOADED & DOOR OPENED
; MSG B0 ERROR
; MSG A1 ERROR
; MSG B1 ERROR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
TYPE ,MSG33 ;PRESS 'RUN-STOP' TO 'RUN'
TYPE ,MSG34 ;VERIFY DOES NOT START
TYPE ,MSG37
JSR PC,GETSP ;GET SPACE FROM TTY

```

E06

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 68
T10 INTERLOCKS TESTS

SEQ 0068

3563								
3564	013176	004737	025760			JSR	PC,GSTAT	
3565	013202	032737	010000	005434		BIT	#D.SPIN,HMR2	
3566	013210	001401				BEQ	2\$	
3567	013212	104065				ERROR	65	;SPIN SET IN MSGAO
3568								
3569	013214	012737	000140	005476	2\$:	MOV	#(D.DRA!D.VV),E.A0	;EXPECTED MSG AO
3570	013222	005037	005500			CLR	E.B0	
3571	013226	012737	000540	005502		MOV	#(D.HOHM!D.BRHM!D.CART),E.A1	
3572	013234	012737	000001	005504		MOV	#1,E.B1	
3573								
3574	013242	004737	025144			JSR	PC,CHKMSG	;CHECK MSGS AO, BO, A1, B1
3575	013246	000000				.WORD	0!0!0	; & MSGS SPECIFIED HERE
3576	013250	104066				ERROR	66	;MSG AO ERROR AFTER ATTEMPT TO START SPIN & DOOR OPEN
3577	013252	104067				ERROR	67	;MSG BO ERROR
3578	013254	104070				ERROR	70	;MSG A1 ERROR
3579	013256	104071				ERROR	71	;MSG B1 ERROR
3580								
3581	013260	004737	026332			JSR	PC,SUBCLR	
3582	013264	104024				ERROR	24	;CERR AFTER SCLR
3583								
3584	013266	104401	043323			TYPE	,MSG65	;DEPRESS 'RUN-STOP' SW TO STOP
3585	013272	104401	040075			TYPE	,MSG32	;REMOVE PACK & CLOSE DOOR
3586	013276	104401	040502			TYPE	,MSG37	
3587	013302	004737	030346			JSR	PC,GETSP	;GET SPACE FROM TTY
3588								
3589	013306	012765	000001	000026		MOV	#1,RKMR1(R5)	;SELECT WORD 1
3590	013314	004737	025760			JSR	PC,GSTAT	
3591	013320	032737	000400	005434		BIT	#D.CART,HMR2	
3592	013326	001401				BEQ	3\$	
3593	013330	104072				ERROR	72	;CARTRIDGE STATUS BIT NOT RESET
3594								
3595								
3596	013332	004737	026332		3\$:	JSR	PC,SUBCLR	
3597	013336	104024				ERROR	24	;CERR AFTER SCLR
3598								
3599	013340	104401	040310			TYPE	,MSG35	;PRESS 'RUN-STOP' TO 'RUN'
3600	013344	104401	040222			TYPE	,MSG34	;VERIFY DOES NOT START
3601	013350	104401	040502			TYPE	,MSG37	
3602	013354	004737	030346			JSR	PC,GETSP	;GET SPACE FROM TTY
3603								
3604	013360	004737	025760			JSR	PC,GSTAT	
3605	013364	032737	010000	005434		BIT	#D.SPIN,HMR2	
3606	013372	001401				BEQ	5\$	
3607	013374	104100				ERROR	100	;SPIN SET IN MSG AO
3608								
3609	013376	104401	043323		5\$:	TYPE	,MSG65	;PRESS 'RUN-STOP' TO 'STOP'
3610	013402	104401	040405			TYPE	,MSG36	;INSERT CARTRIDGE, SET 'RUN' SW. & CLOSE DOOR
3611	013406	104401	043563			TYPE	,MSG70	;VERIFY HEADS LOAD
3612	013412	104401	043506			TYPE	,MSG69	;DEPRESS SPACE WHEN READY GOES ON
3613	013416	004737	030346			JSR	PC,GETSP	;GET SPACE FROM TTY
3614	013422	005037	005370			CLR	UNLD	;FOR VALID HALT
3615								
3616	013426	012765	100000	000000		MOV	#CCLR,RKCS1(R5)	
3617	013434	005065	000026			CLR	RKMR1(R5)	;SELECT WORD 0
3618	013440	004737	025760			JSR	PC,GSTAT	

```

3619 013444 032737 000100 005434 BIT #D.VV,HMR2
3620 013452 001401 BEQ 75
3621 013454 104033 ERROR 33 ;VV NOT CLEARED AFTER PACK RE-INSERTED
3622
3623 013456 012737 050240 005476 75: MOV #<D.DSC!D.SPIN!D.DRDY!D.DRA>,E.A0 ;EXPECTED A0
3624 013464 005037 005500 CLR E.B0
3625 013470 012737 001720 005502 MOV #<D.SPOK!D.CART!D.SSP!D.BRHM!D.DOOR>,E.A1
3626 013476 012737 000001 005504 MOV #1,E.B1
3627
3628 013504 004737 025144 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
3629 013510 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
3630 013512 104074 ERROR 74 ;MSG A0 ERROR AFTER PACK REINSERTED & HEADS LOADED
3631 013514 104075 ERROR 75 ;MSG B0 ERROR
3632 013516 104076 ERROR 76 ;MSG A1 ERROR
3633 013520 104077 ERROR 77 ;MSG B1 ERROR
3634
3635 013522 012765 100000 000000 MOV #CLR,RKCS1(R5)
3636 013530 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE #
3637 013536 012765 000003 000000 MOV #PACK,RKCS1(R5) ;PACK CMD
3638 013544 013737 001426 005444 MOV T10,TEMP1
3639 013552 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
3640 013556 104116 ERROR 116 ;CONTR NOT RDY
3641
3642 013560 032737 000100 005434 BIT #D.VV,HMR2
3643 013566 001001 BNE 645
3644 013570 104027 ERROR 27 ;VOLUME VALID NOT SET AFTER PACK CMD
3645 013572 645:
3646 013572 005237 005370 INC UNLD ;FOR VALID HALT
3647
3648 013576 004737 026332 JSR PC,SUBCLR
3649 013602 104024 ERROR 24 ;CERR AFTER SCLR
3650
3651 013604 012765 000007 000000 MOV #UNLOAD,RKCS1(R5) ;UNLOAD CMD
3652 013612 013737 001426 005444 MOV T10,TEMP1
3653 013620 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
3654 013624 104017 ERROR 17 ;NO RDY AFTER UNLD CMD
3655 013626 004737 024704 JSR PC,TSTATN
3656 013632 104020 ERROR 20 ;NO ATTN AFTER UNLOAD CMD
3657 013634 85:
3658
3659
3660 :*****
3661 :*TEST 11 UNIT SELECT PLUG TEST
3662 :*
3663 :* THIS TEST VERIFIES THAT WHEN THE UNIT SELECT PLUG IS PULLED
3664 :* OUT, THE QUAL LOGIC RESETS ATTN & VOLUME VALID, THAT
3665 :* THE DRIVE DE-SELECTS & NON EXISTENT DRIVE ASSERTS.
3666 :* FURTHER, THE OPERATOR IS ASKED TO INSERT ANY NUMBER OF
3667 :* UNIT SELECT PLUGS. THE PROGRAM WILL RESPOND BY TYPING
3668 :* THE PLUG CODE NUMBER AS SOON AS IT IS INSERTED.
3669 :* THIS PORTION OF THE TEST IS TERMINATED AT ANY TIME BY A CONT-E
3670 :*
3671 :*****
3671 013634 000004 ST11: SCOPE
3672 013636 012737 000001 001174 MOV #1,STIMES ;DO 1 ITERATION
3673 013644 012706 001100 MOV #STACK,SP ;RESTORE STACK PTR
3674

```

```

3675 013650 004737 026332      JSR    PC,SUBCLR
3676 013654 104024              ERROR  24      ;CERR AFTER SCLR
3677
3678 013656 104401 037355      TYPE   ,MSG18      ;UNIT SELECT PLUG TEST
3679 013662 104401 041656      TYPE   ,MSG52      ;CONT-E TO EXIT OR SPACE TO CONTINUE
3680 013666 004737 030306      JSR    PC,CCSP     ;INPUT CONT-E OR SPACE
3681 013672 000137 014504      JMP    12$        ;RETURN HERE FOR CONT-E
3682
3683 013676 012765 000020 000026      MOV    #PAT,RKMR1(R5) ;EVEN PARITY TO GET DSC & ATTN
3684 013704 004737 025760      JSR    PC,GSTAT
3685 013710 032737 001000 005436      BIT    #D.PAR,HMR3 ;SEE IF PARITY SET
3686 013716 001001              BNE    2$         ;BR IF YES
3687 013720 104114              ERROR  114       ;PARITY BIT NOT SET AFTER SEL DRIVE CMD
3688
3689 013722 032737 040000 005434 2$:      BIT    #D.DSC,HMR2 ;SEE IF DSC SET
3690 013730 001001              BNE    1$         ;
3691 013732 104107              ERROR  107       ;DSC NOT SET AFTER SEL DRV WITH EVEN PARITY
3692
3693 013734 012765 100000 000000 1$:      MOV    #CCLR,RKCS1(R5)
3694 013742 004737 025760      JSR    PC,GSTAT
3695 013746 004737 024704      JSR    PC,TSTATN
3696 013752 104113              ERROR  113       ;NO ATTN AFTER SEL DRV WITH EVEN PARITY
3697
3698 013754 104401 037030      TYPE   ,MSG9       ;REMOVE UNIT SELECT PLUG
3699 013760 104401 040502      TYPE   ,MSG37      ;DEPRESS SPACE WHEN DONE
3700 013764 004737 030346      JSR    PC,GETSP    ;GET SPACE
3701
3702 013770 012765 100000 000000      MOV    #CCLR,RKCS1(R5)
3703 013776 004737 025760      JSR    PC,GSTAT
3704 014002 004737 024704      JSR    PC,TSTATN  ;RETURN HERE FOR SPACE
3705 014006 000401              BR     3$         ;NO ATTN
3706 014010 104111              ERROR  111       ;REMOVING UNIT SEL PLUG, DID NOT
3707
3708 014012 012765 100000 000000 3$:      MOV    #CCLR,RKCS1(R5)
3709 014020 004737 025760      JSR    PC,GSTAT
3710 014024 032765 010000 000010      BIT    #NED,RKCS2(R5)
3711 014032 001001              BNE    4$         ;
3712 014034 104112              ERROR  112       ;REMOVING UNIT SEL PLUG DID NOT
3713
3714
3715 014036 104401 043371      TYPE   ,MSG67      ;DESELECT ALL OTHER PORTS
3716 014042 104401 040502      TYPE   ,MSG37      ;TYPE SPACE WHEN DONE
3717 014046 004737 030346      JSR    PC,GETSP    ;GET SPACE
3718
3719 014052 104401 040546      TYPE   ,MSG38      ;INSERT UNIT SELECT PLUGS
3720 014056 104401 040704      TYPE   ,MSG39      ;DEPRESS CONTROL-E WHEN FINISHED
3721 014062 104401 042137      TYPE   ,MSG55      ;EXIT WITH CORRECT UNIT SELECT PLUG #
3722 014066 013746 001222      MOV    $UNIT,-(SP) ;SAVE $UNIT FOR TYPEOUT
3723
3724 014072 104403              TYPOS
3725 014074 001              .BYTE 1          ;GO TYPE--OCTAL ASCII
3726 014075 000              .BYTE 0          ;TYPE 1 DIGIT(S)
3727 014076 104401 001205      TYPE   ,$SCRLF    ;SUPPRESS LEADING ZEROS
3728
3729 014102 105037 033260      CLPB  $TKQSRT     ;CLEAR PREVIOUS INFO
3730 014106 005037 001160      CLR   $TMP0

```



```

3731
3732 014112 113737 033260 001160 5$: MOVB $TKQSRT,$TMP0 ;GET CHAR IF THERE
3733 014120 042737 177600 001160 BIC #C(177),$TMP0 ;GET RID OF JUNK, IF ANY
3734 014126 005737 001160 TST $TMP0
3735 014132 001422 BEQ 6$ ;BR IF NOTHING TYPED YET
3736 014134 023727 001160 000005 CMP $TMP0,#5 ;SEE IF CONT-E
3737 014142 001476 BEQ 9$ ;BR IF YES
3738 014144 023727 001160 000003 CMP $TMP0,#3 ;SEE IF CONT-C
3739 014152 001004 BNE 13$ ;BR IF NO
3740 014154 004737 033262 JSR PC,$TKINT ;ENABLE KB INT
3741 014160 000137 030376 JMP STOI ;ELSE DO VALID HALT
3742 014164 104401 040067 13$: TYPE MSG31 ;?
3743 014170 105037 033260 CLRB $TKQSRT
3744 014174 004737 033262 JSR PC,$TKINT
3745
3746 014200 005000 6$: CLR RO
3747 014202 012765 100000 000000 7$: MOV #CCLR,RKCS1(R5)
3748 014210 010065 000010 MOV RO,RKCS2(R5) ;DRIVE NO.
3749 014214 012765 000001 000000 MOV #SELDV,RKCS1(R5) ;SELECT DRIVE CMD
3750 014222 013737 001426 005444 MOV T10,TEMP1
3751 014230 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
3752 014234 104117 ERROR 117 ;CONTR RDY NOT SET
3753
3754 014236 032765 010000 000010 BIT #NED,RKCS2(R5) ;SEE IF UNIT SELECT PLUG INSERTED
3755 014244 001405 BEQ 8$ ;& IF MATCH DRIVE NO IN CS2. BR IF YES
3756 014246 005200 INC RO
3757 014250 020027 000010 CMP RO,#8. ;ALL 8 DRIVE NOS TRIED?
3758 014254 001352 BNE 7$ ;BR IF NO
3759 014256 000715 BR 5$ ;ELSE RETURN
3760
3761 014260 032737 000100 005434 8$: BIT #D.VV,HMR2
3762 014266 001311 BNE 5$ ;TYPE SAME UNIT SELECT PLUG RDY ONCE
3763 014270 012765 100000 000000 MOV #CCLR,RKCS1(R5)
3764 014276 010065 000010 MOV RO,RKCS2(R5) ;DRIVE ADDR
3765 014302 012765 000003 000000 MOV #PACK,RKCS1(R5) ;PACK CMD
3766 014310 013737 001426 005444 MOV T10,TEMP1
3767 014316 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
3768 014322 104116 ERROR 116 ;CONTR NOT RDY
3769 014324 010046 MOV RO,-(SP) ;SAVE RO FOR TYPEOUT
3770 ;TYPE UNIT SEL PLUG NO.
3771 014326 104403 TYPOS ;GO TYPE--OCTAL ASCII
3772 014330 001 .BYTE 1 ;TYPE 1 DIGIT(S)
3773 014331 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
3774 014332 104401 00.205 TYPE $CRLF
3775 014336 000665 BR 5$
3776
3777 014340 004737 033262 9$: JSR PC,$TKINT ;ENABLE KB INT
3778 014344 012765 100000 000000 MOV #CCLR,RKCS1(R5)
3779 014352 013765 001222 000010 MOV $UNIT,RKCS2(R5)
3780 014360 012765 000001 000000 MOV #SELDV,RKCS1(R5)
3781 014366 013737 001426 005444 MOV T10,TEMP1
3782 014374 004737 024422 JSR PC,FRDY
3783 014400 104117 ERROR 117 ;CONT NOT RDY AFTER SEL DRV CMD
3784
3785 014402 032765 010000 000010 BIT #NED,RKCS2(R5) ;SEE IF CORRECT PLUG FOR EXIT
3786 014410 001411 BEQ 10$ ;BR IF YES

```

```

3787 014412 104401 042137      TYPE      MSG55      ;EXIT WITH CORRECT UNIT SELECT PLUG NO.
3788 014416 013746 001222      MOV      $UNIT,-(SP) ;SAVE $UNIT FOR TYPEOUT
3789                                ;TYPE CORRECT UNIT SEL PLUG #
3790 014422 104403                                ;GO TYPE--OCTAL ASCII
3791 014424      001      .BYTE      1      ;TYPE 1 DIGIT(S)
3792 014425      000      .BYTE      0      ;SUPPRESS LEADING ZEROS
3793 014426 104401 040704      TYPE      MSG39      ;DEPRESS CONT-E WHEN DONE
3794 014432 000627      BR      $$
3795
3796 014434 004737 026332      10$:     JSR      PC,SUBCLR
3797 014440 104024      ERROR    24      ;CERR AFTER SCLR
3798
3799 014442 004737 025760      JSR      PC,GSTAT
3800 014446 032737 000100 005434      BIT      #D.VV,HMR2
3801 014454 001005      BNE     11$
3802 014456 104401 040747      TYPE     ,MSG40      ;VV NOT SET, INSERT UNIT SELECT PLUG
3803 014462 104401 040704      TYPE     ,MSG39      ;DEPRESS CONT-E TO EXIT TEST
3804 014466 000611      BR      $$
3805
3806 014470 104401 043611      11$:     TYPE     ,MSG71      ;SELECT CORRECT PORT ON OTHER DRIVES
3807 014474 104401 040502      TYPE     ,MSG37      ;TYPE SPACE WHEN DONE
3808 014500 004737 030346      JSR      PC,GETSP    ;GET SPACE
3809 014504
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820 014504 000004      ;*****
3821 014506 012737 000001 001174      ;*TEST 12      PORT SELECTION TESTS
3822 014514 012706 001100      ;*
3823                                ;*      THE OPERATOR IS ASKED TO SWITCH TO THE WRONG PORT
3824 014520 004737 026332      ;*      & THEN DESELECT BOTH PORTS.
3825 014524 104024      ;*      IN BOTH CASES, NON EXISTENT DRIVE SHOULD ASSERT IN RKCS2
3826                                ;*
3827                                ;*****
3828 014526 104401 037407      ;*
3829 014532 104401 041656      ;*
3830 014536 004737 030306      ;*
3831 014542 000137 014722      ;*
3832                                ;*****
3833 014546 104401 041061      ;*
3834 014552 104401 040502      ;*
3835 014556 004737 030346      ;*
3836                                ;*****
3837 014562 004737 025760      ;*
3838 014566 032765 010000 000010      ;*
3839 014574 001001      ;*
3840 014576 104122      ;*
3841
3842 014600 104401 041141      1$:     TYPE     ,MSG42      ;SELECT CORRECT PORT

```

```

3843 014604 104401 040502          TYPE      MSG37          ;DEPRESS SPACE BAR WHEN DONE
3844 014610 004737 030346          JSR        PC,GETSP      ;GET SPACE
3845
3846 014614 004737 026332          JSR        PC,SUBCLR
3847 014620 104024          ERROR     24            ;CERR AFTER SCLR
3848
3849 014622 032765 010000 000010        BIT        #NED,RKCS2(R5)
3850 014630 001401          BEQ       25
3851 014632 104130          ERROR     130          ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
3852
3853 014634 104401 041217          25:      TYPE      ,MSG43          ;Deselect BOTH PORTS
3854 014640 104401 040502          TYPE      MSG37          ;DEPRESS SPACE BAR WHEN DONE
3855 014644 004737 030346          JSR        PC,GETSP
3856 014650 004737 025760          JSR        PC,GSTAT
3857 014654 032765 010000 000010        BIT        #NED,RKCS2(R5)
3858 014662 001001          BNE       35
3859 014664 104133          ERROR     133          ;NED NOT SET AFTER BOTH PORTS DESELECTED
3860
3861
3862 014666 104401 041247          35:      TYPE      ,MSG44          ;SELECT CORRECT PORT
3863 014672 104401 040502          TYPE      MSG37          ;DEPRESS SPACE BAR WHEN DONE
3864 014676 004737 030346          JSR        PC,GETSP
3865
3866 014702 004737 026332          JSR        PC,SUBCLR
3867 014706 104024          ERROR     24            ;CERR AFTER SCLR
3868
3869 014710 032765 010000 000010        BIT        #NED,RKCS2(R5)
3870 014716 001421          BEQ       TST13        ;GOTO NEXT TEST
3871 014720 104130          ERROR     130          ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
3872
3873 014722 012765 100000 000000        45:      MOV        #CCLR,RKCS1(R5)
3874 014730 004737 025760          JSR        PC,GSTAT
3875 014734 032765 010000 000010        BIT        #NED,RKCS2(R5)
3876 014742 001407          BEQ       TST13        ;GOTO NEXT TEST
3877 014744 104401 041277          TYPE      ,MSG45          ;CORRECT PORT NOT SELECTED, TRY AGAIN
3878 014750 104401 040502          TYPE      MSG37          ;DEPRESS SPACE BAR WHEN DONE
3879 014754 004737 030346          JSR        PC,GETSP
3880 014760 000760          BR        45
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895 014762 000004          TST13:   SCOPE
3896 014764 012737 000001 001174        MOV        #1,$TIMES      ;DO 1 ITERATION
3897 014772 012706 001100          MOV        #STACK,SP     ;RESTORE STK PTR
3898

```

```

*****
*TEST 13      AC LOW DETECTION PART 1
*
*      A PRELIMINARY AC LOW TEST IS PERFORMED HERE WHILE HEADS ARE UNLOADED.
*      BATTERY RETRACT WILL BE TESTED LATER.
*      ACLO CANNOT BE TESTED DIRECTLY SINCE IT REQUIRES SECTOR PULSES
*      I.E. HEADS LOADED.
*      THEREFORE THE INDICATOR CHECKED IS NON-EXISTENT DRIVE (NED)
*      ASSERTING IN RKCS2 AS A RESULT OF DCLO.
*      AFTER POWER UP, VOLUME VALID IS CHECKED TO BE CLEARED.
*****

```

```

TST13:  SCOPE
        MOV        #1,$TIMES      ;DO 1 ITERATION
        MOV        #STACK,SP     ;RESTORE STK PTR

```

3899	014776	004737	026332			JSR	PC, SUBCLR	
3900	015002	104024				ERROR	24	;CERR AFTER SCLR
3901								
3902	015004	104401	037440			TYPE	,MSG21	;AC LOW TEST
3903	015010	104401	041656			TYPE	,MSG52	;CONT-E TO BYPASS TEST
3904								;OR SPACE TO CONTINUE
3905	015014	004737	030306			JSR	PC, CCSP	;INPUT CONT-E OR SPACE
3906	015020	000137	015302			JMP	10\$;RETURN HERE IF CONT-E
3907	015024	104401	041430			TYPE	,MSG49	;TURN OFF AC (RET HERE IF SPACE)
3908								
3909	015030	005037	005446			CLR	TEMP2	
3910	015034	012737	177777	005444	12\$:	MOV	#-1, TEMP1	;SETUP TIMEOUT
3911	015042	004737	025760		1\$:	JSR	PC, GSTAT	
3912	015046	032737	100000	005406		BIT	#CERR, HCS1	
3913	015054	001012				BNE	3\$	
3914	015056	005337	005444			DEC	TEMP1	
3915	015062	001367				BNE	1\$	
3916	015064	005237	005446			INC	TEMP2	
3917	015070	023727	005446	000002		CMP	TEMP2, #2	;SEE IF GONE THRU 2 TIMES
3918	015076	001356				BNE	12\$;BR IF NO
3919	015100	104146				ERROR	146	;CERR NOT DETECTED BEFORE TIMEOUT
3920	015102	013737	001440	005444	3\$:	MOV	T50000, TEMP1	
3921	015110	012765	100000	000000	4\$:	MOV	#CCLR, RKCS1(R5)	
3922	015116	004737	025760			JSR	PC, GSTAT	
3923	015122	032765	010000	000010		BIT	#NED, RKCS2(R5)	
3924	015130	001004				BNE	5\$	
3925	015132	005337	005444			DEC	TEMP1	
3926	015136	001364				BNE	4\$	
3927	015140	104150				ERROR	150	;NED NOT SET BEFORE TIMEOUT
3928								
3929	015142	104401	041503		5\$:	TYPE	,MSG50	;SWITCH AC BACK ON
3930	015146	104401	043563			TYPE	,MSG70	;VERIFY HEADS LOAD
3931	015152	104401	043506			TYPE	,MSG69	;PRESS SPACE AFTER READY ON
3932	015156	004737	030346			JSR	PC, GETSP	;GET SPACE
3933								
3934	015162	004737	026332			JSR	PC, SUBCLR	
3935	015166	104024				ERROR	24	;CERR AFTER SCLR
3936								
3937	015170	032737	000200	005434		BIT	#D.DRDY, HMR2	;SEE IF LOADED
3938	015176	001001				BNE	6\$;BR IN YES
3939	015200	104141				ERROR	141	;DRV NOT RDY AFTER AC UP
3940								
3941	015202	005037	005370		6\$:	CLR	UNLD	;FOR VALID HALT
3942	015206	032737	000170	005436		BIT	#D.ACLO, HMR3	
3943	015214	001401				BEQ	7\$	
3944	015216	104152				ERROR	152	;ACLO NOT RESET AFTER POWER UP
3945								
3946	015220	032737	000100	005434	7\$:	BIT	#D.VV, HMR2	
3947	015226	001401				BEQ	8\$	
3948	015230	104153				ERROR	153	;VV NOT RESET AFTER POWER UP
3949								
3950	015232				8\$:			
3951	015232	012765	100000	000000		MOV	#CCLR, RKCS1(R5)	
3952	015240	013765	001222	000010		MOV	\$UNIT, RKCS2(R5)	;DRIVE #
3953	015246	012765	000003	000000		MOV	#PACK, RKCS1(R5)	;PACK CMD
3954	015254	013737	001426	005444		MOV	T10, TEMP1	

```

3955 015262 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
3956 015266 104116 ERROR 116 ;CONTR NOT RDY
3957
3958 015270 032737 000100 005434 BIT #D.VV,HMR2
3959 015276 001001 BNE 64$
3960 015300 104027 ERROR 27 ;VOLUME VALID NOT SET AFTER PACK CMD
3961 015302 64$:
3962 015302 10$:
3963
3964
3965 *****
3966 *TEST 14 CHECK NXF LOGIC
3967 *
3968 * THIS TEST VERIFIES NON-EXECUTABLE FUNCTION (NXF) IS DETECTED
3969 * AS A RESULT OF DOING A SEEK WITH VOLUME VALID RESET.
3970 *
3971 *****
3972 ST14: SCOPE
3973 MOV #1,STIMES ;DO 1 ITERATION
3974 MOV #STACK,SP ;RESTORE STACK PTR
3975
3976 015316 004737 026332 JSR PC,SUBCLR
3977 015322 104024 ERROR 24 ;CERR AFTER SCLR
3978
3979 015324 032737 010000 005434 BIT #D.SPIN,HMR2 ;SEE IF SPINDLE ALREADY ON
3980 015332 001023 BNE 64$ ;BR IF YES
3981 015334 104401 037741 TYPE ,MSG29 ;PLEASE WAIT, HEADS BEING LOADED
3982
3983 015340 012765 000011 000000 MOV #SRTSPL,RKCS1(R5) ;START SPINDLE CMD
3984 015346 013737 001426 005444 MOV T10,TEMP1
3985 015354 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
3986 015360 104143 ERROR 143 ;CONTR RDY NOT SET AFTER CMD
3987
3988 015362 013737 001434 005446 MOV T100,TEMP2
3989 015370 004737 024736 JSR PC,FATT1 ;FIND ATTN
3990 015374 104144 ERROR 144 ;NO ATTN AFTER CMD
3991
3992 015376 005037 005370 CLR UNLD
3993 015402 005037 005370 64$: CLR UNLD ;FOR VALID HALT
3994
3995 015406 004737 026332 JSR PC,SUBCLR
3996 015412 104024 ERROR 24 ;CERR AFTER SCLR
3997
3998 015414 104401 037501 TYPE ,MSG22 ;NXF TEST
3999 015420 104401 041656 TYPE ,MSG52 ;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
4000 015424 004737 030306 JSR PC,CCSP ;INPUT CONT-E OR SPACE
4001 015430 000137 015752 JMP 7$ ;RETURN HERE FOR CONT-E
4002
4003 015434 104401 041535 TYPE ,MSG51 ;REMOVE UNIT SELECT PLUG TO CLEAR VV
4004 015440 104401 040502 TYPE ,MSG37 ;PRESS SPACE WHEN DONE
4005 015444 004737 030346 JSR PC,GETSP ;GET SPACE
4006
4007 015450 004737 025760 JSR PC,GSTAT
4008 015454 032737 000100 005434 BIT #D.VV,HMR2
4009 015462 001403 BEQ 1$
4010 015464 104177 ERROR 177 ;VV NOT CLEARED AFTER UNIT SEL PLUG

```

M06

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 76
T14 CHECK NXF LOGIC

SEQ 0076

```

4011 015466 000137 015752          JMP      7$          ;EXIT TEST
4012
4013 015472 032737 000100 005434 1$:  BIT      #D.VV,HMR2
4014 015500 001406          BEQ      2$
4015 015502 104155          ERROR   155        ;VV SET AFTER HEADS LOADED
4016 015504 000137 015752          JMP      7$          ;EXIT TEST
4017
4018 015510 004737 026332          JSR     PC,SUBCLR
4019 015514 104024          ERROR   24         ;CERR AFTER SCLR
4020
4021 015516 012765 000001 000020 2$:  MOV      #1,RKDC(R5)
4022 015524 012765 000017 000000          MOV      #SEEK,RKCS1(R5) ;SEEK CMD
4023 015532 013737 001426 005444          MOV      T10,TEMP1
4024 015540 004737 024422          JSR     PC,FRDY
4025 015544 104131          ERROR   131        ;FIND RDY
4026                                     ;NO RDY AFTER SEEK CMD
4027 015546 013737 001440 005444          MOV      T50000,TEMP1
4028 015554 004737 025032          JSR     PC,FATT2
4029 015560 104132          ERROR   132        ;FIND ATTN
4030                                     ;NO ATTN AFTER SEEK CMD
4031 015562 032737 000400 005436          BIT      #D.NXF,HMR3
4032 015570 001003          BNE     3$
4033 015572 104156          ERROR   156        ;NXF NOT SET AFTER SEEK WITH .V=L
4034 015574 000137 015752          JMP      7$          ;EXIT TEST
4035
4036 015600 032737 000200 005436 3$:  BIT      #D.FLT,HMR3
4037 015606 001003          BNE     4$
4038 015610 104172          ERROR   172        ;NXF DID NOT SET FAULT
4039 015612 000137 015752          JMP      7$          ;EXIT TEST
4040
4041 015616 012737 050240 005476 4$:  MOV      #<D.DSC!D.SPIN!D.DRDY!D.DRA>,E.A0 ;EXPECTED AO
4042 015624 012737 000600 005500          MOV      #<D.NXF!D.FLT>,E.B0
4043 015632 012737 001720 005502          MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4044 015640 012737 000001 005504          MOV      #1,E.B1
4045
4046 015646 004737 025144          JSR     PC,CHKMSG
4047 015652 000003          .WORD   T.A2!T.B2!0 ;CHECK MSGS AO, BO, A1, B1
4048 015654 104160          ERROR   160        ;& MSGS SPECIFIED HERE
4049 015656 104161          ERROR   161        ;MSG AO ERROR AFTER SEEK WITH VV=0
4050 015660 104162          ERROR   162        ;MSG BO ERROR
4051 015662 104163          ERROR   163        ;MSG A1 ERROR
4052 015664 005737 001400          TST     CYLADD
4053 015670 001401          BEQ     5$
4054 015672 104157          ERROR   157        ;MSG B1 ERROR
4055                                     ;HEADS MOVED WITH SEEK & DXF
4056 015674 004737 026332          JSR     PC,SUBCLR
4057 015700 104024          ERROR   24         ;CERR AFTER SCLR
4058
4059 015702 012765 100000 000000          MOV      #CCLR,RKCS1(R5)
4060 015710 013765 001222 000010          MOV      $UNIT,RKCS2(R5) ;DRIVE #
4061 015716 012765 000003 000000          MOV      #PACK,RKCS1(R5) ;PACK CMD
4062 015724 013737 001426 005444          MOV      T10,TEMP1
4063 015732 004737 024422          JSR     PC,FRDY
4064 015736 104116          ERROR   116        ;FIND CONTR RDY
4065                                     ;CONTR NOT RDY
4066 015740 032737 000100 005434          BIT      #D.VV,HMR2

```

4067 015746 001001
4068 015750 104027
4069 015752
4070 015752

BNE 655
ERROR 27

; VOLUME VALID NOT SET AFTER PACK CMD

655:
75:

*TEST 15 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

* THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
* THIS AREA CONTAINS BAD SECTOR INFOR WHICH IS WRITTEN BY THE
* FACTORY DURING MANF. ALL BAD SECTOR INFOR (BSE) WILL BE STORED
* AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.

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

* IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
* IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
* A MESSAGE WILL BE TYPED INDICATING THAT ALL
* FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
* THIS IS DONE SO AS NOT TO DESTROY BSE INFOR OR AN ALIGNMENT PACK BY WRITING

* THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.

4091
4092 015752 000004
4093 015754 012737 000001 001174
4094 015762 012706 001100
4095
4096 015766 004737 026332
4097 015772 104024
4098 015774 005037 005446
4099 016000 005037 005450
4100
4101
4102 016004 012737 002370 005452
4103 016012 013765 005452 000004
4104 016020 012737 001000 005454
4105 016026 013765 005454 000006
4106
4107 016034 012765 000632 000020 18:
4108 016042 012765 177400 000002
4109 016050 012765 000021 000000
4110 016056 013737 001440 005444
4111 016064 004737 024422
4112 016070 104226
4113 016072 004737 025760
4114 016076 032737 100000 005406
4115 016104 001470
4116 016106 104227
4117
4118 016110 012737 010340 005476
4119 016116 005037 005500
4120 016122 012737 001720 005502
4121 016130 012737 000001 005504
4122 016136 005037 005506

*T15: SCOPE
MOV #1,STIMES ;:CO 1 ITERATION
MOV #STACK,SP ;:RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;:CERR AFTER SCLR
CLR TEMP2 ;:SECTOR CTR
CLR TEMP3 ;:0=22 SECTOR HARDWARE DETECTED TABLE
;:1=22 SECTOR SOFTWARE DETECTED TABLE
;:2=DONE
MOV #BSE22H,TEMP4 ;:STORE 22 SECTOR HARDWARE BSE ADDR.
MOV TEMP4,RKBA(R5)
MOV #1000,TEMP5 ;:TRACK 2, SECTOR 0
MOV TEMP5,RKDA(R5)
MOV #410.,RKDC(R5) ;:CYL 410
MOV #-256.,RKWC(R5) ;:LOAD WORD CT
MOV #RODATA,RKCS1(R5) ;:READ DATA COMMAND
MOV 15000,TEMP1 ;:SETUP TIMEOUT
JSR PC,FRDY ;:FIND RDY
ERROR 226 ;:NO RDY AFTER READ DATA CMD
JSR PC,GSTAT ;:GET FRESH DATA
BIT #CEP?,HCS1
BEQ B5
ERROR 227 ;:CERR AFTER READ DATA CMD
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

```

4123 016142 012737 000002 005510      MOV      #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
4124 016157 012737 000003 005514      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
4125
4126 016156 004737 025144      JSR      PC,CHKMSG    ;CHECK MSGS A0, B0, A1, B1
4127 016162 000000      .WORD    0!0!0        ;& MSGS SPECIFIED HERE
4128 016164 104054      ERROR    54           ;MSG A0 ERROR AFTER READ DATA CMD
4129 016166 104026      ERROR    26           ;MSG B0 ERROR
4130 016170 104056      ERROR    56           ;MSG A1 ERROR
4131 016172 104030      ERROR    30           ;MSG B1 ERROR
4132 016174 004737 026332      JSR      PC,SUBCLR    ;
4133 016200 104024      ERROR    24           ;CERR AFTER SUBCLR
4134 016202 005237 005446      INC      TEMP2        ;
4135 016206 023727 005446 000005      CMP      TEMP2,#5     ;READ ALL 5 SECTORS?
4136 016214 001007      BNE      5$           ;
4137 016216 005737 005450      TST      TEMP3        ;
4138 016222 001002      BNE      2$           ;
4139 016224 104165      ERROR    165         ;CANT READ SECTORS 0,2,4,6,8
4140 016226 000414      BR       3$           ;
4141 016230 104167      2$: ERROR    167         ;CANT READ SECTORS 10,12,14,16,18,20
4142 016232 000412      BR       3$           ;
4143
4144 016234 013765 005452 000004 5$: MOV      TEMP4,RKBA(R5) ;RESTORE TABLE ADDR
4145 016242 062737 000002 005454      ADD      #2,TEMP5     ;SETUP TO READ 2 SECTORS FROM LAST
4146 016250 013765 005454 000006      MOV      TEMPS,RKDA(R5)
4147 016256 000666      BR       1$           ;
4148
4149 016260 005237 001542 3$: INC      BSERR      ;SET BSE FLAG
4150 016264 000504      BR       TST16       ;GO TO NEXT TEST
4151 016266 005737 002376 8$: TST      BSE22H+6   ;TEST CARTRIDGE TYPE
4152 016272 001405      BEQ      9$         ;BRANCH IF DATA CARTRIDGE
4153 016274 104401 042217      TYPE     MSG56       ;ALIGNMENT CARTRIDGE USED
4154 016300 005237 001542      INC      BSERR      ;SET BSE ERROR FLAG
4155 016304 000426      BR       10$        ;
4156
4157 016306 005237 005450 9$: INC      TEMP3        ;
4158 016312 023727 005450 000001      CMP      TEMP3,#1    ;
4159 016320 001020      BNE      10$        ;
4160 016322 005037 005446      CLR      TEMP2        ;
4161 016326 012737 003370 005452      MOV      #BSE22S,TEMP4 ;STORE 22 SECTOR SOFTWARE BSE ADDR
4162 016334 013765 005452 000004      MOV      TEMP4,RKBA(R5)
4163 016342 012737 001012 005454      MOV      #1012,TEMP5 ;TRACK 2, SECTOR 12
4164 016350 013765 005454 000006      MOV      TEMPS,RKDA(R5)
4165 016356 000137 016034      JMP      1$         ;REPEAT
4166
4167 016362 005737 001216 10$: TST      $PASS      ;
4168
4169
4170 016366 001014      BNE      13$        ;TYPE CART # ONLY ON 1'ST PASS
4171 016370 104401 037325      TYPE     MSG17       ;CART SERIAL #
4172 016374 012746 002370      MOV      #BSE22H,-(SP)
4173 016400 004737 034662      JSR      PC,$DB20    ;CONVERT DBL BINARY WORD TO OCTAL
4174 016404 004737 035232      JSR      PC,$SUPRS   ;TYPE SERIAL #
4175 016410 104401 001205      TYPE     $CRLF
4176 016414 104401 001205      TYPE     $CRLF
4177
4178 016420 004737 026332 13$: JSR      PC,SUBCLR

```



```

4179 016424 104024          ERROR 24          ;CERR AFTER SCLR
4180                                     ;RETURN TO CYL 0
4181
4182 016426 012765 000017 000000      MOV      #SEEK,RKCS1(R5) ;SEEK CMD
4183 016434 013737 001430 005444      MOV      T50,TEMP1      ;SETUP TIMEOUT
4184 016-42 004737 024422      JSR      PC,FRDY        ;FIND ROY
4185 016446 104131          ERROR 131          ;NO ROY AFTER SEEK CMD
4186
4187 016450 013737 001440 005444      MOV      T50000,TEMP1   ;SETUP TIMEOUT
4188 016456 004737 025032      JSR      PC,FATT2       ;FIND ATTN
4189 016462 104132          ERROR 132          ;NO ATTN AFTER SEEK CMD
4190
4191 016464 032737 100000 005406      BIT      #CERR,HCS1     .
4192 016472 001401          BEQ      64$
4193 016474 104210          ERROR 210          ;CERR AFTER SEEK CMD
4194
4195 016476          64$:
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207 016476 000004          *TEST 16 WRITE LOCK TEST
4208 016500 012737 000001 001174      *
4209 016506 012706 001100          * THIS TEST VERIFIES THAT DATA WRITTEN ON A SECTOR CANNOT BE
4210 016512 005737 001542          * ALTERED ONCE THE WRITE LOCK SWITCH HAS BEEN ENABLED.
4211 016516 001402          * IT ALSO CHECKS THAT WRITE PROTECT TAKES EFFECT ONLY AT
4212 016520 000137 021426          * SECTOR BOUNDARIES WHEN DOING CONTINUOUS WRITING ON CYL 0, HEAD 0
4213
4214 016524 004737 026332      *
4215 016530 104024          * *****
4216          *
4217 016532 104401 037562      *
4218 016536 104401 041656      *
4219 016542 004737 030306      *
4220 016546 000137 021426      *
4221          *
4222 016552 004737 025760      *
4223 016556 032737 004000 005434      *
4224 016564 001416          *
4225 016566 104401 041757      *
4226 016572 104401 040502      *
4227 016576 004737 030346      *
4228          *
4229 016602 004737 025760      *
4230 016606 032737 004000 005434      *
4231 016614 001402          *
4232 016616 104110          *
4233 016620 000762          *
4234          *

```

4235	016622	005037	001366		25:	CLR	TOCYL	; SETUP
4236	016626	005037	001402			CLR	CALADD	; FOR
4237	016632	005037	001510			CLR	HEAD	; FILL HEADER TABLE
4238	016636	005037	001516			CLR	FORMAT	; ROUTINE
4239								
4240	016642	004737	027314		165:	JSR	PC, FHDTAB	; BUILD STD 22 SECTOR HEADER TABLE
4241								
4242	016646	012765	001554	000004		MOV	#HDTAB, RKBA(R5)	
4243	016654	012765	177676	000002		MOV	#-66., RKWC(R5)	
4244	016662	000337	001510			SWAB	HEAD	
4245	016666	013765	001510	000006		MOV	HEAD, RKDA(R5)	; HEAD ADDR
4246	016674	000337	001510			SWAB	HEAD	
4247								
4248								
4249	016700	012765	000027	000000		MOV	#(WRHEAD), RKCS1(R5)	; WRITE HEADER CMD
4250	016706	013737	001440	005444		MOV	T50000, TEMP1	; SETUP TIMEOUT
4251	016714	004737	024422			JSR	PC, FRDY	; FIND RDY
4252	016720	104200				ERROR	200	; NO RDY AFTER WRITE HEADER CMD
4253	016722	004737	025760			JSR	PC, GSTAT	; GET FRESH STATUS
4254	016726	032737	100000	005406		BIT	#CERR, HCS1	
4255	016734	001401				BEQ	645	
4256	016736	104201				ERROR	201	; CERR AFTER WRITE HEADER CMD
4257	016740				645:			
4258								
4259								
4260	016740	005237	001510			INC	HEAD	
4261	016744	023727	001510	000003		CMP	HEAD, #3	; SEE IF ALL HEADS DONE
4262	016752	001333				BNE	165	; BR IF NO
4263								
4264	016754	004737	026332			JSR	PC, SUBCLR	
4265	016760	104024				ERROR	24	; CERR AFTER SCLR
4266								
4267	016762	005037	001422			CLR	SECTOR	
4268	016766	013765	001422	000006	145:	MOV	SECTOR, RKDA(R5)	
4269								
4270	016774	012765	001524	000004		MOV	#DATA0, RKBA(R5)	; WRITE ALL 0'S
4271	017002	052765	000020	000010		BIS	#BA1, RKCS2(R5)	
4272	017010	012765	177400	000002		MOV	#-256., RKWC(R5)	; CYL 0, TRK 0, SECTOR 0
4273								
4274	017016	012765	000023	000000		MOV	#(WRDATA), RKCS1(R5)	; WRITE DATA CMD
4275	017024	013737	001440	005444		MOV	T50000, TEMP1	; SETUP TIMEOUT
4276	017032	004737	024422			JSR	PC, FRDY	; FIND RDY
4277	017036	104011				ERROR	11	; NO RDY AFTER WRITE DATA CMD
4278	017040	004737	025760			JSR	PC, GSTAT	; GET FRESH STATUS
4279	017044	032737	100000	005406		BIT	#CERR, HCS1	
4280	017052	001465				BEQ	685	; BR IF NO ERRORS
4281								
4282	017054	032737	000200	005422		BIT	#BSE, HER	; SEE IF BAD SECTOR FLAG
4283	017062	001421				BEQ	665	; BR IF NO
4284	017064	004737	027752			JSR	PC, TRUERR	; ELSE SEE IF SECTOR LISTED IN BSE TABLE
4285	017070	000455				BR	675	; RETURN HERE IF NO
4286								
4287	017072	005237	001422			INC	SECTOR	; RETURN HERE IF YES
4288	017076	023727	001422	000012		CMP	SECTOR, #10.	; ARE 10 CONSEC. SECTORS BAD
4289	017104	001003				BNE	655	; BR IF NO
4290	017106	104040				ERROR	40	; ABORTING TEST DETECTED 10 BAD SECTORS

E07

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05MACY11 27(1006) 31-JAN-77 18:30 PAGE 81
T16 WRITE LOCK TEST

SEQ 0081

```

4291 017110 000137 021426          JMP      12$          ;BYPASS TEST
4292
4293 017114 012765 100000 000000 65$:  MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4294 017122 000137 016766          JMP      14$
4295
4296 017126 104012          66$:  ERROR    12          ;CERR WITH WRITE DATA CMD
4297
4298 017130 012737 010340 005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4299 017136 005037 005500          CLR     E.B0          ;EXPECTED MSG B0
4300 017142 012737 001720 005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4301 017150 012737 000001 005504          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
4302 017156 005037 005506          CLR     E.A2          ;EXPECTED MSG A2
4303 017162 012737 000002 005510          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
4304 017170 012737 000003 005514          MOV     #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
4305
4306 017176 004737 025144          JSR     PC,CHKMSG      ;CHECK MSGS A0 B0 A1 B1
4307 017202 000003          .WORD   T.A2!T.B2!0    ;& MSGS SPECIFIED HERE
4308 017204 104052          ERROR   52          ;MSG A0 ERROR AFTER WRITE DATA CMD
4309 017206 104023          ERROR   23          ;MSG B0 ERROR
4310 017210 104053          ERROR   53          ;MSG A1 ERROR
4311 017212 104025          ERROR   25          ;MSG B1 ERROR
4312 017214 104401 043674          TYPE    MSG72        ;ABORTING BALANCE OF TESTS
4313 017220 000137 023742          JMP     $EOP
4314 017224 104047          67$:  ERROR    47          ;BAD SECTOR NOT LISTED IN TABLE
4315 017226          68$:
4316
4317 017226 004737 026332          JSR     PC,SUBCLR
4318 017232 104024          ERROR   24          ;CERR AFTER SCLR
4319
4320 017234 012765 001524 000004          MOV     #DATA0,RKBA(R5)
4321 017242 052765 000020 000010          BIS     #BA1,RKCS2(R5)
4322 017250 012765 177400 000002          MOV     #-256.,RKWC(R5)
4323 017256 013737 001524 001504          MOV     DATA0,W02    ;EXPECTED WORD FOR TRUERR TYPEOUT
4324 017264 013765 001422 000006          MOV     SECTOR,RKDA(R5)
4325
4326 017272 012765 000031 000000          MOV     #<WRTCHK>,RKCS1(R5) ;WRITE CHECK CMD
4327 017300 013737 001440 005444          MOV     T5000,TEMP1  ;SETUP TIMEOUT
4328 017306 004737 024422          JSR     PC,FRDY      ;FIND RDY
4329 017312 104015          ERROR   15          ;NO RDY AFTER WRITE CHECK CMD
4330 017314 004737 025760          JSR     PC,GSTAT     ;GET FRESH STATUS
4331 017320 032737 100000 005406          BIT     #CERR,HCS1
4332 017326 001450          BEQ     70$
4333 017330 032737 040000 005410          BIT     #WCE,HCS2    ;SEE IF WRITE CHECK ERROR
4334 017336 001405          BEQ     69$
4335 017340 016537 000024 001502          MOV     RKDB(R5),W01 ;ACTUAL WORD FOR PRINTOUT
4336 017346 104016          ERROR   16          ;WCE AFTER WRITE CMD
4337 017350 000437          BR      70$
4338
4339 017352 104022          69$:  ERROR    22          ;CERR AFTER WRITE CHECK CMD
4340
4341 017354 012737 010340 005476          MOV     #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4342 017362 005037 005500          CLR     E.B0          ;EXPECTED MSG B0
4343 017366 012737 001720 005502          MOV     #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4344 017374 012737 000001 005504          MOV     #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
4345 017402 005037 005506          CLR     E.A2          ;EXPECTED MSG A2
4346 017406 012737 000002 005510          MOV     #2,E.B2          ;MSG ID FOR EXPECTED MSG B2

```

```

4347 017414 012737 000003 005514      MOV      #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
4348
4349 017422 004737 025144      JSR      PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4350 017426 000003                .WORD    T.A2!T.B2!0     ;& MSGS SPECIFIED HERE
4351 017430 104057                ERROR    57              ;MSG A0 ERROR AFTER WRITE CHECK CMD
4352 017432 104031                ERROR    31              ;MSH B0 ERROR
4353 017434 104060                ERROR    60              ;MSG A1 ERROR
4354 017436 104032                ERROR    32              ;MSG B1 ERROR
4355 017440 104401 043674      TYPE     ,MSG72          ;ABORTING BALANCE OF TESTS
4356 017444 000137 023416      JMP      ENDRV
4357
4358 017450                          70$:
4359
4360 017450 004737 026332      3$:      JSR      PC,SUBCLR       ;CERR AFTER SCLR
4361 017454 104024                ERROR    24
4362
4363 017456 104401 042050      TYPE     ,MSG54          ;ENABLE WRITE LOCK
4364 017462 104401 040502      TYPE     ,MSG37          ;PRESS SPACE WHEN DONE
4365 017466 004737 030346      JSR      PC,GETSP        ;GET SPACE
4366
4367 017472 012765 100000 000000      MOV      #CCLR,RKCS1(R5)
4368 017500 004737 025760      JSR      PC,GSTAT
4369 017504 032737 004000 005434      BIT      #D.WRL,HMR2     ;SEE IF WRITE LOCK IS ON
4370 017512 001002                BNE     4$
4371 017514 104115                ERROR    115            ;WRITE LOCK NOT ENABLED
4372 017516 000754                BR      3$
4373
4374 017520 012765 001530 000004      4$:      MOV      #DATA1,RKBA(R5) ;ATTEMPT TO WRITE ALL 1'S WITH WRITE LOCK SET
4375 017526 052765 000020 000010      BIS      #BA1,RKCS2(R5)
4376 017534 012765 177400 000002      MOV      #-256.,RKWC(R5) ;CYLINDER 0, HEAD 0
4377 017542 013765 001422 000006      MOV      SECTOR,RKDA(R5)
4378 017550 012765 000023 000000      MOV      #WRDATA,RKCS1(R5)
4379 017556 013737 001440 005444      MOV      T50000,TEMP1    ;SETUP TIMEOUT
4380 017564 004737 024422      JSR      PC,FRDY         ;FIND RDY
4381 017570 104116                ERROR    116            ;CONTR NOT RDY
4382
4383 017572 004737 025760      JSR      PC,GSTAT       ;GET FRESH STATUS
4384 017576 032737 100000 005406      BIT      #CERR,HCS1
4385 017604 001001                BNE     17$
4386 017606 104207                ERROR    207            ;CERR NOT SET AFTER WRITE WITH WRL SET
4387 017610 032737 004000 005436      17$:     BIT      #D.WLE,HMR3     ;CHECK WRITE LOCK ERROR SET
4388 017616 001001                BNE     5$              ;BR IF SET
4389 017620 104121                ERROR    121            ;WRITE LOCK ERROR NOT SET
4390
4391 017622 012737 054340 005476      5$:      MOV      #<D.DSC!D.SPIN!D.WRL!D.DRDY!D.VV!D.DRA>,E.A0 ;EXP A0
4392 017630 012737 004200 005500      MOV      #<D.WLE!D.FLT>,E.B0
4393 017636 012737 001720 005502      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4394 017644 012737 000001 005504      MOV      #1,E.B1
4395
4396 017652 004737 025144      JSR      PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4397 017656 000000                .WORD    0!0!0          ;& MSGS SPECIFIED HERE
4398 017660 104123                ERROR    123            ;MSG A0 ERROR AFTER WRITE WITH WRITE LOCK SET
4399 017662 104125                ERROR    125            ;MSH B0 ERROR
4400 017664 104126                ERROR    126            ;MSG A1 ERROR
4401 017666 104127                ERROR    127            ;MSG B1 ERROR
4402

```

4403	017670	004737	026332		JSR	PC, SUBCLR	
4404	017674	104024			ERROR	24	;CERR AFTER SCLR
4405							
4406	017676	012765	001524	000004	MOV	#DATA0, RKBA(R5)	;CHECK THAT NONE OF ORIG DATA
4407	017704	052765	000020	000010	BIS	#BAI, RKCS2(R5)	;HAS CHANGED
4408	017712	012765	177400	000002	MOV	#-256, RKWC(R5)	;AS RESULT OF WRITE WITH WRITE LOCK SET
4409	017720	013737	001524	001504	MOV	DATA0, WD2	;EXPECTED WORD FOR TRUERR TYPEOUT
4410	017726	013765	001422	000006	MOV	SECTOR, RKDA(R5)	
4411							
4412	017734	012765	000031	000000	MOV	#(WRTCHK), RKCS1(R5)	;WRITE CHECK CMD
4413	017742	013737	001440	005444	MOV	T5000, TEMP1	;SETUP TIMEOUT
4414	017750	004737	024422		JSR	PC, FRDY	;FIND RDY
4415	017754	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
4416	017756	004737	025760		JSR	PC, GSTAT	;GET FRESH STATUS
4417	017762	032737	100000	005406	BIT	#CERR, HCS1	
4418	017770	001450			BEQ	72\$	
4419	017772	032737	040000	005410	BIT	#WCE, HCS2	;SEE IF WRITE CHECK ERROR
4420	020000	001405			BEQ	71\$	
4421	020002	016537	000024	001502	MOV	RKDB(R5), WD1	;ACTUAL WORD FOR PRINTOUT
4422	020010	104016			ERROR	16	;WCE AFTER WRITE CMD
4423	020012	000437			BR	72\$	
4424							
4425	020014	104022			71\$: ERROR	22	;CERR AFTER WRITE CHECK CMD
4426							
4427	020016	012737	010340	005476	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA), E.A0	;EXPECTED MSG A0
4428	020024	005037	005500		CLR	E.B0	;EXPECTED MSG B0
4429	020030	012737	001720	005502	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP), E.A1	;EXPECTED A1
4430	020036	012737	000001	005504	MOV	#1, E.B1	;MSG ID FOR EXPECTED MSG B1
4431	020044	005037	005506		CLR	E.A2	;EXPECTED MSG A2
4432	020050	012737	000002	005510	MOV	#2, E.B2	;MSG ID FOR EXPECTED MSG B2
4433	020056	012737	000003	005514	MOV	#3, E.B3	;MSG ID FOR EXPECTED MSG B3
4434							
4435	020064	004737	025144		JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
4436	020070	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4437	020072	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
4438	020074	104031			ERROR	31	;MSG B0 ERROR
4439	020076	104060			ERROR	60	;MSG A1 ERROR
4440	020100	104032			ERROR	32	;MSG B1 ERROR
4441	020102	104401	043674		TYPE	MSG72	;ABORTING BALANCE OF TESTS
4442	020106	000137	023416		JMP	ENDRV	
4443							
4444	020112				72\$:		
4445							
4446	020112	004737	026332		6\$: JSR	PC, SUBCLR	
4447	020116	104024			ERROR	24	;CERR AFTER SCLR
4448							
4449	020120	104401	041757		TYPE	MSG53	;DISABLE WRITE LOCK
4450	020124	104401	040502		TYPE	MSG37	;SPACE BAR WHEN DONE
4451	020130	004737	030346		JSR	PC, GETSP	;GET SPACE
4452	020134	004737	025760		JSR	PC, GSTAT	
4453	020140	032737	004000	005434	BIT	#D.WRL, HMR2	;SEE IF WRITE LOCK OFF
4454	020146	001361			BNE	6\$	
4455							
4456							
4457	020150	104401	042050		TYPE	MSG54	;TEST FOR WRITE LOCK AT SECTOR BOUNDRIES
4458	020154	013737	001436	001160	MOV	T5000, \$TMP0	;FOR CONTINUOUS WRITING ON CYL 0, TRACK 0,1,2 ;SET WRITE LOCK

H07

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 84
T16 WRITE LOCK TEST

SEQ 0084

```

4459 020162 005037 001162          CLR      $TMP1          ;BITO=0:WRITE 0'S; BITO=1;WRITE 1'S
4460 020166 005037 001366          CLR      TOCYL
4461
4462 020172 004737 026332          7$:     JSR      PC,SUBCLR
4463 020176 104024                    ERROR    24          ;CERR AFTER SCLR
4464
4465 020200 032737 000001 001162      BIT      #BITO,$TMP1
4466 020206 001004                    BNE     8$          ;BR IF WRITING 1'S
4467 020210 012765 001524 000004      MOV     #DATA0,RKBA(R5) ;SETUP ALL 0'S
4468 020216 000403                    BR      9$
4469
4470 020220 012765 001530 000004      8$:     MOV     #DATA1,RKBA(R5) ;SETUP ALL 1'S
4471 020226 052765 000020 000010      9$:     BIS     #BA1,RKCS2(R5)
4472 020234 012765 140400 000002      MOV     #-63,*256.,RKWC(R5) ;WRITE TRACK 0,1,2 63 SECTORS
4473 020242 005065 000006          CLR     RKDA(R5)      ;BEGIN AT TRACK AND SEC 0
4474 020246 013765 001366 000020      MOV     TOCYL,RKDC(R5)
4475 020254 012765 000023 000000      MOV     #WRDATA,RKCS1(R5)
4476 020262 013737 001440 005444      MOV     T5000,$TMP1
4477 020270 004737 024422          JSR     PC,FROY
4478 020274 104011                    ERROR    11          ;FIND CONTR ROY
4479                                     ;CONTR NOT ROY
4480 020276 032737 100000 005406      BIT     #CERR,HCS1
4481 020304 001017                    BNE     13$
4482 020306 005337 001160          DEC     $TMP0
4483 020312 001011                    BNE     10$
4484 020314 104204                    ERROR    204          ;CERR NOT SET BY TIMEOUT
4485 020316 104401 041757          TYPE    ,MSG53        ;DISABLE WRITE LOCK
4486 020322 104401 040502          TYPE    ,MSG37        ;PRESS SPACE WHEN DONE
4487 020326 004737 030346          JSR     PC,GETSP      ;INPUT SPACE
4488 020332 000137 021426          JMP     12$          ;EXIT TEST
4489
4490 020336 005237 001162          10$:    INC     $TMP1
4491 020342 000713                    BR      7$          ;GO WRITE OPPOSITE DATA
4492 020344 032737 000200 005422      13$:    BIT     #BSE,HER    ;SEE IF BAD SECTOR
4493 020352 001411                    BEQ     21$          ;BR IF NO
4494 020354 005237 001366          INC     TOCYL        ;ELSE TRY ANOTHER CYL
4495 020360 023727 001366 000012      CMP     TOCYL,#10.   ;SEE IF 10 CONSEC CYL BAD
4496 020366 001301                    BNE     7$          ;BR IF NO & DO AGAIN
4497 020370 104062                    ERROR    62          ;10 BAD CONSEC CYLINDERS
4498 020372 000137 021426          JMP     12$          ;EXIT TEST
4499
4500 020376 032737 004000 005436      21$:    BIT     #D.WLE,HMR3
4501 020404 001001                    BNE     11$
4502 020406 104205                    ERROR    205          ;NO WRL BY TIMEOUT
4503
4504 020410 104401 041757          11$:    TYPE    ,MSG53        ;DISABLE WRITE LOCK
4505 020414 104401 040502          TYPE    ,MSG37        ;TYPE SPACE WHEN DONE
4506 020420 004737 030346          JSR     PC,GETSP
4507
4508 020424 013737 005416 001164      MOV     HDA,$TMP2    ;STORE RKDA OF ERROR
4509 020432 013737 005416 001166      MOV     HDA,$TMP3    ;STORE FOR ERROR TYPEOUT
4510
4511 020440 023727 001164 000001      CMP     $TMP2,#1     ;SEE IF WRL ON TRK 0, SEC 1
4512 020446 001003                    BNE     22$          ;BR IF NO
4513 020450 104401 044072          TYPE    ,MSG100      ;ELSE WRL OCCURRED ON TRK 2, SEC 21
4514 020454 000616                    BR      6$          ;& NO NEW DATA XFER TOOK PLACE

```

```

4515
4516 020456 005737 001164      22$:  TST      $TMP2      ;SEE IF WRL ON TRK 0, SEC 0
4517 020462 001004                BNE      23$      ;BR IF NO
4518 020464 104401 044072      TYPE    MSG100    ;ELSE WRL ON TRK 2, SEC 20
4519 020470 000137 020112      JMP      6$      ;& NO NEW DATA XFER TOOK PLACE
4520
4521 020474 023727 001164 001025 23$:  CMP      $TMP2,#1025 ;SEE IF WRL ON TRK 2, SEC 21
4522 020502 001004                BNE      24$      ;BR IF NO
4523 020504 104401 044072      TYPE    MSG100    ;ELSE WRL ON TRK 2, SEC 19
4524 020510 000137 020112      JMP      6$      ;& NO NEW DATA XFER TOOK PLACE
4525
4526 020514 023727 001164 001024 24$:  CMP      $TMP2,#1024 ;SEE IF WRL ON TRK 2, SEC 20
4527 020522 001004                BNE      25$      ;BR IF NO
4528 020524 104401 044072      TYPE    MSG100    ;ELSE WRL ON TRK 2, SEC 18
4529 020530 000137 020112      JMP      6$      ;& NO OLD DATA TO CHECK AGAINST
4530
4531 020534 023727 001164 000400 25$:  CMP      $TMP2,#400  ;SEE IF WRL AT TRK 1, SEC 0
4532 020542 001004                BNE      18$      ;BR IF NO
4533 020544 012765 000025 000006  MOV      #21.,RKDA(R5) ;ELSE SECTOR AT WRL IS TRK 0, SEC 21
4534 020552 000415                BR
4535
4536 020554 023727 001164 001000 18$:  CMP      $TMP2,#1000 ;SEE IF WRL AT TRK 2, SEC 0
4537 020562 001004                BNE      19$      ;BR IF NO
4538 020564 012765 000425 000006  MOV      #425,RKDA(R5) ;ELSE SECTOR AT WRL IS TRK 1, SEC 21
4539 020572 000405                BR
4540
4541 020574 005337 001164      19$:  DEC      $TMP2      ;GET SECTOR AT WRL
4542 020600 013765 001164 000006  MOV      $TMP2,RKDA(R5)
4543
4544 020606 016537 000006 001166 20$:  MOV      RKDA(R5),$TMP3 ;FOR ERROR PRINTOUT
4545
4546
4547
4548 020614 004737 026332      JSR      PC,SUBCLR  ;CERR AFTER SCLR
4549 020620 104024      ERROR    24
4550
4551 020622 005737 001164      TST      $TMP2      ;SEE IF TRK/SECTOR 0
4552 020626 001414                BEQ      80$      ;REPEAT,NO NEW DATA XFER TOOK PLACE
4553 020630 023727 001164 001023  CMP      $TMP2,#1023 ;SEE IF TRK 2,SECTOR 19
4554 020636 001410                BEQ      80$      ;REPEAT,NO OLD DATA TO CHECK AGAINST
4555 020640 032737 000001 001162  BIT      #BIT0,$TMP1
4556 020646 001006                BNE      73$      ;BR IF WRITING 1'S WHEN WLE OCCURRED
4557 020650 012765 001530 000004  MOV      #DATA1,RKBA(R5) ;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4558 020656 000405                BR      74$
4559
4560 020660 000137 020112      80$:  JMP      6$
4561
4562 020664 012765 001524 000004 73$:  MOV      #DATA0,RKBA(R5) ;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4563 020672 052765 000020 000010 74$:  BIS      #BA1,RKCS2(R5)
4564 020700 012765 177400 000002  MOV      #-256.,RKWC(R5)
4565 020706 013765 001166 000006  MOV      $TMP3,RKDA(R5) ;REFRESH RKDA
4566 020714 017537 000004 001504  MOV      @RKBA(R5),WD2 ;EXPECTED WORD FOR TRUERR TYPEOUT
4567 020722 013765 001366 000020  MOV      TOCYL,RKDC(R5)
4568
4569 020730 012765 000031 000000  MOV      #<WRTCHK>,RKCS1(R5) ;WRITE CHECK CMD
4570 020736 013737 001440 005444  MOV      T5000,TEMP1 ;SETUP TIMEOUT

```

4571	020744	004737	024422		JSR	PC,FRDY		;FIND ROY
4572	020750	104015			ERROR	15		;NO ROY AFTER WRITE CHECK CMD
4573	020752	004737	025760		JSR	PC,GSTAT		;GET FRESH STATUS
4574	020756	032737	100000	005406	BIT	#CERR,HCS1		
4575	020764	001450			BEQ	82\$		
4576	020766	032737	040000	005410	BIT	#WCE,HCS2		;SEE IF WRITE CHECK ERROR
4577	020774	001405			BEQ	81\$		
4578	020776	016537	000024	001502	MOV	RKDB(R5),WD1		;ACTUAL WORD FOR PRINTOUT
4579	021004	104134			ERROR	134		;WCE AFTER WRITE CMD
4580	021006	000437			BR	82\$		
4581								
4582	021010	104022			81\$:	ERROR	22	;CERR AFTER WRITE CHECK CMD
4583								
4584	021012	012737	010340	005476	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
4585	021020	005037	005500		CLR	E.B0		;EXPECTED MSG B0
4586	021024	012737	001720	005502	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1		;EXPECTED A1
4587	021032	012737	000001	005504	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
4588	021040	005037	005506		CLR	E.A2		;EXPECTED MSG A2
4589	021044	012737	000002	005510	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
4590	021052	012737	000003	005514	MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
4591								
4592	021060	004737	025144		JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
4593	021064	000003			.WORD	T.A2!T.B2!0		;8 MSGS SPECIFIED HERE
4594	021066	104057			ERROR	57		;MSG A0 ERROR AFTER WRITE CHECK CMD
4595	021070	104031			ERROR	31		;MSG B0 ERROR
4596	021072	104060			ERROR	60		;MSG A1 ERROR
4597	021074	104032			ERROR	32		;MSG B1 ERROR
4598	021076	104401	043674		TYPE	MSG72		;ABORTING BALANCE OF TESTS
4599	021102	000137	023416		JMP	ENDRV		
4600								
4601	021106				82\$:			
4602	021106	000240			NOP			
4603	021110	000240			NOP			
4604								
4605	021112	023727	001164	000400	CMP	\$TMP2,#400		;SEE IF WRL AT TRK 1, SECTOR 0
4606	021120	001004			BNE	75\$;BR IF NO
4607	021122	012765	000025	000006	MOV	#21.,RKDA(R5)		;ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
4608	021130	000415			BR	77\$		
4609	021132	023727	001164	001000	75\$:	CMP	\$TMP2,#1000	;SEE IF WRL AT TRK 2,SECTOR 0
4610	021140	001004			BNE	76\$;BR IF NO
4611	021142	012765	000425	000006	MOV	#425,RKDA(R5)		;ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
4612	021150	000405			BR	77\$		
4613								
4614	021152	005337	001164		76\$:	DEC	\$TMP2	;GET SECTOR BEFORE WRL
4615	021156	013765	001164	000006	MOV	\$TMP2,RKDA(R5)		
4616	021164	016537	000006	001166	77\$:	MOV	RKDA(R5),\$TMP3	;FOR ERROR PRINTOUT
4617	021172	032737	000001	001162	BIT	#BIT0,\$TMP1		
4618	021200	001004			BNE	78\$;BR IF WRITING 1'S WHEN WLE OCCURRED
4619	021202	012765	001524	000004	MOV	#DATA0,RKBA(R5)		;WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
4620	021210	000403			BR	79\$		
4621								
4622	021212	012765	001530	000004	78\$:	MOV	#DATA1,RKBA(R5)	;WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
4623	021220	052765	000020	000010	79\$:	BIS	#BAI,RKCS2(R5)	
4624	021226	012765	177400	000002	MOV	#-256.,RKWC(R5)		
4625	021234	017537	000004	001504	MOV	#RKBA(R5),WD2		;EXPECTED WORD FOR TRUERR TYPEOUT
4626	021242	013765	001366	000020	MOV	TOCYL,RKDC(R5)		


```

4627
4628 021250 012765 000031 000000 MOV      #<WRTCHK>,RKCS1(R5)      ;WRITE CHECK CMD
4629 021256 013737 001440 005444 MOV      T50000,TEMP1          ;SETUP TIMEOUT
4630 021264 004737 024422 JSR      PC,FRDY              ;FIND RDY
4631 021270 104015 ERROR    15                    ;NO RDY AFTER WRITE CHECK CMD
4632 021272 004737 025760 JSR      PC,GSTAT            ;GET FRESH STATUS
4633 021276 032737 100000 005406 BIT      #CERR,HCS1
4634 021304 001450 BEQ      84$
4635 021306 032737 040000 005410 BIT      #WCE,HCS2          ;SEE IF WRITE CHECK ERROR
4636 021314 001405 BEQ      83$
4637 021316 016537 000024 001502 MOV      RKDB(R5),WD1       ;ACTUAL WORD FOR PRINTOUT
4638 021324 104135 ERROR    13                    ;WCE AFTER WRITE CMD
4639 021326 000437 BR       84$
4640
4641 021330 104022 83$: ERROR 22          ;CERR AFTER WRITE CHECK CMD
4642
4643 021332 012737 010340 005476 MOV      #<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4644 021340 005037 005500 CLR      E.B0                ;EXPECTED MSG B0
4645 021344 012737 001720 005502 MOV      #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4646 021352 012737 000001 005504 MOV      #1,E.B1            ;MSG ID FOR EXPECTED MSG B1
4647 021360 005037 005506 CLR      E.A2                ;EXPECTED MSG A2
4648 021364 012737 000002 005510 MOV      #2,E.B2            ;MSG ID FOR EXPECTED MSG B2
4649 021372 012737 000003 005514 MOV      #3,E.B3            ;MSG ID FOR EXPECTED MSG B3
4650
4651 021400 004737 025144 JSR      PC,CHKMSG          ;CHECK MSGS A0, B0, A1, B1
4652 021404 000003 .WORD T.A2!T.B2!0          ;& MSGS SPECIFIED HERE
4653 021406 104057 ERROR    57                    ;MSG A0 ERROR AFTER WRITE CHECK CMD
4654 021410 104031 ERROR    31                    ;MSG B0 ERROR
4655 021412 104060 ERROR    60                    ;MSG A1 ERROR
4656 021414 104032 ERROR    32                    ;MSG B1 ERROR
4657 021416 104401 043674 TYPE      MSG72          ;ABORTING BALANCE OF TESTS
4658 021422 000137 023416 JMP      ENDRV
4659
4660 021426 84$:
4661
4662 021426 12$:
4663
4664 ;*****
4665 ;*TEST 17 AC LOW DETECTION PART 2
4666 ;*
4667 ;* THIS TEST VERIFIES THAT WHEN AC POWER IS LOST, THAT WRITING CEASES
4668 ;* AT SECTOR BOUNDRIES & THAT THE BATTERY RETRACT IS FUNCTIONAL.
4669 ;* THERE IS APPROX 4 MS BETWEEN AC LOW ASSERTING AND NED ASSERTING
4670 ;* WHEN THE INTERFACE SHUTS DOWN.
4671 ;*
4672 ;*****
4673 021426 000004 †ST17: SCOPE
4674 021430 012737 000001 001174 MOV      #1,$TIMES          ;DO 1 ITERATION
4675 021436 012706 001100 MOV      #STACK,SP         ;RESTORE STK PTR
4676
4677 021442 004737 026332 JSR      PC,SUBCLR
4678 021446 104024 ERROR    24                    ;CERR AFTER SCLR
4679
4680 021450 104401 037606 TYPE      ,MSG26          ;AC LOW-PART 2
4681 021454 104401 041656 TYPE      ,MSG52          ;CONT-E TO BYPASS TEST
4682 ;OR SPACE TO CONINUE

```

```

4683 021460 004737 030306      JSR    PC,CCSP      ;INPUT CONT-E OR SPACE
4684 021464 000137 023416      JMP    12$         ;RETURN HERE FOR CONT-E
4685                                     ;RETURN HERE FOR SPACE
4686 021470 005737 001542      TST    BSERR       ;TEST FOR ALIGN CARTRIDGE
4687 021474 001402 001542      BEQ    1$         ;BR IF NOT ALIGN CART.
4688 021476 000137 022134      JMP    2$
4689
4690 021502 004737 025760      1$:  JSR    PC,GSTAT
4691 021506 032737 004000 005434  BIT    #D.WRL,HMR2 ;SEE IF WRITE LOCK
4692 021514 001417 004000 005434  BEQ    11$        ;BR IF NO
4693
4694 021516 104401 041757      TYPE   ,MSG53      ;DISABLE WRITE LOCK
4695 021522 104401 040502      TYPE   ,MSG37      ;PRESS SPACE WHEN DONE
4696 021526 004737 030346      JSR    PC,GETSP    ;GET SPACE
4697
4698 021532 004737 025760      JSR    PC,GSTAT
4699 021536 032737 004000 005434  BIT    #D.WRL,HMR2 ;SEE IF STILL WRITE LOCK
4700 021544 001403 004000 005434  BEQ    11$        ;BR IF NO
4701 021546 104110 004000 005434  ERROR  110        ;WRITE LOCK NOT DISABLED
4702 021550 000137 023416      JMP    12$        ;EXIT TEST
4703
4704 021554 005037 001366      11$: CLR    TOCYL     ;SETUP
4705 021560 005037 001402      CLR    CALADD     ;FOR
4706 021564 005037 001510      CLR    HEAD       ;FILL HEADER TABLE
4707 021570 005037 001516      CLR    FORMAT     ;ROUTINE
4708
4709 021574 004737 027314      13$: JSR    PC,FHDTAB ;BUILD STD 22 SECTOR HEADER TABLE
4710
4711 021600 012765 001554 000004  MOV    #HDTAB,RKBA(R5)
4712 021606 012765 177676 000002  MOV    #-66.,RKWC(R5)
4713 021614 000337 001510      SWAB   HEAD
4714 021620 013765 001510 000006  MOV    HEAD,RKDA(R5) ;HEAD ADDR
4715 021626 000337 001510      SWAB   HEAD
4716
4717 021632 012765 000027 000000  MOV    #<WRHEAD>,RKCS1(R5) ;WRITE HEADER CMD
4718 021640 013737 001440 005444  MOV    T5000,TEMP1 ;SETUP TIMEOUT
4719 021646 004737 024422      JSR    PC,FROY     ;FIND ROY
4720 021652 104200 004422      ERROR  200        ;NO ROY AFTER WRITE HEADER CMD
4721 021654 004737 025760      JSR    PC,GSTAT    ;GET FRESH STATUS
4722 021660 032737 100000 005406  BIT    #CERR,HCS1
4723 021666 001401 004422      BEQ    64$
4724 021670 104201 004422      ERROR  201        ;CERR AFTER WRITE HEADER CMD
4725 021672                                     64$:
4726
4727 021672 005237 001510      INC    HEAD
4728 021676 023727 001510 000003  CMP    HEAD,#3    ;SEE IF ALL HEADS DONE
4729 021704 001333 001510 000003  BNE   13$        ;BR IF NO
4730 021706 005037 001366      CLR    TOCYL
4731
4732 021712 004737 026332      16$: JSR    PC,SUBCLR
4733 021716 104024 004024      ERROR  24         ;CERR AFTER SCLR
4734
4735 021720 012765 001530 000004  MOV    #DATA1,RKBA(R5) ;RETURN HERE FOR SPACE
4736 021726 052765 000020 000010  BIS    #BA1,RKCS2(R5) ;WRITE INITIAL BACKGROUND OF 1'S
4737 021734 012765 137000 000002  MOV    #-66.*256.,RKWC(R5) ;TRACK 0,1,2 ALL SECTORS
4738 021742 013765 001366 000020  MOV    TOCYL,RKDC(R5)

```

4739	021750	012765	000023	000000		MOV	#WRDATA,RKCS1(R5)	;WRITE DATA CMD
4740	021756	013737	001440	005444		MOV	T50000,TEMP1	;SETUP TIMEOUT
4741	021764	004737	024422			JSR	PC,FRDY	;FIND RDY
4742	021770	104011				ERROR	11	;NO RDY AFTER WRITE DATA CMD
4743	021772	004737	025760			JSR	PC,GSTAT	;GET FRESH STATUS
4744	021776	032737	100000	005406		BIT	#CERR,HCS1	
4745	022004	001453				BEQ	25	
4746	022006	032737	000200	005422		BIT	#BSE,HER	;SEE IF BAD SECTOR
4747	022014	001034				BNE	175	
4748								
4749	022016	012737	010340	005476		MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4750	022024	005037	005500			CLR	E.B0	;EXPECTED MSG B0
4751	022030	012737	001720	005502		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1	;EXPECTED A1
4752	022036	012737	000001	005504		MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4753	022044	005037	005506			CLR	E.A2	;EXPECTED MSG A2
4754	022050	012737	000002	005510		MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4755	022056	012737	000003	005514		MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4756								
4757	022064	004737	025144			JSR	PC,CHKMSG	;CHECK MSGS A0 B0 A1 B1
4758	022070	000003				.WORD	T.A2!T.B2!0	;8 MSGS SPECIFIED HERE
4759	022072	104052				ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
4760	022074	104023				ERROR	23	;MSG B0 ERROR
4761	022076	104053				ERROR	53	;MSG A1 ERROR
4762	022100	104025				ERROR	25	;MSG B1 ERROR
4763	022102	000137	023416			JMP	125	
4764								
4765	022106	005237	001366		175:	INC	TOCYL	
4766	022112	023727	001366	000012		CMP	TOCYL,#10.	;TRIED 10 CYLINDERS?
4767	022120	001003				BNE	185	;BR IF NO
4768	022122	104062				ERROR	62	;CANNOT WRITE ON 10 CONSEC CYL
4769	022124	000137	023416			JMP	125	
4770								
4771	022130	000137	021712		185:	JMP	165	
4772								
4773	022134	004737	026332		25:	JSR	PC,SUBCLR	
4774	022140	104024				ERROR	24	;CERR AFTER SCLR
4775								
4776	022142	104401	041430			TYPE	,MSG49	;TURN OFF AC
4777	022146	104401	042423			TYPE	,MSG57	;VERIFY BATTERY RETRACT FUNCTIONAL
4778								
4779	022152	005737	001542			TST	BSERR	;SEE IF ALIGN CART USED
4780	022156	001405				BEQ	155	;BR IF NO
4781	022160	104401	040502			TYPE	,MSG37	;PRESS SPACE WHEN DONE
4782	022164	004737	030346			JSR	PC,GETSP	;GET SPACE
4783	022170	000526				BR	85	;SKIP ALL WRITING
4784								
4785	022172	013737	001440	001160	155:	MOV	T50000,\$TMP0	
4786	022200	005037	001162			CLR	\$TMP1	;BIT0=0;WRITE 0'S; BIT0=1:WRITE 1'S
4787								
4788	022204	004737	026332		45:	JSR	PC,SUBCLR	
4789	022210	104024				ERROR	24	;CERR AFTER SCLR
4790								
4791	022212	032737	000001	001162		BIT	#BIT0,\$TMP1	
4792	022220	001004				BNE	55	;BR IF WRITING 1'S
4793	022222	012765	001524	000004		MOV	#DATA0,RKBA(R5)	;SETUP ALL 0'S
4794	022230	000403				BR	65	

```

4795
4796 022232 012765 001530 000004 5$: MOV #DATA1,RKBA(R5) ;SETUP ALL 1'S
4797 022240 052765 000020 000010 6$: BIS #BAI,RKCS2(R5)
4798 022246 012765 140400 000002 MOV #63.*256.,RKWC(R5) ;TRACK 0,1,2 63 SECTORS
4799 022254 005065 000006 CLR RKDA(R5) ;BEGIN AT TRK AND SECTOR 0
4800 022260 013765 001366 000020 MOV TOCYL,RKDC(R5)
4801 022266 012765 000023 000000 MOV #WRDATA,RKCSI(R5)
4802 022274 013737 001440 005444 MOV T5000,TEMP1
4803 022302 004737 024422 JSR PC,FRDY ;FIND CONTR RDY
4804 022306 104011 ERROR 11 ;CONTR NOT RDY
4805
4806 022310 004737 025760 JSR PC,GSTAT
4807 022314 032737 100000 005406 BIT #CERR,HCS1
4808 022322 001017 BNE 3$
4809 022324 005337 001160 DEC $TMP0
4810 022330 001011 BNE 7$
4811 022332 104146 ERROR 146 ;CERR NOT SET BY TIMEOUT
4812 022334 104401 041503 TYPE ,MSG50 ;TURN AC BACK ON
4813 022340 104401 043506 TYPE ,MSG69 ;DEPRESS SPACE AFTER 'READY' LIGHT ON
4814 022344 004737 030346 JSR PC,GETSP ;GET SPACE
4815 022350 000137 023416 JMP 12$ ;EXIT TEST
4816
4817 022354 005237 001162 7$: INC $TMP1
4818 022360 000711 BR 4$ ;GO WRITE OPPOSITE DATA
4819
4820 022362 032737 000100 005436 3$: BIT #D.ACLO,HMR3
4821 022370 001001 BNE 10$
4822 022372 104147 ERROR 147 ;AC LOW NOT SET
4823
4824 022374 005237 005370 10$: INC UNLD ;FOR VALID HALT
4825 022400 012737 070140 005476 MOV #<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4826 022406 012737 010300 005500 MOV #<D.SPLS!D.ACLO!D.FLT>,E.B0
4827 022414 012737 044720 005502 MOV #<D.UNLD!D.REV!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4828 022422 012737 000001 005504 MOV #1,E.B1
4829
4830 022430 004737 025144 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4831 022434 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4832 022436 104035 ERROR 35 ;MSG A0 ERROR AFTER AC SWITCH OFF FROM HEADS LOADED
4833 022440 104203 ERROR 203 ;MSG B0 ERROR
4834 022442 104036 ERROR 36 ;MSG A1 ERROR
4835 022444 104037 ERROR 37 ;MSG B1 ERROR
4836
4837 022446 013737 005416 001164 8$: MOV HDA,$TMP2 ;SAVE RKDA
4838 022454 013737 005416 001166 MOV HDA,$TMP3 ;SAVE FOR TYPEOUT
4839
4840 022462 104401 041503 TYPE ,MSG50 ;TURN AC BACK ON
4841 022466 104401 043506 TYPE ,MSG69 ;DEPRESS SPACE AFTER 'READY' LIGHT ON
4842 022472 004737 030346 JSR PC,GETSP ;GET SPACE
4843
4844 022476 004737 026332 JSR PC,SUBCLR
4845 022502 104024 ERROR 24 ;CERR AFTER SCLR
4846
4847 022504 032737 000100 005436 BIT #D.ACLO,HMR3
4848 022512 001401 BEQ 9$
4849 022514 104152 ERROR 152 ;ACLO NOT RESET AFTER POWER UP
4850

```

4851	022516	005037	005370		95:	CLR	UNLD		;FOR VALID HALT
4852	022522	012765	100000	000000		MOV	#CCLR,RKCS1(R5)		
4853	022530	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)		;DRIVE #
4854	022536	012765	000003	000000		MOV	#PACK,RKCS1(R5)		;PACK CMD
4855	022544	013737	001426	005444		MOV	T10,TEMP1		
4856	022552	004737	024422			JSR	PC,FRDY		;FIND CONTR RDY
4857	022556	104116				ERROR	116		;CONTR NOT RDY
4858									
4859	022560	032737	000100	005434		BIT	#D.VV,HMR2		
4860	022566	001001				BNE	65\$		
4861	022570	104027				ERROR	27		;VOLUME VALID NOT SET AFTER PACK CMD
4862	022572				65\$:				
4863	022572	005737	001542			TST	BSERR		;SEE IF ALIGN CART USED
4864	022576	001402				BEQ	14\$;BR IF NO
4865	022600	000137	023416			JMP	12\$;ELSE EXIT TEST
4866									
4867	022604				14\$:				
4868									
4869	022604	004737	026332			JSR	PC,SUBCLR		
4870	022610	104024				ERROR	24		;CERR AFTER SCLR
4871									
4872	022612	005737	001164			TST	\$TMP2		;SEE IF TRK/SECTOR 0
4873	022616	001414				BEQ	73\$;REPEAT,NO NEW DATA XFER TOOK PLACE
4874	022620	023727	001164	001023		CMP	\$TMP2,#1023		;SEE IF TRK 2,SECTOR 19
4875	022626	001410				BEQ	73\$;REPEAT,NO OLD DATA TO CHECK AGAINST
4876	022630	032737	000001	001162		BIT	#BIT0,\$TMP1		
4877	022636	001006				BNE	66\$;BR IF WRITING 1'S WHEN WLE OCCURRED
4878	022640	012765	001530	000004		MOV	#DATA1,RKBA(R5)		;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4879	022646	000405				BR	67\$		
4880									
4881	022650	000137	022134		73\$:	JMP	2\$		
4882									
4883	022654	012765	001524	000004	66\$:	MOV	#DATA0,RKBA(R5)		;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4884	022662	052765	000020	000010	67\$:	BIS	#BAI,RKCS2(R5)		
4885	022670	012765	177400	000002		MOV	#-256,RKWC(R5)		
4886	022676	013765	001166	000006		MOV	\$TMP3,RKDA(R5)		;REFRESH RKDA
4887	022704	017537	000004	001504		MOV	\$RKBA(R5),WD2		;EXPECTED WORD FOR TRUERR TYPEOUT
4888	022712	013765	001366	000020		MOV	TOCYL,RKDC(R5)		
4889									
4890	022720	012765	000031	000000		MOV	#(WRTCHK),RKCS1(R5)		;WRITE CHECK CMD
4891	022726	013737	001440	005444		MOV	T5000,TEMP1		;SETUP TIMEOUT
4892	022734	004737	024422			JSR	PC,FRDY		;FIND RDY
4893	022740	104015				ERROR	15		;NO RDY AFTER WRITE CHECK CMD
4894	022742	004737	025760			JSR	PC,GSTAT		;GET FRESH STATUS
4895	022746	032737	100000	005406		BIT	#CERR,HCS1		
4896	022754	001450				BEQ	75\$		
4897	022756	032737	040000	005410		BIT	#WCE,HCS2		;SEE IF WRITE CHECK ERROR
4898	022764	001405				BEQ	74\$		
4899	022766	016537	000024	001502		MOV	RKDB(R5),WD1		;ACTUAL WORD FOR PRINTOUT
4900	022774	104136				ERROR	136		;WCE AFTER WRITE CMD
4901	022776	000437				BR	75\$		
4902									
4903	023000	104022			74\$:	ERROR	22		;CERR AFTER WRITE CHECK CMD
4904									
4905	023002	012737	010340	005476		MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0		;EXPECTED MSG A0
4906	023010	005037	005500			CLR	E.B0		;EXPECTED MSG B0

```

4907 023014 012737 001720 005502      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4908 023022 012737 000001 005504      MOV      #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4909 023030 005037 005506      CLR      E.A2 ;EXPECTED MSG A2
4910 023034 012737 000002 005510      MOV      #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4911 023042 012737 000003 005514      MOV      #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4912
4913 023050 004737 025144      JSR      PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4914 023054 000003      .WORD   T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4915 023056 104057      ERROR   57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
4916 023060 104031      ERROR   31 ;MSG B0 ERROR
4917 023062 104060      ERROR   60 ;MSG A1 ERROR
4918 023064 104032      ERROR   32 ;MSG B1 ERROR
4919 023066 104401 043674      TYPE    ,MSG72 ;ABORTING BALANCE OF TESTS
4920 023072 000137 023416      JMP      ENDRV
4921
4922 023076      75$:
4923 023076 000240      NOP
4924 023100 000240      NOP
4925
4926 023102 023727 001164 000400      CMP      $TMP2,#400 ;SEE IF WRL AT TRK 1, SECTOR 0
4927 023110 001004      BNE     68$ ;BR IF NO
4928 023112 012765 000025 000006      MOV      #21,RKDA(R5) ;ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
4929 023120 000415      BR      70$
4930 023122 023727 001164 001000 68$:      CMP      $TMP2,#1000 ;SEE IF WRL AT TRK 2,SECTOR 0
4931 023130 001004      BNE     69$ ;BR IF NO
4932 023132 012765 000425 000006      MOV      #425,RKDA(R5) ;ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
4933 023140 000405      BR      70$
4934
4935 023142 005337 001164      69$:      DEC      $TMP2 ;GET SECTOR BEFORE WRL
4936 023146 013765 001164 000006      MOV      $TMP2,RKDA(R5)
4937 023154 016537 000006 001166 70$:      MOV      RKDA(R5),$TMP3 ;FOR ERROR PRINTOUT
4938 023162 032737 000001 001162      BIT      #BIT0,$TMP1
4939 023170 001004      BNE     71$ ;BR IF WRITING 1'S WHEN WLE OCCURRED
4940 023172 012765 001524 000004      MOV      #DATA0,RKBA(R5) ;WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
4941 023200 000403      BR      72$
4942
4943 023202 012765 001530 000004 71$:      MOV      #DATA1,RKBA(R5) ;WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
4944 023210 052765 000020 000010 72$:      BIS      #BA1,RKCS2(R5)
4945 023216 012765 177400 000002      MOV      #-256,RKWC(R5)
4946 023224 017537 000004 001504      MOV      #RKBA(R5),WD2 ;EXPECTED WORD FOR TRUERR TYPEOUT
4947 023232 013765 001366 000020      MOV      TOCYL,RKDC(R5)
4948
4949 023240 012765 000031 000000      MOV      #<WATCH>,RKCS1(R5) ;WRITE CHECK CMD
4950 023246 013737 001440 005444      MOV      T5000,TEMP1 ;SETUP TIMEOUT
4951 023254 004737 024422      JSR      PC,FRDY ;FIND RDY
4952 023260 104015      ERROR   15 ;NO RDY AFTER WRITE CHECK CMD
4953 023262 004737 025760      JSR      PC,GSTAT ;GET FRESH STATUS
4954 023266 032737 100000 005406      BIT      #CERR,HCS1
4955 023274 001450      BEQ     77$
4956 023276 032737 040000 005410      BIT      #WCE,HCS2 ;SEE IF WRITE CHECK ERROR
4957 023304 001405      BEQ     76$
4958 023306 016537 000024 001502      MOV      RKDB(R5),WD1 ;ACTUAL WORD FOR PRINTOUT
4959 023314 104137      ERROR   137 ;WCE AFTER WRITE CMD
4960 023316 000437      BR      77$
4961
4962 023320 104022      76$:      ERROR   22 ;CERR AFTER WRITE CHECK CMD

```


E08

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 94
T21 MULTIPLE DRIVE DETECTION TEST

SEQ 0094

OR A CONT-E TO EXIT THE TEST

THE PROGRAM VERIFIES THAT MULTIPLE DRIVE SELECT & DRIVE UNSAFE
BOTH SET & THAT THE DRIVE UNLOADS

THE OPERATOR IS ASKED TO VERIFY THAT HEADS UNLOAD FROM BOTH DRIVES

```

5019
5020
5021
5022
5023
5024
5025
5026
5027 023452 000004
5028 023454 012737 000001 001174
5029 023462 012706 001100
5030
5031 023466 005737 001216
5032 023472 001402
5033 023474 000137 023730
5034
5035 023500 023727 005526 000001 1$:
5036 023506 001004
5037 023510 104401 043034
5038 023514 000137 023730
5039
5040 023520 104401 037677 2$:
5041 023524 104401 041656
5042 023530 004737 030306
5043 023534 000137 023730
5044
5045 023540 104401 042471
5046 023544 104401 040502
5047 023550 004737 030346
5048
5049 023554 004737 026332 3$:
5050 023560 104024
5051
5052 023562 104401 042553
5053 023566 104401 040502
5054 023572 004737 030346
5055
5056 023576 005000
5057
5058 023600 012765 100000 000000 6$:
5059 023606 010065 000010
5060 023612 012765 000001 000000
5061 023620 012737 141520 005444
5062 023626 004737 025112
5063
5064 023632 032765 001000 000010
5065 023640 001006
5066 023642 005200
5067 023644 020027 000010
5068 023650 001353
5069 023652 104176
5070 023654 000411
5071
5072 023656 104401 042766 7$:
5073 023662 010046
5074

```

```

TST21: SCOPE
MOV #1,$TIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STACK PTR
TST $PASS
BEQ 1$ ;DO TEST ONLY IN 1ST PASS
JMP 11$ ;ELSE EXIT TEST
1$: CMP DRIVS,#1
BNE 2$ ;BR IF MORE THAN 1 DRIVE PRESENT
TYPE ,MSG62 ;BYPASS TEST, ONLY 1 DRIVE PRESENT
JMP 11$ ;ELSE EXIT TEST
2$: TYPE ,MSG28 ;MULT DRV DETECTION TEST
TYPE ,MSG52 ;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
JSR PC,CCSP ;INPUT CONT-E OR SPACE
JMP 11$ ;RETURN HERE FOR CONT-E
TYPE ,MSG58 ;LOAD HEADS ON ALL DRIVES
TYPE ,MSG37 ;PRESS SPACE WHEN DONE
JSR PC,GETSP ;GET SPACE
3$: JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
TYPE ,MSG59 ;INSERT SAME UNIT SEL PLUG # IN 2 DRIVES
TYPE ,MSG37 ;DEPRESS SPACE BAR WHEN DONE
JSR PC,GETSP ;GET SPACE
CLR RO ;DRIVE # COUNTER
6$: MOV #CCLR,RKCS1(R5)
MOV RO,RKCS2(R5) ;DRIVE #
MOV #SELDIV,RKCS1(R5) ;SELECT DRIVE CMD TO GET STATUS
MOV #50000.,TEMP1
JSR PC,DLY ;REQ 2 MS DELAY BEFORE MDS DETECTED
BIT #MDS,RKCS2(R5) ;SEE IF THAT DRIVE HAS MDS
BNE 7$ ;BR IF YES
RO ;ELSE TRY ANOTHER DRIVE
INC RO
CMP RO,#8. ;SEE IF ALL DRIVES TESTED
BNE 6$ ;BR IF NO
ERROR 176 ;CANNOT FIND MDS
BR 10$ ;TRY AGAIN
7$: TYPE ,MSG61 ;MULT DRIVES FOUND ON DRIVE #
MOV RO,-(SP) ;SAVE RO FOR TYPEOUT
;TYPE UNIT NO

```


F08

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 95
21 MULTIPLE DRIVE DETECTION TEST

SEQ 0095

5075	023664	104403	
5076	023666	001	
5077	023667	000	
5078	023670	104401	043446
5079	023674	005237	005370
5080	023700	104401	042656
5081	023704	104401	043130
5082	023710	004737	030306
5083	023714	000137	023730
5084	023720	005037	005370
5085	023724	000137	023554
5086	023730	005037	005370
5087			
5088	023734	004737	026332
5089	023740	104024	

	TYP05	
	.BYTE	1
	.BYTE	0
	TYPE	,MSG68
	INC	UNLD
10\$:	TYPE	,MSG60
	TYPE	,MSG63
	JSR	^C,CCSP
	JMP	11\$
	CLR	UNLD
11\$:	JMP	3\$
	CLR	UNLD
	JSR	PC,SUBCLR
	ERROR	24

```

;;GO TYPE--OCTAL ASCII
;;TYPE 1 DIGIT(S)
;;SUPPRESS LEADING ZEROS
;VERIFY BOTH DRIVES UNLOADED
;FOR VALID HALT
;INSERT CORRECT UNIT SEL PLUG & LOAD HEADS
;DEPRESS CONT-E OR SPACE BAR WHEN DONE
;INPUT CONT-E OR SPACE
;RETURN HERE FOR CONTROL-E (EXIT)
;RETURN HERE FOR SPACE (DO AGAIN)

```

;CERR AFTER SCLR

```

5090
5091
5092
5093
5094
5095
5096
5097
5098 023742
5099 023742 000004
5100 023744 005037 001102
5101 023750 005037 001174
5102 023754 005237 001216
5103 023760 042737 100000 001216
5104 023766 005327
5105 023770 000001
5106 023772 003022
5107 023774 012737
5108 023776 000001
5109 024000 023770
5110 024002 104401 024047
5111 024006 013746 001216
5112 024012 104405
5113 024014 104401 024044
5114 024020 013700 000042
5115 024024 001405
5116 024026 000005
5117 024030 004710
5118 024032 000240
5119 024034 000240
5120 024036 000240
5121 024040
5122 024040 000137
5123 024042 010656
5124 024044 377 377 000
5125 024047 015 042412 042116
5126 024054 050040 051501 020123
5127 024062 000043

```

```

.SBTTL END OF PASS ROUTINE

*****
; INCREMENT THE PASS NUMBER ($PASS)
; TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
; IF THERES A MONITOR GO TO IT
; IF THERE ISN'T JUMP TO ST5

$EOP:
SCOPE
CLR $STNM ; ZERO THE TEST NUMBER
CLR $TIMES ; ZERO THE NUMBER OF ITERATIONS
INC $PASS ; INCREMENT THE PASS NUMBER
BIC #100000,$PASS ; DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ; LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ; YES
MOV (PC)+,@(PC)+ ; RESTORE COUNTER
$ENDCT: .WORD 1
TYPE $SENDMG ; TYPE "END PASS #"
MOV $PASS,-(SP) ; SAVE $PASS FOR TYPEOUT
TYPDS ; GO TYPE--DECIMAL ASCII WITH SIGN
TYPE $ENULL ; TYPE A NULL CHARACTER
$GET42: MOV @#42,RO ; GET MONITOR ADDRESS
BEQ $DOAGN ; BRANCH IF NO MONITOR
RESET ; CLEAR THE WORLD
$ENDAD: JSR PC,(RO) ; GO TO MONITOR
NOP ; SAVE ROOM
NOP ; FOR
NOP ; ACT11
$DOAGN: JMP @(PC)+ ; RETURN
$RTNAD: .WORD ST5
$ENULL: .BYTE -1,-1,0 ; NULL CHARACTER STRING
$SENDMG: .ASCIZ '<15><12>/END PASS #/'

```

```

S128      .SBTTL SUBROUTINES
S129
S130      ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
S131      ;
S132
S133      024064 012700 005516 CLRFLG: MOV      #DDUMP, R0
S134      024070 012701 177757      MOV      #-17., R1
S135      024074 005020      1$:      CLR      (R0)+
S136      024076 005201      INC      R1
S137      024100 001375      BNE     1$
S138      024102 000207      RTS     PC
S139
S140
S141      ;TYPE PROGRAM ID IF FTITLE=0
S142      ;
S143
S144      024104 005737 001360 TITLE: TST     FTITLE
S145      024110 001024      BNE     1$
S146      024112 005237 001360      INC     FTITLE
S147      024116 104401 035570      TYPE   MSG1      ;PROGRAM ID
S148      .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
S149      024122 005737 000042      TST     2#42      ;;ARE WE RUNNING UNDER XXDP/ACT?
S150      024126 001012      BNE     64$      ;;BRANCH IF YES
S151      024130 123727 001230 000001      CMPB   $ENV, #1      ;;ARE WE RUNNING UNDER APT?
S152      024136 001406      BEQ     64$      ;;BRANCH IF YES
S153      024140 023727 001140 000176      CMP    SWR, #SWREG      ;;SOFTWARE SWITCH REG SELECTED?
S154      024146 001005      BNE     65$      ;;BRANCH IF NO
S155      024150 104406      GTSWR      ;;GET SOFT-SWR SETTINGS
S156      024152 000403      BR     65$
S157      024154 112737 000001 001134 64$:  MOVB   #1, $AUTOB      ;;SET AUTO-MODE INDICATOR
S158      024162      65$:
S159      024162 000207      1$:  RTS     PC
S160
S161
S162      ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
S163      ;DRIVS, DRIVO-DRIV7 REGISTERS APPROPRIATELY
S164      ;
S165
S166      024164 104411      GDRVS: RDLIN
S167      024166 012600      MOV     (SP)+, R0      ;GET STARTING ADDR OF ASCII STRING
S168      024170 012701 177770      MOV     #-8., R1      ;SET UP COUNT
S169      024174 112002      1$:  MOVB   (R0)+, R2      ;GET ASCII CHAR
S170      024176 042702 177400      BIC     #177400, R2      ;MASK HI BYTE
S171      024202 012703 005530      MOV     #DRIVO, R3      ;DRIVE FLAG ADDR
S172      024206 012704 000060      MOV     #60, R4
S173
S174      024212 020402      2$:  CMP     R4, R2      ;WAS TYPED CHAR 0 THRU 7?
S175      024214 001415      BEQ     3$      ;BRANCH IF YES
S176      024216 005723      TST     (R3)+      ;NO, INCREMENT DR FLAG ADDR
S177      024220 005204      INC     R4
S178      024222 020427 000070      CMP     R4, #70
S179      024226 001371      BNE     2$      ;S/B 0-7 OR TERMINATOR
S180      024230 005702      TST     R2
S181      024232 001022      BNE     4$
S182      024234 020127 177770      CMP     R1, #-8.
S183      024240 001426      BEQ     5$      ;DEFAULT ALL DRIVES

```

```

5184 024242 005037 005556 7$: CLR      SIZFLG      ;BYPASS TEST 1 (SIZING)
5185 024246 000207          RTS      PC           ;FOUND TERMINATOR, EXIT
5186
5187 024250 005213          3$: INC      DR3        ;SET UP FLAG FOR THE DRIVE
5188 024252 005237 005526      INC      DRVS        ;INCREMENT TOTAL # DRIVES TO BE TESTED
5189 024256 112002          MOV      (R0)+,R2    ;GET NEXT ASCII CHAR.
5190 024260 042702 177400      BIC      #177400,R2  ;MASK
5191 024264 022702 000054      CMP      #54,R2     ;IS IT A COMMA?
5192 024270 001407          BEQ      5$         ;YES, GO TO NEXT WORD.
5193 024272 005702          TST      R2         ;NO, IS IT A TERMINATOR?
5194 024274 001001          BNE      4$         ;IF NOT, SOMETHING WRONG.
5195 024276 000761          BR       7$         ;FOUND TERMINATOR, EXIT
5196
5197 024300 104401 044241 4$: TYPE     EM1        ;ONLY 0-7 ALLOWED.
5198 024304 000137 010064      JMP      PRGSRT     ;START ALL OVER
5199
5200 024310 005201          5$: INC      R1        ;S/B NO MORE THAN 8 DIFF
5201 024312 001330          BNE      1$         ;DRIVES TYPED IN.
5202 024314 000771          BR       4$         ;IF NORE, HAVE ERROR.
5203
5204 024316 005237 005556 6$: INC      SIZFLG    ;DO TEST 1 (SIZING)
5205 024322 000207          RTS      PC         ;EXIT.
5206
5207
5208 ;ROUTINE TO INPUT RKBAS OR DEFAULT.
5209 ;
5210
5211 024324 104412  GBA: RDOCT
5212 024326 012600      MOV      (SP)+,R0   ;GET LOW ORDER FROM STACK
5213 024330 005700      TST      R0
5214 024332 001403          BEQ      1$         ;BRANCH IF DEFAULT.
5215 024334 010037 001264      MOV      R0,$BASE
5216 024340 000207          RTS      PC
5217 024342 012737 177440 001264 1$: MOV      #177440,$BASE ;DEFAULT VALUE
5218 024350 000207          RTS      PC
5219
5220 ;ROUTINE TO INPUT RKVEC OR DEFAULT
5221 ;
5222 ;
5223
5224 024352 104412  GINT: RDOCT
5225 024354 012600      MOV      (SP)+,R0   ;GET LOW ORDER FROM STACK
5226 024356 005700      TST      R0
5227 024360 001405          BEQ      1$         ;BRANCH IF DEFAULT
5228 024362 010037 001334      MOV      R0,RKVEC
5229 024366 004737 024404 2$: JSR      PC,SETINT
5230 024372 000207          RTS      PC
5231 024374 012737 000210 001334 1$: MOV      #210,RKVEC  ;DEFAULT VALUE
5232 024402 000771          BR       2$
5233
5234 ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
5235 ;
5236 ;
5237
5238 024404 013700 001334  SETINT: MOV      RKVEC,R0
5239 024410 012720 031070      MOV      #INTER,(R0)+ ;INTER ADDR TO RKVEC

```

JOB

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 99
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0099

```

5240 024414 013710 001336          MOV   RKPRI,(R0)      ;PRS TO RKVEC+2
5241 024420 000207          RTS   PC
5242
5243
5244
5245
5246
5247          ; ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
5248          ; ENTER WITH A COUNT IN TEMP1
5249          ; RETURN IF RDY NOT PRESENT (ERROR CONDITION)
5250          ; RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
5251          ; STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
5252 024422 032765 000200 000000 FRDY:  BIT   #RDY,RKCS1(R5)
5253 024430 001010          BNE   IS
5254 024432 005337 005444          DEC   TEMP1
5255 024436 001371          BNE   FRDY
5256 024440 004737 024556          JSR   PC,HOLD        ;STORE ALL RK611 REGS IN HOLDING REGS.
5257 024444 004737 025676          JSR   PC,CKCERR     ;CHECK FOR SPECIAL CERR
5258 024450 000207          RTS   PC            ;NO RDY, EXIT
5259 024452 062716 000002          IS:   ADD   #2,(SP)  ;SKIP OVER ERROR
5260 024456 004737 024556          JSR   PC,HOLD
5261 024462 004737 025676          JSR   PC,CKCERR     ;CHECK FOR SPECIAL CERR
5262 024466 000207          RTS   PC
5263
5264          ; ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
5265
5266 024470 032765 000200 000000 FRDY1: BIT   #RDY,RKCS1(R5)
5267 024476 001014          BNE   IS
5268 024500 005337 005444          DEC   TEMP1
5269 024504 001371          BNE   FRDY1
5270 024506 016537 000034 005434          MOV   RKMR2(R5),HMR2
5271 024514 016537 000036 005436          MOV   RKMR3(R5),HMR3
5272 024522 004737 025676          JSR   PC,CKCERR     ;CHECK FOR SPECIAL CERR CONDITIONS
5273 024526 000207          RTS   PC            ;NO RDY, EXIT
5274 024530 062716 000002          IS:   ADD   #2,(SP)  ;SKIP OVER ERROR
5275 024534 016537 000034 005434          MOV   RKMR2(R5),HMR2
5276 024542 016537 000036 005436          MOV   RKMR3(R5),HMR3
5277 024550 004737 025676          JSR   PC,CKCERR     ;CHECK FOR SPECIAL CERR CONDITIONS
5278 024554 000207          RTS   PC
5279
5280          ; STORE ALL RK611 REGISTERS IN HOLDING REGS
5281
5282
5283
5284 024556 016537 000000 005406 HOLD:  MOV   RKCS1(R5),HCS1
5285 024564 016537 000010 005410          MOV   RKCS2(R5),HCS2
5286 024572 016537 000002 005412          MOV   RKWC(R5),HWC
5287 024600 016537 000004 005414          MOV   RKBA(R5),HBA
5288 024606 016537 000006 005416          MOV   RKDA(R5),HDA
5289 024614 016537 000012 005420          MOV   RKDS(R5),HDS
5290 024622 016537 000014 005422          MOV   RKER(R5),HER
5291 024630 016537 000016 005424          MOV   RKASOF(R5),HASOF
5292 024636 016537 000020 005426          MOV   RKDC(R5),HDC
5293 024644 016537 000026 005432          MOV   RKMR1(R5),HMR1
5294 024652 016537 000034 005434          MOV   RKMR2(R5),HMR2
5295 024660 016537 000036 005436          MOV   RKMR3(R5),HMR3

```

K08

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05MACY11 27(1006) 31-JAN-77 18:30 PAGE 100
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 010C

```

5296 024666 016537 000030 005440      MOV    RKECPS(R5),HPOS
5297 024674 016537 000032 005442      MOV    RKECPT(R5),HPAT
5298 024702 000207                    RTS    PC
5299
5300
5301      ; ROUTINE TO CHECK FOR CORRECT ATTN
5302      ; RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
5303      ; RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
5304
5305 024704 010446      ;STATN: MOV    R4, -(SP)          ; SAV R4
5306 024706 013704 001222      MOV    $UNIT, R4
5307 024712 136437 005376 005425      BITB  ATTN(R4), HASOF+1
5308 024720 001404      BEQ    1$          ; BRANCH IF ATTN NOT PRESENT
5309 024722 012604      MOV    (SP)+, R4      ; RESTOR R4
5310 024724 062716 000002      ADD    #2, (SP)      ; INCR RET ADDR TO JUMP OVER ERROR.
5311 024730 000207      RTS    PC
5312 024732 012604      1$:  MOV    (SP)+, R4      ; RESTOR R4
5313 024734 000207      RTS    PC
5314
5315
5316      ; ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
5317      ; ENTER WITH TIME IN SECONDS IN TEMP2
5318      ; RETURN IF NO ATTN (ERROR CONDITION)
5319      ; RETURN +2 IF ATTN FOUND
5320      ; STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
5321
5322
5323
5324 024736 010446      FATT1: MOV    R4, -(SP)          ; SAV R4
5325 024740 012737 177777 005444      3$:  MOV    #-1, TEMP1
5326 024746 013704 001222      MOV    $UNIT, R4
5327 024752 136465 005376 000017      1$:  BITB  ATTN(R4), RKASOF+1(R5) ; FIND CORRECT ATTN
5328 024760 001014      BNE    2$
5329 024762 005337 005444      DEC    TEMP1
5330 024766 001371      BNE    1$
5331 024770 005337 005446      DEC    TEMP2
5332 024774 001361      BNE    3$
5333 024776 005065 000026      CLR    RKMR1(R5)      ; SELECT WORD 0
5334 025002 004737 025760      JSR    PC, GSTAT      ; GET LATEST STATUS
5335 025006 012604      MOV    (SP)+, R4      ; RESTOR R4
5336 025010 000207      RTS    PC
5337
5338 025012 005065 000026      2$:  CLR    RKMR1(R5)
5339 025016 004737 025760      JSR    PC, GSTAT      ; GET STATUS AFTER ATTN SEEN
5340 025022 012604      MOV    (SP)+, R4      ; RESTOR R4
5341 025024 062716 000002      ADD    #2, (SP)      ; SKIP OVER ERROR
5342 025030 000207      RTS    PC
5343
5344
5345      ; ROUTINE TO FIND ATTN WITHIN 1 SEC
5346      ; ENTER WITH COUNT IN TEMP1
5347      ; RETURN IF NO ATTN (ERROR)
5348      ; RETURN +2 IF ATTN FOUND
5349      ; STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
5350
5351

```

L08

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 101
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0101

5352	025032	010446		
5353	025034	013704	001222	
5354	025040	136465	005376	000017
5355	025046	001011		
5356	025050	005337	005444	
5357	025054	001367		
5358	025056	005065	000026	
5359	025062	004737	025760	
5360	025066	012604		
5361	025070	000207		
5362	025072	005065	000026	
5363	025076	004737	025760	
5364	025102	012604		
5365	025104	062716	000002	
5366	025110	000207		
5367				
5368				
5369				
5370				
5371				
5372	025112	005737	005444	
5373	025116	001403		
5374	025120	005337	005444	
5375	025124	000772		
5376	025126	000207		
5377				
5378				
5379				
5380				
5381	025130	104401	037244	
5382	025134	010046		
5383				
5384	025136	104403		
5385	025140	001		
5386	025141	000		
5387	025142	000207		
5388				
5389				
5390				
5391	025144	017637	000000	001552
5392	025152	062716	000002	
5393	025156	004737	026022	
5394				
5395	025162	053737	001222	005476
5396	025170	053737	001222	005502
5397	025176	053737	001222	005506
5398	025204	053737	001222	005512
5399				
5400	025212	013746	005444	
5401				
5402	025216	013737	005476	005444
5403	025224	004737	030202	
5404	025230	013737	005444	005476
5405				
5406	025236	013737	005502	005444
5407	025244	004737	030202	

```

FATT2: MOV R4, -(SP) ; SAV R4
2S: MOV $UNIT, R4
BITB ATTN(R4), RKASOF+1(R5) ; FIND CORRECT ATTN
BNE 1S
DEC TEMP1
BNE 2S
CLR RKMR1(R5) ; SELECT WORD 0
JSR PC, GSTAT ; GET LATEST STATUS.
MOV (SP)+, R4 ; RESTOR R4
RTS PC
1S: CLR RKMR1(R5)
JSR PC, GSTAT
MOV (SP)+, R4 ; RESTOR R4
ADD #2, (SP) ; SKIP OVER ERROR
RTS PC

; ENTER WITH A COUNT IN TEMP1
; THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT
; WHEN COUNT IS 0. BASED ON AN 11/05
DLY: TST TEMP1 ; 5.6 US
BEQ 1S ; 2.5 US
DEC TEMP1 ; 6.8 US
BR DLY ; 2.5 US
1S: RTS PC ; 3.8 US

; THIS ROUTINE TYPES BYPASSED DRIVE#. ENTER WITH DRIVE# IN RO
BYP: TYPE MSG14 ; BYPASS DRIVE
MOV RO, -(SP) ; SAVE RO FOR TYPEOUT
; TYPE DR#
; GO TYPE--OCTAL ASCII
; TYPE 1 DIGIT(S)
; SUPPRESS LEADING ZEROS
RTS PC

; THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
CHKMSG: MOV #2(S), CHKFLG ; PASS MSGS TO BE TESTED
ADD #2, (SP) ; BUMP RETURN ADDR TO 1ST ERROR
JSR PC, GSTAT1 ; GET ALL ACTUAL DRIVE & CONTR STATUS
BIS $UNIT, E.A0 ; SET UNIT #
BIS $UNIT, E.A1
BIS $UNIT, E.A2
BIS $UNIT, E.A3
MOV TEMP1, -(SP) ; SAVE TEMP1
MOV E.A0, TEMP1
JSR PC, SBPAR ; GET PARITY FOR MSG A0
MOV TEMP1, E.A0
MOV E.A1, TEMP1
JSR PC, SBPAR ; GET PARITY FOR MSG A1
    
```

M08

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05MACY11 27(1006) 31-JAN-77 18:30 PAGE 102
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0102

```

5408 025250 013737 005444 005502      MOV      TEMP1,E.A1
5409
5410 025256 013737 005506 005444      MOV      E.A2,TEMP1
5411 025264 004737 030202      JSR      PC,SBPARG      ;GET PARITY FOR MSG A2
5412 025270 013737 005444 005506      MOV      TEMP1,E.A2
5413
5414 025276 013737 005500 005444      MOV      E.B0,TEMP1
5415 025304 004737 030202      JSR      PC,SBPARG      ;GET PARITY FOR MSG B0
5416 025310 013737 005444 005500      MOV      TEMP1,E.B0
5417
5418 025316 013737 005504 005444      MOV      E.B1,TEMP1
5419 025324 004737 030202      JSR      PC,SBPARG      ;GET PARITY FOR MSG B1
5420 025330 013737 005444 005504      MOV      TEMP1,E.B1
5421
5422 025336 013737 005510 005444      MOV      E.B2,TEMP1
5423 025344 004737 030202      JSR      PC,SBPARG      ;GET PARITY FOR MSG B2
5424 025350 013737 005444 005510      MOV      TEMP1,E.B2
5425
5426 025356 013737 005514 005444      MOV      E.B3,TEMP1
5427 025364 004737 030202      JSR      PC,SBPARG      ;GET PARITY FOR MSG B3
5428 025370 013737 005444 005514      MOV      TEMP1,E.B3
5429
5430 025376 012637 005444      MOV      (SP)+,TEMP1      ;RESTORE TEMP1
5431 025402 013737 001176 001172      MOV      $ESCAPE,$TMP5      ;SAVE ESCAPE
5432
5433 025410 023737 005456 005476      CMP      H.A0,E.A0      ;TEST MSG A0
5434 025416 001411      BEQ      2$      ;BR IF OK
5435 025420 012737 025432 001176      MOV      #1$,$ESCAPE      ;ELSE SETUP ESCAPE
5436 025426 011646      MOV      (SP),-(SP)      ;COPY RET ADDR.
5437 025430 000207      RTS      PC      ;& RETURN TO MAINLINE ERROR
5438
5439 025432 032777 001000 153500 1$:      BIT      #SW9,$SWR      ;RET HERE FROM MAINLINE ERROR
5440 025440 001107      BNE      20$      ;& BR IF LOOP ON ERROR
5441 025442 062716 000002      ADD      #2,(SP)      ;BUMP RET ADDR TO NEXT ERROR
5442
5443 025446 023737 005460 005500      CMP      H.B0,E.B0      ;TEST MSG B0
5444 025454 001411      BEQ      5$      ;BR IF OK
5445 025456 012737 025470 001176      MOV      #4$,$ESCAPE      ;ELSE SETUP ESCAPE
5446 025464 011646      MOV      (SP),-(SP)      ;COPY RET ADDR
5447 025466 000207      RTS      PC      ;& RETURN TO MAINLINE ERROR
5448
5449 025470 032777 001000 153442 4$:      BIT      #SW9,$SWR      ;RETURN HERE FROM MAINLINE ERROR
5450 025476 001070      BNE      20$      ;& BR IF LOOP ON ERROR
5451 025500 062716 000002      ADD      #2,(SP)      ;BUMP RET ADDR TO NEXT ERROR
5452
5453 025504 023737 005462 005502      CMP      H.A1,E.A1      ;TEST MSG A1
5454 025512 001411      BEQ      8$      ;BR IF OK
5455 025514 012737 025526 001176      MOV      #7$,$ESCAPE
5456 025522 011646      MOV      (SP),-(SP)
5457 025524 000207      RTS      PC
5458
5459 025526 032777 001000 153404 7$:      BIT      #SW9,$SWR
5460 025534 001051      BNE      20$
5461 025536 062716 000002      ADD      #2,(SP)
5462
5463 025542 023737 005464 005504      CMP      H.B1,E.B1      ;TEST MSG B1

```



```

5464 025550 001411          BEQ      11$          ;BR IF OK
5465 025552 012737 025564 001176  MOV     #10$, $ESCAPE
5466 025560 011646          MOV     (SP), -(SP)
5467 025562 000207          RTS     PC
5468
5469 025564 032777 001000 153346 10$:  BIT     #SW9, $SWR
5470 025572 001032          BNE     20$
5471 025574 062716 000002          AND     #2, (SP)
5472
5473 025600 032737 000001 001552 12$:  BIT     #T.A2, CHKFLG ;TEST MSG A2?
5474 025606 001402          BEQ     13$          ;BR IF NO
5475 025610 004737 026616          JSR     PC, RCYLD    ;PUT INFO CYLDIF, DO NOT CHECK
5476 025614 032737 000002 001552 13$:  BIT     #T.B2, CHKFLG ;TEST MSG B2?
5477 025622 001402          BEQ     14$          ;BR IF NO
5478 025624 004737 026670          JSR     PC, RCYLA    ;PUT INFO IN CYLADD, DO NOT CHECK
5479
5480 025630 032737 000004 001552 14$:  BIT     #T.B3, CHKFLG ;TEST MSG B3?
5481 025636 001404          BEQ     15$
5482 025640 004737 026726          JSR     PC, RSEC     ;PUT INFO IN SECTOR, DO NOT CHECK
5483 025644 004737 026764          JSR     PC, RHEAD    ;PUT INFO IN HEADA, DO NOT CHECK
5484
5485 025650 013737 001172 001176 15$:  MOV     $TMP5, $ESCAPE ;RESTORE ESCAPE
5486 025656 000207          RTS     PC
5487
5488 025660 012706 001100          MOV     #STACK, SP   ;RESET STACK PTR
5489 025664 013737 001172 001176 20$:  MOV     $TMP5, $ESCAPE ;RESTORE ESCAPE
5490 025672 000177 153212          JMP     $SLPERR
5491
5492 ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
5493 ; I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
5494
5495 025676 005737 001550          CKCERR: TST     BYPCERR
5496 025702 001025          BNE     4$
5497 025704 032737 100000 005406          BIT     #CERR, HCS1
5498 025712 001001          BNE     1$          ;BR IF CERR
5499 025714 000207          RTS     PC
5500
5501 025716 032737 004000 005406 1$:  BIT     #CTO, HCS1
5502 025724 001402          BEQ     2$          ;BR IF NOT CTO
5503 025726 104211          ERROR   211         ;CTO ERROR, MSG A & B INVALID
5504 025730 000207          RTS     PC
5505
5506 025732 032737 010000 005410 2$:  BIT     #NED, HCS2
5507 025740 001401          BEQ     3$          ;BR IF NOT NED
5508 025742 104212          ERROR   212         ;NED ERROR, MSG A & B INVALID
5509
5510 025744 032737 001000 005410 3$:  BIT     #MDS, HCS2
5511 025752 001401          BEQ     4$
5512 025754 104213          ERROR   213         ;MDS ERROR, MSG A & B INVALID
5513
5514 025756 000207          4$:    RTS     PC
5515
5516 ; THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
5517 ; IT THEN WAITS FOR CONTROLLER READY
5518
5519

```

```

5520
5521
5522
5523 025760 013746 005444
5524 025764 013765 001222 000010
5525 025772 012765 000001 000000
5526 026000 013737 001426 005444
5527 026006 004737 024422
5528 026012 104117
5529 026014 012637 005444
5530 026020 000207
5531
5532
5533
5534
5535 026022 013746 005444
5536 026026 004737 024556
5537 026032 012765 100000 000000
5538 026040 013765 001222 000010
5539 026046 012765 000003 000026
5540 026054 012765 000001 000000
5541 026062 013737 001426 005444
5542 026070 004737 024470
5543 026074 104117
5544 026076 013737 005434 005472
5545 026104 013737 005436 005474
5546
5547 026112 012765 100000 000000
5548 026120 013765 001222 000010
5549 026126 012765 000002 000026
5550 026134 012765 000001 000000
5551 026142 013737 001426 005444
5552 026150 004737 024470
5553 026154 104117
5554 026156 013737 005434 005466
5555 026164 013737 005436 005470
5556
5557 026172 012765 100000 000000
5558 026200 013765 001222 000010
5559 026206 012765 000001 000026
5560 026214 012765 000001 000000
5561 026222 013737 001426 005444
5562 026230 004737 024470
5563 026234 104117
5564 026236 013737 005434 005462
5565 026244 013737 005436 005464
5566
5567 026252 012765 100000 000000
5568 026260 013765 001222 000010
5569 026266 012765 000001 000000
5570 026274 013737 001426 005444
5571 026302 004737 024470
5572 026306 104117
5573 026310 013737 005434 005456
5574 026316 013737 005436 005460
5575
    
```

;IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED

```

GSTAT: MOV TEMP1, -(SP) ;SAVE TEMP1
        MOV $UNIT, RKCS2(R5) ;CURRENT DRIVE #
        MOV #SELDRV, RKCS1(R5) ;GET STATUS WITH SELECT DRIVE CMD
        MOV T10, TEMP1
        JSR PC, FRDY ;FIND RDY
        ERROR 117 ;RDY NOT SET BY END OF SELECT DRIVE CMD
        MOV (SP)+, TEMP1 ;RESTOR TEMP1.
        RTS PC
    
```

;THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
; & ALL CONTROLLER REGISTERS.

```

GSTAT1: MOV TEMP1, -(SP) ;SAVE TEMP1
         JSR PC, HOLD ;GET ALL CONTR REG
         MOV #CCLR, RKCS1(R5) ;CLEAR CONTR
         MOV $UNIT, RKCS2(R5) ;CURRENT DRIVE #
         MOV #3, RMR1(R5) ;SELECT WORD 3
         MOV #SELDRV, RKCS1(R5) ;GET STATUS
         MOV T10, TEMP1
         JSR PC, FRDY1 ;FIND RDY & STORE DRIVE REGS ONLY
         ERROR 117 ;RDY NOT SET BY END OF SELECT DRV CMD
         MOV HMR2, H.A3 ;STORE MSG A3
         MOV HMR3, H.B3 ;STORE MSG B3
         MOV #CCLR, RKCS1(R5)
         MOV $UNIT, RKCS2(R5)
         MOV #2, RMR1(R5) ;SELECT WORD 2
         MOV #SELDRV, RKCS1(R5)
         MOV T10, TEMP1
         JSR PC, FRDY1 ;FIND RDY & STORE DRIVE REGS ONLY
         ERROR 117 ;RDY NOT SET BY END OF SELECT DRV CMD
         MOV HMR2, H.A2 ;STORE MSG A2
         MOV HMR3, H.B2 ;STORE MSG B2
         MOV #CCLR, RKCS1(R5)
         MOV $UNIT, RKCS2(R5)
         MOV #1, RMR1(R5) ;SELECT WORD 1
         MOV #SELDRV, RKCS1(R5)
         MOV T10, TEMP1
         JSR PC, FRDY1 ;FIND RDY & STORE DRIVE REGS ONLY
         ERROR 117 ;RDY NOT SET BY END OF SELECT DRV CMD
         MOV HMR2, H.A1 ;STORE MSG A1
         MOV HMR3, H.B1 ;STORE MSG B1
         MOV #CCLR, RKCS1(R5)
         MOV $UNIT, RKCS2(R5)
         MOV #SELDRV, RKCS1(R5) ;SELECT WORD 0
         MOV T10, TEMP1
         JSR PC, FRDY1 ;FIND RDY & STORE DRIVE REGS ONLY
         ERROR 117 ;RDY NOT SET BY END OF SEL DRV CMD
         MOV HMR2, H.A0 ;STORE MSG A0
         MOV HMR3, H.B0 ;STORE MSG B0
    
```

```

5576 026324 012637 005444      MOV    (SP)+,TEMP1    ;RESTORE TEMP1
5577 026330 000207      RTS     PC
5578
5579
5580      ; THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
5581      ; IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
5582      ; THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
5583      ; RETURN IF CERR SET
5584      ; RETURN +2 IF CERR CLEAR
5585
5586 026332 012765 000040 000010  SUBCLR: MOV    #SCLR,RKCS2(R5) ;SUBSYS CLEAR
5587 026340 013737 001426 005444      MOV    T10,TEMP1
5588 026346 004737 024422      JSR    PC,FRDY        ;FIND RDY
5589 026352 104120      ERROR 120            ;RDY NOT SET BY END OF SCLR
5590 026354 013765 001222 000010  MOV    #UNIT,RKCS2(R5) ;CURRENT DRIVE #
5591 026362 005065 000026      CLR    RKMR1(R5)      ;SELECT WORD 0
5592 026366 004737 025760      JSR    PC,GSTAT      ;GET STATUS
5593 026372 032737 100000 005406  BIT    #CERR,HCS1     ;CHECK FOR CONT ERROR
5594 026400 001401      BEQ    1$
5595 026402 000207      RTS     PC
5596 026404 062716 000002 1$: ADD    #2,(SP)      ;SKIP OVER ERROR
5597 026410 000207      RTS     PC
5598
5599
5600      ; READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
5601
5602 026412 012765 000003 000026  RDSEC: MOV    #3,RKMR1(R5) ;WORD 3
5603 026420 004737 025760      JSR    PC,GSTAT
5604 026424 013737 005436 001422  MOV    HMR3,SECTOR
5605 026432 042737 177017 001422  BIC    #1C<M.SECT>,SECTOR
5606 026440 006237 001422      ASR    SECTOR         ;RIGHT JUSTIFY
5607 026444 006237 001422      ASR    SECTOR         ;SECTOR
5608 026450 006237 001422      ASR    SECTOR         ;INFO
5609 026454 006237 001422      ASR    SECTOR
5610 026460 000207      RTS     PC
5611
5612      ; READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
5613
5614 026462 012765 000002 000026  RDCYLD: MOV    #2,RKMR1(R5) ;WORD 2
5615 026470 004737 025760      JSR    PC,GSTAT
5616 026474 013737 005434 001376  MOV    HMR2,CYLDIF
5617
5618      BIC    #1C<M.CDIF>,CYLDIF
5619 026510 006237 001376      ASR    CYLDIF         ;RIGHT JUSTIFY
5620 026514 006237 001376      ASR    CYLDIF         ;CYL DIFF/OFFSET
5621 026520 006237 001376      ASR    CYLDIF         ;INFO
5622 026524 006237 001376      ASR    CYLDIF
5623 026530 023727 001376 000777  CMP    CYLDIF,#777    ;CHK TO SEE IF RET IN COMPL. FORM
5624 026536 001002      BNE    1$            ;BR IF NOT
5625 026540 005037 001376      CLR    CYLDIF        ;CLR IF YES
5626 026544 000207 1$: RTS     PC
5627
5628      ; READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
5629
5630 026546 012765 000002 000026  RDCYLA: MOV    #2,RKMR1(R5) ;WORD 2
5631 026554 004737 025760      JSR    PC,GSTAT

```

```

5632 026560 013737 005436 001400      MOV      HMR3,CYLADD
5633 026566 042737 160017 001400      BIC      #1C<M.CADD>,CYLADD
5634 026574 006237 001400      ASR      CYLADD          ;RIGHT JUSTIFY
5635 026600 006237 001400      ASR      CYLADD          ;CYL ADDR
5636 026604 006237 001400      ASR      CYLADD          ;INFO
5637 026610 006237 001400      ASR      CYLADD
5638 026614 000207                RTS      PC
5639
5640                ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
5641
5642 026616 013737 005466 001376  RCYLD:  MOV      H.A2,CYLDIF
5643 026624 042737 160017 001376      BIC      #1C<M.CDIF>,CYLDIF ;CLEAR UNWANTED INFO
5644 026632 006237 001376      ASR      CYLDIF          ;RIGHT JUSTIFY
5645 026636 006237 001376      ASR      CYLDIF
5646 026642 006237 001376      ASR      CYLDIF
5647 026646 006237 001376      ASR      CYLDIF
5648 026652 023727 001376 000777      CMP      CYLDIF,#777      ;CHK TO SEE IF RET IN COMPL. FORM
5649 026660 001002                BNE      IS              ;BR IF NO
5650 026662 005037 001376      CLR      CYLDIF          ;ELSE CLEAR
5651 026666 000207                IS:      RTS      PC
5652
5653                ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
5654
5655 026670 013737 005470 001400  RCYLA:  MOV      H.B2,CYLADD
5656 026676 042737 160017 001400      BIC      #1C<M.CADD>,CYLADD ;CLEAR UNWANTED INFO
5657 026704 006237 001400      ASR      CYLADD          ;RIGHT JUSTIFY
5658 026710 006237 001400      ASR      CYLADD
5659 026714 006237 001400      ASR      CYLADD
5660 026720 006237 001400      ASR      CYLADD
5661 026724 000207                RTS      PC
5662
5663                ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
5664
5665 026726 013737 005474 001422  RSEC:   MOV      H.B3,SECTOR
5666 026734 042737 177017 001422      BIC      #1C<M.SECT>,SECTOR ;CLEAR UNWANTED INFO
5667 026742 006237 001422      ASR      SECTOR          ;RIGHT JUSTIFY
5668 026746 006237 001422      ASR      SECTOR
5669 026752 006237 001422      ASR      SECTOR
5670 026756 006237 001422      ASR      SECTOR
5671 026762 000207                RTS      PC
5672
5673                ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
5674
5675 026764 013737 005474 001512  RHEAD:  MOV      H.B3,HEADA
5676 026772 042737 170777 001512      BIC      #1C<M.HEAD>,HEADA ;CLEAR UNWANTED INFO
5677 027000 006237 001512      ASR      HEADA          ;RIGHT JUSTIFY IT
5678 027004 000337 001512      SWAB    HEADA
5679 027010 000207                RTS      PC
5680
5681                ;FIND SECTOR 17
5682                ;RETURN IF NOT FOUND
5683                ;RETURN +4 IF FOUND
5684
5685 027012 013737 001436 005444  FSEC17: MOV      T5000,TEMP1 ;SETUP TIMEOUT
5686 027020 004737 026412                IS:      JSR      PC,RDSEC ;READ SECTOR
5687 027024 023727 001422 000021      CMP      SECTOR,#17.    ;TEST FOR SECTOR 17

```

E09

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05MACY11 27(1006) 31-JAN-77 18:30 PAGE 107
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0107

```

5688 027032 001014          BNE      2$          ;BR IF NOT 17
5689
5690 027034 004737 026412    JSR      PC,RDSEC
5691 027040 023727 001422 000021  CMP      SECTOR,#17.
5692 027046 001412          BEQ      3$          ;BR IF READ SAME TWICE
5693 027050 004737 026412    JSR      PC,RDSEC    ;ELSE TRY 1 MORE TIME
5694 027054 023727 001422 000021  CMP      SECTOR,#17.
5695 027062 001404          BEQ      3$          ;BR IF 17
5696
5697 027064 005337 005444    2$:     DEC      TEMP1
5698 027070 001353          BNE      1$          ;TRY AGAIN
5699 027072 000207          RTS      PC
5700
5701 027074 062716 000004    3$:     ADD      #4,(SP) ;SKIP OVER ERROR
5702 027100 000207          RTS      PC
5703
5704          ;FIND DESIRED CYL DIFF
5705          ;RETURN IF NOT FOUND
5706          ;RETURN+6 IF FOUND
5707
5708
5709 027102 013737 001436 005444  FCYL:   MOV      T5000,TEMP1 ;SETUP TIMEOUT
5710 027110 004737 026462    1$:     JSR      PC,RDCYLD
5711 027114 023737 001376 005446    CMP      CYLDIF,TEMP2 ;TEST FOR CYL DIFF
5712 027122 001014          BNE      2$          ;BR IF NOT FOUND
5713
5714 027124 004737 026462    JSR      PC,RDCYLD
5715 027130 023737 001376 005446    CMP      CYLDIF,TEMP2
5716 027136 001412          BEQ      3$          ;BR IF READ SAME TWICE
5717 027140 004737 026462    JSR      PC,RDCYLD    ;ELSE TRY 1 MORE TIME
5718 027144 023737 001376 005446    CMP      CYLDIF,TEMP2
5719 027152 001404          BEQ      3$
5720
5721 027154 005337 005444    2$:     DEC      TEMP1
5722 027160 001353          BNE      1$          ;TRY AGAIN
5723 027162 000207          RTS      PC
5724
5725 027164 062716 000006    3$:     ADD      #6,(SP) ;SKIP OVER ERROR
5726 027170 000207          RTS      PC
5727
5728
5729          ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
5730          ;ENTER WITH TIME IN SECONDS IN TEMP2
5731          ;RETURN IF NOT FOUND
5732          ;RETURN+2 IF FOUND - SKIP OVER ERROR
5733
5734 027172 012737 177777 005444  FHDHM:  MOV      #-1,TEMP1 ;ALL 1'S
5735 027200 012765 000001 000026  MOV      #1,RKMR1(R5) ;WORD 1
5736 027206 004737 025760    1$:     JSR      PC,GSTAT
5737 027212 032737 000040 005434    BIT      #0,HDHM,HMR2
5738 027220 001007          BNE      2$
5739 027222 005337 005444    DEC      TEMP1
5740 027226 001367          BNE      1$
5741
5742 027230 005337 005446    DEC      TEMP2
5743 027234 001356          BNE      FHDHM

```

```

5744 027236 000207
5745 027240 062716 000002
5746 027244 000207
5747
5748
5749
5750
5751
5752 027246 012737 000372 005444
5753 027254 012765 000001 000026
5754 027262 004737 025760
5755 027266 032737 010000 005434
5756 027274 001004
5757 027276 005337 005444
5758 027302 001367
5759 027304 000207
5760 027306 062716 000002
5761 027312 000207
5762
5763
5764
5765
5766
5767
5768 027314 010046
5769 027316 010146
5770 027320 012700 001554
5771 027324 005001
5772 027326 013737 001510 001514
5773 027334 006337 001514
5774 027340 006337 001514
5775 027344 006337 001514
5776 027350 006337 001514
5777 027354 006337 001514
5778 027360 013737 001516 001520
5779 027366 000337 001520
5780 027372 006337 001520
5781
5782 027376 013720 001402
5783 027402 010110
5784 027404 053710 001514
5785 027410 053710 001520
5786 027414 004737 027474
5787
5788 027420 013737 001402 005444
5789 027426 011037 005446
5790 027432 043737 001402 005446
5791 027440 042037 005444
5792 027444 053737 005444 005446
5793 027452 013720 005446
5794
5795 027456 005201
5796 027460 020127 000026
5797 027464 001344
5798
5799 027466 012601

RTS PC
2$: ADD #2,(SP) ;SKIP OVER ERROR
RTS PC

;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE SMS
;RETURN IF NOT FOUND
;RETURN+2 IF FOUND: SKIP OVER ERROR

FLOAD: MOV #250,TEMP1
MOV #1,RKMR1(R5) ;WORD 1
1$: JSR PC,GSTAT
BIT #D.LOAD,HMR2
BNE 2$
DEC TEMP1
BNE 1$
RTS PC
2$: ADD #2,(SP) ;SKIP OVER ERROR
RTS PC

;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
;ENTER WITH CYL # IN 'CALADD'
;ENTER WITH HEAD # IN 'HEAD'
;ENTER WITH FORMAT IN 'FORMAT'

FHDTAB: MOV RO,-(SP) ;SAV RO
MOV R1,-(SP) ;SAV R1
MOV #HDTAB,RO ;HEADER WORD TABLE ADDR
CLR R1 ;SECTOR COUNTER
MOV HEAD,HD1
ASL HD1
ASL HD1
ASL HD1
ASL HD1
ASL HD1 ;SETUP HEAD # FOR WORD 2 OF HEADER
MOV FORMAT,FMT1
SWAB FMT1
ASL FMT1 ;SETUP FORMAT FOR WORD 2 OF HEADER

1$: MOV CALADD,(RO)+ ;HEADER WORD 1-CYL ADDR
MOV R1,(RO) ;HEADER WORD 2-SECTOR NO
BIS HD1,(RO) ; -HEAD NO
BIS FMT1,(RO) ; -FORMAT
JSR PC,SECFLG ;GET SECTOR FLAGS

MOV CALADD,TEMP1
MOV (RO),TEMP2
BIC CALADD,TEMP2
BIC (RO)+,TEMP1
BIS TEMP1,TEMP2
MOV TEMP2,(RO)+ ;HEADER WORD 3-HEADER CHECK

INC R1 ;SECTOR CTR
CMP R1,#22. ;ALL 22 SECTORS DONE? (66 WORDS)
BNE 1$ ;BR IF NO

MOV (SP)+,R1 ;RESTOR R1

```

```

5800 027470 012600          MOV    (SP)+,R0      ;RESTOR R0
5801 027472 000207          RTS     PC
5802
5803
5804
5805
5806 027474 010246          SECFLG: MOV   R2,-(SP)      ;SAVE R2
5807 027476 005737 001516   TST    FORMAT
5808 027502 001016          BNE    1$              ;BR IF 20 SECTOR FORMAT
5809 027504 012702 002400   MOV    #BSE22H+8.,R2
5810 027510 004737 027544   JSR    PC,FLGTST      ;GET HARDWARE DETECTED FLAG
5811 027514 052710 100000   BIS    #BIT15,(R0)    ;RETURN HERE IF GOOD SECTOR
5812
5813 027520 012702 003400   MOV    #BSE22S+8.,R2  ;ELSE RETURN HERE
5814 027524 004737 027544   JSR    PC,FLGTST      ;GET SOFTWARE DETECTED FLAG
5815 027530 052710 040000   BIS    #BIT14,(R0)    ;RETURN HERE IF GOOD SECTOR
5816
5817 027534 012602          MOV    (SP)+,R2      ;ELSE RETURN HERE
5818 027536 000207          RTS     PC
5819
5820 027540 012602          1$:   MOV    (SP)+,R2      ;RESTORE R2
5821 027542 000207          RTS     PC
5822
5823
5824
5825
5826
5827
5828
5829 027544 010346          ; THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES.
5830
5831 027546 021227 177777   ; ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
5832 027552 001002          ; RETURN IF NO COMPARE
5833 027554 012603          ; RETURN+4 IF COMPARE
5834 027556 000207          FLGTST: MOV   R3,-(SP)    ;SAVE R3
5835
5836 027560 022237 001402   1$:   CMP    (R2),#-1      ;SEE IF ALL 1'S
5837 027564 001403          BNE    2$              ;BR IF NO
5838 027566 062702 000002   MOV    (SP)+,R3      ;RESTORE R3
5839 027572 000765          RTS     PC
5840
5841 027574 013703 001510   2$:   CMP    (R2)+,CALADD ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
5842 027600 000303          BEQ    3$              ;GO TO NEXT CYL WORD IN TABLE
5843 027602 050103          ADD    #2,R2
5844 027604 022203          BR     1$
5845
5846 027606 001401          3$:   MOV    HEAD,R3        ;GET HEAD # FROM FHDTAB ROUTINE
5847 027610 000756          SWAB   R3
5848
5849 027612 012603 000004   BIS    R1,R3          ;ADD SECTOR # FROM FHDTAB ROUTINE
5850 027614 062716          CMP    (R2)+,R3      ;SEE IF SECTOR/HEAD COMPARE
5851 027620 000207          ; & INCR PTR TO NEXT CYL WORD
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000

```

```

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

```



```

5912
5913 027776 012704 003400      MOV      #BSE225+8.,R4      ;ELSE RETURN HERE
5914 030002 004737 030024      JSR      PC,TERR1          ;& SEE IF ON SOFTWARE DETECTED TABLE
5915 030006 000402                BR       3$                ;RETURN HERE IF YES
5916
5917 030010 012604                1$:     MOV      (SP)+,R4      ;RESTORE R4
5918 030012 000207                RTS      PC                ;RETURN WITHOUT JUMPING OVER ERROR
5919
5920
5921 030014 012604                3$:     MOV      (SP)+,R4      ;RESTORE R4
5922 030016 062716 000002      ADD      #2,(SP)          ;SKIP OVER ERROR ON RETURN
5923 030022 000207                RTS      PC

```

```

: THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
: THE BSE TABLE FOR THE ABOVE SUBROUTINE.
: RETURN IF FOUND ON TABLE
: RETURN+2 IF NOT FOUND

```

```

5930
5931 030024 021427 177777      TERR1:  CMP      (R4),#-1        ;SEE IF ALL 1'S
5932 030030 001405                BEQ      1$                ;BR IF YES, NOT ON TABLE
5933 030032 022437 005426      CMP      (R4)+,HDC        ;SEE IF CYL MATCH
5934 030036 001405                BEQ      2$                ;BR IF YES
5935 030040 005724                TST      (R4)+            ;ELSE ADV TO NEXT CYL WORD
5936 030042 000770                BR       TERR1            ;& TRY AGAIN.
5937
5938 030044 062716 000002      1$:     ADD      #2,(SP)
5939 030050 000207                RTS      PC
5940
5941 030052 022437 005416      2$:     CMP      (R4)+,HDA        ;SEE IF SECTOR & TRACK MATCH
5942 030056 001401                BEQ      3$                ;BR IF YES
5943 030060 000761                BR       TERR1            ;OR TRY AGAIN
5944
5945 030062 000207                3$:     RTS      PC

```

```

: ROUTINE TO TURN L OR P CLOCK INTERRUPT ON

```

```

5950
5951
5952
5953
5954 030064 005037 001412      CLKON:  CLR      TIMUP
5955 030070 005737 005552      TST      PCLKF
5956 030074 001004                BNE      1$                ;BRANCH IF P-CLOCK PRESENT
5957 030076 012777 000100 151242  MOV      #100,ALKS        ;L-CLOCK, ENABLE INT
5958 030104 000207                RTS      PC
5959 030106 012777 177777 151226  1$:     MOV      #-1,PKSMB        ;P-CLOCK, ALL 1'S
5960 030114 012777 000135 151216  MOV      #135,PKS        ;ENABLE INT, CT UP, REP INT
5961 030122 000207                RTS      PC                ;LINE FREQ & RUN

```

```

: KW11-L & KW11-P INTERRUPT HANDLER

```

```

5962
5963
5964
5965 030124 005037 001412      CLOCK:  CLR      TIMUP
5966 030130 005337 001406      DEC      COUNT
5967 030134 001010                BNE      1$

```

```

5968 030136 013737 001404 001406      MOV      HZ COUNT
5969 030144 005337 001410      DEC      SEC
5970 030150 001002      BNE     1$
5971 030152 005237 001412      INC     TIMUP      ;SORRY, TIME IS UP
5972 030156 000002      1$:     RTI
5973
5974      ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
5975
5976 030160 005737 005552      CLKOF:  TST     PCLKF
5977 030164 001003      BNE     1$      ;BRACH IF P-CLOCK PRESENT
5978 030166 005077 151154      CLR     ALKS     ;L-CLOCK, CLEAR INTERRUPT
5979 030172 000207      RTS     PC
5980 030174 005077 151140      1$:     CLR     APKS     ;P-CLOCK, CLEAR INTERRUPT
5981 030200 000207      RTS     PC
5982
5983      ;
5984      ;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MESSAGES
5985      ;ENTER WITH THE EXPECTED WORD IN TEMP1
5986      ;TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET
5987      ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES (TEMP1 BACK TO ORIG),
5988      ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
5989      ;THE PARITY BIT IS NOT SET IN B
5990      ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
5991      ;SET IN TEMP1
5992
5993 030202 010046      SBPAR:  MOV     R0, -(SP)      ;SAVE R0
5994 030204 010146      MOV     R1, -(SP)      ;SAVE R1
5995 030206 012700 000021      MOV     #17, R0      ;SHIFT COUNTER
5996 030212 005001      CLR     R1      ;COUNT # OF 1'S IN TEMP1
5997 030214 000241      CLC      ;CLEAR CARRY
5998
5999 030216 006137 005444      1$:     ROL     TEMP1
6000 030222 103001      BCC     2$      ;BR IF CARRY CLEAR
6001 030224 005201      INC     R1      ;COUNT # OF 1'S
6002 030226 005300      2$:     DEC     R0      ;SHIFT COUNTER
6003 030230 001372      BNE     1$
6004
6005 030232 032701 000001      BIT     #BIT0, R1
6006 030236 001003      BNE     3$      ;BR IF ODD # IN R0
6007 030240 052737 100000 005444      BIS     #M.PAR, TEMP1 ;SET PARITY BIT
6008 030246 012601      3$:     MOV     (SP)+, R1      ;RESTORE R1
6009 030250 012600      MOV     (SP)+, R0      ;RESTORE R0
6010 030252 000207      RTS     PC
6011
6012      ;
6013      ;ROUTINE TO ENABLE LOOPING ON INTERMITTANT ERRORS
6014      ;WHEN SLPERR SET BY OTHER THAN SCOPE ROUTINE
6015      ;IE: MY LOOP MACRO
6016
6017 030254 032777 001000 150656      SCOPE:  BIT     #SW9, ASWR      ;LOOP ON ERROR?
6018 030262 001406      BEQ     1$      ;BR IF NO
6019 030264 105737 001103      TSTB   $ERFLG      ;HAD ERROR?
6020 030270 001403      BEQ     1$      ;BR IF NO
6021 030272 013716 001110      MOV     SLPERR, (SP)
6022 030276 000002      RTI
6023

```

6024	030300	011637	001110	1S:	MOV	(SP), \$LPERR	;SET LOOP ADDR FOR TIGHT SCOPE LOOP
6025	030304	000002			RTI		
6026							
6027							
6028							
6029							
6030							
6031							
6032							
6033							
6034	030306	005777	150634	CCSP:	TST	2\$TKB	;CLEAR DONE FLAG
6035	030312	104410			RDCHR		;READ CHAR FROM TTY
6036	030314	012600			MOV	(SP)+, RO	;GET CHAR OFF STACK
6037	030316	020027	000040		CMP	RO, #SPBAR	;SEE IF SPACE
6038	030322	001406			BEQ	1\$;BR IF YES.
6039							
6040	030324	020027	000005		CMP	RO, #5	;SEE IF CONTROL-E
6041	030330	001405			BEQ	2\$;BR IF YES
6042	030332	104401	040067		TYPE	MSG31	;?"
6043	030336	000763			BR	CCSP	;TRY AGAIN
6044							
6045	030340	062716	000004	1\$:	ADD	#4, (SP)	
6046	030344	000207		2\$:	RTS	PC	
6047							
6048							
6049							
6050							
6051	030346	005777	150574	GETSP:	TST	2\$TKB	;CLEAR DONE FLAG
6052	030352	104410			RDCHR		;READ CHAR OFF TTY
6053	030354	012600			MOV	(SP)+, RO	;GET CHAR OFF STACK
6054	030356	020027	000040		CMP	RO, #SPBAR	;SEE IF SPACE
6055	030362	001403			BEQ	1\$;EXIT IF YES
6056	030364	104401	040067		TYPE	MSG31	;?
6057	030370	000766			BR	GETSP	;TRY AGAIN
6058	030372	000207		1\$:	RTS	PC	
6059							
6060							
6061							
6062							
6063							
6064							
6065							
6066	030374	022626		STOP:	CMP	(SP)+, (SP)+	;RESTORE STACK FROM INTERRUPT
6067							
6068	030376	004737	026332	STOP1:	JSR	PC, SUBCLR	
6069	030402	104024			ERROR	24	;CERR AFTER
6070							
6071	030404	005737	005370		TST	UNLD	;SEE IF HEADS UNLOADED
6072	030410	001442			BEQ	3\$;BR IF NO
6073	030412	005737	000042		TST	42	;SEE IF MANUAL OR AUTO MODE
6074	030416	001403			BEQ	1\$;BR IF MANUAL MODE
6075	030420	104401	043730		TYPE	MSG74	;PGM ABORT PENDING
6076	030424	000402			BR	2\$	
6077	030426	104401	043776	1\$:	TYPE	,MSG75	;HALT PENDING
6078	030432			2\$:			
6079							

```

;ROUTINE TO INPUT A 'SPACE' OR 'CONTROL-E' FROM TTY
;RETURN IF CONTROL-E
;RETURN +4 IF SPACE

```

```

CCSP: TST 2$TKB ;CLEAR DONE FLAG
      RDCHR ;READ CHAR FROM TTY
      MOV (SP)+, RO ;GET CHAR OFF STACK
      CMP RO, #SPBAR ;SEE IF SPACE
      BEQ 1$ ;BR IF YES.

      CMP RO, #5 ;SEE IF CONTROL-E
      BEQ 2$ ;BR IF YES
      TYPE MSG31 ;?"
      BR CCSP ;TRY AGAIN

```

```

1$: ADD #4, (SP)
2$: RTS PC

```

```

;ROUTINE TO INPUT A 'SPACE' FROM TTY

```

```

GETSP: TST 2$TKB ;CLEAR DONE FLAG
      RDCHR ;READ CHAR OFF TTY
      MOV (SP)+, RO ;GET CHAR OFF STACK
      CMP RO, #SPBAR ;SEE IF SPACE
      BEQ 1$ ;EXIT IF YES
      TYPE MSG31 ;?
      BR GETSP ;TRY AGAIN
1$: RTS PC

```

```

;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.

```

```

STOP: CMP (SP)+, (SP)+ ;RESTORE STACK FROM INTERRUPT

```

```

STOP1: JSR PC, SUBCLR
      ERROR 24 ;CERR AFTER

```

```

      TST UNLD ;SEE IF HEADS UNLOADED
      BEQ 3$ ;BR IF NO
      TST 42 ;SEE IF MANUAL OR AUTO MODE
      BEQ 1$ ;BR IF MANUAL MODE
      TYPE MSG74 ;PGM ABORT PENDING
      BR 2$
1$: TYPE ,MSG75 ;HALT PENDING
2$:

```

6080	030432	004737	026332		JSR	PC, SUBCLR	
6081	030436	104024			ERROR	24	;CERR AFTER SCLR
6082							
6083	030440	032737	010000	005434	BIT	#D.SPIN, HMR2	;SEE IF SPINDLE ALREADY ON
6084	030446	001023			BNE	64\$;BR IF YES
6085	030450	104401	037741		TYPE	,MSG29	;PLEASE WAIT, HEADS BEING LOADED
6086							
6087	030454	012765	000011	000000	MOV	#SRTSPL, RKCS1(R5)	;START SPINDLE CMD
6088	030462	013737	001426	005444	MOV	T10, TEMP1	
6089	030470	004737	024422		JSR	PC, FRDY	;FIND CONTR RDY
6090	030474	104143			ERROR	143	;CONTR RDY NOT SET AFTER CMD
6091							
6092	030476	013737	001434	005446	MOV	T100, TEMP2	
6093	030504	004737	024736		JSR	PC, FATT1	;FIND ATTN
6094	030510	104144			ERROR	144	;NO ATTN AFTER CMD
6095							
6096	030512	005037	005370		CLR	UNLD	
6097	030516						64\$:
6098							
6099	030516	005737	005372		TST	BADHDR	;SEE IF HEADERS VALID
6100	030522	001466			BEQ	4\$;BR IF YES
6101	030524	005237	005374		INC	HPEND	
6102	030530	012765	100000	000000	MOV	#CCLR, RKCS1(R5)	
6103	030536	013765	001222	000010	MOV	\$UNIT, RKCS2(R5)	
6104	030544	012765	000013	000000	MOV	#RECAL, RKCS1(R5)	;RECAL CMD
6105							;RESET CYL DIFF/OFFSET & CYL ADDR REG
6106							;IN RKMR2 & RKMR3 RESP.
6107	030552	013737	001426	005444	MOV	T10, TEMP1	;SETUP TIMEOUT
6108	030560	004737	024422		JSR	PC, FRDY	;FIND RDY
6109	030564	104124			ERROR	124	;RDY NOT SET AFTER RECAL CMD
6110							
6111	030566	012765	000001	000026	MOV	#1, RKMR1(R5)	;SELECT WORD 1
6112	030574	004737	025700		JSR	PC, GSTAT	
6113	030600	032737	020000	005434	BIT	#D.RTZ, HMR2	
6114	030606	001001			BNE	65\$	
6115	030610	104214			ERROR	214	;RTZ NOT SET DURING RECAL CMD
6116	030612	013737	001426	005446	MOV	T10, TEMP2	;SETUP TIMEOUT
6117	030620	004737	024736		JSR	PC, FATT1	;FIND ATTN
6118	030624	104055			ERROR	55	;NO ATTN AFTER RECAL CMD
6119							
6120	030626	012765	100000	000000	MOV	#CCLR, RKCS1(R5)	
6121	030634	013765	001222	000010	MOV	\$UNIT, RKCS2(R5)	;DRIVE#
6122	030642	012765	000005	000000	MOV	#CLEAR, RKCS1(R5)	;DRIVE CLEAR CMD
6123	030650	013737	001426	005444	MOV	T10, TEMP1	;SETUP TIMEOUT
6124	030656	004737	024422		JSR	PC, FRDY	;FIND RDY
6125	030662	104151			ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD
6126	030664	004737	024704		JSR	PC, TSTATN	;TEST FOR ATTN
6127	030670	000401			BR	66\$	
6128	030672	104154			ERROR	154	;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6129	030674						66\$:
6130							
6131							
6132	030674	000137	030740		JMP	FORM	;WRITE VALID FORMATS
6133							
6134	030700	005737	000042		TST	42	;SEE IF MANUAL OR AUTO MODE
6135	030704	001410			BEQ	5\$;BR IF MANUAL MODE


```

6192 031056 000002          RTI          ;ELSE RETURN
6193
6194 031060 012706 001100  25:  MOV      #STACK,SP      ;INIT STACK
6195 031064 000177 150020  JMP      @SLPERR
6196
6197          .SBTTL  RK06 INTERRUPT HANDLER
6198
6199 031070 000240  INTER:  NOP
6200 031072 000240  NOP
6201 031074 000240  NOP
6202 031076 011600  MOV      (SP),RO      ;SAVE PC WHERE INT OCCURRED.
6203 031100 005740  TST     -(RO)        ;GET PC BEFORE UPDATE.
6204 031102 104401 036713  TYPE    MSG6         ;INT AT PC=
6205 031106 010046  MOV     RO,-(SP)     ;SAVE RO FOR TYPEOUT
6206                                     ;TYPE PC
6207 031110 104403  TYPOS   GO TYPE--OCTAL ASCII
6208 031112      006   .BYTE   6            ;TYPE 6 DIGIT(S)
6209 031113      000   .BYTE   0            ;SUPPRESS LEADING ZEROS
6210 031114 000000  HALT
6211 031116 000240  NOP
6212 031120 000240  NOP
6213 031122 000002  RTI
6214
6215          .SBTTL  POWER DOWN AND UP ROUTINES
6216
6217          ;POWER DOWN ROUTINE
6218
6219 031124 012737 031136 000024  SPWRDN: MOV     #SPWRUP,PWRVEC ;SET UP VECTOR
6220 031132 000000  HALT
6221 031134 000776  BR      -2          ;HANG UP.
6222
6223          ;POWER UP ROUTINE
6224
6225 031136 005037 031210  SPWRUP: CLR     SPWRCT      ;WAIT LOOP FOR TTY
6226 031142 005237 031210  15:  INC     SPWRCT      ;WAIT FOR THE INCR
6227 031146 001375  BNE     15          ;OF WORD
6228 031150 012737 031124 000024  MOV     #SPWRDN,PWRVEC ;SET POWER DOWN VECTOR
6229 031156 012737 000340 000026  MOV     #PR7,PWRVEC+2 ;PRIORITY 7
6230 031164 012737 000340 000036  MOV     #PR7,TRAPVEC+2 ;LOCKOUT ALL INTERRUPTS FOR TRAPS
6231 031172 012706 001100  MOV     #STACK,SP    ;INITIALIZE STACK
6232 031176 104401 037107  TYPE    ,MSG11      ;REPORT POWER FAIL
6233 031202 000005  RESET
6234 031204 000137 012440  JMP     PFSRT
6235
6236 031210 000000  SPWRCT: 0          ;WAIT COUNT FOR TTY
6237
6238          ;DIVISION UTILITY ROUTINE
6239          ;R0-R1-R2-R3=DIVIDEND
6240          ;R4-R5=DIVISOR
6241          ;R0-R1=REMAINDER AFTER DIVISION
6242          ;R2-R3=QUOTIENT AFTER DIVISION
6243          ;ENTER WITH JSR PC,M.DPID
6244
6245          ;
6246          ;
6247 031212 012746 000040  M.DPID: MOV     #40,-(SP) ;COUNTER FOR DIVISION CYCLES

```

6248	031216	010446		MOV	R4, -(SP)	;HI ORDER
6249	031220	010546		MOV	R5, -(SP)	;LO ORDER TO THE STACK
6250	031222	005466	000002	NEG	2(SP)	;FORM NEGATIVE
6251	031226	005416		NEG	2SP	;VERSION OF DIVISOR
6252	031230	005666	000002	SBC	2(SP)	
6253	031234	061601		ADD	2SP, R1	
6254	031236	005500		ADC	RO	;PERFORM INIT SUBT.
6255	031240	066600	000002	ADD	2(SP), RO	
6256	031244	103445		BCS	M.DP50	;IF CARRY THEN OVERFLOW HAS OCCURRED
6257	031246	005046		CLR	-(SP)	;THIS IS A LONGER LASTING CARRY BIT
6258	031250	006103		M.DP40: ROL	R3	
6259	031252	006102		ROL	R2	
6260	031254	006101		ROL	R1	
6261	031256	006100		ROL	RO	
6262	031260	005716		TST	2SP	;TEST CARRY INDICATOR
6263	031262	001410		BEQ	M.DP41	;IF TO CARRY THEN ADD, ELSE SUBT.
6264	031264	005016		CLR	2SP	;CLEAR UP FOR NEXT TIME
6265	031266	066601	000002	ADD	2(SP), R1	
6266	031272	005500		ADC	RO	;ADD -(DIVISOR)
6267	031274	005516		ADC	2SP	;SET CARRY
6268	031276	066600	000004	ADD	4(SP), RO	
6269	031302	000404		BR	M.DP42	
6270						
6271	031304	060501		M.DP41: ADD	R5, R1	
6272	031306	005500		ADC	RO	;ADD +(DIVISOR)
6273	031310	005516		ADC	2SP	;SET CARRY
6274	031312	060400		ADD	R4, RO	
6275	031314	005516		M.DP42: ADC	2SP	;SET CARRY
6276	031316	005716		TST	2SP	;TEST THE UPDATE INDICATOR
6277	031320	001401		BEQ	.+4	;IF 0 FORGET IT
6278	031322	005203		INC	R3	;NO CARRY POSSIBLE HERE
6279	031324	005366	000006	DEC	6(SP)	;DECREMENT CTR
6280	031330	003347		BGT	M.DP40	;BR IF MORE TO DO
6281	031332	006003		ROR	R3	
6282	031334	103404		BCS	M.DP44	
6283	031336	060501		ADD	R5, R1	
6284	031340	005500		ADC	RO	
6285	031342	060400		ADD	R4, RO	
6286	031344	000241		CLC		
6287						
6288	031346	006103		M.DP44: ROL	R3	
6289	031350	062706	000010	ADD	#10, SP	;ADJUST STACK BY 4 WORDS
6290	031354	000242		CLV		
6291	031356	000207		RTS	PC	
6292						
6293	031360	062706	000006	M.DP50: ADD	#6, SP	
6294	031364	000262		SEV		
6295	031366	000207		RTS	PC	
6296						

.SBTTL SCOPE HANDLER ROUTINE

```

6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311 031370
6312 031370 104407
6313 031372 032777 040000 147540
6314 031400 001114
6315
6316 031402 000416
6317
6318 031404 013746 000004
6319 031410 012737 031430 000004
6320 031416 005737 177060
6321 031422 012637 000004
6322 031426 000463
6323 031430 022626
6324 031432 012637 000004
6325 031436 000423
6326 031440
6327 031440 032777 000400 147472
6328 031446 001404
6329 031450 127737 147464 001102
6330 031456 001465
6331 031460 105737 001103
6332 031464 001421
6333 031466 123737 001115 001103
6334 031474 101015
6335 031476 032777 001000 147434
6336 031504 001404
6337 031506 013737 001110 001106
6338 031514 000446
6339 031516 105037 001103
6340 031522 005037 001174
6341 031526 000415
6342 031530 032777 004000 147402
6343 031536 001011
6344 031540 005737 001216
6345 031544 001406
6346 031546 005237 001104
6347 031552 023737 001174 001104
6348 031560 002024
6349 031562 012737 000001 001104
6350 031570 013737 031646 001174
6351 031576 105237 001102
6352 031602 113737 001102 C01214

```

```

*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
*AND LOAD THE ERROR FLAG (SERFLG) 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
        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*****
$XTSTR: BR      6$
        MOV     @#ERRVEC,-(SP)        ;;IF RUNNING ON THE "XOR" TESTER CHANGE
        MOV     @#ERRVEC              ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
        TST    @#177060              ;;SAVE THE CONTENTS OF THE ERROR VECTOR
        MOV     (SP)+,@#ERRVEC        ;;SET FOR TIMEOUT
        BR     $$VLAD                ;;TIME OUT ON XOR?
        BR     $S                    ;;RESTORE THE ERROR VECTOR
        CMP     (SP)+,(SP)+          ;;GO TO THE NEXT TEST
        MOV     (SP)+,@#ERRVEC        ;;CLEAR THE STACK AFTER A TIME OUT
        BR     7$                    ;;RESTORE THE ERROR VECTOR
        ;;*****END OF CODE FOR THE XOR TESTER*****
        BIT     #BIT08,@SWR          ;;LOOP ON SPEC. TEST?
        BEQ    2$                    ;;BR IF NO
        CMPB   @SWR,$TSTNM           ;;ON THE RIGHT TEST? SWR<7:0>
        BEQ    $OVER                ;;BR IF YES
        TSTB   $SERFLG              ;;HAS AN ERROR OCCURRED?
        BEQ    3$                    ;;BR IF NO
        CMPB   $SERMAX,$SERFLG       ;;MAX. ERRORS FOR THIS TEST OCCURRED?
        BHI    3$                    ;;BR IF NO
        BIT     #BIT09,@SWR          ;;LOOP ON ERROR?
        BEQ    4$                    ;;BR IF NO
        MOV     $LPERR,$LPAOR        ;;SET LOOP ADDRESS TO LAST SCOPE
        BR     $OVER
        CLRB   $SERFLG              ;;ZERO THE ERROR FLAG
        CLR    $TIMES                ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
        BR     1$                    ;;ESCAPE TO THE NEXT TEST
        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    $OVER                ;;BR IF MORE ITERATION REQUIRED
        MOV    #1,$ICNT              ;;REINITIALIZE THE ITERATION COUNTER
        MOV    $MXCNT,$TIMES         ;;SET NUMBER OF ITERATIONS TO DO
        INCB   $TSTNM               ;;COUNT TEST NUMBERS
        MOVB  $TSTNM,$TESTN         ;;SET TEST NUMBER IN APT MAILBOX

```


6353	031610	011637	001106		MOV	(SP), \$LPADR	:: SAVE SCOPE LOOP ADDRESS	
6354	031614	011637	001110		MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS	
6355	031620	005037	001176		CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS	
6356	031624	112737	000001	001115	MOVB	#1, \$ERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST	
6357	031632	013777	001102	147302	\$OVER	MOV	\$STNM, @DISPLAY	:: DISPLAY TEST NUMBER
6358	031640	013716	001106		MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS	
6359	031644	000002			RTI		:: FIXES PS	
6360	031646	003720			\$MXCNT:	2000.	:: MAX. NUMBER OF ITERATIONS	
6361					.SBTTL	ERROR HANDLER ROUTINE		
6362								
6363								
6364								
6365								
6366								
6367								
6368								
6369								
6370								
6371								
6372								
6373								
6374								
6375	031650				\$ERROR:			
6376	031650	104407			CKSWR		:: TEST FOR CHANGE IN SOFT-SWR	
6377	031652	105237	001103		7\$:	INCB	\$ERFLG	:: SET THE ERROR FLAG
6378	031656	001775				BEQ	7\$:: DON'T LET THE FLAG GO TO ZERO
6379	031660	013777	001102	147254		MOV	\$STNM, @DISPLAY	:: DISPLAY TEST NUMBER AND ERROR FLAG
6380	031666	032777	002000	147244		BIT	#BIT10, @SWR	:: BELL ON ERROR?
6381	031674	001402				BEQ	1\$:: NO - SKIP
6382	031676	104401	001200			TYPE	\$BELL	:: RING BELL
6383	031702	005237	001112		1\$:	INC	\$ERTTL	:: COUNT THE NUMBER OF ERRORS
6384	031706	011637	001116			MOV	(SP), \$ERRPC	:: GET ADDRESS OF ERROR INSTRUCTION
6385	031712	162737	000002	001116		SUB	#2, \$ERRPC	
6386	031720	117737	147172	001114		MOVB	@\$ERRPC, \$ITEMB	:: STRIP AND SAVE THE ERROR ITEM CODE
6387	031726	032777	020000	147204		BIT	#BIT13, @SWR	:: SKIP TYPEOUT IF SET
6388	031734	001004				BNE	20\$:: SKIP TYPEOUTS
6389	031736	004737	054772			JSR	PC, TYPERR	:: GO TO USER ERROR ROUTINE
6390	031742	104401	001205			TYPE	, \$CRLF	
6391	031746				20\$:			
6392	031746	122737	000001	001230		CMPB	#APTENV, \$ENV	:: RUNNING IN APT MODE
6393	031754	001007				BNE	2\$:: NO SKIP APT ERROR REPORT
6394	031756	113737	001114	031770		MOVB	\$ITEMB, 21\$:: SET ITEM NUMBER AS ERROR NUMBER
6395	031764	004737	032574			JSR	PC, \$ATY4	:: REPORT FATAL ERROR TO APT
6396	031770	000			21\$:	.BYTE	0	
6397	031771	000				.BYTE	0	
6398	031772	000777			22\$:	BR	22\$:: APT ERROR LOOP
6399	031774	005777	147140		2\$:	TST	@SWR	:: HALT ON ERROR
6400	032000	100002				BPL	3\$:: SKIP IF CONTINUE
6401	032002	000000				HALT		:: HALT ON ERROR!
6402	032004	104407				CKSWR		:: TEST FOR CHANGE IN SOFT-SWR
6403	032006	032777	001000	147124	3\$:	BIT	#BIT09, @SWR	:: LOOP ON ERROR SWITCH SET?
6404	032014	001402				BEQ	4\$:: BR IF NO
6405	032016	013716	001110			MOV	\$LPERR, (SP)	:: FUDGE RETURN FOR LOOPING
6406	032022	005737	001176		4\$:	TST	\$ESCAPE	:: CHECK FOR AN ESCAPE ADDRESS
6407	032026	001402				BEQ	5\$:: BR IF NONE
6408	032030	013716	001176			MOV	\$ESCAPE, (SP)	:: FUDGE RETURN ADDRESS FOR ESCAPE

```

6409 032034
6410 032034 022737 024030 000042
6411 032042 001001
6412 032044 000000
6413 032046
6414 032046 000002
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432 032050 105737 001157
6433 032054 100002
6434 032056 000000
6435 032060 000430
6436 032062 010046
6437 032064 017600 000002
6438 032070 122737 000001 001230
6439 032076 001011
6440 032100 132737 000100 001231
6441 032106 001405
6442 032110 010037 032120
6443 032114 004737 032564
6444 032120 000000
6445 032122 132737 000040 001231
6446 032130 001003
6447 032132 112046
6448 032134 001005
6449 032136 005726
6450 032140 012600
6451 032142 062716 000002
6452 032146 000002
6453 032150 122716 000011
6454 032154 001430
6455 032156 122716 000200
6456 032162 001006
6457 032164 005726
6458 032166 104401
6459 032170 001205
6460 032172 105037 032326
6461 032176 000755
6462 032200 004737 032262
6463 032204 123726 001156
6464 032210 001350

```

```

5$:      CMP      #SENDAD,2#42      ;;ACT-11 AUTO-ACCEPT?
        BNE      6$                ;;BRANCH IF NO
        HALT                      ;;YES
6$:      RTI                          ;;RETURN
.SBTTL   TYPE ROUTINE

;*****
;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
;NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
;
;CALL:
;1) USING A TRAP INSTRUCTION
;      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;OR
;      TYPE
;      MESADR
;
$TYPE:   TSTB      $TPFLG      ;; IS THERE A TERMINAL?
        BPL      1$          ;; BR IF YES
        HALT      HERE IF NO TERMINAL
        BR       3$          ;; LEAVE
1$:      MOV      RO,-(SP)      ;; SAVE RO
        MOV      22(SP),RO     ;; GET ADDRESS OF ASCIZ STRING
        CMPB     #APTENV,$ENV   ;; RUNNING IN APT MODE
        BNE     62$          ;; NO GO CHECK FOR APT CONSOLE
        BITB     #APTSPool,$ENVM ;; SPOOL MESSAGE TO APT
        BEQ     62$          ;; NO GO CHECK FOR CONSOLE
        MOV      RO,61$        ;; SETUP MESSAGE ADDRESS FOR APT
        JSR     PC,$ATY3      ;; SPOOL MESSAGE TO APT
        .WORD    0            ;; MESSAGE ADDRESS
        BITB     #APTCSUP,$ENVM ;; APT CONSOLE SUPPRESSED
        BNE     60$          ;; YES, SKIP TYPE OUT
        MOVB     (RO)+,-(SP)   ;; PUSH CHARACTER TO BE TYPED ONTO STACK
        BNE     4$           ;; BR IF IT ISN'T THE TERMINATOR
        TST     (SP)+         ;; IF TERMINATOR POP IT OFF THE STACK
        MOV      (SP)+,RO     ;; RESTORE RO
        ADD     #2,(SP)       ;; ADJUST RETURN PC
        RTI                    ;; RETURN
        CMPB     #HT,(SP)     ;; BRANCH IF <HT>
        BEQ     8$           ;;
        CMPB     #CRLF,(SP)   ;; BRANCH IF NOT <CRLF>
        BNE     5$           ;;
        TST     (SP)+         ;; POP <CR><LF> EQUIV
        TYPE     ;; TYPE A CR AND LF
        CLRB     $CHARCNT     ;; CLEAR CHARACTER COUNT
        BR       2$          ;; GET NEXT CHARACTER
        JSR     PC,$TYPEC     ;; GO TYPE THIS CHARACTER
        CMPB     $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
        BNE     2$          ;; IF NO GO GET NEXT CHAR.

```

```

6465 032212 013746 001154          MOV      $NULL,-(SP)          ;; GET # OF FILLER CHARS. NEEDED
6466                                     ;; AND THE NULL CHAR.
6467 032216 105366 000001      7$:     DECB      1(SP)          ;; DOES A NULL NEED TO BE TYPED?
6468 032222 002770                                     BLT      6$                    ;; BR IF NO--GO POP THE NULL OFF OF STACK
6469 032224 004737 032262          JSR      PC,$TYPEC          ;; GO TYPE A NULL
6470 032230 105337 032326          DECB      $CHARCNT          ;; DO NOT COUNT AS A COUNT
6471 032234 000770          BR       7$                    ;; LOOP
6472
6473                                     ;HORIZONTAL TAB PROCESSOR
6474
6475 032236 112716 000040      8$:     MOVVB     #' ,(SP)          ;; REPLACE TAB WITH SPACE
6476 032242 004737 032262      9$:     JSR      PC,$TYPEC          ;; TYPE A SPACE
6477 032246 132737 000007 032326  BITB     #',$CHARCNT          ;; BRANCH IF NOT AT
6478 032254 001372                                     BNE     9$                    ;; TAB STOP
6479 032256 005726          TST      (SP)+              ;; POP SPACE OFF STACK
6480 032260 000724          BR       2$                    ;; GET NEXT CHARACTER
6481 032262 105777 146662      $TYPEC: TSTB     @$TPS          ;; WAIT UNTIL PRINTER IS READY
6482 032266 100375          BPL     $TYPEC
6483 032270 116677 000002 146654  MOVVB     2(SP),@STPB          ;; LOAD CHAR TO BE TYPED INTO DATA REG.
6484 032276 122766 000015 000002  CMPB     #CR,2(SP)          ;; IS CHARACTER A CARRIAGE RETURN?
6485 032304 001003          BNE     1$                    ;; BRANCH IF NO
6486 032306 105037 032326          CLRB     $CHARCNT          ;; YES--CLEAR CHARACTER COUNT
6487 032312 000406          BR       $TYPEX              ;; EXIT
6488 032314 122766 000012 000002  1$:     CMPB     #LF,2(SP)          ;; IS CHARACTER A LINE FEED?
6489 032322 001402          BEQ     $TYPEX              ;; BRANCH IF YES
6490 032324 105227          INCB     (PC)+              ;; COUNT THE CHARACTER
6491 032326 000000      $CHARCNT: .WORD 0          ;; CHARACTER COUNT STORAGE
6492 032330 000207      $TYPEX:  RTS      PC
6493
6494                                     .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6495
6496                                     ;*****
6497                                     ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
6498                                     ;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
6499                                     ;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
6500                                     ;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
6501                                     ;REPLACED WITH SPACES.
6502                                     ;CALL:
6503                                     ;
6504                                     ;      MOV      NUM,-(SP)          ;; PUT THE BINARY NUMBER ON THE STACK
6505                                     ;      TYPDS          ;; GO TO THE ROUTINE
6506
6506 032332                                     $TYPDS:
6507 032332 010046          MOV      R0,-(SP)          ;; PUSH R0 ON STACK
6508 032334 010146          MOV      R1,-(SP)          ;; PUSH R1 ON STACK
6509 032336 010246          MOV      R2,-(SP)          ;; PUSH R2 ON STACK
6510 032340 010346          MOV      R3,-(SP)          ;; PUSH R3 ON STACK
6511 032342 010546          MOV      R5,-(SP)          ;; PUSH R5 ON STACK
6512 032344 012746 020200          MOV      #20200,-(SP)      ;; SET BLANK SWITCH AND SIGN
6513 032350 016605 000020          MOV      20(SP),R5          ;; GET THE INPUT NUMBER
6514 032354 100004          BPL     1$                    ;; BR IF INPUT IS POS.
6515 032356 005405          NEG      R5                    ;; MAKE THE BINARY NUMBER POS.
6516 032360 112766 000055 000001  MOVVB     #'-,1(SP)          ;; MAKE THE ASCII NUMBER NEG.
6517 032366 005000      1$:     CLR      R0                    ;; ZERO THE CONSTANTS INDEX
6518 032370 012703 032546          MOV      #SDBLK,R3          ;; SETUP THE OUTPUT POINTER
6519 032374 112723 000040          MOVVB     #' ,(R3)+          ;; SET THE FIRST CHARACTER TO A BLANK
6520 032400 005002      2$:     CLR      R2                    ;; CLEAR THE BCD NUMBER

```


6577	032634	017600	000004		MOV	24(SP),RO	:: GET MESSAGE ADDR.
6578	032640	062766	000002	000004	ADD	#2,4(SP)	:: BUMP RETURN ADDR.
6579	032646	005737	001210		1S: TST	\$MSGTYPE	:: SEE IF DONE W/ LAST XMISSION?
6580	032652	001375			BNE	1S	:: IF NOT: WAIT
6581	032654	010037	001224		MOV	RO,\$MSGAD	:: PUT ADDR IN MAILBOX
6582	032660	105720			2S: TSTB	(RO)+	:: FIND END OF MESSAGE
6583	032662	001376			BNE	2S	
6584	032664	163700	001224		SUB	\$MSGAD,RO	:: SUB START OF MESSAGE
6585	032670	006200			ASR	RO	:: GET MESSAGE LNTH IN WORDS
6586	032672	010037	001226		MOV	RO,\$MSGLGT	:: PUT LENGTH IN MAILBOX
6587	032676	012737	000004	001210	MOV	#4,\$MSGTYPE	:: TELL APT TO TAKE MSG.
6588	032704	000413			BR	5S	
6589	032706	017637	000004	032732	3S: MOV	24(SP),4S	:: PUT MSG ADDR IN JSR LINKAGE
6590	032714	062766	000002	000004	ADD	#2,4(SP)	:: BUMP RETURN ADDRESS
6591	032722	013746	177776		MOV	177776,-(SP)	:: PUSH 177776 ON STACK
6592	032726	004737	032050		JSR	PC,\$TYPE	:: CALL TYPE MACRO
6593	032732	000000			4S: .WORD	0	
6594	032734				5S:		
6595	032734	105737	033022		10S: TSTB	\$FFLG	:: SHOULD REPORT FATAL ERROR?
6596	032740	001416			BEQ	12S	:: IF NOT: BR
6597	032742	005737	001230		TST	\$ENV	:: RUNNING UNDER APT?
6598	032746	001413			BEQ	12S	:: IF NOT: BR
6599	032750	005737	001210		11S: TST	\$MSGTYPE	:: FINISHED LAST MESSAGE?
6600	032754	001375			BNE	11S	:: IF NOT: WAIT
6601	032756	017637	000004	001212	MOV	24(SP),\$FATAL	:: GET ERROR #
6602	032764	062766	000002	000004	ADD	#2,4(SP)	:: BUMP RETURN ADDR.
6603	032772	005237	001210		INC	\$MSGTYPE	:: TELL APT TO TAKE ERROR
6604	032776	105037	033022		12S: CLRB	\$FFLG	:: CLEAR FATAL FLAG
6605	033002	105037	033021		CLRB	\$LFLG	:: CLEAR LOG FLAG
6606	033006	105037	033020		CLRB	\$MFLG	:: CLEAR MESSAGE FLAG
6607	033012	012601			MOV	(SP)+,R1	:: POP STACK INTO R1
6608	033014	012600			MOV	(SP)+,RO	:: POP STACK INTO RO
6609	033016	000207			RTS	PC	:: RETURN
6610	033020	000			\$MFLG: .BYTE	0	:: MESSG. FLAG
6611	033021	000			\$LFLG: .BYTE	0	:: LOG FLAG
6612	033022	000			\$FFLG: .BYTE	0	:: FATAL FLAG

```

6613      033024
6614      000200
6615      000001
6616      000100
6617      000040
6618      APTSIZE=200
6619      APTENV=001
6620      APTSPool=100
6621      APTCSUP=040
6622      .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
6623      *****
6624      *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
6625      *OCTAL (ASCII) NUMBER AND TYPE IT.
6626      *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
6627      *CALL:
6628      *      MOV      NUM,-(SP)      :: NUMBER TO BE TYPED
6629      *      TYPOS
6630      *      .BYTE  N                :: CALL FOR TYPEOUT
6631      *      .BYTE  M                :: N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
6632      *                                          :: M=1 OR 0
6633      *                                          :: 1=TYPE LEADING ZEROS
6634      *                                          :: 0=SUPPRESS LEADING ZEROS
6635      *
6636      *$STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST

```

```

6633 ;*STYPOS OR STYPOC
6634 ;*CALL:
6635 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
6636 ;*      TYPON      ;;CALL FOR TYPEOUT
6637 ;*
6638 ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
6639 ;*CALL:
6640 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
6641 ;*      TYPOC      ;;CALL FOR TYPEOUT
6642
6643 033024 017646 000000 STYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
6644 033030 116637 000001 033247 MOVVB 1(SP),%OFILL ;;LOAD ZERO FILL SWITCH
6645 033036 112637 033251 MOVVB (SP)+,%OMODE+1 ;;NUMBER OF DIGITS TO TYPE
6646 033042 062716 000002 ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
6647 033046 000406 BR      $TYPON
6648 033050 112737 000001 033247 STYPOC: MOVVB #1,%OFILL ;;SET THE ZERO FILL SWITCH
6649 033056 112737 000006 033251 MOVVB #6,%OMODE+1 ;;SET FOR SIX(6) DIGITS
6650 033064 112737 000005 033246 STYPON: MOVVB #5,%OCNT ;;SET THE ITERATION COUNT
6651 033072 010346 MOV      R3,-(SP)      ;;SAVE R3
6652 033074 010446 MOV      R4,-(SP)      ;;SAVE R4
6653 033076 010546 MOV      R5,-(SP)      ;;SAVE R5
6654 033100 113704 033251 MOVVB %OMODE+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
6655 033104 005404 NEG      R4
6656 033106 062704 000006 ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
6657 033112 110437 033250 MOVVB R4,%OMODE ;;SAVE IT FOR USE
6658 033116 113704 033247 MOVVB %OFILL,R4 ;;GET THE ZERO FILL SWITCH
6659 033122 016605 000012 MOV      12(SP),R5 ;;PICKUP THE INPUT NUMBER
6660 033126 005003 CLR      R3      ;;CLEAR THE OUTPUT WORD
6661 033130 006105 1$: ROL      R5      ;;ROTATE MSB INTO "C"
6662 033132 000404 BR      3$      ;;GO DO MSB
6663 033134 006105 2$: ROL      R5      ;;FORM THIS DIGIT
6664 033136 006105 ROL      R5
6665 033140 006105 ROL      R5
6666 033142 010503 MOV      R5,R3
6667 033144 006103 3$: ROL      R3      ;;GET LSB OF THIS DIGIT
6668 033146 105337 033250 DECB    %OMODE ;;TYPE THIS DIGIT?
6669 033152 100016 BPL     7$      ;;BR IF NO
6670 033154 042703 177770 BIC     #177770,R3 ;;GET RID OF JUNK
6671 033160 001002 BNE     4$      ;;TEST FOR 0
6672 033162 005704 TST     R4      ;;SUPPRESS THIS 0?
6673 033164 001403 BEQ     5$      ;;BR IF YES
6674 033166 005204 4$: INC     R4      ;;DON'T SUPPRESS ANYMORE 0'S
6675 033170 052703 BIS     #'0,R3 ;;MAKE THIS DIGIT ASCII
6676 033174 052703 5$: BIS     #' ,R3 ;;MAKE ASCII IF NOT ALREADY
6677 033200 110337 033244 MOVVB  R3,%S      ;;SAVE FOR TYPING
6678 033204 104401 033244 TYPE   ,%S      ;;GO TYPE THIS DIGIT
6679 033210 105337 033246 7$: DECB    %OCNT ;;COUNT BY 1
6680 033214 003347 BGT     2$      ;;BR IF MORE TO DO
6681 033216 002402 BLT     6$      ;;BR IF DONE
6682 033220 005204 INC     R4      ;;INSURE LAST DIGIT ISN'T A BLANK
6683 033222 000744 BR      2$      ;;GO DO THE LAST DIGIT
6684 033224 012605 6$: MOV     (SP)+,R5 ;;RESTORE R5
6685 033226 012604 MOV     (SP)+,R4 ;;RESTORE R4
6686 033230 012603 MOV     (SP)+,R3 ;;RESTORE R3
6687 033232 016666 000002 000004 MOV     2(SP),4(SP) ;;SET THE STACK FOR RETURNING
6688 033240 012616 MOV     (SP)+,(SP)

```

```

6689 033242 000002
6690 033244 000
6691 033245 000
6692 033246 000
6693 033247 000
6694 033250 000000
6695
6696
6697
6698
6699 033252 000000
6700 033254 000000
6701 033256 000000
6702 033260 000001
6703 033261 033261
6704 033262 033262
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714 033262 005037 033252
6715 033266 012737 033260 033254
6716 033274 013737 033254 033256
6717 033302 012737 033332 000060
6718 033310 012737 000200 000062
6719 033316 005777 145624
6720 033322 012777 000100 145614
6721 033330 000207
6722
6723
6724
6725
6726
6727
6728
6729
6730 033332 117746 145610
6731 033336 042716 177600
6732 033342 021627 000003
6733 033346 001007
6734 033350 104401 034460
6735 033354 004737 033262
6736 033360 005726
6737 033362 000137 030374
6738 033366 021627 000007
6739 033372 001004
6740 033374 022737 000176 001140
6741 033402 001500
6742
6743 033404
6744 033404 022737 000001 033252

```

```

RTI ;: RETURN
BS: .BYTE 0 ;: STORAGE FOR ASCII DIGIT
;: 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 #↑C17,(SP) ;: STRIP THE JUNK
CMP (SP),#3 ;: IS IT A CONTROL C?
BNE IS ;: 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
IS: 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?

```

K10

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 126
TTY INPUT ROUTINE

SEQ 0126

6745	033412	001004			BNE	3\$:: BRANCH IF NO
6746	033414	104401	001200		TYPE	, \$BELL	:: RING THE TTY BELL
6747	033420	005726			TST	(SP)+	:: CLEAN CHARACTER OFF OF STACK
6748	033422	000451			BR	5\$:: EXIT
6749	033424	021627	000023		3\$: CMP	(SP), #23	:: IS IT A CONTROL-S?
6750	033430	001021			BNE	32\$:: BRANCH IF NO
6751	033432	005077	145506		CLR	2\$TKS	:: DISABLE TTY KEYBOARD INTERRUPTS
6752	033436	005726			TST	(SP)+	:: CLEAN CHAR OFF STACK
6753	033440	105777	145500		31\$: TSTB	2\$TKS	:: WAIT FOR A CHAR
6754	033444	100375			BPL	31\$:: LOOP UNTIL ITS THERE
6755	033446	117746	145474		MOV B	2\$TKB, -(SP)	:: GET THE CHARACTER
6756	033452	042716	177600		BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
6757	033456	022627	000021		CMP	(SP)+, #21	:: IS IT A CONTROL-Q?
6758	033462	001366			BNE	31\$:: BRANCH IF NO
6759	033464	012777	000100	145452	MOV	#100, 2\$TKS	:: REENABLE TTY KEYBOARD INTERRUPTS
6760	033472	001002			RTI		:: RETURN
6761	033474	005237	033252		32\$: INC	\$TKCNT	:: COUNT THIS CHARACTER
6762	033500	021627	000140		CMP	(SP), #140	:: IS IT UPPER CASE?
6763	033504	002405			BLT	4\$:: BRANCH IF YES
6764	033506	021627	000175		CMP	(SP), #175	:: IS IT A SPECIAL CHAR?
6765	033512	003002			BGT	4\$:: BRANCH IF YES
6766	033514	042716	000040		BIC	#40, (SP)	:: MAKE IT UPPER CASE
6767	033520	112677	177530		4\$: MOV B	(SP)+, 2\$TKQIN	:: AND PUT IT IN QUEUE
6768	033524	005237	033254		INC	\$TKQIN	:: UPDATE THE POINTER
6769	033530	023727	033254	033261	CMP	\$TKQIN, #2\$TKQEND	:: GO OFF THE END?
6770	033536	001003			BNE	5\$:: BRANCH IF NO
6771	033540	012737	033260	033254	MOV	#2\$TKQSR, \$TKQIN	:: RESET THE POINTER
6772	033546	000002			5\$: RTI		:: RETURN
6773							
6774							
6775							
6776							
6777							
6778							
6779	033550	022737	000176	001140	\$CKSWR: CMP	#SWREG, SWR	:: IS THE SOFT-SWR SELECTED
6780	033556	001124			BNE	15\$:: EXIT IF NOT
6781	033560	105777	145360		TSTB	2\$TKS	:: IS A CHAR WAITING?
6782	033564	100121			BPL	15\$:: IF NOT, EXIT
6783	033566	117746	145354		MOV B	2\$TKB, -(SP)	:: YES
6784	033572	042716	177600		BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
6785	033576	021627	000007		CMP	(SP), #7	:: IS IT A CONTROL-G?
6786	033602	001300			BNE	2\$:: IF NOT, PUT IT IN THE TTY QUEUE
6787							:: AND EXIT
6788							
6789							
6790							
6791							
6792							
6793	033604	123727	001134	000001	6\$: CMPB	\$AUTOB, #1	:: ARE WE RUNNING IN AUTO-MODE?
6794	033612	001674			BEQ	2\$:: BRANCH IF YES
6795	033614	005726			TST	(SP)+	:: CLEAR CONTROL-G OFF STACK
6796	033616	004737	033262		JSR	PC, \$TKINT	:: FLUSH THE TTY INPUT QUEUE
6797	033622	005077	145316		CLR	2\$TKS	:: DISABLE TTY KEYBOARD INTERRUPTS
6798	033626	112737	000001	001135	MOV B	#1, \$INTAG	:: SET INTERRUPT MODE INDICATOR
6799							
6800	033634	104401	034472		TYPE	, \$CNTLG	:: ECHO THE CONTROL-G (+G)

6801	033640	104401	034477	SGTSWR:	TYPE	\$MSWR	:: TYPE CURRENT CONTENTS
6802	033644	013746	000176		MOV	\$WREG, -(SP)	:: SAVE SWREG FOR TYPEOUT
6803	033650	104402			TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
6804	033652	104401	034510		TYPE	, \$MNEW	:: PROMPT FOR NEW SWR
6805	033656	005046		19\$:	CLR	-(SP)	:: CLEAR COUNTER
6806	033660	005046			CLR	-(SP)	:: THE NEW SWR
6807	033662	105777	145256	7\$:	TSTB	\$STKS	:: CHAR THERE?
6808	033666	100375			BPL	7\$:: IF NOT TRY AGAIN
6809							
6810	033670	117746	145252		MOVB	\$STKB, -(SP)	:: PICK UP CHAR
6811	033674	042716	177600		BIC	\$IC177, (SP)	:: MAKE IT 7-BIT ASCII
6812							
6813	033700	021627	000003		CMP	(SP), #3	:: IS IT A CONTROL-C?
6814	033704	001015			BNE	9\$:: BRANCH IF NOT
6815	033706	104401	034460		TYPE	, \$CNTLC	:: YES, ECHO CONTROL-C (↑C)
6816	033712	062706	000006		ADD	#6, SP	:: CLEAN UP STACK
6817	033716	123727	001135	000001	CMPB	\$INTAG, #1	:: REENABLE TTY KEYBOARD INTERRUPTS?
6818	033724	001003			BNE	8\$:: BRANCH IF NO
6819	033726	012777	000100	145210	MOV	#100, \$STKS	:: ALLOW TTY KEYBOARD INTERRUPTS
6820	033734	000137	030374	8\$:	JMP	STOP	:: CONTROL-C RESTART
6821							
6822							
6823	033740	021627	000025	9\$:	CMP	(SP), #25	:: IS IT A CONTROL-U?
6824	033744	001005			BNE	10\$:: BRANCH IF NOT
6825	033746	104401	034465		TYPE	, \$CNTLU	:: YES, ECHO CONTROL-U (↑U)
6826	033752	062706	000006	20\$:	ADD	#6, SP	:: IGNORE PREVIOUS INPUT
6827	033756	000737			BR	19\$:: LET'S TRY IT AGAIN
6828							
6829							
6830	033760	021627	000015	10\$:	CMP	(SP), #15	:: IS IT A <CR>?
6831	033764	001022			BNE	16\$:: BRANCH IF NO
6832	033766	005766	000004		TST	4(SP)	:: YES, IS IT THE FIRST CHAR?
6833	033772	001403			BEQ	11\$:: BRANCH IF YES
6834	033774	016677	000002	145136	MOV	2(SP), \$SWR	:: SAVE NEW SWR
6835	034002	062706	000006	11\$:	ADD	#6, SP	:: CLEAN UP STACK
6836	034006	104401	001205	14\$:	TYPE	, \$CRLF	:: ECHO <CR> AND <LF>
6837	034012	123727	001135	000001	CMPB	\$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS?
6838	034020	001003			BNE	15\$:: BRANCH IF NOT
6839	034022	012777	000100	145114	MOV	#100, \$STKS	:: RE-ENABLE TTY KBD INTERRUPTS
6840	034030	000002		15\$:	RTI		:: RETURN
6841	034032	004737	032262	16\$:	JSR	PC, \$TYPEC	:: ECHO CHAR
6842	034036	021627	000060		CMP	(SP), #60	:: CHAR < 0?
6843	034042	002420			BLT	18\$:: BRANCH IF YES
6844	034044	021627	000067		CMP	(SP), #67	:: CHAR > 7?
6845	034050	003015			BGT	18\$:: BRANCH IF YES
6846	034052	042726	000060		BIC	#60, (SP)+	:: STRIP-OFF ASCII
6847	034056	005766	000002		TST	2(SP)	:: IS THIS THE FIRST CHAR
6848	034062	001403			BEQ	17\$:: BRANCH IF YES
6849	034064	006316			ASL	(SP)	:: NO, SHIFT PRESENT
6850	034066	006316			ASL	(SP)	:: CHAR OVER TO MAKE
6851	034070	006316			ASL	(SP)	:: ROOM FOR NEW ONE.
6852	034072	005266	000002	17\$:	INC	2(SP)	:: KEEP COUNT OF CHAR
6853	034076	056616	177776		BIS	-2(SP), (SP)	:: SET IN NEW CHAR
6854	034102	000667			BR	7\$:: GET THE NEXT ONE
6855	034104	104401	001204	18\$:	TYPE	, \$QUES	:: TYPE ?<CR><LF>
6856	034110	000720			BR	20\$:: SIMULATE CONTROL-U

M10

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 128
TTY INPUT ROUTINE

SEQ 0128

6857
6858
6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912

034112 011646
034114 016666 000004 000002
034122 005066 000004
034126 005046
034130 012746 034136
034134 000002
034136
034136 005737 033252
034142 001775
034144 005337 033252
034150 117766 177102 000004
034156 005237 033256
034162 023727 033256 033261
034170 001003
034172 012737 033260 033256
034200 000002

010346
005046
012703 034436
022703 034460
101456
104410
112613
122713 000177
001022
005716
001007
112737 000134 034434
104401 034434
012716 177777
005303
020327 034436
103434
111337 034434
104401 034434
000746
005716
001406

```
.DSABL LSB

*****
*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
*CALL:
*   RDCHR          ;; GET A CHARACTER FROM THE QUEUE
*   RETURN HERE   ;; CHARACTER IS ON THE STACK
*                ;; WITH PARITY BIT STRIPPED OFF
*

SRDCHR: MOV      (SP), -(SP) ;; PUSH DOWN THE PC AND
        MOV      4(SP), 2(SP) ;; THE PS
        CLR      4(SP)       ;; GET READY FOR A CHARACTER
        CLR      -(SP)      ;; PUT NEW PS ON STACK
        MOV      #64$, -(SP) ;; PUT NEW PC ON STACK
        RTI          ;; POP NEW PC AND PS

64$:
1$:     TST      $TKCNT      ;; WAIT ON A CHARACTER
        BEQ      1$
        DEC      $TKCNT     ;; DECREMENT THE COUNTER
        MOVB    2($TKQOUT, 4(SP) ;; GET ONE CHARACTER
        INC     $TKQOUT     ;; UPDATE THE POINTER
        CMP     $TKQOUT, #($TKQEND) ;; DID IT GO OFF OF THE END?
        BNE     2$         ;; BRANCH IF NO
        MOV     #($TKQSR, $TKQOUT) ;; RESET THE POINTER
        RTI          ;; RETURN

2$:
*****
*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
*CALL:
*   ROLIN
*   RETURN HERE
*                ;; INPUT A STRING FROM THE TTY
*                ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
*                ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
*

SRDOLIN: MOV     R3, -(SP)   ;; SAVE R3
         CLR     -(SP)     ;; CLEAR THE RUBOUT KEY
1$:     MOV     #($TTYIN, R3 ;; GET ADDRESS
2$:     CMP     #($TTYIN+22, R3 ;; BUFFER FULL?
         BLOS   4$         ;; BR IF YES
         RDCHR  ;; GO READ ONE CHARACTER FROM THE TTY
         MOVB  (SP)+, (R3) ;; GET CHARACTER
10$:    CMPB   #177, (R3)   ;; IS IT A RUBOUT
         BNE   5$         ;; BR IF NO
         TST   (SP)       ;; IS THIS THE FIRST RUBOUT?
         BNE   6$         ;; BR IF NO
         MOVB #'\, 9$     ;; TYPE A BACK SLASH
         TYPE  9$
6$:     MOV     #-1, (SP)   ;; SET THE RUBOUT KEY
         DEC   R3         ;; BACKUP BY ONE
         CMP   R3, #($TTYIN ;; STACK EMPTY?
         BLO  4$         ;; BR IF YES
         BLO  4$         ;; BR IF YES
         MOVB (R3), 9$     ;; SETUP TO TYPEOUT THE DELETED CHAR.
         TYPE  9$
         BR   2$         ;; GO TYPE
5$:     TST   (SP)       ;; GO READ ANOTHER CHAR.
         BEQ  7$         ;; RUBOUT KEY SET?
         BR   7$         ;; BR IF NO
```

```

6913 034302 112737 000134 034434      MOVB      #' \, 95      ;; TYPE A BACK SLASH
6914 034310 104401 034434      TYPE      95
6915 034314 005016      CLR      (SP)      ;; CLEAR THE RUBOUT KEY
6916 034316 122713 000025      7$: CMPB      #25, (R3)  ;; IS CHARACTER A CTRL U?
6917 034322 001003      BNE      8$      ;; NO IF NO
6918 034324 104401 034465      TYPE      SCNTLU      ;; TYPE A CONTROL "U"
6919 034330 000726      BR      1$      ;; GO START OVER
6920 034332 122713 000022      8$: CMPB      #22, (R3)  ;; IS CHARACTER A "r"?
6921 034336 001011      LNE      3$      ;; BRANCH IF NO
6922 034340 105013      CLRB      (R3)      ;; CLEAR THE CHARACTER
6923 034342 104401 001205      TYPE      , SCRLF      ;; TYPE A "CR" & "LF"
6924 034346 104401 034436      TYPE      , $TTYIN      ;; TYPE THE INPUT STRING
6925 034352 000717      BR      2$      ;; GO PICKUP ANOTHER CHACTER
6926 034354 104401 001204      4$: TYPE      , $QUES      ;; TYPE A '?'
6927 034360 000712      BR      1$      ;; CLEAR THE BUFFER AND LOOP
6928 034362 111337 034434      3$: MOVB      (R3), 95  ;; ECHO THE CHARACTER
6929 034366 104401 034434      TYPE      95
6930 034372 122723 000015      CMPB      #15, (R3)+  ;; CHECK FOR RETURN
6931 034376 001305      BNE      2$      ;; LOOP IF NOT RETURN
6932 034400 105063 177777      CLRB      -1(R3)      ;; CLEAR RETURN (THE 15)
6933 034404 104401 001206      TYPE      , $LF      ;; TYPE A LINE FEED
6934 034410 005726      TST      (SP)+      ;; CLEAN RUBOUT KEY FROM THE STACK
6935 034412 012603      MOV      (SP)+, R3      ;; RESTORE R3
6936 034414 011646      MOV      (SP), -(SP)  ;; ADJUST THE STACK AND PUT ADDRESS OF THE
6937 034416 016666 000004 000002      MOV      4(SP), 2(SP)  ;; FIRST ASCII CHARACTER ON IT
6938 034424 012766 034436 000004      MOV      #TTYIN, 4(SP)
6939 034432 000002      RTI
6940 034434 000      9$: .BYTE      0      ;; RETURN
6941 034435 000      .BYTE      0      ;; STORAGE FOR ASCII CHAR. TO TYPE
6942 034436 000022      $TTYIN: .BLKB      22  ;; TERMINATOR
6943 034460 041536 005015 000      $CNTLC: .ASCIZ      /TC<<15><12>  ;; RESERVE 22 BYTES FOR TTY INPUT
6944 034465 136 006525 000012      $CNTLU: .ASCIZ      /TU<<15><12>  ;; CONTROL "C"
6945 034472 043536 005015 000      $CNTLG: .ASCIZ      /TG<<15><12>  ;; CONTROL "U"
6946 034477 015 051412 051127      $MSWR: .ASCIZ      <15><12>/SWR = /  ;; CONTROL "G"
6947 034504 036440 000040
6948 034510 020040 042516 020127      $MNEW: .ASCIZ      / NEW = /
6949 034516 020075 000
6950 034522
6951 .EVEN
6952 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
6953
6954 *****
6955 *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
6956 *CHANGE IT TO BINARY.
6957 *THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
6958 *OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
6959 *FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
6960 *THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
6961 *CALL:
6962 *      RDOCT      ;; READ AN OCTAL NUMBER
6963 *      RETURN HERE  ;; LOW ORDER BITS ARE ON TOP OF THE STACK
6964 *      ;; HIGH ORDER BITS ARE IN $HIOCT
6965 $RDOCT: MOV      (SP), -(SP)  ;; PROVIDE SPACE FOR THE
6966 034524 016666 000004 000002      MOV      4(SP), 2(SP)  ;; INPUT NUMBER
6967 034532 010046      MOV      R0, -(SP)    ;; PUSH R0 ON STACK
6968 034534 010146      MOV      R1, -(SP)    ;; PUSH R1 ON STACK

```

B11

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 130
READ AN OCTAL NUMBER FROM THE TTY

SEQ 0130

6969 034536 010246
6970 034540 104411
6971 034542 012600
6972 034544 010037 034650
6973 034550 005001
6974 034552 005002
6975 034554 112046
6976 034556 001420
6977 034560 122716 000060
6978 034564 003026
6979 034566 122716 000067
6980 034572 002423
6981 034574 006301
6982 034576 006102
6983 034600 006301
6984 034602 006102
6985 034604 006301
6986 034606 006102
6987 034610 042716 177770
6988 034614 062601
6989 034616 000756
6990 034620 005726
6991 034622 010166 000012
6992 034626 010237 034660
6993 034632 012602
6994 034634 012601
6995 034636 012600
6996 034640 000002
6997 034642 005726
6998 034644 105010
6999 034646 104401
7000 034650 000000
7001 034652 104401 001204
7002 034656 000730
7003 034660 000000
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015 034662 104413
7016 034664 016601 000002
7017 034670 012705 035001
7018 034674 012704 000014
7019 034700 012703 177770
7020 034704 012100
7021 034706 012101
7022 034710 005002
7023 034712 110245
7024 034714 010002

```

MOV R2,-(SP) ;; PUSH R2 ON STACK
1$: RDLIN ;; READ AN ASCII LINE
MOV (SP)+,R0 ;; GET ADDRESS OF 1ST CHARACTER
MOV R0,$$ AND SAVE IT
CLR R1 ;; CLEAR DATA WORD
CLR R2
2$: MOV B (R0)+,-(SP) ;; PICKUP THIS CHARACTER
BEQ 3$ ;; IF ZERO GET OUT
CMPB #'0,(SP) ;; MAKE SURE THIS CHARACTER
BGT 4$ ;; IS AN OCTAL DIGIT
CMPB #'7,(SP)
BLT 4$
ASL R1 ;; *2
ROL R2
ASL R1 ;; *4
ROL R2
ASL R1 ;; *8
ROL R2
BIC #'C7,(SP) ;; STRIP THE ASCII JUNK
ADD (SP)+,R1 ;; ADD IN THIS DIGIT
BR 2$ ;; LOOP
3$: TST (SP)+ ;; CLEAN TERMINATOR FROM STACK
MOV R1,12(SP) ;; SAVE THE RESULT
MOV R2,$HIOCT
MOV (SP)+,R2 ;; POP STACK INTO R2
MOV (SP)+,R1 ;; POP STACK INTO R1
MOV (SP)+,R0 ;; POP STACK INTO R0
RTI ;; RETURN
4$: TST (SP)+ ;; CLEAN PARTIAL FROM STACK
CLRB (R0) ;; SET A TERMINATOR
TYPE ;; TYPE UP THRU THE BAD CHAR.
5$: .WORD 0
TYPE $QUES ;; "?" "CR" & "LF"
BR 1$ ;; TRY AGAIN
$HIOCT: .WORD 0 ;; HIGH ORDER BITS GO HERE
.SBTL DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
*****
;THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
;UNSIGNED OCTAL ASCII NUMBER.
;CALL
; * MOV #PNTR,-(SP) ;; POINTER TO LOW WORD OF BINARY NUMBER
; * JSR PC,$SDB20 ;; CALL THE ROUTINE
; * RETURN ;; THE ADDRESS OF THE FIRST ASCII CHAR. IS ON THE STACK
$SDB20: SAVREG ;; SAVE ALL REGISTERS
MOV 2(SP),R1 ;; PICKUP THE POINTER TO LOW WORD
MOV #OCTVL+13.,R5 ;; POINTER TO DATA TABLE
MOV #12.,R4 ;; DO ELEVEN CHARACTERS
MOV #'C7,R3 ;; MASK
MOV (R1)+,R0 ;; LOWER WORD
MOV (R1)+,R1 ;; HIGH WORD
CLR R2 ;; TERMINATOR
1$: MOV B R2,-(R5) ;; PUT CHARACTER IN DATA TABLE
MOV R0,R2 ;; GET THIS DIGIT

```

```

7025 034716 005304          DEC      R4          ;; COUNT THIS CHARACTER
7026 034720 003007          BGT      3$          ;; BR IF NOT THE LAST DIGIT
7027 034722 001405          BEQ      2$          ;; BR IF IT IS THE LAST DIGIT
7028 034724 005205          INC      R5          ;; ALL DIGITS DONE-ADJUST POINTER FOR FIRST
7029 034726 010566 000002  MOV      R5,2(SP)    ;; ASCIZ CHAR. & PUT IT ON THE STACK
7030 034732 104414          RESREG                    ;; RESTORE ALL REGISTERS
7031 034734 000207          RTS      PC          ;; RETURN TO USER
7032 034736 006203 2$:    ASR      R3          ;; POSITION THE MASK FOR THE LAST DIGIT
7033 034740 006001 3$:    ROR      R1          ;; POSITION THE BINARY NUMBER FOR
7034 034742 006000          ROR      R0          ;; THE NEXT OCTAL DIGIT
7035 034744 006001          ROR      R1
7036 034746 006000          ROR      R0
7037 034750 006001          ROR      R1
7038 034752 006000          ROR      R0
7039 034754 040302          B'0C     R3,R2      ;; MASK OUT ALL JUNK
7040 034756 062702 000060  ADD      #'0,R2      ;; MAKE THIS CHAR. ASCII
7041 034762 000753          BR       1$          ;; GO PUT IT IN THE DATA TABLE
7042 034764 000016          $OCTVL: .BLKB 14.    ;; RESERVE DATA TABLE
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080

```

```

.SBTTL DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
;*****
;THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
;DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
;POSITIVE.
;CALL

```

```

*   MOV      #PNTR, -(SP)    ;; POINTER TO LOW WORD OF BINARY NUMBER
*   JSR      PC, @#$DB2D
*   RETURN                    ;; THE FIRST ADDRESS OF ASCIZ
;; IS ON THE STACK

```

```

$DB2D: SAVREG                    ;; SAVE REGISTERS
MOV      2(SP),R2            ;; PICKUP THE DATA POINTER
MOV      #$DECVL,R0         ;; GET ADDRESS OF "$DECVL" STRING
MOV      R0,2(SP)          ;; PUT ADDRESS OF ASCIZ STRING ON STACK
MOV      (R2)+,R1          ;; PICKUP THE BINARY NUMBER
MOV      (R2)+,R2
MOV      #10,4$            ;; SET UP TO DO 10 CONVERSIONS
MOV      #$STNPNR,R4       ;; ADDRESS OF TEN POWER
MOV      #$STNPNR+2,R5
1$:    CLR      R3          ;; CLEAR PARTIAL
2$:    SUB      (R4),R1     ;; SUBTRACT TEN POWER
SBC      R2
SUB      (R5),R2
BLT      3$              ;; BR IF TEN POWER TOO LARGE
INC      R3              ;; ADD 1 TO PARTIAL
BR       2$              ;; LOOP
3$:    ADD      (R4)+,R1    ;; RESTORE SUBTRACTED VALUE
ADC      R2
ADD      (R4)+,R2
CMP      (R5)+,(R5)+      ;; MOVE TO NEXT TEN POWER
BIS      #'0,R3          ;; CHANGE PARTIAL TO ASCII
MOVB     R3,(R0)+        ;; SAVE IT
DEC      (PC)+          ;; DONE?
4$:    .WORD   0
BNE     1$              ;; BR IF NO

```

D11

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
 DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 132
 DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE

SEQ 0132

```

7081 035104 105020          CLRB      (R0)+      ;; TERMINATOR
7082 035106 104414          RESREG                     ;; RESTORE REGISTERS
7083 035110 000207          RTS        PC          ;; RETURN
7084 035112 145000          STNPNR: 145000          ;; 1.0E09
7085 035114 035632          35632                      ;;
7086 035116 160400          160400                      ;; 1.0E08
7087 035120 002765          2765                        ;;
7088 035122 113200          113200                      ;; 1.0E07
7089 035124 000230          230                          ;;
7090 035126 041100          041100                      ;; 1.0E06
7091 035130 000017          17                            ;;
7092 035132 103240          103240                      ;; 1.0E05
7093 035134 000001          1                             ;;
7094 035136 023420          23420                       ;; 1.0E04
7095 035140 000000          0                             ;;
7096 035142 001750          1750                        ;; 1.0E03
7097 035144 000000          0                             ;;
7098 035146 000144          144                          ;; 1.0E02
7099 035150 000000          0                             ;;
7100 035152 000012          12                           ;; 1.0E01
7101 035154 000000          0                             ;;
7102 035156 000001          1                             ;; 1.0E00
7103 035160 000000          0
7104 035162 000014          $DECVL: .BLKB 12.          ;; RESERVE STORAGE FOR ASCIZ STRING
7105          .SBTTL SINGLE LENGTH BINARY TO DECIMAL ASCII ROUTINE
7106
7107          ;*****
7108          ;THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
7109          ;UNSIGNED DECIMAL ASCII NUMBER.
7110          ;CALL
7111          ;      MOV      NUMBER, -(SP)      ;; PUT BINARY NUMBER ON THE STACK
7112          ;      JSR      PC, @#$SB2D      ;; CALL
7113          ;      RETURN                      ;; ADDRESS OF THE 1ST ASCII CHAR. IS ON THE STACK
7114
7115
7116 035176 016637 000002 035226 $SB2D:  MOV      2(SP), 1$      ;; SAVE BINARY NUMBER
7117 035204 012746 035226      MOV      #1$, -(SP)      ;; SET POINTER
7118 035210 004737 035002      JSR      PC, @#$SB2D      ;; CALL DOUBLE LENGTH CONVERT
7119 035214 062716 000005      ADD      #5, (SP)        ;; ONLY ALLOW FIVE CHARACTERS
7120 035220 012666 000002      MOV      (SP)+, 2(SP)    ;; PICKUP POINTER
7121 035224 000207          RTS        PC          ;; RETURN
7122 035226 000000 000000      1$:      .WORD 0,0
7123          .SBTTL TYPE NUMERICAL ASCII STRING SUPPRESS LEADING ZEROS
7124
7125          ;*****
7126          ;THIS ROUTINE IS USED TO TYPE AN ASCII NUMBER SUPPRESSING THE
7127          ;LEADING NUMBERS.
7128          ;CALL
7129          ;      MOV      #NUMADR, -(SP)    ;; FIRST ADDRESS OF ASCII STRING
7130          ;      JSR      PC, @#$SUPRS
7131
7132
7133 035232 010046          $SUPRS: MOV      R0, -(SP)    ;; SAVE R0
7134 035234 016600 000004      MOV      4(SP), R0      ;; PICKUP THE POINTER
7135 035240 105710          1$:      TSTB      (R0)        ;; TERMINATEOR?
7136 035242 001403          BEQ      2$            ;; BR IF YES
  
```

```

7137 035244 122720 000060          CMPB    #'0,(RO)+      ;; IS THIS AN ASCII "0" ?
7138 035250 001773          BEQ     1$             ;; BR IF YES
7139 035252 005300          2$:    DEC     RO      ;; BACKUP BY "1"
7140 035254 010037 035262          MOV     RO,3$        ;; SAVE FOR TYPING
7141 035260 104401          TYPE   0             ;; GO TYPE
7142 035262 000000          3$:    .WORD  0       ;; ASCIZ POINTER GOES HERE
7143 035264 012600          MOV     (SP)+,RO     ;; RESTORE RO
7144 035266 012616          MOV     (SP)+,(SP)  ;; RESTORE THE STACK
7145 035270 000207          RTS     PC           ;; RETURN
7146
7147          .SBTTL  INTEGER MULTIPLY ROUTINE
7148
7149          ;*****
7150          ;CALL
7151          ;   MOV     MULTIPLIER,-(SP)
7152          ;   MOV     MULTIPLICAND,-(SP)
7153          ;   JSR     PC,2$MULT
7154          ;   RETURN  ;; PRODUCT IS ON THE STACK
7155          ;
7156          ;   STACK  PRODUCT
7157          ;   -----
7158          ;   TOP    LSB'S
7159          ;   +2     MSB'S
7160
7160 035272          $MULT:
7161 035272 010046          MOV     RO,-(SP)    ;; PUSH RO ON STACK
7162 035274 010146          MOV     R1,-(SP)    ;; PUSH R1 ON STACK
7163 035276 010246          MOV     R2,-(SP)    ;; PUSH R2 ON STACK
7164 035300 005046          CLR     -(SP)       ;; CLEAR THE SIGN KEY
7165 035302 016601 000012          MOV     12(SP),R1  ;; GET THE MULTIPLICAND
7166 035306 100002          BPL     1$          ;; BR IF PLUS
7167 035310 005216          INC     (SP)        ;; SET THE SIGN KEY
7168 035312 005401          NEG     R1          ;; MAKE THE MULTIPLICAND POSTIVE
7169 035314 016602 000014          1$:    MOV     14(SP),R2  ;; GET THE MULTIPLIER
7170 035320 100002          BPL     2$          ;; BR IF PLUS
7171 035322 005316          DEC     (SP)        ;; UPDATE THE SIGN KEY
7172 035324 005402          NEG     R2          ;; MAKE THE MULTIPLIER POSTIVE
7173 035326 012746 000021          2$:    MOV     #17,-(SP)  ;; SET THE LOOP COUNT
7174 035332 005000          CLR     RO          ;; SETUP FOR THE MULTIPLY LOOP
7175 035334 103001          3$:    BCC     4$          ;; DON'T ADD IF MULTIPLICAND = 0
7176 035336 060200          ADD     R2,RO
7177 035340 006000          4$:    ROR     RO
7178 035342 006001          ROR     R1
7179 035344 005316          DEC     (SP)        ;; POSITION THE PARTIAL PRODUCT AND
7180 035346 001372          BNE     3$          ;; THE MULTIPLICAND
7181 035350 022616          CMP     (SP)+,(SP)  ;; HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
7182 035352 001403          BEQ     5$          ;; BR IF NO
7183 035354 005400          NEG     RO          ;; SHOULD PRODUCT BE NEGATIVE?
7184 035356 005401          NEG     R1          ;; GO TO EXIT IF NO
7185 035360 005600          SBC     RO          ;; YES--SO MAKE IT SO
7186 035362 005726          5$:    TST     (SP)+      ;; CLEAR SIGN INFO. OFF OF STACK
7187 035364 010066 000012          MOV     RO,12(SP)   ;; PUT THE PRODUCT ON THE STACK (MSB'S)
7188 035370 010166 000010          MOV     R1,10(SP)   ;; LSB'S
7189 035374 012602          MOV     (SP)+,R2    ;; POP STACK INTO R2
7190 035376 012601          MOV     (SP)+,R1    ;; POP STACK INTO R1
7191 035400 012600          MOV     (SP)+,RO    ;; POP STACK INTO RO
7192 035402 000207          RTS     PC

```

```

7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210 035404
7211 035404 010046
7212 035406 010146
7213 035410 010246
7214 035412 010346
7215 035414 010446
7216 035416 010546
7217 035420 016646 000022
7218 035424 016646 000022
7219 035430 016646 000022
7220 035434 016646 000022
7221 035440 000002
7222
7223
7224
7225
7226 035442
7227 035442 012666 000022
7228 035446 012666 000022
7229 035452 012666 000022
7230 035456 012666 000022
7231 035462 012605
7232 035464 012604
7233 035466 012603
7234 035470 012602
7235 035472 012601
7236 035474 012600
7237 035476 000002
7238
7239
7240
7241
7242
7243
7244
7245
7246 035500 010046
7247 035502 016600 000002
7248 035506 005740
    
```

```

.SBTTL SAVE AND RESTORE RO-R5 ROUTINES

;*****
;SAVE RO-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 RO,-(SP) ;: PUSH RO 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 RO-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

;*****
;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
    
```



```

7249 035510 111000          MOVB   (RO),RO          ;;GET RIGHT BYTE OF TRAP
7250 035512 006300          ASL    RO              ;;POSITION FOR INDEXING
7251 035514 016000 035534   MOV    $TRPAD(RO),RO   ;;INDEX TO TABLE
7252 035520 000200          RTS    RO              ;;GO TO ROUTINE

```

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

```

7257 035522 011646 000004 000002 $TRAP2: MOV   (SP),-(SP)  ;;MOVE THE PC DOWN
7258 035524 016666          MOV   4(SP),2(SP)     ;;MOVE THE PSW DOWN
7259 035532 000002          RTI                    ;;RESTORE THE PSW

```

.SBTTL TRAP TABLE

;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;*BY THE "TRAP" INSTRUCTION.

```

;          ROUTINE
;          -----
7268 035534 035522 $TRPAD: .WORD  $TRAP2
7269 035536 032050   $TYPE  ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
7270 035540 033050   $TYPOC ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
7271 035542 033024   $TYPOS ;;CALL=TYPOS     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
7272 035544 033064   $TYPON ;;CALL=TYPON     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
7273 035546 032332   $TYPDS ;;CALL=TYPDS    TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
7274
7275 035550 033640   $GTSWR ;;CALL=GTSWR     TRAP+6(104406) GET SOFT-SWR SETTING
7276
7277 035552 033550   $CKSWR ;;CALL=CKSWR     TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
7278 035554 034112   $R0CHR ;;CALL=R0CHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
7279 035556 034202   $RDLIN ;;CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
7280 035560 034522   $RDOCT ;;CALL=RDOCT     TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
7281 035562 035404   $SAVREG ;;CALL=SAVREG    TRAP+13(104413) SAVE R0-R5 ROUTINE
7282 035564 035442   $RESREG ;;CALL=RESREG    TRAP+14(104414) RESTORE R0-R5 ROUTINE
7283 035566 030254   $COPI5 ;;CALL=SCOPI     TRAP+15(104415) INTERNAL LOOP ON ERROR
7284

```

Address	Hex 1	Hex 2	Hex 3	Hex 4	Text
7285					
7286					
7287					
7288					
7289	035570	005015	047125	041111	.SBTTL SERVICE MESSAGES
7290	035576	051525	051040	030113	MSG1: .ASCII <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/
7291	035604	020066	051104	053111	
7292	035612	020105	044504	043501	
7293	035620	047516	052123	041511	
7294	035626	005015	040515	047111	.ASCII <CR><LF>/MAINDEC-11-DZR6J-D-PB/<CR><LF>
7295	035634	042504	026503	030461	
7296	035642	042055	051132	045066	
7297	035650	042055	050055	006502	
7298	035656	012			
7299	035657	015	004412	025052	.ASCII <CR><LF>/ ***** CAUTION *****/<CR><LF>
7300	035664	025052	020052	040503	
7301	035672	052125	047511	020116	
7302	035700	025052	025052	006452	
7303	035706	012			
7304	035707	015	052012	044510	.ASCII <CR><LF>/THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING CONTROL-C/
7305	035714	020123	051120	043517	
7306	035722	040522	020115	044123	
7307	035730	052517	042114	020040	
7308	035736	042502	044040	046101	
7309	035744	042524	020104	047117	
7310	035752	054514	041040	020131	
7311	035760	054524	044520	043516	
7312	035766	041440	047117	051124	
7313	035774	046117	041455		
7314	036000	005015	052117	042510	.ASCII <CR><LF>/OTHERWISE CARTRIDGE FORMATTING AND, OR THE DEVICE/
7315	036006	053522	051511	020105	
7316	036014	040503	052122	044522	
7317	036022	043504	020105	047506	
7318	036030	046522	052101	044524	
7319	036036	043516	040440	042116	
7320	036044	020054	051117	052040	
7321	036052	042510	042040	053105	
7322	036060	041511	105		
7323	036063	015	046412	054501	.ASCII <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/<CR><LF>
7324	036070	041040	020105	042514	
7325	036076	052106	044440	020116	
7326	036104	047101	052440	042116	
7327	036112	052105	051105	044515	
7328	036120	042516	020104	052123	
7329	036126	052101	006505	012	
7330	036133	015	044412	044516	.ASCII <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE: /<CR><LF>
7331	036140	044524	046101	054514	
7332	036146	020054	051104	053111	
7333	036154	051505	052040	020117	
7334	036162	042502	052040	051505	
7335	036170	042524	020104	044123	
7336	036176	052517	042114	044040	
7337	036204	053101	035105	005015	
7338	036212	005015	027101	044040	.ASCII <CR><LF>/A. HEADS MANUALLY LOADED/
7339	036220	040505	051504	046440	
7340	036226	047101	040525	046114	

7341	036234	020131	047514	042101	
7342	036242	042105			
7343	036244	005015	027102	041440	.ASCII <CR><LF>/B. CORRECT PORT SELECTED/
7344	036252	051117	042522	052103	
7345	036260	050040	051117	020124	
7346	036266	042523	042514	052103	
7347	036274	042105			
7348	036276	005015	027103	053440	.ASCII <CR><LF>/C. WRITE LOCK DISABLED/
7349	036304	044522	042524	046040	
7350	036312	041517	020113	044504	
7351	036320	040523	046102	042105	
7352	036326	005015	027104	042040	.ASCII <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
7353	036334	044522	042526	051040	
7354	036342	040505	054504	044440	
7355	036350	042116	041511	052101	
7356	036356	051117	047440	006516	
7357	036364	012			
7358	036365	015	042012	044522	.ASCII <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
7359	036372	042526	020123	047516	
7360	036400	020124	047524	041040	
7361	036406	020105	042524	052123	
7362	036414	042105	046440	051525	
7363	036422	020124	040510	042526	
7364	036430	005015	047502	044124	.ASCIZ <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
7365	036436	050040	051117	051524	
7366	036444	042040	051505	046105	
7367	036452	041505	042524	006504	
7368	036460	000012			
7369					
7370	036462	005015	042502	051440	MSG2: .ASCIZ <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE D/
7371	036470	051125	020105	047524	
7372	036476	050040	052125	051440	
7373	036504	051103	052101	044103	
7374	036512	050040	041501	020113	
7375	036520	047111	042040	044522	
7376	036526	042526	030040	000	
7377	036533	015	042012	044522	MSG3: .ASCIZ <CR><LF>/DRIVE(S) TO BE TESTED: /
7378	036540	042526	051450	020051	
7379	036546	047524	041040	020105	
7380	036554	042524	052123	042105	
7381	036562	020072	000		
7382	036565	015	052012	050131	MSG4: .ASCIZ <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440: /
7383	036572	020105	052502	051523	
7384	036600	040440	042104	042522	
7385	036606	051523	044440	020106	
7386	036614	047516	020124	033461	
7387	036622	032067	030064	020072	
7388	036630	000040			
7389	036632	005015	054524	042520	MSG5: .ASCIZ <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210: /
7390	036640	041440	047117	051124	
7391	036646	046117	042514	020122	
7392	036654	047111	042524	051122	
7393	036662	050125	020124	042525	
7394	036670	052103	051117	044440	
7395	036676	020106	047516	020124	
7396	036704	030462	035060	020040	

7397	036712	000				
7398	036713	015	044412	052116	MSG6:	.ASCIZ <CR><LF>/INTERRUPT OCCURRED AT PC=/
7399	036720	051105	052522	052120		
7400	036726	047440	041503	051125		
7401	036734	042522	020104	052101		
7402	036742	050040	036503	000		
7403	036747	015	042012	044522	MSG7:	.ASCIZ <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
7404	036754	042526	030040	053440		
7405	036762	046111	020114	047516		
7406	036770	020124	042502	052040		
7407	036776	051505	042524	000104		
7408	037004	005015	047111	042524	MSG8:	.ASCIZ <CR><LF>/INTERLOCKS TEST/<CR><LF>
7409	037012	046122	041517	051513		
7410	037020	052040	051505	006524		
7411	037026	000012				
7412	037030	005015	042522	047515	MSG9:	.ASCIZ <CR><LF>/REMOVE UNIT SELECT PLUG/
7413	037036	042526	052440	044516		
7414	037044	020124	042523	042514		
7415	037052	052103	050040	052514		
7416	037060	000107				
7417	037062	005015	053412	046111	MSG10:	.ASCIZ <CR><LF><LF>/WILL TEST DRIVES:/
7418	037070	020114	042524	052123		
7419	037076	042040	044522	042526		
7420	037104	035123	000			
7421	037107	015	005012	047520	MSG11:	.ASCIZ <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
7422	037114	042527	020122	050125		
7423	037122	051040	051505	040524		
7424	037130	052122	052040	020117		
7425	037136	042524	052123	030440		
7426	037144	005015	000			
7427	037147	116	047117	006505	MSG12:	.ASCIZ /NONE/<CR><LF>
7428	037154	000012				
7429	037156	005015	047516	046040	MSG13:	.ASCII <CR><LF>/NO L OR P CLOCKS PRESENT/
7430	037164	047440	020122	020120		
7431	037172	046103	041517	051513		
7432	037200	050040	042522	042523		
7433	037206	052116				
7434	037210	005015	046101	020114		.ASCIZ <CR><LF>/ALL TIMING TESTS BYPASSED/
7435	037216	044524	044515	043516		
7436	037224	052040	051505	051524		
7437	037232	041040	050131	051501		
7438	037240	042523	000104			
7439	037244	005015	054502	040520	MSG14:	.ASCIZ <CR><LF>/BYPASSING DRIVE /
7440	037252	051523	047111	020107		
7441	037260	051104	053111	020105		
7442	037266	000				
7443	037267	015	005012	051104	MSG15:	.ASCIZ <CR><LF><LF>/DRIVE /
7444	037274	053111	020105	000		
7445	037301	015	042012	044522	MSG16:	.ASCIZ <CR><LF>/DRIVE SERIAL NO. /
7446	037306	042526	051440	051105		
7447	037314	040511	020114	047516		
7448	037322	020056	000			
7449	037325	015	041412	051101	MSG17:	.ASCIZ <CR><LF>/CARTRIDGE SERIAL NO. /
7450	037332	051124	042111	042507		
7451	037340	051440	051105	040511		
7452	037346	020114	047516	020056		

7453	037354	000				
7454	037355	015	052412	044516	MSG18:	.ASCIZ <CR><LF>/UNIT SELECT PLUG TEST/<CR><LF>
7455	037362	020124	042523	042514		
7456	037370	052103	050040	052514		
7457	037376	020107	042524	052123		
7458	037404	005015	000			
7459	037407	015	050012	051117	MSG19:	.ASCIZ <CR><LF>/PORT SELECTION TESTS/<CR><LF>
7460	037414	020124	042523	042514		
7461	037422	052103	047511	020116		
7462	037430	042524	052123	006523		
7463	037436	000012				
7464	037440	005015	041501	046040	MSG21:	.ASCIZ <CR><LF>/AC LOW DETECTION TEST-PART 1/<CR><LF>
7465	037446	053517	042040	052105		
7466	037454	041505	044524	047117		
7467	037462	052040	051505	026524		
7468	037470	040520	052122	030440		
7469	037476	005015	000			
7470	037501	015	047012	047117	MSG22:	.ASCIZ <CR><LF>/NON-EXECUTABLE FUNCTION (NXF) DETECTION TEST/<CR><LF>
7471	037506	042455	042530	052503		
7472	037514	040524	046102	020105		
7473	037522	052506	041516	044524		
7474	037530	047117	024040	054116		
7475	037536	024506	042040	052105		
7476	037544	041505	044524	047117		
7477	037552	052040	051505	006524		
7478	037560	000012				
7479	037562	005015	051127	052111	MSG24:	.ASCIZ <CR><LF>/WRITE LOCK TEST/<CR><LF>
7480	037570	020105	047514	045503		
7481	037576	052040	051505	006524		
7482	037604	000012				
7483	037606	005015	041501	046040	MSG26:	.ASCIZ <CR><LF>/AC LOW DETECTION TEST-PART 2/<CR><LF>
7484	037614	053517	042040	052105		
7485	037622	041505	044524	047117		
7486	037630	052040	051505	026524		
7487	037636	040520	052122	031040		
7488	037644	005015	000			
7489	037647	015	005012	046101	MSG27:	.ASCIZ <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
7490	037654	020114	051104	053111		
7491	037662	051505	052040	051505		
7492	037670	042524	006504	005012		
7493	037676	000				
7494	037677	015	046412	046125	MSG28:	.ASCIZ <CR><LF>/MULTIPLE DRIVE DETECTION TEST/<CR><LF>
7495	037704	044524	046120	020105		
7496	037712	051104	053111	020105		
7497	037720	042504	042524	052103		
7498	037726	047511	020116	042524		
7499	037734	052123	005015	000		
7500	037741	015	050012	042514	MSG29:	.ASCIZ <CR><LF>/PLEASE WAIT, HEADS BEING LOADED/<CR><LF>
7501	037746	051501	020105	040527		
7502	037754	052111	020054	042510		
7503	037762	042101	020123	042502		
7504	037770	047111	020107	047514		
7505	037776	042101	042105	005015		
7506	040004	000				
7507	040005	015	053012	051105	MSG30:	.ASCIZ <CR><LF>/VERIFY DOOR CAN NOW BE OPENED & LEAVE IT OPEN/<CR><LF>
7508	040012	043111	020131	047504		

7509	040020	051117	041440	047101	
7510	040026	047040	053517	041140	
7511	040034	020105	050117	047105	
7512	040042	042105	023040	046040	
7513	040050	040505	042526	044440	
7514	040056	020124	050117	047105	
7515	040064	005015	000		
7516	040067	015	037412	005015	MSG31: .ASCIZ <CR><LF>/?<CR><LF>
7517	040074	000			
7518	040075	015	051012	046505	MSG32: .ASCIZ <CR><LF>/REMOVE DISK PACK & CLOSE DOOR/
7519	040102	053117	020105	044504	
7520	040110	045523	050040	041501	
7521	040116	020113	020046	046103	
7522	040124	051517	020105	047504	
7523	040132	051117	000		
7524	040135	015	042012	050105	MSG33: .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE DOOR OPEN/
7525	040142	042522	051523	023440	
7526	040150	052522	026516	052123	
7527	040156	050117	020047	053523	
7528	040164	052111	044103	052040	
7529	040172	020117	051047	047125	
7530	040200	020047	044127	046111	
7531	040206	020105	047504	051117	
7532	040214	047440	042520	000116	
7533	040222	005015	042526	044522	MSG34: .ASCIZ <CR><LF>/VERIFY SPINDLE DOES NOT START & HEADS DO NOT LOAD/<CR><LF>
7534	040230	054506	051440	044520	
7535	040236	042116	042514	042040	
7536	040244	042517	020123	047516	
7537	040252	020124	052123	051101	
7538	040260	020124	020046	042510	
7539	040266	042101	020123	047504	
7540	040274	047040	052117	046040	
7541	040302	040517	006504	000012	
7542	040310	005015	042504	051120	MSG35: .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE CARTRIDGE REMOVED/
7543	040316	051505	020123	051047	
7544	040324	047125	051455	047524	
7545	040332	023520	051440	044527	
7546	040340	041524	020110	047524	
7547	040346	023440	052522	023516	
7548	040354	053440	044510	042514	
7549	040362	041440	051101	051124	
7550	040370	042111	042507	051040	
7551	040376	046505	053117	042105	
7552	040404	000			
7553	040405	015	044412	051516	MSG36: .ASCIZ <CR><LF>/INSERT DISK PACK, DEPRESS 'RUN-STOP' TO 'RUN' & CLOSE DOOR/
7554	040412	051105	020124	044504	
7555	040420	045523	050040	041501	
7556	040426	026113	042040	050105	
7557	040434	042522	051523	023440	
7558	040442	052522	026516	052123	
7559	040450	050117	020047	047524	
7560	040456	023440	052522	023516	
7561	040464	023040	041440	047514	
7562	040472	042523	042040	047517	
7563	040500	000122			
7564	040502	005015	042504	051120	MSG37: .ASCIZ <CR><LF>/DEPRESS SPACE BAR WHEN FINISHED/<CR><LF>

7565	040510	051505	020123	050123
7566	040516	041501	020105	040502
7567	040524	020122	044127	047105
7568	040532	043040	047111	051511
7569	040540	042510	006504	000012
7570	040546	005015	047111	042523
7571	040554	052122	052440	044516
7572	040562	020124	042523	042514
7573	040570	052103	050040	052514
7574	040576	051507	020054	047111
7575	040604	040440	054516	047440
7576	040612	042122	051105	
7577	040616	005015	044124	020105
7578	040624	051120	043517	040522
7579	040632	020115	044527	046114
7580	040640	042440	044103	020117
7581	040646	044124	020105	047125
7582	040654	052111	051440	046105
7583	040662	041505	020124	046120
7584	040670	043525	047040	046525
7585	040676	042502	006522	000012
7586	040704	005015	042504	051120
7587	040712	051505	020123	047503
7588	040720	052116	047522	026514
7589	040726	020105	047524	042440
7590	040734	044530	020124	042524
7591	040742	052123	005015	000
7592	040747	015	053012	046117
7593	040754	046525	020105	040526
7594	040762	044514	020104	047516
7595	040770	020124	042523	124
7596	040775	015	046412	045501
7597	041002	020105	052523	042522
7598	041010	047440	044522	044507
7599	041016	040516	020114	047125
7600	041024	052111	051440	046105
7601	041032	041505	020124	046120
7602	041040	043525	044440	020123
7603	041046	047111	042523	052122
7604	041054	042105	005015	000
7605	041061	015	042012	051505
7606	041066	046105	041505	020124
7607	041074	047520	052122	044440
7608	041102	020116	051525	020105
7609	041110	020046	042523	042514
7610	041116	052103	047440	050120
7611	041124	051517	052111	020105
7612	041132	047520	052122	005015
7613	041140	000		
7614	041141	015	042012	051505
7615	041146	046105	041505	020124
7616	041154	051127	047117	020107
7617	041162	047520	052122	023040
7618	041170	051440	046105	041505
7619	041176	020124	047503	051122
7620	041204	041505	020124	047520

MSG38: .ASCII <CR><LF>/INSERT UNIT SELECT PLUGS, IN ANY ORDER/

.ASCIZ <CR><LF>/THE PROGRAM WILL ECHO THE UNIT SELECT PLUG NUMBER/<CR><LF>

MSG39: .ASCIZ <CR><LF>/DEPRESS CONTROL-E TO EXIT TEST/<CR><LF>

MSG40: .ASCII <CR><LF>/VOLUME VALID NOT SET/

.ASCIZ <CR><LF>/MAKE SURE ORIGINAL UNIT SELECT PLUG IS INSERTED/<CR><LF>

MSG41: .ASCIZ <CR><LF>/DESELECT PORT IN USE & SELECT OPPOSITE PORT/<CR><LF>

MSG42: .ASCIZ <CR><LF>/DESELECT WRONG PORT & SELECT CORRECT PORT/<CR><LF>

7621	041212	052122	005015	000	
7622	041217	015	042012	051505	MSG43: .ASCIZ <CR><LF>/DESELECT BOTH PORTS/<CR><LF>
7623	041224	046105	041505	020124	
7624	041232	047502	044124	050040	
7625	041240	051117	051524	005015	
7626	041246	000			
7627	041247	015	051412	046105	MSG44: .ASCIZ <CR><LF>/SELECT CORRECT PORT/<CR><LF>
7628	041254	041505	020124	047503	
7629	041262	051122	041505	020124	
7630	041270	047520	052122	005015	
7631	041276	000			
7632	041277	015	041412	051117	MSG45: .ASCIZ <CR><LF>/CORRECT PORT NOT SELECTED, TRY AGAIN/<CR><LF>
7633	041304	042522	052103	050040	
7634	041312	051117	020124	047516	
7635	041320	020124	042523	042514	
7636	041326	052103	042105	020054	
7637	041334	051124	020131	043501	
7638	041342	044501	006516	000012	
7639	041350	005015	042526	044522	MSG47: .ASCIZ <CR><LF>/VERIFY DOOR CANNOT BE OPENED (DO NOT FORCE)/<CR><LF>
7640	041356	054506	042040	047517	
7641	041364	020122	040503	047116	
7642	041372	052117	041040	020105	
7643	041400	050117	047105	042105	
7644	041406	024040	047504	047040	
7645	041414	052117	043040	051117	
7646	041422	042503	006451	000012	
7647	041430	005015	052524	047122	MSG49: .ASCIZ <CR><LF>/TURN OFF AC POWER FROM BEHIND THE RK06/<CR><LF>
7648	041436	047440	043106	040440	
7649	041444	020103	047520	042527	
7650	041452	020122	051106	046517	
7651	041460	041040	044105	047111	
7652	041466	020104	044124	020105	
7653	041474	045522	033060	005015	
7654	041502	000			
7655	041503	015	051412	044527	MSG50: .ASCIZ <CR><LF>/SWITCH AC POWER BACK ON/
7656	041510	041524	020110	041501	
7657	041516	050040	053517	051105	
7658	041524	041040	041501	020113	
7659	041532	047117	000		
7660	041535	015	046412	046517	MSG51: .ASCII <CR><LF>/MOMENTARILY REMOVE & INSERT THE SAME UNIT SELECT PLUG/
7661	041542	047105	040524	044522	
7662	041550	054514	051040	046505	
7663	041556	053117	020105	020046	
7664	041564	047111	042523	052122	
7665	041572	052040	042510	051440	
7666	041600	046501	020105	047125	
7667	041606	052111	051440	046105	
7668	041614	041505	020124	046120	
7669	041622	043525			
7670	041624	005015	047524	051040	.ASCIZ <CR><LF>/TO RESET VOLUME VALID/<CR><LF>
7671	041632	051505	052105	053040	
7672	041640	046117	046525	020105	
7673	041646	040526	044514	006504	
7674	041654	000012			
7675	041656	005015	042504	051120	MSG52: .ASCII <CR><LF>/DEPRESS SPACE TO DO TEST/
7676	041664	051505	020123	050123	

7677	041672	041501	020105	047524	
7678	041700	042040	020117	042524	
7679	041706	052123			
7680	041710	005015	051117	041440	.ASCIZ <CR><LF>/OR CONTROL-E TO BYPASS ENTIRE TEST/<CR><LF>
7681	041716	047117	051124	046117	
7682	041724	042455	052040	020117	
7683	041732	054502	040520	051523	
7684	041740	042440	052116	051111	
7685	041746	020105	042524	052123	
7686	041754	005015	000		
7687	041757	015	042012	051511	MSG53: .ASCIZ <CR><LF>/DISABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES OFF/<CR><LF>
7688	041764	041101	042514	052040	
7689	041772	042510	053440	044522	
7690	042000	042524	046040	041517	
7691	042006	020113	053523	052111	
7692	042014	044103	020054	042526	
7693	042022	044522	054506	046040	
7694	042030	043511	052110	043440	
7695	042036	042517	020123	043117	
7696	042044	006506	000012		
7697	042050	005015	047105	041101	MSG54: .ASCIZ <CR><LF>/ENABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES ON/<CR><LF>
7698	042056	042514	052040	042510	
7699	042064	053440	044522	042524	
7700	042072	046040	041517	020113	
7701	042100	053523	052111	044103	
7702	042106	020054	042526	044522	
7703	042114	054506	046040	043511	
7704	042122	052110	043440	042517	
7705	042130	020123	047117	005015	
7706	042136	000			
7707	042137	015	042412	044530	MSG55: .ASCIZ <CR><LF>/EXIT TEST WITH ORIGINAL UNIT SELECT PLUG NO. /
7708	042144	020124	042524	052123	
7709	042152	053440	052111	020110	
7710	042160	051117	043511	047111	
7711	042166	046101	052440	044516	
7712	042174	020124	042523	042514	
7713	042202	052103	050040	052514	
7714	042210	020107	047516	020056	
7715	042216	000			
7716	042217	015	040412	044514	MSG56: .ASCII <CR><LF>/ALIGNMENT CARTRIDGE USED/
7717	042224	047107	042515	052116	
7718	042232	041440	051101	051124	
7719	042240	042111	042507	052440	
7720	042246	042523	104		
7721	042251	015	050012	047522	.ASCII <CR><LF>/PROGRAM WILL BYPASS THE WRITE LOCK TEST/
7722	042256	051107	046501	053440	
7723	042264	046111	020114	054502	
7724	042272	040520	051523	052040	
7725	042300	042510	053440	044522	
7726	042306	042524	046040	041517	
7727	042314	020113	042524	052123	
7728	042322	005015	047101	020104	.ASCIZ <CR><LF>/AND READ-WRITE DATA PORTION OF AC LOW DETECTION TEST-PART 2/<CR>
7729	042330	042522	042101	053455	
7730	042336	044522	042524	042040	
7731	042344	052101	020101	047520	
7732	042352	052122	047511	020116	

7733	042360	043117	040440	020103	
7734	042366	047514	020127	042504	
7735	042374	042524	052103	047511	
7736	042402	020116	042524	052123	
7737	042410	050055	051101	020124	
7738	042416	006462	005012	000	
7739	042423	015	053012	051105	MSG57: .ASCIZ <CR><LF>/VERIFY BATTERY RETRACT FUNCTIONAL/<CR><LF>
7740	042430	043111	020131	040502	
7741	042436	052124	051105	020131	
7742	042444	042522	051124	041501	
7743	042452	020124	052506	041516	
7744	042460	044524	047117	046101	
7745	042466	005015	000		
7746	042471	015	046012	040517	MSG58: .ASCIZ <CR><LF>/LOAD HEADS ON ALL DRIVES TO BE TESTED FOR MDS/<CR><LF>
7747	042476	020104	042510	042101	
7748	042504	020123	047117	040440	
7749	042512	046114	042040	044522	
7750	042520	042526	020123	047524	
7751	042526	041040	020105	042524	
7752	042534	052123	042105	043040	
7753	042542	051117	046440	051504	
7754	042550	005015	000		
7755	042553	015	044412	051516	MSG59: .ASCIZ <CR><LF>/INSERT SAME UNIT SELECT PLUG NUMBER IN ANY 2 DRIVES TO BE TESTE
7756	042560	051105	020124	040523	
7757	042566	042515	052440	044516	
7758	042574	020124	042523	042514	
7759	042602	052103	050040	052514	
7760	042610	020107	052516	041115	
7761	042616	051105	044440	020116	
7762	042624	047101	020131	020062	
7763	042632	051104	053111	051505	
7764	042640	052040	020117	042502	
7765	042646	052040	051505	042524	
7766	042654	000104			
7767	042656	005015	047111	042523	MSG60: .ASCII <CR><LF>/INSERT CORRECT UNIT SELECT PLUGS & LOAD HEADS/
7768	042664	052122	041440	051117	
7769	042672	042522	052103	052440	
7770	042700	044516	020124	042523	
7771	042706	042514	052103	050040	
7772	042714	052514	051507	023040	
7773	042722	046040	040517	020104	
7774	042730	042510	042101	123	
7775	042735	015	047412	020116	.ASCIZ <CR><LF>/ON PREVIOUS 2 DRIVES/<CR><LF>
7776	042742	051120	053105	047511	
7777	042750	051525	031040	042040	
7778	042756	044522	042526	006523	
7779	042764	000012			
7780	042766	005015	052515	052114	MSG61: .ASCIZ <CR><LF>/MULTIPLE DRIVES FOUND ON DRIVE NO. /
7781	042774	050111	042514	042040	
7782	043002	044522	042526	020123	
7783	043010	047506	047125	020104	
7784	043016	047117	042040	044522	
7785	043024	042526	047040	027117	
7786	043032	000040			
7787	043034	005015	054502	040520	MSG62: .ASCII <CR><LF>/BYPASSING MULT. DRIVE SELECT TEST/
7788	043042	051523	047111	020107	

7789	043050	052515	052114	020056	
7790	043056	051104	053111	020105	
7791	043064	042523	042514	052103	
7792	043072	052040	051505	124	
7793	043077	015	047412	046116	.ASCIZ <CR><LF>/ONLY 1 DRIVE PRESENT/<CR><LF>
7794	043104	020131	020061	051104	
7795	043112	053111	020105	051120	
7796	043120	051505	047105	006524	
7797	043126	000012			
7798	043130	005015	042504	051120	MSG63: .ASCII <CR><LF>/DEPRESS SPACE BAR TO DO ANOTHER 2 DRIVES/
7799	043136	051505	020123	050123	
7800	043144	041501	020105	040502	
7801	043152	020122	047524	042040	
7802	043160	020117	047101	052117	
7803	043166	042510	020122	020062	
7804	043174	051104	053111	051505	
7805	043202	005015	051117	041440	.ASCII <CR><LF>/OR CONTROL-E TO EXIT TEST/
7806	043210	047117	051124	046117	
7807	043216	042455	052040	020117	
7808	043224	054105	052111	052040	
7809	043232	051505	124		
7810	043235	015	024012	047111	.ASCIZ <CR><LF>/((INSERT CORRECT UNIT SELECT PLUGS BEFORE EXITING))<CR><LF>
7811	043242	042523	052122	041440	
7812	043250	051117	042522	052103	
7813	043256	052440	044516	020124	
7814	043264	042523	042514	052103	
7815	043272	050040	052514	051507	
7816	043300	041040	043105	051117	
7817	043306	020105	054105	052111	
7818	043314	047111	024507	005015	
7819	043322	000			
7820	043323	015	042012	050105	MSG65: .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'STOP'/'
7821	043330	042522	051523	023440	
7822	043336	052522	026516	052123	
7823	043344	050117	020047	053523	
7824	043352	052111	044103	052040	
7825	043360	020117	051447	047524	
7826	043366	023520	000		
7827	043371	015	042012	051505	MSG67: .ASCIZ <CR><LF>/DESELECT PORT SWITCH ON ALL OTHER DRIVES/<CR><LF>
7828	043376	046105	041505	020124	
7829	043404	047520	052122	051440	
7830	043412	044527	041524	020110	
7831	043420	047117	040440	046114	
7832	043426	047440	044124	051105	
7833	043434	042040	044522	042526	
7834	043442	006523	000012		
7835	043446	005015	042526	044522	MSG68: .ASCIZ <CR><LF>/VERIFY BOTH DRIVES UNLOADED/<CR><LF>
7836	043454	054506	041040	052117	
7837	043462	020110	051104	053111	
7838	043470	051505	052440	046116	
7839	043476	040517	042504	006504	
7840	043504	000012			
7841	043506	005015	042504	051120	MSG69: .ASCIZ <CR><LF>/DEPRESS SPACE WHEN 'READY' LIGHT GOES ON/<CR><LF>
7842	043514	051505	020123	050123	
7843	043522	041501	020105	044127	
7844	043530	047105	023440	042522	

7845	043536	042101	023531	046040	
7846	043544	043511	052110	043440	
7847	043552	042517	020123	047117	
7848	043560	005015	000		
7849	043563	015	053012	051105	MSG70: .ASCIZ <CR><LF>/VERIFY HEADS LOAD/<CR><LF>
7850	043570	043111	020131	042510	
7851	043576	042101	020123	047514	
7852	043604	042101	005015	000	
7853	043611	015	051412	046105	MSG71: .ASCIZ <CR><LF>/SELECT CORRECT PORT SWITCH ON ALL OTHER DRIVES/<CR><LF>
7854	043616	041505	020124	047503	
7855	043624	051122	041505	020124	
7856	043632	047520	052122	051440	
7857	043640	044527	041524	020110	
7858	043646	047117	040440	046114	
7859	043654	047440	044124	051105	
7860	043662	042040	044522	042526	
7861	043670	006523	000012		
7862	043674	005015	041101	051117	MSG72: .ASCIZ <CR><LF>/ABORTING BALANCE OF TESTS/
7863	043702	044524	043516	041040	
7864	043710	046101	047101	042503	
7865	043716	047440	020106	042524	
7866	043724	052123	000123		
7867					
7868	043730	005015	051120	043517	MSG74: .ASCIZ <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
7869	043736	040522	020115	041101	
7870	043744	051117	020124	042520	
7871	043752	042116	047111	027107	
7872	043760	027056	046120	040505	
7873	043766	042523	053440	044501	
7874	043774	000124			
7875	043776	005015	040510	052114	MSG75: .ASCIZ <CR><LF>/HALT PENDING...PLEASE WAIT/
7876	044004	050040	047105	044504	
7877	044012	043516	027056	050056	
7878	044020	042514	051501	020105	
7879	044026	040527	052111	000	
7880	044033	015	050012	047522	MSG76: .ASCIZ <CR><LF>/PROGRAM ABORTED/
7881	044040	051107	046501	040440	
7882	044046	047502	052122	042105	
7883	044054	000			
7884	044055	015	041412	052520	MSG77: .ASCIZ <CR><LF>/CPU HALTED/
7885	044062	044040	046101	042524	
7886	044070	000104			
7887	044072	005015	047523	051122	MSG100: .ASCII <CR><LF>/SORRY, WRITE LOCK SHOULD NOT BE DEPRESSED/
7888	044100	026131	053440	044522	
7889	044106	042524	046040	041517	
7890	044114	020113	044123	052517	
7891	044122	042114	047040	052117	
7892	044130	041040	020105	042504	
7893	044136	051120	051505	042523	
7894	044144	104			
7895	044145	015	053412	044510	.ASCII <CR><LF>/WHILE ON TRACK 2 SECTORS 19, 20 OR 21/
7896	044152	042514	047440	020116	
7897	044160	051124	041501	020113	
7898	044166	020062	042523	052103	
7899	044174	051117	020123	034461	
7900	044202	020054	030062	047440	

7901	044210	020122	030462		
7902	044214	005015	046120	040505	.ASCIZ <CR><LF>/PLEASE TRY AGAIN/<CR><LF>
7903	044222	042523	052040	054522	
7904	044230	040440	040507	047111	
7905	044236	005015	000		
7906					
7907					.SBTTL ERROR MESSAGES
7908					
7909	044241	015	042412	051122	EM1: .ASCIZ <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
7910	044246	051117	020054	047117	
7911	044254	054514	030040	052040	
7912	044262	051110	020125	020067	
7913	044270	046101	047514	042527	
7914	044276	026104	052040	054522	
7915	044304	040440	040507	047111	
7916	044312	005015	000		
7917	044315	104	044522	042526	EM2: .ASCIZ /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/
7918	044322	021440	044440	020116	
7919	044330	045522	051503	020062	
7920	044336	040503	047116	052117	
7921	044344	041040	020105	042522	
7922	044352	042101	041040	041501	
7923	044360	020113	047503	051122	
7924	044366	041505	046124	020131	
7925	044374	047111	051040	046513	
7926	044402	031122	000		
7927	044405	015	040412	047502	EM3: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=
7928	044412	052122	052040	051505	
7929	044420	051524	027056	052456	
7930	044426	042516	050130	041505	
7931	044434	042524	020104	044524	
7932	044442	042515	047440	052125	
7933	044450	040440	020124	041520	
7934	044456	000075			
7935	044460	005015	041101	051117	EM4: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=
7936	044466	020124	042524	052123	
7937	044474	027123	027056	047125	
7938	044502	054105	042520	052103	
7939	044510	042105	044440	052116	
7940	044516	051105	052522	052120	
7941	044524	040440	020124	041520	
7942	044532	000075			
7943	044534	042115	020123	042523	EM5: .ASCIZ /MDS SET IN RKCS2/
7944	044542	020124	047111	051040	
7945	044550	041513	031123	000	
7946	044555	125	042506	051440	EM6: .ASCIZ /UFE SET IN RKCS2/
7947	044562	052105	044440	020116	
7948	044570	045522	051503	000062	
7949	044576	051104	020101	047111	EM7: .ASCIZ /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
7950	044604	051040	042113	020123	
7951	044612	020046	042516	020104	
7952	044620	047111	051040	041513	
7953	044626	031123	051040	051505	
7954	044634	052105	020073	051127	
7955	044642	047117	020107	047520	
7956	044650	052122	051440	046105	

7957	044656	041505	042524	037504	
7958	044664	000			
7959	044665	104	044522	042526	EM8: .ASCIZ /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
7960	044672	050040	042522	042523	
7961	044700	052116	041040	052125	
7962	044706	047040	052117	051440	
7963	044714	042520	044503	044506	
7964	044722	042105	041040	020131	
7965	044730	050117	051105	052101	
7966	044736	051117	000		
7967	044741	104	044522	042526	EM9: .ASCIZ /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
7968	044746	047040	052117	050040	
7969	044754	042522	042523	052116	
7970	044762	041040	052125	051440	
7971	044770	042520	044503	044506	
7972	044776	042105	041040	020131	
7973	045004	050117	051105	052101	
7974	045012	051117	000		
7975	045015	101	047502	052122	EM10: .ASCIZ /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
7976	045022	052040	051505	051524	
7977	045030	027056	041456	047101	
7978	045036	047516	020124	042522	
7979	045044	042506	042522	041516	
7980	045052	020105	047503	052116	
7981	045060	047522	046114	051105	
7982	045066	051040	043505	051511	
7983	045074	042524	000122		
7984	045100	051104	020101	047111	EM11: .ASCIZ /DRA IN RKDS & NED IN RKCS2 BOTH SET/
7985	045106	051040	042113	020123	
7986	045114	020046	042516	020104	
7987	045122	047111	051040	041513	
7988	045130	031123	041040	052117	
7989	045136	020110	042523	000124	
7990	045144	047503	052116	047522	EM12: .ASCIZ /CONTROLLER NOT READY IN RKCS1/
7991	045152	046114	051105	047040	
7992	045160	052117	051040	040505	
7993	045166	054504	044440	020116	
7994	045174	045522	051503	000061	
7995	045202	047516	040440	052124	EM13: .ASCIZ /NO ATTN IN RKASOF/
7996	045210	020116	047111	051040	
7997	045216	040513	047523	000106	
7998	045224	051127	047117	020107	EM14: .ASCIZ /WRONG ATTN IN RKASOF/
7999	045232	052101	047124	044440	
8000	045240	020116	045522	051501	
8001	045246	043117	000		
8002	045251	104	042122	020131	EM15: .ASCIZ /DRDY NOT CLEARED IN RKMR2/
8003	045256	047516	020124	046103	
8004	045264	040505	042522	020104	
8005	045272	047111	051040	046513	
8006	045300	031122	000		
8007	045303	104	041523	047040	EM16: .ASCIZ /DSC NOT SET IN RKMR2/
8008	045310	052117	051440	052105	
8009	045316	044440	020116	045522	
8010	045324	051115	000062		
8011	045330	042515	051523	043501	EM17: .ASCIZ /MESSAGE AD ERROR/
8012	045336	020105	030101	042440	

H12

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 149
ERROR MESSAGES

SEQ 0149

8013	045344	051122	051117	000	
8014	045351	115	051505	040523	EM18: .ASCIZ /MESSAGE B0 ERROR/
8015	045356	042507	041040	020060	
8016	045364	051105	047522	000122	
8017	045372	042515	051523	043501	EM19: .ASCIZ /MESSAGE A1 ERROR/
8018	045400	020105	030501	042440	
8019	045406	051122	051117	000	
8020	045413	115	051505	040523	EM20: .ASCIZ /MESSAGE B1 ERROR/
8021	045420	042507	041040	020061	
8022	045426	051105	047522	000122	
8023	045434	042503	051122	051440	EM21: .ASCIZ /CERR SET IN RKCS1/
8024	045442	052105	044440	020116	
8025	045450	045522	051503	000061	
8026	045456	047504	051117	051440	EM22: .ASCIZ /DOOR STATUS IN RKMR2 NOT CLEARED/
8027	045464	040524	052524	020123	
8028	045472	047111	051040	046513	
8029	045500	031122	047040	052117	
8030	045506	041440	042514	051101	
8031	045514	042105	000		
8032	045517	123	044520	042116	EM23: .ASCIZ /SPINDLE ON SET IN RKMR2
8033	045524	042514	047440	020116	
8034	045532	042523	020124	047111	
8035	045540	051040	046513	031122	
8036	045546	000			
8037	045547	126	046117	046525	EM24: .ASCIZ /VOLUME VALID NOT SET IN RKMR2/
8038	045554	020105	040526	044514	
8039	045562	020104	047516	020124	
8040	045570	042523	020124	047111	
8041	045576	051040	046513	031122	
8042	045604	000			
8043	045605	103	051101	051124	EM25: .ASCIZ /CARTRIDGE STATUS IN RKMR2 NOT CLEARED/
8044	045612	042111	042507	051440	
8045	045620	040524	052524	020123	
8046	045626	047111	051040	046513	
8047	045634	031122	047040	052117	
8048	045642	041440	042514	051101	
8049	045650	042105	000		
8050	045653	126	046117	046525	EM26: .ASCIZ /VOLUME VALID NOT CLEARED IN RKMR2/
8051	045660	020105	040526	044514	
8052	045666	020104	047516	020124	
8053	045674	046103	040505	042522	
8054	045702	020104	047111	051040	
8055	045710	046513	031122	000	
8056	045715	116	042105	047040	EM27: .ASCIZ /NED NOT SET IN RKCS2/
8057	045722	052117	051440	052105	
8058	045730	044440	020116	045522	
8059	045736	051503	000062		
8060	045742	047526	052514	042515	EM28: .ASCIZ /VOLUME VALID SET IN RKMR2/
8061	045750	053040	046101	042111	
8062	045756	051440	052105	044440	
8063	045764	020116	045522	051115	
8064	045772	000062			
8065	045774	051504	020103	047516	EM29: .ASCIZ /DSC NOT SET IN RKMR2/
8066	046002	020124	042523	020124	
8067	046010	047111	051040	046513	
8068	046016	031122	000		

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 150
ERROR MESSAGES

SEQ 0150

8069	046021	101	052124	020116	EM30:	.ASCIZ	/ATTN NOT RESET IN RKASOF/
8070	046026	047516	020124	042522			
8071	046034	042523	020124	047111			
8072	046042	051040	040513	047523			
8073	046050	000106					
8074	046052	042516	020104	047516	EM31:	.ASCIZ	/NED NOT CLEARED IN RKCS2/
8075	046060	020124	046103	040505			
8076	046066	042522	020104	047111			
8077	046074	051040	041513	031123			
8078	046102	000					
8079	046103	123	044520	042116	EM32:	.ASCIZ	/SPINDLE ON NOT SET IN RKMR2/
8080	046110	042514	047440	020116			
8081	046116	047516	020124	042523			
8082	046124	020124	047111	051040			
8083	046132	046513	031122	000			
8084	046137	104	044522	042526	EM33:	.ASCIZ	/DRIVE NOT READY IN RKMR2/
8085	046144	047040	052117	051040			
8086	046152	040505	054504	044440			
8087	046160	020116	045522	051115			
8088	046166	000062					
8089	046170	047504	051117	051440	EM34:	.ASCIZ	/DOOR STATUS BIT NOT SET IN RKMR2/
8090	046176	040524	052524	020123			
8091	046204	044502	020124	047516			
8092	046212	020124	042523	020124			
8093	046220	047111	051040	046513			
8094	046226	031122	000				
8095	046231	110	040505	051504	EM35:	.ASCIZ	/HEADS HOME NOT SET IN RKMR2/
8096	046236	044040	046517	020105			
8097	046244	047516	020124	042523			
8098	046252	020124	047111	051040			
8099	046260	046513	031122	000			
8100	046265	101	020103	047514	EM37:	.ASCIZ	/AC LOW NOT SET IN RKMR3/
8101	046272	020127	047516	020124			
8102	046300	042523	020124	047111			
8103	046306	051040	046513	031522			
8104	046314	000					
8105	046315	116	042105	047040	EM42:	.ASCIZ	/NED NOT SET IN RKCS2/
8106	046322	052117	051440	052105			
8107	046330	044440	020116	045522			
8108	046336	051503	000062				
8109	046342	041501	047514	047040	EM43:	.ASCIZ	/ACLO NOT CLEARED IN RKMR3/
8110	046350	052117	041440	042514			
8111	046356	051101	042105	044440			
8112	046364	020116	045522	051115			
8113	046372	000063					
8114	046374	047526	052514	042515	EM44:	.ASCIZ	/VOLUME VALID NOT CLEARED IN RKMR2/
8115	046402	053040	046101	042111			
8116	046410	047040	052117	041440			
8117	046416	042514	051101	042105			
8118	046424	044440	020116	045522			
8119	046432	051115	000062				
8120	046436	047526	052514	042515	EM45:	.ASCIZ	/VOLUME VALID SET IN RKMR2 AFTER HEADS LOADED/
8121	046444	053040	046101	042111			
8122	046452	051440	052105	044440			
8123	046460	020116	045522	051115			
8124	046466	020062	043101	042524			

8125	046474	020122	042510	042101	
8126	046502	020123	047514	042101	
8127	046510	042105	000		
8128	046513	116	047117	042455	EM46: .ASCIZ /NON-EXECUTABLE FUNCTION (NXF) NOT SET IN RKMR3/
8129	046520	042530	052503	040524	
8130	046526	046102	020105	052506	
8131	046534	041516	044524	047117	
8132	046542	024040	054116	024506	
8133	046550	047040	052117	051440	
8134	046556	052105	044440	020116	
8135	046564	045522	051115	000063	
8136	046572	054503	044514	042116	EM47: .ASCIZ /CYLINDER ADDRESS CHANGED FROM 0/
8137	046600	051105	040440	042104	
8138	046606	042522	051523	041440	
8139	046614	040510	043516	042105	
8140	046622	043040	047522	020115	
8141	046630	000060			
8142	046632	051127	052111	020105	EM48: .ASCIZ /WRITE LOCK IN RKMR2 NOT CLEARED/
8143	046640	047514	045503	044440	
8144	046646	020116	045522	051115	
8145	046654	020062	047516	020124	
8146	046662	046103	040505	042522	
8147	046670	000104			
8148	046672	051127	052111	020105	EM49: .ASCIZ /WRITE LOCK IN RKMR2 NOT SET/
8149	046700	047514	045503	044440	
8150	046706	020116	045522	051115	
8151	046714	020062	047516	020124	
8152	046722	042523	000124		
8153	046726	051127	052111	020105	EM50: .ASCIZ /WRITE LOCK ERROR IN RKMR3 NOT SET/
8154	046734	047514	045503	042440	
8155	046742	051122	051117	044440	
8156	046750	020116	045522	051115	
8157	046756	020063	047516	020124	
8158	046764	042523	000124		
8159	046770	051127	052111	020105	EM51: .ASCIZ /WRITE LOCK DID NOT OCCUR AT SECTOR BOUNDARY/
8160	046776	047514	045503	042040	
8161	047004	042111	047040	052117	
8162	047012	047440	041503	051125	
8163	047020	040440	020124	042523	
8164	047026	052103	051117	041040	
8165	047034	052517	042116	054522	
8166	047042	000			
8167	047043	125	05 .16	047040	EM52: .ASCIZ /UNS NOT SET IN RKMR3/
8168	047050	052117	051440	052105	
8169	047056	044440	020116	045522	
8170	047064	051115	000063		
8171					
8172	047070	047125	047514	042101	EM53: .ASCIZ /UNLOAD NOT SET IN RKMR2/
8173	047076	047040	052117	051440	
8174	047104	052105	044440	020116	
8175	047112	045522	051115	000062	
8176	047120	040503	047116	052117	EM54: .ASCIZ /CANNOT FIND MULT. DRIVE SELECT IN RKCS2/
8177	047126	043040	047111	020104	
8178	047134	052515	052114	020056	
8179	047142	051104	053111	020105	
8180	047150	042523	042514	052103	

8181	047156	044440	020116	045522	
8182	047164	051503	000062		
8183	047170	052101	047124	047040	EM55: .ASCIZ /ATTN NOT CLEARED IN RKASOF/
8184	047176	052117	041440	042514	
8185	047204	051101	042105	044440	
8186	047212	020116	045522	051501	
8187	047220	043117	000		
8188	047223	125	042516	050130	EM56: .ASCIZ /UNEXPECTED MEMORY PARITY ERROR TRAP/
8189	047230	041505	042524	020104	
8190	047236	042515	047515	054522	
8191	047244	050040	051101	052111	
8192	047252	020131	051105	047522	
8193	047260	020122	051124	050101	
8194	047266	000			
8195	047267	103	051105	020122	EM57: .ASCIZ /CERR IN RKCS1 NOT SET/
8196	047274	047111	051040	041513	
8197	047302	030523	047040	052117	
8198	047310	051440	052105	000	
8199	047315	120	051101	052111	EM58: .ASCIZ /PARITY NOT SET IN RKMR3/
8200	047322	020131	047516	020124	
8201	047330	042523	020124	047111	
8202	047336	051040	046513	031522	
8203	047344	000			
8204	047345	103	047524	051440	EM59: .ASCIZ /CTO SET IN RKCS1/
8205	047352	052105	044440	020116	
8206	047360	045522	051503	000061	
8207	047366	054116	020106	044504	EM61: .ASCIZ /NXF DID NOT SET FAULT/
8208	047374	020104	047516	020124	
8209	047402	042523	020124	040506	
8210	047410	046125	000124		
8211	047414	046104	020124	042523	EM63: .ASCIZ /DLT SET IN RKCS2/
8212	047422	020124	047111	051040	
8213	047430	041513	031123	000	
8214	047435	122	042113	020103	EM64: .ASCII /RKDC & RKDA INDICATE THAT WRITE CHECK ERROR/
8215	047442	020046	045522	040504	
8216	047450	044440	042116	041511	
8217	047456	052101	020105	044124	
8218	047464	052101	053440	044522	
8219	047472	042524	041440	042510	
8220	047500	045503	042440	051122	
8221	047506	051117			
8222	047510	005015	041517	052503	.ASCIZ <CR><LF>/OCCURRED AT CYL 411, TRACK 2, SECTOR 21/
8223	047516	051122	042105	040440	
8224	047524	020124	054503	020114	
8225	047532	030464	026061	052040	
8226	047540	040522	045503	031040	
8227	047546	020054	042523	052103	
8228	047554	051117	031040	000061	
8229	047562	042516	020104	042523	EM67: .ASCIZ /NED SET IN RKCS2/
8230	047570	020124	047111	051040	
8231	047576	041513	031123	000	
8232	047603	103	047101	047516	EM68: .ASCIZ /CANNOT READ BAD SECTOR INFORMATION/
8233	047610	020124	042522	042101	
8234	047616	041040	042101	051440	
8235	047624	041505	047524	020122	
8236	047632	047111	047506	046522	

8237	047640	052101	047511	000116	
8238	047646	047516	042040	044522	EM69: .ASCII /NO DRIVES FOUND ON BUSS/
8239	047654	042526	020123	047506	
8240	047662	047125	020104	047117	
8241	047670	041040	051525	123	
8242	047675	015	051412	052105	.ASCIZ <CR><LF>/SETUP CORRECTLY & PRESS 'CONTINUE'/<CR><LF>
8243	047702	050125	041440	051117	
8244	047710	042522	052103	054514	
8245	047716	023040	050040	042522	
8246	047724	051523	023440	047503	
8247	047732	052116	047111	042525	
8248	047740	006447	000012		
8249	047744	044127	046111	020105	EM70: .ASCIZ /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
8250	047752	040527	052111	047111	
8251	047760	020107	047506	020122	
8252	047766	047503	052116	020122	
8253	047774	042522	042101	020131	
8254	050002	051117	040440	052106	
8255	050010	051105	041440	047117	
8256	050016	051124	051040	040505	
8257	050024	054504	051040	041505	
8258	050032	042047	000		
8259	050035	104	052105	041505	EM71: .ASCIZ /DETECTED 10 BAD SECTORS...ABORTING TEST/
8260	050042	042524	020104	030061	
8261	050050	041040	042101	051440	
8262	050056	041505	047524	051522	
8263	050064	027056	040456	047502	
8264	050072	052122	047111	020107	
8265	050100	042524	052123	000	
8266	050105	104	052105	041505	EM72: .ASCIZ /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
8267	050112	042524	020104	051502	
8268	050120	020105	052502	020124	
8269	050126	047516	020124	044514	
8270	050134	052123	042105	044440	
8271	050142	020116	040502	020104	
8272	050150	042523	052103	051117	
8273	050156	043040	046111	000105	
8274	050164	042504	042524	052103	EM73: .ASCII /DETECTED BSE IN READ COMMAND/
8275	050172	042105	041040	042523	
8276	050200	044440	020116	042522	
8277	050206	042101	041440	046517	
8278	050214	040515	042116		
8279	050220	005015	052502	020124	.ASCIZ <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
8280	050226	047516	020124	047111	
8281	050234	050040	042522	044526	
8282	050242	052517	020123	051127	
8283	050250	052111	020105	047503	
8284	050256	046515	047101	020104	
8285	050264	047524	051440	046501	
8286	050272	020105	042523	052103	
8287	050300	051117	000		
8288	050303	122	055124	047040	EM74: .ASCIZ /RTZ NOT SET IN RKMR2/
8289	050310	052117	051440	052105	
8290	050316	044440	020116	045522	
8291	050324	051115	000062		
8292	050330	005015	042504	042524	EM75: .ASCIZ <CR><LF>/DETECTED 10 BAD CYLINDERS...ABORTING TEST/

8293	050336	052103	042105	030440	
8294	050344	020060	040502	020104	
8295	050352	054503	044514	042116	
8296	050360	051105	027123	027056	
8297	050366	041101	05117	044524	
8298	050374	043516	052040	051505	
8299	050402	000124			
8300	050404	047516	042040	044522	EM76: .ASCII /NO DRIVES FOUND IN DEVICE MAP (\$DEVN)/
8301	050412	042526	020123	047506	
8302	050420	047125	020104	047111	
8303	050426	042040	053105	041511	
8304	050434	020105	040515	020120	
8305	050442	022050	042504	046526	
8306	050450	051			
8307	050451	015	051412	052105	.ASCIZ <CR><LF>/SETUP CORRECTLY & RESTART/<CR><LF>
8308	050456	050125	041440	051117	
8309	050464	042522	052103	054514	
8310	050472	023040	051040	051505	
8311	050500	040524	052122	005015	
8312	050506	000			
8313	050507	127	044522	042524	EM80: .ASCIZ /WRITE CHECK ERROR SET IN RKCS2/
8314	050514	041440	042510	045503	
8315	050522	042440	051122	051117	
8316	050530	051440	052105	044440	
8317	050536	020116	045522	051503	
8318	050544	000062			
8319	050546	040504	040524	041440	EM83: .ASCIZ /DATA CHECK ERROR SET IN RKER/
8320	050554	042510	045503	042440	
8321	050562	051122	051117	051440	
8322	050570	052105	044440	020116	
8323	050576	045522	051105	000	
8324	050603	122	040505	044504	EM93: .ASCIZ /READING WRONG CYLINDER # IN HEADER/
8325	050610	043516	053440	047522	
8326	050616	043516	041440	046131	
8327	050624	047111	042504	020122	
8328	050632	020043	047111	044040	
8329	050640	040505	042504	000122	
8330					
8331					.SBTTL DATA HEADERS
8332					
8333	050646	042524	052123	047040	DH1: .ASCIZ /TEST NO. PC/
8334	050654	027117	020040	041520	
8335	050662	000			
8336	050663	122	046513	030522	DH2: .ASCIZ /RKMR1 RKMR2 RKMR3 RKER RKDS RKCS1 RKCS2/
8337	050670	051011	046513	031122	
8338	050676	051011	046513	031522	
8339	050704	051011	042513	004522	
8340	050712	045522	051504	051011	
8341	050720	041513	030523	051011	
8342	050726	041513	031123	000	
8343	050733	122	053513	004503	DH3: .ASCIZ /RKWC RKBA RKDA RKASOF RKDC RKECPS RKECPT/
8344	050740	045522	040502	051011	
8345	050746	042113	004501	045522	
8346	050754	051501	043117	051011	
8347	050762	042113	004503	045522	
8348	050770	041505	051520	051011	

8349	050776	042513	050103	000124	
8350	051004	043101	042524	020122	DH4: .ASCIZ /AFTER PACK RE-INSERTED & HEADS LOADED/
8351	051012	040520	045503	051040	
8352	051020	026505	047111	042523	
8353	051026	052122	042105	023040	
8354	051034	044040	040505	051504	
8355	051042	046040	040517	042504	
8356	051050	000104			
8357	051052	043101	042524	020122	DH5: .ASCIZ /AFTER AC SWITCHED OFF/
8358	051060	041501	051440	044527	
8359	051066	041524	042510	020104	
8360	051074	043117	000106		
8361	051100	043101	042524	020122	DH8: .ASCIZ /AFTER DRIVE UNLOADED & DOOR OPENED/
8362	051106	051104	053111	020105	
8363	051114	047125	047514	042101	
8364	051122	042105	023040	042040	
8365	051130	047517	020122	050117	
8366	051136	047105	042105	000	
8367	051143	101	052106	051105	DH11: .ASCIZ /AFTER MANUALLY LOADING HEADS WITH DOOR OPEN/
8368	051150	046440	047101	040525	
8369	051156	046114	020131	047514	
8370	051164	042101	047111	020107	
8371	051172	042510	042101	020123	
8372	051200	044527	044124	042040	
8373	051206	047517	020122	050117	
8374	051214	047105	000		
8375	051217	101	052106	051105	DH12: .ASCIZ /AFTER DISK PACK REMOVED/
8376	051224	042040	051511	020113	
8377	051232	040520	045503	051040	
8378	051240	046505	053117	042105	
8379	051246	000			
8380	051247	101	052106	051105	DH13: .ASCIZ /AFTER MANUALLY LOADING HEADS WITH DISK PACK REMOVED/
8381	051254	046440	047101	040525	
8382	051262	046114	020131	047514	
8383	051270	042101	047111	020107	
8384	051276	042510	042101	020123	
8385	051304	044527	044124	042040	
8386	051312	051511	020113	040520	
8387	051320	045103	051040	046505	
8388	051326	053117	042105	000	
8389	051333	127	052111	047510	DH15: .ASCIZ /WITHOUT PACK COMMAND/
8390	051340	052125	050040	041501	
8391	051346	020113	047503	046515	
8392	051354	047101	000104		
8393	051360	043101	042524	020122	DH16: .ASCIZ /AFTER UNIT SELECT PLUG REMOVED/
8394	051366	047125	052111	051440	
8395	051374	046105	041505	020124	
8396	051402	046120	043525	051040	
8397	051410	046505	053117	042105	
8398	051416	000			
8399	051417	101	052106	051105	DH17: .ASCIZ /AFTER RECAL COMMAND/
8400	051424	051040	041505	046101	
8401	051432	041440	046517	040515	
8402	051440	042116	000		
8403	051443	101	052106	051105	DH18: .ASCIZ /AFTER UNLOAD COMMAND/
8404	051450	052440	046116	040517	

8405	051456	020104	047503	046515		
8406	051464	047101	000104			
8407	051470	043101	042524	020122	DH19:	.ASCIZ /AFTER PACK COMMAND/
8408	051476	040520	045503	041440		
8409	051504	046517	040515	042116		
8410	051512	000				
8411	051513	101	052106	051105	DH20:	.ASCIZ /AFTER SELECT DRIVE COMMAND/
8412	051520	051440	046105	041505		
8413	051526	020124	051104	053111		
8414	051534	020105	047503	046515		
8415	051542	047101	000104			
8416	051546	043101	042524	020122	DH21:	.ASCIZ /AFTER SUBSYSTEM CLEAR/
8417	051554	052523	051502	051531		
8418	051562	042524	020115	046103		
8419	051570	040505	000122			
8420	051574	043101	042524	020122	DH22:	.ASCIZ /AFTER DRIVE CLEAR COMMAND/
8421	051602	051104	053111	020105		
8422	051610	046103	040505	020122		
8423	051616	047503	046515	047101		
8424	051624	000104				
8425	051626	043101	042524	020122	DH23:	.ASCIZ /AFTER WRONG PORT SELECTED/
8426	051634	051127	047117	020107		
8427	051642	047520	052122	051440		
8428	051650	046105	041505	042524		
8429	051656	000104				
8430	051660	043101	042524	020122	DH25:	.ASCIZ /AFTER SEEK COMMAND/
8431	051666	042523	045505	041440		
8432	051674	046517	040515	042116		
8433	051702	000				
8434	051703	101	052106	051105	DH26:	.ASCIZ /AFTER READ DATA COMMAND/
8435	051710	051040	040505	020104		
8436	051716	040504	040524	041440		
8437	051724	046517	040515	042116		
8438	051732	000				
8439	051733	101	052106	051105	DH27:	.ASCIZ /AFTER WRITE DATA COMMAND/
8440	051740	053440	044522	042524		
8441	051746	042040	052101	020101		
8442	051754	047503	046515	047101		
8443	051762	000104				
8444	051764	043101	042524	020122	DH28:	.ASCIZ /AFTER BOTH PORTS DESELECTED/
8445	051772	047502	044124	050040		
8446	052000	051117	051524	042040		
8447	052006	051505	046105	041505		
8448	052014	042524	000104			
8449	052020	043101	042524	020122	DH29:	.ASCIZ /AFTER CORRECT PORT SELECTED/
8450	052026	047503	051122	041505		
8451	052034	020124	047520	052122		
8452	052042	051440	046105	041505		
8453	052050	042524	000104			
8454	052054	043101	042524	020122	DH30:	.ASCIZ /AFTER READ HEADER COMMAND/
8455	052062	042522	042101	044040		
8456	052070	040505	042504	020122		
8457	052076	047503	046515	047101		
8458	052104	000104				
8459	052106	043101	042524	020122	DH31:	.ASCIZ /AFTER DRIVE MANUALLY LOADED/
8460	052114	051104	053111	020105		

8461	052122	040515	052516	046101	
8462	052130	054514	046040	040517	
8463	052136	042504	000104		
8464	052142	043101	042524	020122	DH32: .ASCIZ /AFTER WRITE CHECK COMMAND/
8465	052150	051127	052111	020105	
8466	052156	044103	041505	020113	
8467	052164	047503	046515	047101	
8468	052172	000104			
8469	052174	043101	042524	020122	DH34: .ASCIZ /AFTER START SPINDLE COMMAND/
8470	052202	052123	051101	020124	
8471	052210	050123	047111	046104	
8472	052216	020105	047503	046515	
8473	052224	047101	000104		
8474	052230	043101	042524	020122	DH35: .ASCIZ /AFTER MANUALLY UNLOADING/
8475	052236	040515	052516	046101	
8476	052244	054514	052440	046116	
8477	052252	040517	044504	043516	
8478	052260	000			
8479	052261	101	052106	051105	DH37: .ASCIZ /AFTER TIMEOUT TO POWER DOWN/
8480	052266	052040	046511	047505	
8481	052274	052125	052040	020117	
8482	052302	047520	042527	020122	
8483	052310	047504	047527	000	
8484	052315	101	052106	051105	DH38: .ASCIZ /AFTER AC POWERED UP/
8485	052322	040440	020103	047520	
8486	052330	042527	042522	020104	
8487	052336	050125	000		
8488	052341	101	052106	051105	DH39: .ASCIZ /AFTER WRITE HEADER COMMAND/
8489	052346	053440	044522	042524	
8490	052354	044040	040505	042504	
8491	052362	020122	047503	046515	
8492	052370	047101	000104		
8493	052374	052504	044522	043516	DH41: .ASCIZ /DURING RECAL COMMAND/
8494	052402	051040	041505	046101	
8495	052410	041440	046517	040515	
8496	052416	042116	000		
8497	052421	117	020116	042523	DH42: .ASCIZ /ON SECTORS 0,2,4,6 OR 8 CYL 410 TRACK 2/
8498	052426	052103	051117	020123	
8499	052434	026060	026062	026064	
8500	052442	020066	051117	034040	
8501	052450	020040	054503	020114	
8502	052456	030464	020060	051124	
8503	052464	041501	020113	000062	
8504	052472	047506	046522	052101	DH44: .ASCIZ /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
8505	052500	023040	040440	046114	
8506	052506	051040	040505	026504	
8507	052514	051127	052111	020105	
8508	052522	042524	052123	020123	
8509	052530	044527	046114	041040	
8510	052536	020105	054502	040520	
8511	052544	051523	042105	000	
8512	052551	101	052106	051105	DH45: .ASCIZ /AFTER SEEK WITH VOLUME VALID=0/
8513	052556	051440	042505	020113	
8514	052564	044527	044124	053040	
8515	052572	046117	046525	020105	
8516	052600	040526	044514	036504	

8517	052606	000060				
8518	052610	043101	042524	020122	DH46:	.ASCIZ /AFTER WRITE DATA WITH VOLUME VALID=0/
8519	052616	051127	052111	020105		
8520	052624	040504	040524	053440		
8521	052632	052111	020110	047526		
8522	052640	052514	042515	053040		
8523	052646	046101	042111	030075		
8524	052654	000				
8525	052655	101	052105	051105	DH48:	.ASCIZ /AFTER WRITE LOCK SWITCH DISABLED/
8526	052662	053440	044522	042524		
8527	052670	046040	041517	020113		
8528	052676	053523	052111	044103		
8529	052704	042040	051511	041101		
8530	052712	042514	000104			
8531	052716	043101	042524	020122	DH49:	.ASCIZ /AFTER WRITE LOCK SWITCH ENABLED/
8532	052724	051127	052111	020105		
8533	052732	047514	045503	051440		
8534	052740	044527	041524	020110		
8535	052746	047105	041101	042514		
8536	052754	000104				
8537	052756	043101	042524	020122	DH50:	.ASCIZ /AFTER WRITING WITH WRITE LOCK ENABLED/
8538	052764	051127	052111	047111		
8539	052772	020107	044527	044124		
8540	053000	053440	044522	042524		
8541	053006	046040	041517	020113		
8542	053014	047105	041101	042514		
8543	053022	000104				
8544	053024	054503	020114	004443	DH56:	.ASCIZ /CYL # HEADER WORD 0/
8545	053032	042510	042101	051105		
8546	053040	053440	051117	020104		
8547	053046	000060				
8548	053050	043101	042524	020122	DH59:	.ASCIZ /AFTER DRIVE SELECT COMMAND WITH EVEN PARITY/
8549	053056	051104	053111	020105		
8550	053064	042523	042514	052103		
8551	053072	041440	046517	040515		
8552	053100	042116	053440	052111		
8553	053106	020110	053105	047105		
8554	053114	050040	051101	052111		
8555	053122	000131				
8556	053124	043101	042524	020122	DH60:	.ASCIZ /AFTER WRITE LOCK ENABLED DURING CONTINUOUS WRITING/
8557	053132	051127	052111	020105		
8558	053140	047514	045503	042440		
8559	053146	040516	046102	042105		
8560	053154	042040	051125	047111		
8561	053162	020107	047503	052116		
8562	053170	047111	047525	051525		
8563	053176	053440	044522	044524		
8564	053204	043516	000			
8565	053207	122	042113	004501	DH61:	.ASCII /RKDA WORD EXPECTED/
8566	053214	047527	042122	042411		
8567	053222	050130	041505	042524		
8568	053230	104				
8569	053231	015	040012	053440		.ASCIZ <CR><LF>/@ WRL WAS WORD/
8570	053236	046122	053411	051501		
8571	053244	053411	051117	000104		
8572	053252	045522	040504	053411	DH62:	.ASCII /RKDA WORD EXPECTED/

8573	053260	051117	004504	054105	
8574	053266	042520	052103	042105	
8575	053274	005015	042502	047506	.ASCII <CR><LF>/BEFORE WAS WORD/
8576	053302	042522	053411	051501	
8577	053310	053411	051117	104	
8578	053315	015	053412	046122	.ASCIZ <CR><LF>/WRL/
8579	053322	000			
8580	053323	101	052106	051105	DH63: .ASCIZ /AFTER WRITE LOCK ENABLED FROM AC FAILURE/
8581	053330	053440	044522	042524	
8582	053336	046040	041517	020113	
8583	053344	047105	041101	042514	
8584	053352	020104	051106	046517	
8585	053360	040440	020103	040506	
8586	053366	046111	051125	000105	
8587	053374	043101	042524	020122	DH64: .ASCIZ /AFTER MDS DETECTED IN RKCS2/
8588	053402	042115	020123	042504	
8589	053410	042524	052103	042105	
8590	053416	044440	020116	045522	
8591	053424	051503	000062		
8592	053430	043101	042524	020122	DH65: .ASCIZ /AFTER SEARCHING ALL DRIVES PRESENT/
8593	053436	042523	051101	044103	
8594	053444	047111	020107	046101	
8595	053452	020114	051104	053111	
8596	053460	051505	050040	042522	
8597	053466	042523	052116	000	
8598	053473	124	051505	020124	DH66: .ASCIZ /TEST NO. TRAP PC/
8599	053500	047516	004456	051124	
8600	053506	050101	050040	000103	
8601	053514	043101	042524	020122	DH67: .ASCIZ /AFTER TIMEOUT TO ENABLE WRITE LOCK/
8602	053522	044524	042515	052517	
8603	053530	020124	047524	042440	
8604	053536	040516	046102	020105	
8605	053544	051127	052111	020105	
8606	053552	047514	045503	000	
8607	053557	011	054105	042520	DH69: .ASCIZ / EXPECT/
8608	053564	052103	000		
8609	053567	011	041501	052524	DH70: .ASCIZ / ACTUAL/
8610	053574	046101	000		
8611	053577	101	004460	030102	DH71: .ASCIZ /A0 B0 A1 B1 A2 B2 B3/
8612	053604	040411	004461	030502	
8613	053612	040411	004462	031102	
8614	053620	041011	000063		
8615	053624	051515	020107	020101	DH73: .ASCIZ /MSG A & B IN RKMR2 & RKMR3 RESP ARE INVALID/
8616	053632	020046	020102	047111	
8617	053640	051040	046513	031122	
8618	053646	023040	051040	046513	
8619	053654	031522	051040	051505	
8620	053662	020120	051101	020105	
8621	053670	047111	040526	044514	
8622	053676	000104			
8623	053700	047117	051440	041505	DH74: .ASCIZ /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
8624	053706	047524	051522	030440	
8625	053714	026060	030440	026062	
8626	053722	030440	026064	030440	
8627	053730	026066	030440	020070	
8628	053736	051117	031040	020060	

8629	053744	054503	020114	030464	
8630	053752	020060	051124	041501	
8631	053760	020113	000062		
8632					
8633					.SBTTL ERROR OUTPUT DATA
8634					
8635					.EVEN
8636	053764	001214	001116		DT1: \$TESTN,\$ERRPC
8637	053770	005432	005434	005436	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8638	053776	005422	005420	005406	
8639	054004	005410			
8640	054006	005412	005414	005416	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8641	054014	005424	005426	005440	
8642	054022	005442			
8643	054024	001214	001116	001166	DT3: \$TESTN,\$ERRPC,\$TMP3,WD1,WD2
8644	054032	001502	001504		
8645	054036	005432	005434	005436	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8646	054044	005422	005420	005406	
8647	054052	005410			
8648	054054	005412	005414	005416	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8649	054062	005424	005426	005440	
8650	054070	005442			
8651	054072	001214	001354		DT6: \$TESTN,TRAPPC
8652	054076	001214	001116	001366	DT9: \$TESTN,\$ERRPC,TOCYL,RHTAB
8653	054104	001760			
8654	054106	005432	005434	005436	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8655	054114	005422	005420	005406	
8656	054122	005410			
8657	054124	005412	005414	005416	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8658	054132	005424	005426	005440	
8659	054140	005442			
8660	054142	001214	001116	005476	DT13: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
8661	054150	005500	005502	005504	
8662	054156	005456	005460	005462	
8663	054164	005464			
8664	054166	005432	005434	005436	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8665	054174	005422	005420	005406	
8666	054202	005410			
8667	054204	005412	005414	005416	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8668	054212	005424	005426	005440	
8669	054220	005442			
8670					
8671	054222	001214	001116	005476	DT14: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
8672	054230	005500	005502	005504	
8673	054236	005506	005510		
8674	054242	005456	005460	005462	H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
8675	054250	005464	005466	005470	
8676	054256	005432	005434	005436	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8677	054264	005422	005420	005406	
8678	054272	005410			
8679	054274	005412	005414	005416	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8680	054302	005424	005426	005440	
8681	054310	005442			
8682					
8683	054312	001214	001116	005476	DT15: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
8684	054320	005500	005502	005504	

8685	054326	005506	005510	005514	
8686	054334	005456	005460	005462	H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
8687	054342	005464	005466	005470	
8688	054350	005474			
8689	054352	005432	005434	005436	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8690	054360	005422	005420	005406	
8691	054366	005410			
8692	054370	005412	005414	005416	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8693	054376	005424	005426	005440	
8694	054404	005442			
8695					
8696					

.SBTTL ERROR DATA FORMATS

8698	054406	000003			DF1: 3
8699	054410	002	000		.BYTE 2,0
8700	054412	050663			DH2
8701	054414	007	000		.BYTE 7,0
8702	054416	050733			DH3
8703	054420	007	000		.BYTE 7,0
8704					
8705					
8706	054422	000005			DF3: 5
8707	054424	000	000		.BYTE 0,0
8708	054426	050646			DH1
8709	054430	002	000		.BYTE 2,0
8710	054432	053207			DH61
8711	054434	003	000		.BYTE 3,0
8712	054436	050663			DH2
8713	054440	007	000		.BYTE 7,0
8714	054442	050733			DH3
8715	054444	007	000		.BYTE 7,0
8716					
8717	054446	000005			DF4: 5
8718	054450	000	000		.BYTE 0,0
8719	054452	050646			DH1
8720	054454	002	000		.BYTE 2,0
8721	054456	053252			DH62
8722	054460	003	000		.BYTE 3,0
8723	054462	050663			DH2
8724	054464	007	000		.BYTE 7,0
8725	054466	050733			DH3
8726	054470	007	000		.BYTE 7,0
8727					
8728	054472	000001			DF5: 1
8729	054474	002	000		.BYTE 2,0
8730					
8731	054476	000003			DF7: 3
8732	054500	002	000		.BYTE 2,0
8733	054502	050663			DH2
8734	054504	007	000		.BYTE 7,0
8735	054506	050733			DH3
8736	054510	007	000		.BYTE 7,0
8737	054512	000004			DF10: 4
8738	054514	000	000		.BYTE 0,0
8739	054516	050646			DH1
8740	054520	002	000		.BYTE 2,0

H13

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 162
ERROR DATA FORMATS

SEQ 0162

8741	054522	050663		DH2	
8742	054524	007	000	.BYTE	7,0
8743	054526	050733		DH3	
8744	054530	007	000	.BYTE	7,0
8745					
8746	054532	000005		DF12:	5
8747	054534	000	000	.BYTE	0,0
8748	054536	053624		DH73	
8749	054540	000	000	.BYTE	0,0
8750	054542	050646		DH1	
8751	054544	002	000	.BYTE	2,0
8752	054546	050663		DH2	
8753	054550	007	000	.BYTE	7,0
8754	054552	050733		DH3	
8755	054554	007	000	.BYTE	7,0
8756					
8757					
8758	054556	000004		DF15:	4
8759	054560	000	000	.BYTE	0,0
8760	054562	050646		DH1	
8761	054564	002	000	.BYTE	2,0
8762	054566	050663		DH2	
8763	054570	007	000	.BYTE	7,0
8764	054572	050733		DH3	
8765	054574	007	000	.BYTE	7,0
8766					
8767					
8768	054576	000005		DF17:	5
8769	054600	000	000	.BYTE	0,0
8770	054602	052472		DH44	
8771	054604	000	000	.BYTE	0,0
8772	054606	050646		DH1	
8773	054610	002	000	.BYTE	2,0
8774	054612	050663		DH2	
8775	054614	007	000	.BYTE	7,0
8776	054616	050733		DH3	
8777	054620	007	000	.BYTE	7,0
8778	054622	000005		DF20:	5
8779	054624	000	000	.BYTE	0,0
8780	054626	050646		DH1	
8781	054630	002	000	.BYTE	2,0
8782	054632	053024		DH56	
8783	054634	002	000	.BYTE	2,0
8784	054636	050663		DH2	
8785	054640	007	000	.BYTE	7,0
8786	054642	050733		DH3	
8787	054644	007	000	.BYTE	7,0
8788					
8789	054646	000007		DF21:	7
8790	054650	000	000	.BYTE	0,0
8791	054652	050646		DH1	
8792	054654	002	000	.BYTE	2,0
8793	054656	053557		DH69	
8794	054660	000	000	.BYTE	0,0
8795	054662	053577		DH71	
8796	054664	004	000	.BYTE	4,0

```

8797 054666 053567
8798 054670 004 000
8799 054672 050663
8800 054674 007 000
8801 054676 050733
8802 054700 007 000
8803
8804 054702 000007
8805 054704 000 000
8806 054706 050646
8807 054710 002 000
8808 054712 053557
8809 054714 000 000
8810 054716 053577
8811 054720 006 000
8812 054722 053567
8813 054724 006 000
8814 054726 050663
8815 054730 007 000
8816 054732 050733
8817 054734 007 000
8818
8819 054736 000007
8820 054740 000 000
8821 054742 050646
8822 054744 002 000
8823 054746 053557
8824 054750 000 000
8825 054752 053577
8826 054754 007 000
8827 054756 053567
8828 054760 007 000
8829 054762 050663
8830 054764 007 000
8831 054766 050733
8832 054770 007 000
8833
8834
8835
8836
8837
8838
8839
8840
8841
8842
8843 054772 104413
8844 054774 032777 020000 124136
8845 055002 001107
8846 055004 113700 001114
8847 055010 042700 177400
8848 055014 005300
8849 055016 006300
8850 055020 006300
8851 055022 006300
8852 055024 062700 005560

```

```

DH70
.BYTE 4,0
DH2
.BYTE 7,0
DH3
.BYTE 7,0
DF22: 7
.BYTE 0,0
DH1
.BYTE 2,0
DH69
.BYTE 0,0
DH71
.BYTE 6,0
DH70
.BYTE 6,0
DH2
.BYTE 7,0
DH3
.BYTE 7,0
DF23: 7
.BYTE 0,0
DH1
.BYTE 2,0
DH69
.BYTE 0,0
DH71
.BYTE 7,0
DH70
.BYTE 7,0
DH2
.BYTE 7,0
DH3
.BYTE 7,0

```

```

*****
.SBTTL TYPE ERROR ROUTINE
*ENTRY JSR PC,TYP ERR
*RETURN RTS PC
*
*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
*THE ERROR.
*****
†TYPERR: SAVREG
BIT #SW13,JSWR ;INHIBIT ERROR TYPEOUTS?
BNE 205 ;YES-BRANCH
MOVB $ITEMB,RO ;ENTER ERROR NUMBER
BIC #177400,RO ;CLEAR SIGN EXTENSION
DEC RO ;FORM INDEX FOR ERROR TABLE
ASL RO
ASL RO
ASL RO
19: ADD #ERRTB,RO ;FORM ADDRESS OF ERROR ENTRY

```

```

8853 055030 012037 055044           MOV      (R0)+,2$      ;GET EM POINTER
8854 055034 001404                   BEQ      3$           ;BRANCH IF THERE ISN'T ONE
8855 055036 104401 001205           TYPE     ,$CRLF      ;TYPE CARRIAGE RETURN LINE FEED
8856 055042 104401                   TYPE     ;TYPE ERROR MESSAGE (EM)
8857 055044 000000 2$: .WORD      0       ;EM POINTER GOES HERE
8858 055046 012037 055062 3$: MOV      (R0)+,4$      ;GET DH POINTER
8859 055052 001404                   BEQ      5$           ;BRANCH IF THERE ISN'T ONE
8860 055054 104401 001205           TYPE     , $CRLF      ;TYPE CR-LF
8861 055060 104401                   TYPE     ;TYPE DATA HEADER
8862 055062 000000 4$: .WORD      0       ;DH POINTER GOES HERE
8863 055064 012001 5$: MOV      (R0)+,R1      ;GET DT POINTER
8864 055066 001455                   BEQ      20$          ;BRANCH IF THERE ARE NONE
8865 055070 005004                   CLR      R4           ;SET INDENT SWITCH
8866 055072 012000                   MOV      (R0)+,R0      ;GET DF POINTER
8867 055074 012002                   MOV      (R0)+,R2      ;STORE NUMBER OF DH'S
8868 055076 001446                   BEQ      17$          ;DH NUM IS 0-BRANCH
8869 055100 005104                   COM      R4           ;NO INDENT
8870 055102 104401 001205           TYPE     , $CRLF      ;
8871 055106 112003 10$: MOV      (R0)+,R3      ;GET & STORE NUMBER OF DATA WORDS
8872 055110 105720                   TSTB    (R0)+         ;BUMP PAST FORMAT WORD
8873 055112 005703                   TST     R3           ;TEST IF ANY DATA FOR THIS HEADER
8874 055114 001407                   BEQ      14$          ;NO - SKIP DATA PRINT
8875 055116 013146 11$: MOV      2(R1)+,-(SP) ;PUT FIRST DATA WORD ON STACK
8876 055120 104402                   TYPOC   ;TYPE IT
8877 055122 005303                   DEC     R3           ;MORE DATA WORDS
8878 055124 001403                   BEQ      14$          ;NO-BRANCH
8879 055126 104401 055256           TYPE     ,SPACE2     ;TYPE SEPARATORS
8880 055132 000771                   BR      11$          ;LOOP
8881 055134 005302 14$: DEC     R2           ;MORE DH'S?
8882 055136 003431                   BLE     20$          ;NO-BRANCH
8883 055140 104401 001205           TYPE     , $CRLF      ;
8884 055144 005760 000002           TST     2(R0)         ;ONLY A DH IN THIS REQUEST?
8885 055150 001404                   BEQ      15$          ;YES-BRANCH BYPASS INDENT
8886 055152 005104                   COM     R4           ;INDENT?
8887 055154 001002                   BNE     15$          ;NO-BRANCH
8888 055156 104401 055256           TYPE     ,SPACE2     ;YES-TYPE SPACES
8889 055162 012037 055170 15$: MOV      (R0)+,16$    ;GET NEXT DH POINTER
8890 055166 104401                   TYPE     ;TYPE DH
8891 055170 000000 16$: .WORD      0       ;DH POINTER GOES HERE
8892 055172 105710                   TSTB    (R0)         ;TYPE A DT?
8893 055174 001003                   BNE     21$          ;YES-BRANCH
8894 055176 062700 000002           ADD     #2,R0         ;INCREMENT DF POINTER
8895 055202 000754                   BR      14$          ;SEE IF END OF DF BLOCK
8896 055204 104401 001205 21$: TYPE     , $CRLF      ;
8897 055210 005704                   TST     R4           ;INDENT?
8898 055212 001335                   BNE     10$          ;NO-BRANCH
8899 055214 104401 055256 17$: TYPE     ,SPACE2     ;YES-TYPE SPACES
8900 055220 000732                   BR      10$          ;LOOP
8901 055222 104414 20$: RESREG   ;
8902 055224 032777 010000 123706       BIT     #SW12,#SWR    ;SEE IF EXIT AFTER 20 ERRORS
8903 055232 001410                   BEQ     25$          ;BR IF NO
8904 055234 023727 001103 000024       CMP     $ERFLG,#20.  ;ELSE SEE IF HAVE 20 ERRORS
8905 055242 001004                   BNE     25$          ;BR IF NO
8906 055244 012706 001100                   MOV     #STACK,SP    ;ELSE RESTORE STACK
8907 055250 000137 023416                   JMP     ENDRV         ;AND BYPASS DRIVE
8908 055254 000207 25$: RTS      PC
    
```

8909	055256	020040	000
8910			
8911			
8912			
8913			
8914			
8915			
8916			
8917			
8918	055262		
8919	055342		
8920	000000		
8921	000001		
8922	000002		
8923	000003		
8924	000004		
8925	000005		
8926	000006		
8927	000007		
8928	177776		
8929			
8930	000014		
8931	000340		
8932	000020		
8933	000003		
8934	000006		
8935			
8936			
8937			
8938			
8939			
8940			
8941	177562		
8942	177560		
8943	177566		
8944	177564		
8945			
8946			
8947			
8948			
8949			
8950			
8951			
8952			
8953			
8954			
8955	055342	000413	
8956	055344	000417	
8957	055346	013737	177776 055322
8958	055354	013737	000016 177776
Z 8959	055362	010737	055320
8960	055366	000137	056520
8961			
8962	055372	012706	055302
8963	055376	010637	055316
8964	055402	000414	

```

SPACE2: .ASCIZ/ / ;2 SPACES
; ODT-11 -- VOOSA
; DEC-11-UODPA-A-LA
; COPYRIGHT 1969,1970,1972
; DIGITAL EQUIPMENT CORPORATION
; MAYNA MASSACHUSETTS 01754
      .ENABL ABS,AMA
      .EVEN
      .+.60
RO    =    %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
;
; RS 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.USP  ; FAKE THE SAVED STACK
      BR   O.RST1   ; CLEAR BREAKPOINT TABLES

```

8965	055404	004037	056726		0.RST: JSR	0,0.SVR	;SAVE REGISTERS
8966	055412	013777	055340	177716	MOV	0.UIN,20.ADR1	;REMOVE THE BREAKPOINT
8967	055416	113734	055324		MOVB	0.PRI,R4	;GET OOT PRIORITY
8968	055422	106004			RORB	R4	;SHIFT
8969	055424	106004			RORB	R4	;INTO
8970	055426	106004			RORB	R4	;POSITION
8971	055430	110437	177776		MOVB	R4,ST	;STORE IN STATUS
Z 8972	055434	000127			0.RST1: JMP	(PC)+	
8973	055436	000403			BR	0.45	
8974	055440	012737	000002	056430	MOV	#RTI,0.RTIT	;SET TO RTI IF 11/20 OR /05
8975	055446	105037	057347		0.45: CLRB	0.P	;DISALLOW PROCEED
8976	055452	012737	000340	000016	MOV	#0.STM,0.TVEC+2	;STATUS WORD TO TRT VECTOR + 2
8977	055460	012737	056510	000014	MOV	#0.BRK,0.TVEC	;PC TO TRT VECTOR
8978	055466	000447			BR	0.RALL	;CLEAR BREAKPOINT TABLES
8979							
8980							
8981							
8982							
8983	055470	004537	057150		0.REGT: JSR	5,0.GET	;SPECIAL NAME, GET ONE MORE CHARACTER
8984	055474	012704	057373		MOV	#0.TL,R4	;TABLE START ADDRESS
8985	055500	120024			0.RSP: CMPB	RD,(R4)+	;IS THIS THE CORRECT CHARACTER?
8986	055502	001413			BEQ	0.SP	;JUMP IF YES
8987	055504	022704	057401		CMP	#0.TL+0.LG,R4	;IS THE SEARCH DONE?
8988	055510	101373			BHI	0.RSP	;BRANCH IF NOT
8989	055512	042700	177770		BIC	#177770,RD	;MASK OFF OCTAL
8990	055516	010004			MOV	RD,R4	
8991	055520	006304			0.SP1: ASL	R4	
8992	055522	062704	055302		ADD	#0.URD,R4	;GENERATE ADDRESS
8993	055526	005202			INC	R2	;SET FOUND FLAG
8994	055530	000444			BR	0.SCAN	;GO FIND NEXT CHARACTER
8995	055532	162704	057364		0.SP: SUB	#0.TL-7,R4	;CORRECT CONSTANT
8996	055536	000770			BR	0.SP1	
8997							
8998							
8999							
9000	055540	004737	057274		0.ORPC: JSR	PC,0.TCLS	
9001	055544	010502			MOV	R5,R2	;CURRENT ADDRESS IN R2
9002	055546	061202			ADD	#R2,R2	;COMPUTE
9003	055550	006202			ASR	R2	;MOVE ONE BIT TO CARRY
9004	055552	103421			BCS	0.ERR	;ERROR IF ODD NUMBER
9005	055554	006302			ASL	R2	;RESTORE WORD
9006	055556	005722			TST	(R2)+	;AND INCREMENT BY TWO
9007	055560	010205			MOV	R2,R5	;UPDATE CAD
9008	055562	000137	056034		JMP	0.OP2	;GO FINISH UP
9009							
9010							
9011							
9012	055566	005702			0.BKPT: TST	R2	;IF NO NUMBER TYPED
9013	055570	001406			BEQ	0.RALL	;REMOVE BREAKPOINT
9014	055572	006204			ASR	R4	;CHECK IF ODD
9015	055574	103410			BCS	0.ERR	;JUMP IF ODD
9016	055576	006304			ASL	R4	;RESTORE ONE BIT
9017	055600	010437	055334		MOV	R4,0.ADR1	;SET A BREAKPOINT
9018							
9019	055604	000412			BR	0.DCD	
9020							

; SPECIAL NAME HANDLER
 ; DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URD

; * HANDLER - OPEN INDEXED ON THE PC

; B HANDLER - SET AND REMOVE BREAKPOINTS


```

9021
9022
9023 055606 012737 057410 055334 0.RALL: MOV #0,TRTC,0.ADR1 ;CLEAR BREAKPOINT
9024 055614 000406 BR 0.DCD
9025
9026 : COMMAND DECODER - ODT11
9027
9028 : REGISTERS R0-R4 MAY BE USED,
9029 : REGISTER R5 WILL BE CONSIDERED SAFE
9030
9031 055616 052705 000001 0.ERR: BIS #1,R5 ;CLOSE EVERYTHING
9032 055622 012700 000077 MOV #1,R0 ; ? TO BE TYPED
9033 055626 004537 057226 JSR 5,0.FTYP ; OUTPUT ?
9034 055632 004537 057326 0.DCD: JSR 5,0.CRLS ;TYPE <CR><LF>*
9035 055636 005004 0.DCD1: CLR R4 ; R4 CONTAINS THE CONVERTED OCTAL
9036 055640 005002 CLR R2 ; R2 IS THE NUMBER FOUND FLAG
9037 055642 004537 057150 0.SCAN: JSR 5,0.GET ;GET A CHAR, RETURN IN R0
9038 055646 022700 000060 CMP #0,R0 ;COMPARE WITH ASCII 0
9039 055652 101013 BHI 0.CLGL ;CHECK LEGALITY IF NON-NUMERIC
9040 055654 022700 000067 CMP #7,R0 ;COMPARE WITH ASCII 7
9041 055660 103410 BLO 0.CLGL ;CHECK LEGALITY IF NOT OCTAL
9042 055662 042700 177770 BIC #177770,R0 ;CONVERT TO BCD
9043 055666 006304 ASL R4 ;MAKE ROOM
9044 055670 006304 ASL R4 ; IN
9045 055672 006304 ASL R4 ; R4
9046 055674 060004 ADD R0,R4 ;PACK THREE BITS IN R4
9047 055676 007002 INC R2 ;R2 HAS NUMERIC FLAG
9048 055680 007000 BR 0.SCAN ;AND TRY AGAIN
9049 055702 005001 0.CLGL: CLR R1 ;CLEAR INDEX
9050 055704 120061 057357 0.LGL1: CMPB R0,0.LGCH(R1) ;DO THE CODES MATCH?
9051 055710 001405 BEQ 0.LGL2 ;JUMP IF YES
9052 055712 005201 INC R1 ;SET INDEX FOR NEXT SEARCH
9053 055714 020127 000014 CMP R1,#0.CLGT ;IS THE SEARCH DONE?
9054 055720 103336 BHS 0.ERR ;OOPS!
9055 055722 000770 BR 0.LGL1 ;RE-LOOP
9056 055724 006301 0.LGL2: ASL R1 ;MULTIPLY BY TWO
9057 055726 000171 055732 JMP #0.LGDR(R1) ;GO TO PROPER ROUTINE
9058
9059
9060 055732 055762 0.LGDR: 0.WRD ; / OPEN WORD
9061 055734 056014 0.CRET ; CARRIAGE RETURN CLOSE
9062 055736 055470 0.REGT ; $ REGISTER OPS
9063 055740 056324 0.GO ; G GO TO ADDRESS K
9064 055742 056026 0.OPI ; <LF> MODIFY, CLOSE, OPEN NEXT
9065 055744 055540 0.ORPC ; + OPEN RELATED, INDEX - PC
9066 055746 056060 0.BACK ; † OPEN PREVIOUS
9067 055750 056070 0.OFST ; 0 OFFSET
9068 055752 056146 0.WSCH ; W SEARCH WORD
9069 055754 056142 0.EFF ; E SEARCH EFFECTIVE ADDRESS
9070 055756 055566 0.BKPT ; B BREAKPOINTS
9071 055760 056432 0.PROC ; P PROCEED
9072 000030 0.LGL = -0.LGDR ;LGL MUST EQUAL 2X CHLGT ALWAYS
9073
9074 : PROCESS / - OPEN WORD
9075
9076 055762 005702 0.WRD: TST R2 ;GET VALUE IF R2 IS NON-ZERO

```

9077	055764	001410		BEQ	0.WRDA		;SKIP OTHERWISE
9078	055766	010405		MOV	R4,R5		; PUT VALUE IN CAD
9079	055770	006205		0.WRD1: ASR	R5		; MOVE ONE BIT TO CARRY
9080	055772	103711		0.ERR2: BCS	0.ERR		; JUMP IF ODD ADDRESS
9081	055774	006305		ASL	R5		; RESTORE THE CARRY BIT
9082	055776	011500		MOV	R5,RO		; GET CONTENTS OF WORD
9083	056000	004537	057064	JSR	5,0.CADV		; GO GET AND TYPE OUT CAD
9084	056004	000714		BR	0.DCD1		; GO BACK TO DECODER
9085	056006	042705	000001	0.WRDA: BIC	#1,R5		; CLEAR CLOSED BIT
9086	056012	000766		BR	0.WRD1		; GO BACK TO MAIN-LINE
9087							
9088							
9089							
9090	056014	004737	057274	0.CRET: JSR	PC,0.TCLS		; CLOSE LOCATION
9091	056020	052705	000001	BIS	#1,R5		; CLOSE EVERYTHING
9092	056024	000702		BR	0.DCD		; RETURN TO DECODER
9093							
9094							
9095							
9096	056026	004737	057274	0.OP1: JSR	PC,0.TCLS		; CLOSE PRESENT CELL
9097	056032	005725		TST	(R5)+		; GENERATE NEW ADDRESS
9098	056034	004537	057320	0.OP2: JSR	5,0.CRLF		; <CR><LF>
9099	056040	010500		MOV	R5,RO		; NUMBER TO TYPE
9100	056042	004537	057064	JSR	5,0.CADV		; TYPE OUT ADDRESS
9101	056046	012700	000057	MOV	#1,RO		; TYPE A /
9102	056052	004537	057226	JSR	5,0.FTYP		
9103	056056	000744		BR	0.WRD1		; GO PROCESS IT
9104							
9105							
9106							
9107	056060	004737	057274	0.BACK: JSR	PC,0.TCLS		; GENERATE NEW ADDRESS
9108	056064	005745		TST	-(R5)		; GO DO THE REST
9109	056066	000762		BR	0.OP2		
9110							
9111							
9112							
9113	056070	006205		0.OFST: ASR	R5		; GET LOW ORDER BIT
9114	056072	103737		BCS	0.ERR2		; ERROR IF CLOSED
9115	056074	006305		ASL	R5		; RESTORE WORD
9116	056076	012700	000040	MOV	#',RO		; TYPE ONE BLANK
9117	056102	004537	057226	JSR	5,0.FTYP		; AS A SEPARATOR
9118	056106	160504		SUB	R5,R4		; COMPUTE
9119	056110	005304		DEC	R4		
9120	056112	005304		DEC	R4		; 16 BIT OFFSET
9121	056114	010400		MOV	R4,RO		; TYPE A
9122	056116	010402		MOV	R4,R2		; SAVE R4
9123	056120	004537	057064	JSR	5,0.CADV		; NUMBER IN RO - WORD MODE
9124	056124	010200		MOV	R2,RO		
9125	056126	006200		ASR	RO		; DIVIDE BY TWO
9126	056130	103402		BCS	0.OF1		; BRANCH IF ODD
9127	056132	004537	057064	JSR	5,0.CADV		; NUMBER IN RO - BYTE MODE
9128	056136	000137	055636	0.OF1: JMP	0.DCD1		; ALL DONE
9129							
9130							
9131							
9132							

SEARCHES - SMSK HAS THE MASK
SMSK+2 HAS THE FWA
SMSK+4 HAS THE LWA

9133					
9134					
9135					
9136					
9137					
9138	056142	005201			
9139	056144	000401			
9140	056146	005001			
9141	056150	005702			
9142	056152	001621			
9143	056154	013702	055330		
9144	056160	013705	055326		
9145	056164	005105			
9146	056166	020237	055332		
9147	056172	101217			
9148	056174	011200			
9149	056176	005701			
9150	056200	001027			
9151	056202	010046			
9152	056204	010403			
9153	056206	040400			
9154	056210	042603			
9155	056212	050003			
9156	056214	040503			
9157	056216	001016			
9158	056220	010446			
9159	056222	004537	057320		
9160	056226	010200			
9161	056230	004537	057064		
9162	056234	012700	000057		
9163	056240	004537	057226		
9164	056244	011200			
9165	056246	004537	057064		
9166	056252	012604			
9167	056254	005722			
9168	056256	000743			
9169	056260	020004			
9170	056262	001755			
9171	056264	010003			
9172	056266	060203			
9173	056270	005203			
9174	056272	005203			
9175	056274	020304			
9176	056276	001747			
9177	056300	042700	177400		
9178	056304	110000			
9179	056306	000257			
9180	056310	006300			
9181	056312	005200			
9182	056314	005200			
9183	056316	060200			
9184	056320	020004			
9185	056322	000735			
9186					
9187					
9188					

0.EFF:	INC	R1		;SET EFFECTIVE SEARCH
	BR	0.WDS		
0.WSCH:	CLR	R1		;SET WORD SEARCH
0.WDS:	TST	R2		;CHECK FOR OBJECT FOUND
0.ERR1:	BEQ	0.ERR		;ERROR IF NO OBJECT
	MOV	0.MSK+2,R2		;SET ORIGIN
	MOV	0.MSK,R5		;SET MASK
	COM	R5		;AND COMPLEMENT IT
0.WDS2:	CMP	R2,0.MSK+4		; IS THE SEARCH ALL DONE?
	BHI	0.DCD		; YES
	MOV	2R2,R0		; GET OBJECT
	TST	R1		; NO
	BNE	0.EFF1		;BRANCH IF EFFECTIVE SEARCH
	MOV	R0,-(SP)		
	MOV	R4,R3		;EXCLUSIVE OR
	BIC	R4,R0		; IS DONE
	BIC	(SP)+,R3		; IN A VERY
	BIS	R0,R3		; FANCY MANNER HERE
	BIC	R5,R3		;AND RESULT WITH MASK
0.WDS3:	BNE	0.WDS4		;RE-LOOP IF NO MATCH
	MOV	R4,-(SP)		;REGISTERS R2,R4, AND R5 ARE SAFE
	JSR	5,0.CRLF		;TYPE (CR,LF)
	MOV	R2,R0		;GET READY TO TYPE
	JSR	5,0.CADV		;TYPE ADDRESS
	MOV	#1,R0		;SLASH TO R0
	JSR	5,0.FTYP		;TYPE IT
	MOV	2R2,R0		;GET CONTENTS
	JSR	5,0.CADV		;TYPE CONTENTS
	MOV	(SP)+,R4		;RESTORE R4
0.WDS4:	TST	(R2)+		;INCREMENT TO NEXT CELL AND
	BR	0.WDS2		; RETURN
0.EFF1:	CMP	R0,R4		; IS (X)=K?
	BEQ	0.WDS3		;TYPE IF EQUAL
	MOV	R0,R3		; (X) TO R3
	ADD	R2,R3		; (X)+X
	INC	R3		
	INC	R3		; (X)+X+2
	CMP	R3,R4		; IS (X)+X+2=K?
	BEQ	0.WDS3		;BRANCH IF EQUAL
	BIC	#177400,R0		;WIPE OUT EXTRA DJS BITS
	MOVB	R0,R0		;EXTEND SIGN
	CC			
	ASL	R0		;MULTIPLY BY TWO
	INC	R0		;ADD TWO
	INC	R0		
	ADD	R2,R0		;ADD PC
	CMP	R0,R4		; IS THE RESULT A PROPER REL. BRANCH?
	BR	0.WDS3		

;	PROCESS G - GO
---	----------------

```

9189 056324 105037 057347 0.G0: CLR B 0.P ;DISALLOW PROCEED
9190 056330 006204 ASR R4 ;CHECK LOW ORDER BIT
9191 056332 103617 BCS 0.ERR2 ;ERROR IF ODD NUMBER
9192 056334 006304 ASL R4 ;RESTORE WORD
9193 056336 010437 055320 MOV R4,0.UPC ;SET UP NEW PC
9194 056342 112737 000340 177776 MOVB #0.STM,ST ;SET HIGH PRIORITY
9195 056350 004537 057016 JSR 5,0.RSTT ;RESTORE TELETYPE
9196 056354 105037 057346 0.TBIT: CLR B 0.T ;CLEAR BOTH
9197 056360 042737 000020 055322 BIC #0.TBT,0.UST ;T-BIT FLAGS
9198 056366 017737 176742 055340 MOV #0.ADR1,0.UIN ;SAVE INSTRUCTION
9199 056374 013777 057410 176732 MOV 0.TRTC,0.ADR1 ;REPLACE WITH TRAP
9200 056402 012600 0.G02: MOV (SP)+,R0 ;RESTORE
9201 056404 012601 MOV (SP)+,R1 ;R0
9202 056406 012602 MOV (SP)+,R2 ;THRU
9203 056410 012603 MOV (SP)+,R3
9204 056412 012604 MOV (SP)+,R4
9205 056414 012605 MOV (SP)+,R5
9206 056416 012606 MOV (SP)+,SP ;R5
9207 056420 013746 055322 MOV 0.UST,-(SP) ;AND SP
9208 056424 013746 055320 MOV 0.UPC,-(SP) ;AND STATUS
9209 056430 000006 0.RTIT: RTT ;AND PC
9210 ;CHANGED TO RTI FOR 11/20 AND /05
9211 ;
9212 ; PROCESS P - PROCEED
9213 ; ONLY ALLOWED AFTER A BREAKPOINT
9214 056432 105737 057347 0.PROC: TST B 0.P ;CHECK LEGALITY OF PROCEED
9215 056436 001645 BEQ 0.ERR1 ;NOT LEGAL
9216 056440 105037 057347 CLR B 0.P ;CLEAR PROCEED FLAG
9217 056444 005702 TST R2 ;WAS COUNT SPECIFIED?
9218 056446 001402 BEQ 0.PRI ;NO
9219 056450 010437 055336 MOV R4,0.CT ;YES, PUT AWAY COUNT
9220 056454 112737 000340 177776 0.PRI: MOVB #0.STM,ST ;FORCE HIGH PRIORITY
9221 056462 004537 057016 JSR 5,0.RSTT ;RESTORE TTY
9222 056466 112737 000340 177776 0.C1: MOVB #0.STM,ST ;SET HIGH PRIORITY
9223 056474 105237 057346 INCB 0.T ;SET T-BIT FLAG
9224 056500 052737 000020 055322 BIS #0.TBT,0.UST ;SET T-BIT
9225 056506 000735 BR 0.G02
9226 ;
9227 ; BREAKPOINT HANDLER
9228 ; A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
9229 ; VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
9230 ; AND GIVES CONTROL TO THE COMMAND DECODER
9231 ;
9232 056510 012637 055320 0.BRK: MOV (SP)+,0.UPC ;PRIORITY IS 7 UPON ENTRY
9233 056514 012637 055322 MOV (SP)+,0.UST ;SAVE STATUS AND PC
9234 056520 004037 056726 0.BK1: JSR 0,0.SVR ;SAVE VARIOUS REGISTERS
9235 056524 105737 057346 TST B 0.T ;CHECK FOR T-BIT SET
9236 056530 001311 BNE 0.TBIT ;JUMP IF SET
9237 056532 013777 055340 176574 MOV 0.UIN,0.ADR1 ;REMOVE BREAKPOINTS
9238 056540 105737 055324 TST B 0.PRI ;CHECK IF PRIORITY
9239 056544 100003 BPL 0.BK2 ;IS AS SAME AS USER PGM
9240 056546 113705 055322 MOVB 0.UST,R5 ;PICK UP USER UST IF SO
9241 056552 000407 BR 0.BK3 ;AND DON'T COMPUTE THE PRIORITY
9242 056554 113705 055324 0.BK2: MOVB 0.PRI,R5 ;OTHERWISE PICK UP ACTUAL PRIORITY
9243 056560 000257 CCC ;CLEAR CARRY
9244 056562 106005 RORB R5 ;SHIFT LOW ORDER BITS

```

```

9245 056564 106005          RORB    R5          ; INTO
9246 056566 106005          RORB    R5          ;   HIGH ORDER
9247 056570 106005          RORB    R5          ;   POSITION
9248 056572 110537 177776    0.BK3: MOVB   R5,ST      ; PUT THE STATUS AWAY WHERE IT BELONGS
9249 056576 013705 055320    MOV    0,UPC,R5     ; GET PC, IT POINTS TO THE TRT
9250 056602 005745          TST    -(R5)        ; SUBTRACT TWO
9251 056604 010537 055320    MOV    R5,0,UPC     ; FROM THE USER'S PC
9252 056610 020537 055334    CMP    R5,0,ADR1    ; COMPARE WITH LIST
9253 056614 001417          BEQ    0,B2         ; JUMP IF FOUND
9254 056616 004537 056764    JSR    5,0,SVTT     ; SAVE TELETYPE STATUS
9255 056622 004537 057320    JSR    5,0,CRLF     ;
9256 056626 012704 057352    MOV    #0,BD,R4     ; ERROR, NOTHING FOUND
9257 056632 012703 057353    MOV    #0,BD+1,R3   ;
9258 056636 004537 057212    JSR    5,0,TYPE     ; OUTPUT "BE" FOR BAD ENTRY
9259 056642 010500          MOV    R5,R0        ;
9260 056644 042737 000020 055322  BIC    #0,TBT,0,UST ; CLEAR OUT ANY POSSIBLE FAKE T-BIT
9261 056652 000420          BR     0,B3         ; AND CONTINUE
9262 056654 005337 055336    0.B2: DEC    0,CT    ;
9263 056660 003302          BGT    0,C1         ; JUMP IF REPEAT
9264 056662 012737 000001 055336  MOV    #1,0,CT      ; RESET COUNT TO 1
9265 056670 105237 057347    INCB   0,P          ; ALLOW PROCEED
9266 056674 004537 056764    JSR    5,0,SVTT     ; SAVE TELETYPE STATUS, R4 IS SAFE
9267 056700 012700 000102    MOV    #8,R0        ;
9268 056704 004537 057226    JSR    5,0,FTYP     ; TYPE "B"
9269 056710 013700 055334    MOV    0,ADR1,R0   ; GET ADDRESS OF BREAK
9270 056714 004537 057064    0.B3: JSR    5,0,CADV ; TYPE ADDRESS
9271 056720 005005          CLR    R5           ; CLEAR CAD
9272 056722 000137 055632    JMP    0,DCD        ; GO TO DECODER
9273
9274          ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
9275
9276 056726 012637 057344    0.SVR: MOV    (SP)+,0,XXX ; PICK REGISTER FROM STACK AND SAVE
9277 056732 010637 055316    MOV    SP,0,USP     ; SAVE USER STACK ADDRESS
9278 056736 012706 055316    MOV    #0,USP,SP    ; SET TO INTERNAL STACK
9279 056742 010546          MOV    R5,-(SP)     ; SAVE
9280 056744 010446          MOV    R4,-(SP)     ; REGISTERS
9281 056746 010346          MOV    R3,-(SP)     ; 1
9282 056750 010246          MOV    R2,-(SP)     ; THRU
9283 056752 010146          MOV    R1,-(SP)     ; 5
9284 056754 013746 057344    MOV    0,XXX,-(SP) ; PUT SAVED REGISTER ON STACK
9285 056760 005746          TST    -(SP)        ;
9286 056762 000200          RTS    R0           ;
9287
9288          ; SAVE TELETYPE STATUS
9289
9290 056764 113737 177560 057350  0.SVTT: MOVB   0,RCSR,0,CSR1 ; SAVE R C/SR
9291 056772 113737 177564 057351  MOVB   0,TCSR,0,CSR2 ; SAVE T C/SR
9292 057000 105037 177560    CLRB   0,RCSR       ; CLEAR ENABLE AND MAINTENANCE
9293 057004 105037 177564    CLRB   0,TCSR       ; BITS IN BOTH C/SR
9294 057010 004537 057320    JSR    5,0,CRLF     ; TYPE <CR,LF>
9295 057014 000205          RTS    R5           ;
9296
9297          ; RESTORE TELETYPE STATUS
9298
9299 057016 004537 057320    0.RSTT: JSR    5,0,CRLF ; <CR,LF> BEFORE RESTORING
9300 057022 105737 177564    TSTB   0,TCSR       ; WAIT READY ON PRINTER

```

```

9301 057026 100375          BPL      -4
9302 057030 032737 004000 177560 BIT      #4000,0.RCSR ;CHECK BUSY FLAG ON READER
9303 057036 001403          BEQ      0.RSE1 ;SKIP READY LOOP IF NOT BUSY
9304 057040 105737 177560 TSTB    0.RCSR ;WAIT READY
9305 057044 100375          BPL      -4 ;ON READER
9306 057046 113737 057350 177560 0.RSE1: MOVB  0.CSR1,0.RCSR ;RESTORE
9307 057054 113737 057351 177564 MOVB    0.CSR2,0.TCSR ;THE STATUS REGISTERS
9308 057062 000205          RTS      R5
9309
9310 ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
9311 ; WORD IS IN RO
9312
9313 057064 010246          0.CADV: MOV   R2,-(SP) ;SAVE R2
9314 057066 012704 057407 MOV   #0.BUF+6,R4 ;BUFFER START ADDRESS
9315 057072 012746 000060 MOV   #'0,-(SP) ;CONSTANT ASCII 0
9316 057076 010002          0.SPC: MOV   R0,R2 ;GET
9317 057100 042702 177770 BIC   #177770,R2 ;OCTAL CHARACTER
9318 057104 061602 ADD    #SP,R2 ;CONVERT TO ASCII
9319 057106 110244 MOVB   R2,-(R4) ;STORE IN BUFFER
9320 057110 006200 ASR    R0 ;SHIFT THIS MESS
9321 057112 006200 ASR    R0 ;RIGHT
9322 057114 006200 ASR    R0 ;THREE WHOLE PLACES
9323 057116 020427 057402 CMP    R4,#0.BUF+1 ;DONE?
9324 057122 101365 BHI    0.SPC ;NO
9325 057124 042700 177776 BIC   #177776,R0 ;GET LAST BIT
9326 057130 062600 ADD    (SP)+,R0 ;CONVERT TO ASCII
9327 057132 110044 MOVB   R0,-(R4) ;AND PUT IT AWAY
9328 057134 012703 057407 MOV   #0.BUF+6,R3 ;LWA
9329 057140 004537 057212 JSR    5,0.TYP ;TYPE WHOLE STRING OF CHARACTERS
9330 057144 012602 MOV    (SP)+,R2 ;RESTORE R2
9331 057146 000205 RTS      R5
9332
9333 ; GENERAL CHARACTER INPUT ROUTINE
9334 ; CHARACTER INPUT GOES TO RO
9335
9336 057150 105737 177560 0.GET: TSTB  0.RCSR ;WAIT FOR
9337 057154 100375          BPL      -4 ;INPUT FROM KEYBOARD
9338 057156 113700 177562 MOVB   0.RDB,R0 ;GET A CHARACTER
9339 057162 004537 057226 JSR    5,0.F1YP ;ECHO CHARACTER
9340 057166 042700 177600 BIC   #177600,R0 ;STRIP OFF PARITY FROM CHARACTER
9341 057172 001766 BEQ    0.GET ;IGNORE NULLS
9342 057174 122700 000040 CMPB   #40,R0 ;CHECK FOR SPACES
9343 057200 001763 BEQ    0.GET ;IGNORE NULLS
9344 057202 122700 000073 CMPB   #' ;,R0 ;CHECK FOR SEMI-COLON
9345 057206 001760 BEQ    0.GET ;IGNORE THEM IF FOUND
9346 057210 000205 RTS      R5
9347
9348 ; GENERAL CHARACTER OUTPUT ROUTINE
9349 ; ADDRESS OF FIRST BYTE IN R4,
9350 ; ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
9351
9352 057212 020304 057226 0.TYPE: CMP    R3,R4 ;CHECK FOR COMPLETION
9353 057214 103426 BLO    0.TYP1 ;EXIT WHEN DONE
9354 057216 112400 MOVB   (R4)+,R0 ;GET A CHARACTER
9355 057220 004537 057226 JSR    5,0.F1YP ;TYPE ONE CHARACTER
9356 057224 000772 BR     0.TYPE ;LOOP UNTIL DONE

```

9357
9358
9359
9360 057226 105737 177564
9361 057232 100375
9362 057234 110037 177566
9363 057240 120037 000045
9364 057244 001012
9365 057246 113746 000044
9366 057252 105737 177564
9367 057256 100375
9368 057260 105037 177566
9369 057264 105316
9370 057266 003371
9371 057270 005726
9372 057272 000205
9373
9374
9375
9376
9377 057274 006205
9378 057276 103405
9379 057300 006305
9380 057302 005702
9381 057304 001401
9382 057306 010415
9383 057310 000207
9384 057312 005746
9385 057314 000137 055616
9386
9387
9388
9389
9390 057320 012703 057355
9391 057324 000402
9392 057326 012703 057356
9393 057332 012704 057354
9394 057336 004537 057212
9395 057342 000205
9396
9397 057344 000000
9398 057346 000
9399 057347 000
9400
9401 057350 000
9402 057351 000
9403
9404
9405 057352 042502
9406
9407 057354 015
9408 057355 012
9409 057356 052
9410
9411 057357 057
9412 057360 015

```

; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
O.FTYP: TSTB 0.TCSR ;CHECK STATUS
        BPL -4 ;WAIT UNTIL READY
        MOVB R0,0.TDB ;TYPE ONE CHARACTER
        CMPB R0,#45 ;IS CHAR TO BE FILLED?
        BNE 0.TYP1 ;NO
        MOVB #44, -(SP) ;YES, INIT THE COUNT
O.TYP2: TSTB 0.TCSR
        BPL 0.TYP2
        CLRB 0.TDB ;GENERATE NULL FILLER
        DECB #SP
        BGT 0.TYP2
        TST (SP)+ ;POP STACK
O.TYP1: RTS R5

; CLOSE WORD OR BYTE AND EXIT
; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
O.TCLS: ASR R5 ;GET LOW ORDER BIT
        BCS 0.TC ;JUMP IF ALREADY CLOSED
        ASL R5
        TST R2 ;IF NO NUMBER WAS TYPED THERE IS
        BEQ 0.CLS1 ;NO CHANGE TO THE OPEN CELL
        MOV R4,#R5 ;STORE WORD
O.CLS1: RTS PC
O.TC: TST -(SP) ;POP EXTRA CELL FROM STACK
        JMP 0.ERR ;AND SCREAM BLOODY MURDER

; O.CRLF - TYPE <CR,LF>
; O.CRLS - TYPE <CR,LF>*
O.CRLF: MOV #0.CR+1,R3 ;LWA <CR,LF>
        BR 0.CRS
O.CRLS: MOV #0.CR+2,R3 ;LWA <CR,LF>*
O.CRS: MOV #0.CR,R4 ;FWA
        JSR 5.0.TYPE ;TYPE SOMETHING
        RTS R5

; O.XXX: .WORD 0 ;TEMPORARY STORAGE
O.T: .BYTE 0 ; T-BIT FLAG
O.P: .BYTE 0 ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
; = 1 IF PROCEED ALLOWED
O.CSR1: .BYTE 0 ;SAVE CELL - R C/SR
O.CSR2: .BYTE 0 ;SAVE CELL - T C/SR
;
; O.BD: .EVEN
; .WORD "BE
;
; O.CR: .BYTE 015 ; <CR>
; .BYTE 012 ; <LF>
; .BYTE '*' ; *
;
; O.LGCH: .BYTE '/' ; /
; .BYTE 015 ; CARRIAGE RETURN

```

```

9413 057361 044
9414 057362 107
9415 057363 012
9416 057364 137
9417 057365 136
9418 057366 117
9419 057367 127
9420 057370 105
9421 057371 102
9422 057372 120
9423 000014
9424
9425 057373 123
9426 057374 120
9427 057375 115
9428 057376 000
9429 057377 000
9430 057400 102
9431 000006
9432
9433 057401
9434 057407
9435 057407 040
9436
9437
9438 057410 000003
9439
9440
9441
9442 055302
9443 055302 000000
9444 055304 000000
9445 055306 000000
9446 055310 000000
9447 055312 000000
9448 055314 000000
9449 055316 000000
9450 055320 000000
9451 055322 000000
9452 055324 000007
9453 055326 000000
9454 055330 000000
9455 055332 000000
9456
9457
9458
9459
9460 055334 000000
9461 055336 000000
9462 055340 000000
9463 000001

```

```

.BYTE '$
.BYTE 'G
.BYTE 012 <LF>
.BYTE '+
.BYTE 'O
.BYTE 'K
.BYTE 'B
.BYTE 'B
.O.CLGT = -.O.LGCH ;TABLE LENGTH
.O.TL: .BYTE 'S ;DO
.BYTE 'P ;NOT
.BYTE 'M ;CHANGE
.BYTE 0 ;THE
.BYTE 0 ;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.UR0: = 0.00T-40
;USER R0
; R1
; R2
; R3
; R4
; R5
.O.USP: 0 ;USER SP
.O.UPC: 0 ;USER PC
.O.UST: 0 ;USER ST
.O.PRI: 7 ;ODT PRIORITY
.O.MSK: 0 ;MASK
;LOW LIMIT
;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	1822	1863	1887*	
ACDW1 = 000000	1822	1865		
ACDW2 = 000000	1822	1866		
ACLO = 000010	1171*			
ACPUOP = 000000	1822	1837		
ACT11 = 005522	2071*	3017*		
ADDW0 = 000000	1822	1867		
ADDW1 = 000000	1822	1868		
ADDW10 = 000000	1822	1877		
ADDW11 = 000000	1822	1878		
ADDW12 = 000000	1822	1879		
ADDW13 = 000000	1822	1880		
ADDW14 = 000000	1822	1881		
ADDW15 = 000000	1822	1882		
ADDW2 = 000000	1822	1869		
ADDW3 = 000000	1822	1870		
ADDW4 = 000000	1822	1871		
ADDW5 = 000000	1822	1872		
ADDW6 = 000000	1822	1873		
ADDW7 = 000000	1822	1874		
ADDW8 = 000000	1822	1875		
ADDW9 = 000000	1822	1876		
ADEVCT = 000000	1822	1828		
ADEVN = 000000	1822	1864		
RENV = 000000	1822	1833		
RENVN = 000000	1822	1834		
AFATAL = 000000	1822	1825		
AMADR1 = 000000	1822	1850		
AMADR2 = 000000	1822	1854		
AMADR3 = 000000	1822	1857		
AMADR4 = 000000	1822	1860		
AMAMS1 = 000000	1822	1844		
AMAMS2 = 000000	1822	1852		
AMAMS3 = 000000	1822	1855		
AMAMS4 = 000000	1822	1858		
AMSGAD = 000000	1822	1830		
AMSGLG = 000000	1822	1831		
AMSGTY = 000000	1822	1824		
AMTYP1 = 000000	1822	1845		
AMTYP2 = 000000	1822	1853		
AMTYP3 = 000000	1822	1856		
AMTYP4 = 000000	1822	1859		
APASS = 000000	1822	1827		
APRIOR = 000000	1822			
APTC SU = 000040	6445	6617*		
APTENV = 000001	6392	6438	6573	6615*
APTSIZ = 000200	2945	6614*		
APTSP0 = 000100	6440	6575	6616*	
ASWREG = 000000	1822	1835		
ATESTN = 000000	1822	1826		
ATTN = 005376	2006*	5307	5327	5354
AUNIT = 000000	1822	1829		
AUSWR = 000000	1822	1830		
AVECT1 = 000000	1822	1861		
AVECT2 = 000000	1822	1862		
BADHDR = 005372	1996*	3025*	6099	

E.A1	005502	2052*	3546*	3571*	3625*	4043*	4120*	4300*	4343*	4393*	4429*	4586*	4645*	4751*
E.A2	005506	4827*	4907*	4966*	5396*	5406	5408*	5453	8660	8671	8683	5397*	5410	5412*
E.A3	005512	2054*	4122*	4302*	4345*	4431*	4588*	4647*	4753*	4909*	4968*			
E.B0	005500	8671	8683											
E.B1	005504	2056*	5398*											
E.B2	005510	2051*	3545*	3570*	3624*	4042*	4119*	4299*	4342*	4392*	4428*	4585*	4644*	4750*
E.B3	005514	4826*	4906*	4965*	5414	5416*	5443	8660	8671	8683				
FATT1	024736	2053*	3547*	3572*	3626*	4044*	4121*	4301*	4344*	4394*	4430*	4587*	4646*	4752*
FATT2	025032	4828*	4908*	4967*	5418	5420*	5463	8660	8671	8683				
FCYL	027102	2055*	4123*	4303*	4346*	4432*	4589*	4648*	4754*	4910*	4969*	5422	5424*	8671*
FHDHM	027172	8683												
FHDTAB	027314	2057*	4124*	4304*	4347*	4433*	4590*	4649*	4755*	4911*	4970*	5426	5428*	8683
FLGTST	027544	3988	5324*	6093	6117									
FLOAD	027246	4028	4188	5352*										
FMT1	001520	5709*												
FORM	030740	5734*	5743											
FORMAT	001516	4240	4709	5768*										
FRCYL	001364	5810	5814	5829*										
FRDY	024422	5752*												
FRDY1	024470	1153*												
FSEC17	027012	1968*	5778*	5779*	5780*	5785								
FTITLE	001360	6132	6145*											
GBA	024324	1967*	4238*	4707*	5778	5807								
GDRVS	024164	1910*												
GETSP	030346	3170	3186	3191	3286	3291	3469	3494	3639	3653	3751	3767	3782	3955
GINT	024352	3984	4024	4063	4111	4184	4251	4276	4328	4380	4414	4477	4571	4630
GNS	***** U	4719	4741	4803	4856	4892	4951	5252*	5255	5527	5588	6089	6108	6124
GO	000001	5266*	5269	5542	5552	5562	5571							
GSTAT	025760	5685*												
GSTAT1	026022	1904*	5144	5146*										
GTSWR	104406	2999	5211*											
HASOF	005424	2997	5166*											
HBA	005414	3532	3537	3562	3587	3602	3613	3700	3717	3808	3834	3844	3855	3864
HCS1	005406	3879	3932	4005	4227	4365	4451	4487	4506	4696	4782	4814	4842	5047
HCS2	005410	5054	6051*	6057										
HDA	005416	3001	5224*											
HDB	005430	1274	7269	7270	7271	7272	7273	7275	7277	7278	7279	7280	7281	7282
HDC	005426	7283												
HDS	005420	1117*												
		3422	3539	3564	3590	3604	3618	3684	3694	3703	3709	3799	3836	3856
		3874	3911	3922	4007	4113	4222	4229	4253	4278	4330	4368	4383	4416
		4452	4573	4632	4690	4698	4721	4743	4806	4894	4953	5334	5339	5359
		5363	5523*	5592	5603	5615	5631	5736	5754	6112				
		5393	5535*											
		5155	7275*											
		2022*	5291*	5307	8640	8648	8657	8667	8679	8692				
		2018*	5287*	8640	8648	8657	8667	8679	8692					
		2015*	3193	3293	3912	4114	4191	4254	4279	4331	4384	4417	4480	4574
		4633	4722	4744	4807	4895	4954	5284*	5497	5501	5593	5904	8637	8645
		8654	8664	8676	8689									
		2016*	3226	3228	3232	3244	3319	3321	3325	3328	4333	4419	4576	4635
		4897	4956	5285*	5506	5510	8637	8645	8654	8664	8676	8689	8679	8692
		2019*	4508	4509	4837	4838	5288*	5941	8640	8648	8657	8667	8679	8692
		2024*												
		2023*	5292*	5933	8640	8648	8657	8667	8679	8692				
		2020*	3230	3323	5289*	8637	8645	8654	8664	8676	8689			

F15

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR&JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 187
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0186

0.BRK	056510	8977	9232#							
0.BUF	057401	9314	9323	9328	9433#					
0.B2	056654	9253	9262#							
0.B3	056714	9261	9270#							
0.CADV	057064	9083	9100	9123	9127	9161	9165	9270	9313#	
0.CLGL	055702	9039	9041	9049#						
0.CLGT=	000014	9053	9423#							
0.CLS1	057310	9381	9383#							
0.CR	057354	9390	9392	9393	9407#					
0.CRET	056014	9061	9090#							
0.CRF	057320	9098	9159	9255	9294	9299	9390#			
0.CRLS	057326	9034	9392#							
0.CRS	057332	9391	9393#							
0.CSR1	057350	9290*	9306	9401#						
0.CSR2	057351	9291*	9307	9402#						
0.CT	055336	9219*	9262*	9264*	9461#					
0.C1	056466	9222#	9263							
0.DCD	055632	9019	9024	9034#	9092	9147	9272			
0.DCD1	055636	9035#	9084	9128						
0.EFF	056142	9069	9138#							
0.EFF1	056260	9150	9169#							
0.ENTR	055346	8957#								
0.ERR	055616	9004	9015	9031#	9054	9070	9142	9385		
0.ERR1	056152	9142#	9215							
0.ERR2	055772	9080#	9114	9191						
0.FTYP	057226	9033	9102	9117	9163	9268	9339	9355	9360#	
0.GET	057150	8983	9037	9336#	9341	9343	9345			
0.GO	056324	9063	9189#							
0.G02	056402	9200#	9225							
0.LG =	000006	8987	9431#							
0.LGCH	057357	9050	9411#	9423						
0.LGDR	055732	9057	9060#	9072						
0.LGL =	000030	9072#								
0.LGL1	055704	9050#	9055							
0.LGL2	055724	9051	9056#							
0.MSK	055326	9143	9144	9146	9453#					
0.OOT	055342	1283	8955#	9442						
0.OFST	056070	9067	9113#							
0.OF1	056136	9126	9128#							
0.OP1	056026	9064	9096#							
0.OP2	056034	9008	9098#	9109						
0.ORPC	055540	9000#	9065							
0.P	057347	8975*	9189*	9214	9216*	9265*	9399#			
0.PRI	055324	8967	9238	9242	9452#					
0.PROC	056432	9071	9214#							
0.PRI	056454	9218	9220#							
0.RALL	055606	8978	9013	9023#						
0.RCSR=	177560	8942#	9290	9292*	9302	9304	9306*	9336		
0.R08 =	177562	8941#	9338							
0.REGT	055470	8983#	9062							
0.RSE1	057046	9303	9306#							
0.RSP	055500	8985#	8988							
0.RST	055404	8956	8965#							
0.RSTT	057016	9195	9221	9299#						
0.RST1	055434	8964	8972#							
0.RTIT	056430	8974*	9209#							

0. SCAN	055642	8994	9037#	9048					
0. SP	055532	8986	8995#						
0. SPC	057076	9316#	9324						
0. SP1	055520	8991#	8996						
0. STM =	000340	8931#	8976	9194	9220	9222			
0. STRT	055372	8955	8962#						
0. SVR	056726	8965	9234	9276#					
0. SVTT	056764	9254	9266	9290#					
0. T	057346	9196*	9223*	9235	9398#				
0. TBIT	056354	9196#	9236						
0. TBT =	000020	8932#	9197	9224	9260				
0. TC	057312	9378	9384#						
0. TCLS	057274	9000	9090	9096	9107	9377#			
0. TCSR=	177564	8944#	9291	9293*	9300	9307*	9360	9366	
0. TDB =	177566	8943#	9362*	9368#					
0. TL	057373	8984	8987	8995	9425#	9431			
0. TRTC	057410	9023	9199	9438#					
0. TVEC=	000014	8930#	8958	8976*	8977*				
0. TYPE	057212	9258	9329	9352#	9356	9394			
0. TYP1	057272	9353	9364	9372#					
0. TYP2	057252	9366#	9367	9370					
0. UIN	055340	8966	9198*	9237	9462#				
0. UPC	055320	8959*	9193*	9208	9232*	9249	9251*	9450#	
0. URO	055302	8962	8992	9443#					
0. USP	055316	8963*	9277*	9278	9449#				
0. UST	055322	8957*	9197*	9207	9224*	9233*	9240	9260*	9451#
0. WDS	056150	9139	9141#						
0. WDS2	056166	9146#	9168						
0. WDS3	056216	9157#	9170	9176	9185				
0. WDS4	056254	9157	9167#						
0. WRD	055762	9060	9076#						
0. WRDA	056006	9077	9085#						
0. WRD1	055770	9079#	9086	9103					
0. WSCH	056146	9068	9140#						
0. XXX	057344	9276*	9284	9397#					
0. 45	055446	8973	8975#						
PACK =	000003	1104#	3467	3637	3765	3953	4061	4854	
PARAM	001356	1903#	2880*	2883*	2993				
PARSRT	010050	1280	2880#						
PAT =	000020	1185#	3683						
PCA =	004000	1192#							
PCD =	010000	1193#							
PCLKF	005552	2088#	3037*	3044*	5955	5976			
PCVEC	001352	1896#	3038	3045					
PCYL	001372	1913#							
PFSRT	012440	3403#	6234						
PGE =	002000	1140#							
PIP =	020000	1178#							
PIRQ =	177772	983#							
PIRQVE=	000240	1077#							
PKRB	001344	1892#							
PKS	001340	1890#	3036	3043	5960*	5980*			
PKSB	001342	1891#	5959*						
PPTP	005524	2072#	2984*						
PRGSRT	010064	2881	2884#	5198					
PRO =	000000	1000#	2886	3075					

N15

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(1006) 31-JAN-77 18:30 PAGE 195
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0194

\$MXCNT 031646	6350	6360#													
\$NULL 001154	1802#	6465	6494												
\$NWTST= 000001	3061#	3063	3106#	3108	3260#	3262	3362#	3364	3404#	3406	3445#	3447	3476#		
	3478	3502#	3504	3659#	3661	3812#	3814	3883#	3885	3964#	3966	4072#	4074		
	4198#	4200	4664#	4666	4987#	4989	5006#	5008							
\$OCNT 033246	6650#	6679#	6692#												
\$OCTL 034764	3424	7017	7042#												
\$UMODE 033250	6645#	6649#	6654	6657#	6668#	6694#									
\$OVER 031632	6314	6330	6338	6348	6357#										
\$PASS 001216	1827#	2944#	3415	4167	5031	5102*	5103*	5111	5124	6344	6361				
\$PASTH 001006	1315#														
\$PWACT 031210	6225#	6226#	6236#												
\$PWCON 031124	2920	6219#	6228												
\$PWUP 031136	6219	6225#													
\$QUES 001204	1815#	6415	6494	6855	6926	6943	7001	7004							
\$RDCHR 034112	6868#	7278													
\$RDEC= ***** U	7281														
\$ROLIN 034202	6891#	7279													
\$ROCT 034522	6965#	7280													
\$ROSZ = 000022	6884#														
\$RESRE 035442	7226#	7282													
\$RTNAD 024042	5123#														
\$R2A = ***** U	7283														
\$SAVRE 035404	7210#	7281													
\$S82D 035176	7116#														
\$SCOPE 031370	2914	6311#													
\$SETUP= 000137	2877#	2913	2914	2916	2918	2920	2922	2923	2924	2926	5100	5148	6312		
	6376	6402	6410	6738	6743	6744	6774	6950							
\$STUP = 177777	2877#														
\$SUPRS 035232	3441	4174	7133#												
\$SVLAD 031576	6322	6351#													
\$SVPC = 000244	1289#	1294													
\$SWR = 167400	933#	947	951	952	953	954	955	956	957	958	1812	1813	1814		
	2923	2924	2926	2927	3072	3121	3276	3372	3412	3452	3484	3517	3672		
	3821	3896	3972	4093	4208	4674	4998	5028	5095	5101	5116	5122	5124		
	6303	6304	6305	6306	6307	6313	6325	6327	6328	6331	6332	6333	6340		
	6341	6342	6354	6357	6360	6367	6368	6369	6370	6371	6380	6387	6399		
	6403	6415													
\$SWREG 001232	1835#	2947													
\$SWRMK= 000000	958	959	6307	6308	6329										
\$TESTN 001214	1826#	3374#	6352#	8636	8643	8651	8652	8660	8671	8683					
\$TIMES 001174	1812#	2923#	3072#	3121#	3276#	3372#	3412#	3452#	3484#	3517#	3672#	3821#	3896#		
	3972#	4093#	4208#	4674#	4998#	5028#	5101#	6340#	6347	6350#	6360				
\$TKB 001146	1799#	6034	6051	6698	6719	6730	6755	6783	6810						
\$TKCNT 033252	6699#	6714#	6744	6761#	6875	6877#									
\$TKINT 033262	2890	3740	3744	3777	6714#	6735	6796								
\$TKGEN= 033261	6703#	6769	6880												
\$TKQIN 033260	6700#	6715#	6716	6767#	6768#	6769	6771#								
\$TKQOU 033260	6701#	6716#	6878	6879#	6880	6882#									
\$TKQSR 033260	3729#	3732	3743#	6702#	6715	6771	6882								
\$TKS 001144	1798#	6698	6720#	6751#	6753	6759#	6781	6797#	6807	6819#	6839#				
\$TKSRV 033332	6717	6730#													
\$TMPD 001160	1806#	3730#	3732#	3733#	3734	3736	3738	4458#	4482#	4785#	4809#				
\$TMP1 001162	1807#	4459#	4465	4490#	4555	4617	4786#	4791	4817#	4876	4938				
\$TMP2 001164	1808#	4508#	4511	4516	4521	4526	4531	4536	4541#	4542	4551	4553	4605		
	4609	4614#	4615	4837#	4872	4874	4926	4930	4935#	4936					

CALIB	1403#	6102													
CHECK	1361#	3548	3573	3627	4045	4125	4305	4348	4395	4434	4591	4650	4756	4829	4912
	4971														
COMMEN	1078#														
CWD2	1376#														
DRCLR	1387#	6119													
ENDCOM	1078#														
ERROR	972#	3102	3159	3171	3187	3192	3214	3222	3237	3241	3252	3256	3287	3292	3308
	3316	3334	3338	3342	3346	3418	3456	3465	3470	3474	3490	3495	3497	3521	3542
	3551	3552	3553	3554	3557	3567	3576	3577	3578	3579	3582	3583	3597	3607	3621
	3630	3631	3632	3633	3640	3644	3649	3654	3656	3676	3687	3691	3696	3706	3712
	3752	3768	3783	3797	3825	3839	3847	3851	3859	3867	3871	3900	3919	3927	3935
	3939	3944	3948	3956	3960	3976	3985	3989	3996	4010	4015	4019	4025	4029	4033
	4038	4048	4049	4050	4051	4054	4057	4064	4068	4097	4112	4116	4128	4129	4130
	4131	4133	4139	4141	4179	4185	4189	4193	4215	4232	4252	4256	4265	4277	4290
	4296	4308	4309	4310	4311	4314	4318	4329	4336	4339	4351	4352	4353	4354	4361
	4371	4381	4386	4389	4398	4399	4400	4401	4404	4415	4422	4425	4437	4438	4439
	4440	4447	4463	4478	4484	4497	4502	4549	4572	4579	4582	4594	4595	4596	4597
	4631	4638	4641	4653	4654	4655	4656	4678	4701	4720	4724	4733	4742	4759	4760
	4761	4762	4768	4774	4789	4804	4811	4822	4832	4833	4834	4835	4845	4849	4857
	4861	4870	4893	4900	4903	4915	4916	4917	4918	4952	4959	4962	4974	4975	4976
	4977	5050	5069	5089	5503	5508	5512	5528	5543	5553	5563	5572	5589	6069	6081
	6090	6094	6109	6115	6118	6125	6128	6187							
ESCAPE	1078#														
F. EAB	1341#	4117	4297	4340	4426	4583	4642	4748	4904	4963					
GETPRI	1078#														
GETSWR	1078#	5148													
HDCHK3	1482#														
HOTBL	1495#														
LOOP	1325#	3180	3280	3461											
MSG	3061#	3063	3106#	3108	3260#	3262	3362#	3364	3404#	3406	3445#	3447	3476#	3478	3502#
	3504	3659#	3661	3812#	3814	3883#	3885	3964#	3966	4072#	4074	4198#	4200	4664#	4666
	4987#	4989	5006#	5008											
MULT	1078#														
NEWTST	1078#	3061	3106	3260	3362	3404	3445	3476	3502	3659	3812	3883	3964	4072	4198
	4664	4987	5006												
OMNTAG	1319#	1887													
POP	1078#	6547	6607	6608	6993	7189	7231								
PUSH	1078#	6506	6568	6570	6591	6967	7150	7211							
QKPACK	1734#	3635	3950	4059	4852										
QKSEEK	1507#	4181													
QKSRT	1709#	3974	6078												
QKUNLD	1751#	3488	3647												
RDATA	1578#														
RHDR	1452#														
REPORT	1078#														
SBOUND	1654#	4547	4867												
SCOPE	973#	3071	3120	3275	3371	3411	3451	3483	3516	3671	3820	3895	3971	4092	4207
	4673	4997	5027	5099											
SETPRI	1078#	6871													
SETTRA	7261#	7270	7271	7272	7273	7275	7277	7278	7279	7280	7281	7282	7283		
SETUP	1078#	2906													
SKIP	1078#	3049	3099	3175	3305	3416	3459	3473	3870	3876	4150	5003			
SLASH	1078#														
SPACE	1078#														
STARS	1078#	1287	1298	1300	1307	1771	1818	1821	3061	3070	3106	3119	3260	3274	3362

	3370	3404	3410	3445	3450	3476	3482	3502	3515	3659	3670	3812	3819	3883	3894
	3964	3970	4072	4091	4198	4206	4664	4672	4987	4996	5006	5026	5092	6299	6363
	6417	6496	6563	6620	6697	6774	6789	6860	6884	6953	7006	7045	7107	7125	7148
	7195	7240	8833	8842											
SWRSU	1078#	2928#													
TRMTRP	7261#														
TYPBIN	1078#														
TYPDEC	1078#	5111													
TYPNAM	1078#														
TYPNUM	1078#														
TYPOCS	1078#	3141	3206	3247	3395	3722	3769	3788	5073	5382	6160	6205			
TYPOCT	1078#	6802													
TYPTXT	1078#														
WDATA	1535#	4273													
WRCHK	1620#	4325	4411	4568	4627	4889	4948								
WRHDR	1430#	4248	4716												
SSCMRE	1769#														
SSCHTH	1769#	1806	1807	1808	1909	1810	1811								
SSESCA	1078#														
SSNEWT	1078#	3061	3106	3260	3362	3404	3445	3476	3502	3659	3812	3883	3964	4072	4198
	4664	4987	5006												
SSSET	7261#	7270	7271	7272	7273	7275	7277	7278	7279	7280	7281	7282	7283		
SSSETH	2944#														
SSSKIP	1078#	3049	3099	3175	3305	3416	3459	3473	3870	3876	4150	5003			
.EQUAT	933#	968													
.HEADE	933#	937													
.SETUP	933#	2877													
.SWRHI	933#	947													
.SWRLO	933#	959#													
.SACT1	933#	1285													
.SAPT8	1819#														
.SAPTH	933#	1296													
.SAPTY	933#	6561													
.SCATC	933#	1268													
.SCHTA	933#	1769													
.SOB2D	933#	7043													
.SOB2O	933#	7004													
.SEOP	933#	5090													
.SERRO	933#	6361													
.SMULT	933#	7146													
.SROOC	933#	6951													
.SREAD	933#	6695													
.SSAVE	933#	7193													
.SSB2D	933#	7105													
.SSCOP	933#	6297													
.SSUPR	933#	7123													
.STRAP	933#	7238													
.STYPD	933#	6494													
.STYPE	933#	6415													
.STYPO	933#	6618													

. ABS. 057412 000

% ERRORS DETECTED: 0 HARD 2 SOFT

E16

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JD.P11 28-JAN-77 11:05

MACY11 27(100%) 31-JAN-77 18:30 PAGE 200
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0198

DEFAULT GLOBALS GENERATED: 0

DZR6JD, DZR6JD.SEQ/SOL/CRF/NL:TOC/DOC=DZR6JD.P11

RUN-TIME: 23 20 2 SECONDS

RUN-TIME RATIO: 482/47=10.2

CORE USED: 31K (62 PAGES)

DOCUMENT PAGES: 198