

RK06

DISK DRIVE DIAG. PART 3 MD-11-DZR6J-C

EP-DZR6J-C-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

DEC 1976
digital
MADE IN USA

This microfiche card contains a grid of 100 frames of technical diagrams and text, arranged in 10 rows and 10 columns. The frames contain various diagrams, including:

- Block diagrams of disk drive components.
- Flowcharts for diagnostic procedures.
- Tables of data and parameters.
- Textual descriptions and instructions.

The diagrams are small and densely packed, typical of microfiche format. The text is also small and difficult to read at this scale.

B01

UNIBUS RK06 DRIVE DIAGNOSTIC PART 3
DZR6JC.P11 06-OCT-76 09:44

MAC:11 27(1006) 06-OCT-76 09:54 PAGE 1

SEG 0001

UNIBUS RK06 DRIVE DIAGNOSTIC PART 3
DZR6JC.P11 06-OCT-76 09:44

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6J-C-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 3
DATE:	DECEMBER 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAIZIAN

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 © 1976 BY DIGITAL EQUIPMENT CORPORATION

1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142

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 SUCCESSFU FOLLOWED BY THE RK06 DRIVE DIAGNOSTICS - PARTS 1 & 2.

3.0 PROGRAM CONSIDERATIONS

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

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 COMPATIBLE 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 COMPATIBLE WITH THE Rk611.

3.2 XXDP

THIS PROGRAM SHOULD NOT BE CHAINED BY XXDP.

CHAIN MODE OPERATION (MONITOR)

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

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 DR0 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

BITS 4-0 = 0 ALLOW CONSOLE OUTPUT
NOT USED

3. SWITCH 1 (SOFTWARE SWITCH REGISTER).
IF ENVIRONMENT MODE BIT 7 (SIZING BIT) IS SET TO 1,
THE SOFTWARE SWITCH REGISTER WILL BE USED, INSTEAD
OF THE HARDWARE CONSOLE SWITCH REGISTER. REGARDLESS
OF WHICH ONE IS USED, ALL BITS DEFINED IN SECTIONS
4.3 & 4.4 (SWITCH REGISTER OPTIONS) MAY USED
WHEN RUNNING IN STANDALONE MODE. IN APT SCRIPT MODE,
HOWEVER, BIT 14 (LOOP ON TEST) MUST ALWAYS BE SET
TO 0.

4. SWITCH 2 (USER SWITCH REGISTER)
NOT USED

5. CPU OPTIONS:
NOT USED

6. MEMORY TYPES 1-4 AND MAX MEMORY ADDRESSES
NOT USED

7. INTERRUPT VECTOR 1:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 210

8. BUS PRIORITY 1:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 5

9. INTERRUPT VECTOR 2:
NOT USED

10. BUS PRIORITY 2:
NOT USED

11. BASE ADDRESS:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. DEFAULT = 177440

12. DEVICE MAP:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. EACH BIT
SET TO 1 IN BITS 0-7 WILL SELECT THE CORRESPONDING
DRIVE TO BE TESTED. BITS 8-15 ARE NOT USED.

13. CONTROLLER DESCRIPTOR WORDS:
NOT USED

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

3.4 DUAL ACCESS

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

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

199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310

3.5 MEMORY MANAGEMENT

MEMORY MANAGEMENT NOT USED

3.6 PARITY CHECK ENABLED

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

3.7 BAD SECTOR

THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR
INFORMATION CONTAINED ON 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 POP 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

011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066

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 XADP.

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 /XDP, 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, IF SW15=0. PRESSING "CONTINUE" RESTORES 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.

367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420

423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478

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 TIMEOUT IS INHIBITED.

4.3.7 SW<09>

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP FOR ERRORS. IF THE PROGRAM DETECTS AN ERROR, IT WILL LOOP BACK TO THE BEGINNING OF TEST.

4.4 '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 RKO 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.

479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534

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
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-C-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)

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-C-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 SERIAL NO. AAA
CARTRIDGE SERIAL NO. BBB

DRIVE 1

DRIVE SERIAL NO. CCC
CARTRIDGE SERIAL NO. DDD

END PASS # 2

(ETC)

535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590

591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646

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 ON UNDETERMINED
STATE; IE: HEADS UNLOADED OR INVALID FORMAT.

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

IF HEADS ARE LOADED & FORMATTING IS VALID,
THE PROGRAM WILL:

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

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

1. ECHO ↑C
2. TYPE 'HALT PENDING, PLEASE WAIT'
3. DO THE TEST(S) THAT LOADS HEADS AND/OR FORMATS
THE INVALID 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.

647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
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

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

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, UFE 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

703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758

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.

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
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 AD & 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 FRONT PANEL RUN/STOP SWITCH TEST

THIS TEST ALLOWS THE HEADS TO LOAD. THE OPERATOR IS
ASKED TO VISUALLY VERIFY THE SEQUENCE OF HEADS LOADING &
UNLOADING BOTH MECHANICALLY & BY THE LIGHTS ON THE FRONT PANEL

TEST 14 AC LOW DETECTION PART 1

A PRELIMINARY AC LOW TEST IS PERFORMED HERE WHILE HEADS ARE UNLOADED.

BATTERY RETRACT WILL BE TESTED LATER.
THE PROGRAM WAITS FOR AC LOW TO ASSERT IN RKMR3.
FROM THIS POINT, THERE IS APPROX 4 MS AVAILABLE BEFORE DC LOW ASSERTS
& THE INTERFACE SHUTS DOWN.
THE INDICATION OF DC LOW WILL BE NON-EXISTENT DRIVE ASSERTING IN RKCS2.
AFTER POWER UP, VOLUME VALID IS CHECKED TO BE CLEARED.

TEST 15 CHECK NXF LOGIC

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

TEST 16 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 17 WRITE LOCK TEST

THIS TEST VERIFIES THAT DATA WRITTEN ON A SECTOR CANNOT BE
ALTERED ONCE THE WRITE LOCK SWITCH HAS BEEN ENABLED.

7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7890
7891
7892
7893
7894
7895
7896
7897
7898
7899
7900

IT ALSO CHECKS THAT WRITE PROTECT TAKES EFFECT ONLY AT
SECTOR BOUNDRIES WHEN DOING CONTINUOUS WRITING ON CYL 0, HEAD 0

TEST 20 AC LOW DETECTION PART 2

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

TEST 21 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 22 MULTIPLE DRIVE DETECTION TEST

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 TES
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

0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070

6.1 ERROR INTERPRETATION

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

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

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLE:

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	RKASCF
050343	100000	000000	140301	040200	000103	004000

MESSAGE AD ERROR
AFTER MANUALLY LOADING HEADS WITH DOOR OPEN

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

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR

UNIBUS RKB DRIVE DIAGNOSTIC PART 3
DZ8610.P11 06-OCT-76 09:44

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]

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200
000201
000202
000203
000204
000205
000206
000207
000208
000209
000210
000211
000212
000213
000214
000215
000216
000217
000218
000219
000220
000221
000222
000223
000224
000225
000226
000227
000228
000229
000230
000231
000232
000233
000234
000235
000236
000237
000238
000239
000240
000241
000242
000243
000244
000245
000246
000247
000248
000249
000250
000251
000252
000253
000254
000255
000256
000257
000258
000259
000260
000261
000262
000263
000264
000265
000266
000267
000268
000269
000270
000271
000272
000273
000274
000275
000276
000277
000278
000279
000280
000281
000282
000283
000284
000285
000286
000287
000288
000289
000290
000291
000292
000293
000294
000295
000296
000297
000298
000299
000300
000301
000302
000303
000304
000305
000306
000307
000308
000309
000310
000311
000312
000313
000314
000315
000316
000317
000318
000319
000320
000321
000322
000323
000324
000325
000326
000327
000328
000329
000330
000331
000332
000333
000334
000335
000336
000337
000338
000339
000340
000341
000342
000343
000344
000345
000346
000347
000348
000349
000350
000351
000352
000353
000354
000355
000356
000357
000358
000359
000360
000361
000362
000363
000364
000365
000366
000367
000368
000369
000370
000371
000372
000373
000374
000375
000376
000377
000378
000379
000380
000381
000382
000383
000384
000385
000386
000387
000388
000389
000390
000391
000392
000393
000394
000395
000396
000397
000398
000399
000400
000401
000402
000403
000404
000405
000406
000407
000408
000409
000410
000411
000412
000413
000414
000415
000416
000417
000418
000419
000420
000421
000422
000423
000424
000425
000426
000427
000428
000429
000430
000431
000432
000433
000434
000435
000436
000437
000438
000439
000440
000441
000442
000443
000444
000445
000446
000447
000448
000449
000450
000451
000452
000453
000454
000455
000456
000457
000458
000459
000460
000461
000462
000463
000464
000465
000466
000467
000468
000469
000470
000471
000472
000473
000474
000475
000476
000477
000478
000479
000480
000481
000482
000483
000484
000485
000486
000487
000488
000489
000490
000491
000492
000493
000494
000495
000496
000497
000498
000499
000500
000501
000502
000503
000504
000505
000506
000507
000508
000509
000510
000511
000512
000513
000514
000515
000516
000517
000518
000519
000520
000521
000522
000523
000524
000525
000526
000527
000528
000529
000530
000531
000532
000533
000534
000535
000536
000537
000538
000539
000540
000541
000542
000543
000544
000545
000546
000547
000548
000549
000550
000551
000552
000553
000554
000555
000556
000557
000558
000559
000560
000561
000562
000563
000564
000565
000566
000567
000568
000569
000570
000571
000572
000573
000574
000575
000576
000577
000578
000579
000580
000581
000582
000583
000584
000585
000586
000587
000588
000589
000590
000591
000592
000593
000594
000595
000596
000597
000598
000599
000600
000601
000602
000603
000604
000605
000606
000607
000608
000609
000610
000611
000612
000613
000614
000615
000616
000617
000618
000619
000620
000621
000622
000623
000624
000625
000626
000627
000628
000629
000630
000631
000632
000633
000634
000635
000636
000637
000638
000639
000640
000641
000642
000643
000644
000645
000646
000647
000648
000649
000650
000651
000652
000653
000654
000655
000656
000657
000658
000659
000660
000661
000662
000663
000664
000665
000666
000667
000668
000669
000670
000671
000672
000673
000674
000675
000676
000677
000678
000679
000680
000681
000682
000683
000684
000685
000686
000687
000688
000689
000690
000691
000692
000693
000694
000695
000696
000697
000698
000699
000700
000701
000702
000703
000704
000705
000706
000707
000708
000709
000710
000711
000712
000713
000714
000715
000716
000717
000718
000719
000720
000721
000722
000723
000724
000725
000726
000727
000728
000729
000730
000731
000732
000733
000734
000735
000736
000737
000738
000739
000740
000741
000742
000743
000744
000745
000746
000747
000748
000749
000750
000751
000752
000753
000754
000755
000756
000757
000758
000759
000760
000761
000762
000763
000764
000765
000766
000767
000768
000769
000770
000771
000772
000773
000774
000775
000776
000777
000778
000779
000780
000781
000782
000783
000784
000785
000786
000787
000788
000789
000790
000791
000792
000793
000794
000795
000796
000797
000798
000799
000800
000801
000802
000803
000804
000805
000806
000807
000808
000809
000810
000811
000812
000813
000814
000815
000816
000817
000818
000819
000820
000821
000822
000823
000824
000825
000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845
000846
000847
000848
000849
000850
000851
000852
000853
000854
000855
000856
000857
000858
000859
000860
000861
000862
000863
000864
000865
000866
000867
000868
000869
000870
000871
000872
000873
000874
000875
000876
000877
000878
000879
000880
000881
000882
000883
000884
000885
000886
000887
000888
000889
000890
000891
000892
000893
000894
000895
000896
000897
000898
000899
000900
000901
000902
000903
000904
000905
000906
000907
000908
000909
000910
000911
000912
000913
000914
000915
000916
000917
000918
000919
000920
000921
000922
000923
000924
000925
000926
000927
000928
000929
000930
000931
000932
000933
000934
000935
000936
000937
000938
000939
000940
000941
000942
000943
000944
000945
000946
000947
000948
000949
000950
000951
000952
000953
000954
000955
000956
000957
000958
000959
000960
000961
000962
000963
000964
000965
000966
000967
000968
000969
000970
000971
000972
000973
000974
000975
000976
000977
000978
000979
000980
000981
000982
000983
000984
000985
000986
000987
000988
000989
000990
000991
000992
000993
000994
000995
000996
000997
000998
000999
001000

.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 RKE 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-C2), SEPT 14, 1976.

.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

.SETTL BASIC DEFINITIONS

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

STACK= 1100

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

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

::MISCELLANEOUS DEFINITIONS

HT= 11 ::CODE FOR HORIZONTAL TAB

LF= 12 ::CODE FOR LINE FEED

CR= 15 ::CODE FOR CARRIAGE RETURN

CR_LF= 200 ::CODE FOR CARRIAGE RETURN-LINE FEED

PS= 177776 ::PROCESSOR STATUS WORD

.EQUIV PS,PSW

STKLMT= 177774 ::STACK LIMIT REGISTER

PIRQ= 177772 ::PROGRAM INTERRUPT REQUEST REGISTER

DSWR= 177570 ::HARDWARE SWITCH REGISTER

DDISP= 177570 ::HARDWARE DISPLAY REGISTER

::GENERAL PURPOSE REGISTER DEFINITIONS

R0= %0 ::GENERAL REGISTER

R1= %1 ::GENERAL REGISTER

R2= %2 ::GENERAL REGISTER

R3= %3 ::GENERAL REGISTER

R4= %4 ::GENERAL REGISTER

R5= %5 ::GENERAL REGISTER

R6= %6 ::GENERAL REGISTER

R7= %7 ::GENERAL REGISTER

SP= %6 ::STACK POINTER

PC= %7 ::PROGRAM COUNTER

::PRIORITY LEVEL DEFINITIONS

PR0= 0 ::PRIORITY LEVEL 0

PR1= 40 ::PRIORITY LEVEL 1

PR2= 100 ::PRIORITY LEVEL 2

PR3= 140 ::PRIORITY LEVEL 3

PR4= 200 ::PRIORITY LEVEL 4

PR5= 240 ::PRIORITY LEVEL 5

PR6= 300 ::PRIORITY LEVEL 6

PR7= 340 ::PRIORITY LEVEL 7

::"SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000

SW14= 40000

SW13= 20000

SW12= 10000

SW11= 4000

SW10= 2000

SW09= 1000

SW08= 400

SW07= 200

SW06= 100

SW05= 40

SW04= 20

SW03= 10

SW02= 4

100000
100001
100002
100003
100004
100005
100006
100007
100008
100009
100010
100011
100012
100013
100014
100015
100016
100017
100018
100019
100020
100021
100022
100023
100024
100025
100026
100027
100028
100029
100030
100031
100032
100033
100034
100035
100036
100037

001100

000011

000012

000015

000200

177776

177774

177772

177570

177570

000000

000001

000002

000003

000004

000005

000006

000007

000006

000007

000000

000040

000100

000140

000200

000240

000300

000340

100000

040000

020000

010000

004000

002000

001000

000400

000200

000100

000040

000020

000010

000004

1038 000002
 1039 000001
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1090
 1091
 1092
 1093

000000
 040000
 020000
 010000
 004000
 002000
 001000
 000400
 000200
 000100
 000040
 000020
 000010
 000004
 000002
 000001
 000004
 000010
 000014
 000014
 000014
 000014
 000020
 000024
 000030
 000034
 000060
 000064
 000240

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

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

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

.*BASIC "CPU" TRAP VECTOR ADDRESSES

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

.SBTTL RK06 CONTROLLER REGISTER DEFINITION

```

:094
1095
1096
1097 000000
1098 000002
1099 000004
1100 000006
1101 000010
1102 000012
1103 000014
1104 000016
1105 000020
1106 000024
1107 000026
1108 000034
1109 000036
1110 000030
1111 000032
1112
1113
1114
1115
1116
1117 000001
1118 000003
1119 000005
1120 000007
1121 000011
1122 000013
1123 000015
1124 000017
1125 000021
1126 000023
1127 000025
1128 000027
1129 000031
1130
1131 000001
1132 000100
1133 000200
1134 000400
1135 001000
1136 002000
1137 004000
1138 010000
1139 020000
1140 040000
1141 100000
1142 100000
1143
1144
1145
1146 000007
1147 000010
1148 000020
1149 000040
:          $BASE=177440
RKCS1= 0      ;CONTROL AND STATUS REGISTER 1
RKWC= 2      ;WORD COUNT REGISTER
RKBA= 4      ;BUS ADDRESS REGISTER
RKDA= 6      ;DESIRED TRACK SECTOR REGISTER
RKCS2= 10    ;CONTROL AND STATUS REGISTER 2
RKDS= 12    ;DRIVE STATUS REGISTER
RKER= 14    ;ERROR REGISTER
RKASOF= 16   ;ATTENTION SUMMARY AND OFFSET REGISTER
RKDC= 20    ;DESIRED CYLINDER REGISTER
RKDB= 24    ;DATA BUFFER
RKMR1= 26   ;MAINTENANCE REGISTER 1
RKMR2= 34   ;MAINTENANCE REGISTER 2 (MESSAGE LINE A)
RKMR3= 36   ;MAINTENANCE REGISTER 3 (MESSAGE LINE B)
RKECPS= 30   ;ECC POSITION INFORMATION
RKECPT= 32   ;ECC PATTERN INFORMATION

.SBTTL CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
;
; DRIVE COMMANDS
SELDRV= 1     ;SELECT DRIVE (GET STATUS)
PACK= 3      ;PACK ACKNOWLEDGE
CLEAR= 5     ;DRIVE CLEAR
UNLOAD= 7    ;UNLOAD
SRTSPL= 11   ;START SPINDLE
RECAL= 13    ;RECALIBRATE
OFFSET= 15   ;OFFSET
SEEK= 17     ;SEEK
RDATA= 21    ;READ DATA
WRDATA= 23   ;WRITE DATA
RDHEAD= 25   ;READ HEADER
WRHEAD= 27   ;WRITE HEADER AND DATA
WRTCHK= 31   ;WRITE CHECK

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

.SBTTL CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
DRVMSK= 7    ;MASK FOR DRIVE SELECTION CODE
RLS= BIT3    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
BAI= BIT4    ;BUS ADDRESS INCREMENT INHIBIT
SCLR= BITS   ;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
  
```

1150	000100	IR=	BIT6	: INPUT READY
1151	000200	OR=	BIT7	: OUTPUT READY
1152	000400	JFE=	BIT8	: UNIT FIELD ERROR
1153	001000	MDS=	BIT9	: MULTIPLE DRIVE SELECT
1154	002000	PGE=	BIT10	: PROGRAMMING ERROR
1155	004000	NEM=	BIT11	: NON-EXISTENT MEMORY
1156	010000	NED=	BIT12	: NON-EXISTENT DRIVE
1157	020000	UPE=	BIT13	: UNIBUS PARITY ERROR
1158	040000	WCE=	BIT14	: WRITE CHECK ERROR
1159	100000	DLT=	BIT15	: DATA LATE ERROR
1160				
1161		.SBTTL	ERROR REGISTER BIT DEFINITION (RKER:14)	
1162				
1163	000001	ILF=	BIT0	: ILLEGAL FUNCTION CODE
1164	000002	SKI=	BIT1	: SEEK INCOMPLETE
1165	000004	NXF=	BIT2	: NON-EXECUTABLE FUNCTION
1166	000010	DRPAR=	BIT3	: DRIVE DETECTED SERCON PARITY ERROR
1167	000020	FMTE=	BIT4	: FORMAT ERROR
1168	000040	DTYPE=	BIT5	: DRIVE TYPE ERROR
1169	000100	ECH=	BIT6	: ECC HARD
1170	000200	BSE=	BIT7	: BAD SECTOR ERROR
1171	000400	HVRC=	BIT8	: HEADER VRC ERROR
1172	001000	COE=	BIT9	: CYLINDER ADDRESS OVERFLOW ERROR
1173	002000	IDAE=	BIT10	: INVALID DISK ADDRESS ERROR: HEAD/CYL
1174	004000	WLE=	BIT11	: WRITE LOCK ERROR
1175	010000	DTE=	BIT12	: DRIVE TIMING ERROR
1176	020000	OPI=	BIT13	: OPERATION (SEARCH) INCOMPLETE
1177	040000	UNS=	BIT14	: DRIVE UNSAFE
1178	100000	DCK=	BIT15	: DATA CHECK
1179				
1180		.SBTTL	STATUS REGISTER BIT DEFINITION (RKDS:12)	
1181				
1182	000001	DRA=	BIT0	: DRIVE AVAILABLE (CONTROLLER IS SET IF : THIS BIT IS RESET)
1183				
1184	000004	OFST=	BIT2	: DRIVE OFFSET
1185	000010	ACLO=	BIT3	: AC LOW
1186	000020	DCLO=	BIT4	: DC LOW
1187	000040	DROT=	BIT5	: DRIVE OFF TRACK
1188	000100	VV=	BIT6	: VOLUME VALID
1189	000200	DRDY=	BIT7	: DRIVE READY
1190	000400	DDT=	BIT8	: DRIVE TYPE (0=RK06)
1191	004000	WRL=	BIT11	: WRITE LOCK
1192	020000	PIP=	BIT13	: POSITIONING IN PROGRESS
1193	040000	DSC=	BIT14	: DRIVE STATUS CHANGE
1194	100000	SVAL=	BIT15	: STATUS VALID
1195				
1196		.SBTTL	MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)	
1197				
1198	000017	MESMSK=	17	: MESSAGE MASK
1199	000020	PAT=	BIT4	: FORCE EVEN PARITY ON SERCON MESSAGE LINES
1200	000040	DMD=	BIT5	: DIAGNOSTIC MODE
1201	000100	MSP=	BIT6	: MAINTENANCE SECTOR PULSE
1202	000200	MIND=	BIT7	: MAINTENANCE INDEX
1203	000400	MCLK=	BIT8	: MAINTENANCE CLOCK
1204	001000	MERD=	BIT9	: MAINTENANCE ENCODED READ DATA
1205	002000	MEWD=	BIT10	: MAINTENANCE ENCODED WRITE DATA

1206	004000	PCA= BIT11	;PRECOMPENSATION ADVANCE
1207	010000	PCD= BIT12	;PRECOMPENSATION DELAY
1208	C20000	ECCW= BIT13	;ECC WORD IS BEING READ OR WRITTEN
1209	040000	WRTGAT= BIT14	;WRITE GATE
1210	100000	RDGATE= BIT15	;READ GATE
1211			
1212		.SBTTL DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)	
1213			
1214	000040	D.DRA= BITS	;DRIVE AVAILABLE
1215	000100	D.VV= BIT6	;VOLUME VALID
1216	000200	D.DRDY= BIT7	;DRIVE READY
1217	030400	D.DDT= BIT8	;DRIVE TYPE (0=RK06)
1218	001000	D.FORM= BIT9	;DRIVE FORMAT
1219	002000	D.OFF= BIT10	;OFFSET ON
1220	004000	D.WRL= BIT11	;WRITE LOCK
1221	010000	D.SPIN= BIT12	;SPINDLE ON
1222	020000	D.PIP= BIT13	;POSITIONING IN PROGRESS
1223	040000	D.DSC= BIT14	;DRIVE STATUS CHANGE
1224			
1225		.SBTTL DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)	
1226			
1227	003020	D.SSP= BIT4	;SERVO SIG PRESENT
1228	000040	D.HDHM= BIT5	;HEADS HOME
1229	000100	D.BRHM= BIT6	;BRUSHES HOME
1230	000200	D.DOOR= BIT7	;DOOR INTERLOCKED
1231	000400	D.CART= BIT8	;CARTRIDGE INTERLOCK
1232	001000	D.SPOK= BIT9	;SPEED OK
1233	002000	D.FWD= BIT10	;FORWARD
1234	004000	D.REV= BIT11	;REVERSE
1235	010000	D.LOAD= BIT12	;HEADS LOADING
1236	020000	D.RTZ= BIT13	;RETURN TO ZERO
1237	040000	D.UNLD= BIT14	;HEADS UNLOADING
1238			
1239		.SBTTL DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)	
1240			
1241	000040	D.IDAE= BITS	;INVALID DISK ADDRESS ERROR:HEAD/CYL
1242	000100	D.ACLO= BIT6	;AC LOW
1243	000200	D.FLT= BIT7	;DRIVE FAULT
1244	000400	D.NXF= BIT8	;NON-EXECUTABLE FUNCTION CODE
1245	001000	D.PAR= BIT9	;DRIVE DETECTED SERCON PARITY ERROR
1246	002000	D.SKI= BIT10	;SEEK INCOMPLETE
1247	004000	D.WLE= BIT11	;WRITE LOCK ERROR
1248	010000	D.SPLS= BIT12	;SPEED LOSS
1249	020000	D.DROT= BIT13	;DRIVE OFF TRACK
1250	040000	D.UNS= BIT14	;R/W UNSAFE
1251			
1252		.SBTTL DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)	
1253			
1254	000020	D.SECT= BIT4	;SECTOR ERROR
1255	000040	D.WCUR= BITS	;WRITE CURRENT AND NO WRITE GATE
1256	000100	D.WGAT= BIT6	;WRITE GATE AND NO TRANSISTIONS
1257	000200	D.HDFL= BIT7	;HEAD FAULT
1258	000400	D.MHD= BIT8	;MULTIPLE HEAD SELECT
1259	001000	D.XERR= BIT9	;INDEX ERROR
1260	002000	D.TIB= BIT10	;TRIBIT ERROR
1261	004000	D.PLO= BIT11	;PLO ERROR

M02

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
 C2R6JC.P11 06-OCT-76 09:44

MACV11 27(1006) 06-OCT-76 09:54 PAGE 25
 DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)

SEQ 0025

1262	010000	D.NMOV= BIT12	:SEEK AND NO MOTION
1263	020000	D.LIMD= BIT13	:LIMIT DETECT ON SEEK
1264	040000	D.SUNS= BIT14	:SERVO UNSAFE
1265			
1266		.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)	
1267			
1268	000007	M.DRV= 7	:DRIVE CODE, ALL BYTES
1269	017760	M.CDIF= 17760	:CYLINDER DIFF, BYTE 10
1270	017760	M.OFST= 17760	:OFFSET VALUE, BYTE 10
1271	077770	M.SER= 77770	:DRIVE SERIAL #, BYTE 11
1272			
1273		.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)	
1274			
1275	000003	M.ID= 3	:BYTE ID, ALL BYTES
1276	017760	M.CADD= 17760	:CYLINDER ADDRESS, BYTE 10
1277	040000	M.ALGN= BIT14	:ALIGN SIGN, BYTE 10
1278	000760	M.SECT= 760	:SECTOR COUNT, BYTE 11
1279	007000	M.HEAD= 7000	:HEAD DECODE, BYTE 11
1280	100000	M.PAR= BIT15	:PARITY, MESS A/B, ALL BYTES

```

1281
1282
1283
1284      000000
1285
1286
1287
1288      000174
1289      000174 000000
1290      000176 000000
1291
1292      000200 000137 010060
1293
1294      000220 000137 010050
1295
1296
1297      000240 000137 062002
1298
1299
1300
1301
1302
1303      000244
1304      000046
1305      000046 024470
1306      000052
1307      000052 120000
1308      000244
1309      001000
1310
1311
1312
1313
1314
1315      001000
1316      000024
1317      000024 000200
1318      000044
1319      000044 001000
1320      001000
1321
1322
1323
1324
1325      001000
1326      001000 000000
1327      001002 001210
1328      001004 000454
1329      001006 001130
1330      001010 001130
1331      001012 000052
1332
1333
1334
1335
1336

```

```

.SBTTL TRAP CATCHER
      .=0
;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
      .=174
DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0          ;;SOFTWARE SWITCH REGISTER
.SBTTL   STARTING ADDRESS(ES)
      JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
      .=220
      JMP PARSRT          ;INPUT ALL PARAMETERS & START TESTING
      .=240
      JMP 0.0DT          ;ENTER ODT11

.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
      $SVPC=.          ;SAVE PC
      .=46
      $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
      .=52
      .WORD 120000      ;;2)SET LOC.52 TO 120000
      .= $SVPC          ;;RESTORE PC
      .=1000

.SBTTL APT PARAMETER BLOCK
;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
      .$X=.          ;;SAVE CURRENT LOCATION
      .=24          ;;SET POWER FAIL TO POINT TO START OF PROGRAM
      200          ;;FOR APT START UP
      .=44          ;;POINT TO APT INDIRECT ADDRESS PNTR.
      $APTHDR          ;;POINT TO APT HEADER BLOCK
      .=.$X          ;;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.
$APTHD:
$HIBTS: .WORD 0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$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 LUNIT
      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

      .LIST MD
;

```


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
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392

```

:USE LOOP X TO OMIT SUBCLR
:
:
:THIS MACRO FILLS EXPECTED MSG AD, B0, A1, B1, A2, B2 & B3 WITH STANDARD
:BITS SEE A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEFS
:NOTE: A CAN BE ANY BIT COMBINATION DESIRED.
:
:
:THIS MACRO ASSUMES DRIVE MSG AD, B0, A1, B1 WILL ALWAYS BE TESTED
:USE A.C.D.E FOR MSG AD, 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>
:
:
:A=CYL DIFF/OFFSET ERROR #
:B=CYL ADDR ERROR #
:C=<ERROR DESCRIPTION>
:
:
:A=WRHEAD/<CFMT!WRHEAD>
:USE WRHDR <A>,X TO OMIT CHECKING AD, B0, A1, B1
:
:
:A=RDHEAD/<CFMT!RDHEAD>
:USE RDHDR <A>,X TO OMIT CHECKING AD, B0, A1, B1
:
:
:A=TOCYL/FRCYL . B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
:
:
:QUICK SEEK. ENTER WITH CYL# IN RKDC
:
:
: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 AD, B0, A1 & B1 AT THE END OF WRITING
:E: IF NON BLANK WILL OMIT CHECKING AD THRU B1
  
```



```

14 001100
141 001100
142 000000
143 000
144 000
145 000000
146 000000
147 000000
148 000000
149 000000
14A 000
14B 001
14C 000000
14D 000000
14E 000000
14F 000000
150 000000
151 000
152 001
153 000000
154 000000
155 000000
156 000000
157 000000
158 000000
159 000000
15A 000
15B 000
15C 000000
15D 001140 177570
15E 001142 177570
15F 001144 177560
160 001146 177562
161 001150 177564
162 001152 177566
163 001154 000
164 001155 002
165 001156 012
166 001157 000
167 001160 000000
168 001162 000000
169 001164 000000
16A 001166 000000
16B 001170 000000
16C 001172 000000
16D 001174 000000
16E 001176 000000
16F 001200 177607 000377
170 001204 077
171 001205 015
172 001206 000012
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190 001210
191 001210 000000

```

.SBTTL COMMON TAGS

```

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

```

SCMTAG: =1100

:: START OF COMMON TAGS

```

$*STNM: .WORD 0
SERFLG: .BYTE 000
$ICNT: .WORD 000000
$LFADR: .WORD 000000
$LPERR: .WORD 000000
$ERTTL: .WORD 000000
$ITEMB: .BYTE 001
$ERMAX: .BYTE 1
$ERRPC: .WORD 0
$GADR: .WORD 000000
$BADR: .WORD 000000
$GDDAT: .WORD 000000
$BDDAT: .WORD 000000
SAUTOB: .BYTE 000
SINTAG: .BYTE 0
$SWR: .WORD 0
DISPLAY: .WORD 0
$TKS: 177560
$TKB: 177562
$TPS: 177564
$TPB: 177566
$NULL: .BYTE 0
$FILLS: .BYTE 2
$FILLC: .BYTE 12
$TPFLG: .BYTE 0
$TMP0: .WORD 0
$TMP1: .WORD 0
$TMP2: .WORD 0
$TMP3: .WORD 0
$TMP4: .WORD 0
$TMP5: .WORD 0
$TIMES: 0
$ESCAPE: 0
$BELL: .ASCIZ <207><377><377>
$QUES: .ASCII /?/
$CRLF: .ASCII <15>
$LF: .ASCIZ <12>

```

```

:: CONTAINS THE TEST NUMBER
:: CONTAINS ERROR FLAG
:: CONTAINS SUBTEST ITERATION COUNT
:: CONTAINS SCOPE LOOP ADDRESS
:: CONTAINS SCOPE RETURN FOR ERRORS
:: CONTAINS TOTAL ERRORS DETECTED
:: CONTAINS ITEM CONTROL BYTE
:: CONTAINS MAX. ERRORS PER TEST
:: CONTAINS PC OF LAST ERROR INSTRUCTION
:: CONTAINS ADDRESS OF 'GOOD' DATA
:: CONTAINS ADDRESS OF 'BAD' DATA
:: CONTAINS 'GOOD' DATA
:: CONTAINS 'BAD' DATA
:: RESERVED--NOT TO BE USED

:: AUTOMATIC MODE INDICATOR
:: INTERRUPT MODE INDICATOR

:: ADDRESS OF SWITCH REGISTER
:: ADDRESS OF DISPLAY REGISTER
:: TTY KBD STATUS
:: TTY KBD BUFFER
:: TTY PRINTER STATUS REG. ADDRESS
:: TTY PRINTER BUFFER REG. ADDRESS
:: CONTAINS NULL CHARACTER FOR FILLS
:: CONTAINS # OF FILLER CHARACTERS REQUIRED
:: INSERT FILL CHARS. AFTER A "LINE FEED"
:: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: USER DEFINED
:: MAX. NUMBER OF ITERATIONS
:: ESCAPE ON ERROR ADDRESS
:: CODE FOR BELL
:: QUESTION MARK
:: CARRIAGE RETURN
:: LINE FEED

```

.SBTTL APT MAILBOX-ETABLE

```

*****
.EVEN
$MAIL: .WORD 0
$MSGTY: .WORD 0

```

:: APT MAILBOX
:: MESSAGE TYPE CODE

E03

1492	001212	000000
1493	001214	000000
1494	001216	000000
1495	001220	000000
1496	001222	000000
1497	001224	000000
1498	001226	000000
1499	001230	000000
1500	001230	000
1501	001231	000
1502	001232	000000
1503	001234	000000
1504	001236	000000
1505		
1506		
1507		
1508		
1509		
1510		
1511		
1512		
1513		
1514	001240	000
1515	001241	000
1516		
1517	001242	000000
1518		
1519	001244	000
1520	001245	000
1521	001246	000000
1522	001250	000
1523	001251	000
1524	001252	000000
1525	001254	000
1526	001255	000
1527	001256	000000
1528	001260	000000
1529	001262	000000
1530	001264	177440
1531	001266	000000
1532	001270	000000
1533	001272	000000
1534	001274	000000
1535	001276	000000
1536	001300	000000
1537	001302	000000
1538	001304	000000
1539	001306	000000
1540	001310	000000
1541	001312	000000
1542	001314	000000
1543	001316	000000
1544	001320	000000
1545	001322	000000
1546	001324	000000
1547	001326	000000

\$FATAL:	.WORD	RFATAL	:: FATAL ERROR NUMBER
\$TESTN:	.WORD	ATESTN	:: TEST NUMBER
\$PASS:	.WORD	APASS	:: PASS COUNT
\$DEVCT:	.WORD	ADEVCT	:: DEVICE COUNT
\$UNIT:	.WORD	RUNIT	:: I/O UNIT NUMBER
\$MSGAD:	.WORD	RMSGAD	:: MESSAGE ADDRESS
\$MSGLG:	.WORD	RMSGLG	:: MESSAGE LENGTH
\$ETABLE:			:: APT ENVIRONMENT TABLE
\$ENV:	.BYTE	AENV	:: ENVIRONMENT BYTE
\$ENVM:	.BYTE	RENVM	:: ENVIRONMENT MODE BITS
\$SWREG:	.WORD	ASWREG	:: APT SWITCH REGISTER
\$USWR:	.WORD	ALSWR	:: USER SWITCHES
\$CFLOP:	.WORD	ACPJOP	:: CPU TYPE, OPTIONS
* * * * *			
BITS 15-11=CPU TYPE			
11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05			
11/70=06, PDQ=07, Q=10			
BIT 10=REAL TIME CLOCK			
BIT 9=FLOATING POINT PROCESSOR			
BIT 8=MEMORY MANAGEMENT			
\$MAMS1:	.BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
\$MTYP1:	.BYTE	AMTYP1	:: MEM. TYPE, BLK#1
* * * * *			
MEM. TYPE BYTE -- (HIGH BYTE)			
900 NSEC CORE=001			
300 NSEC BIPOLAR=002			
500 NSEC MOS=003			
\$MADR1:	.WORD	AMADR1	:: HIGH ADDRESS, BLK#1
* * * * *			
MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE			
\$MAMS2:	.BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
\$MTYP2:	.BYTE	AMTYP2	:: MEM. TYPE, BLK#2
\$MADR2:	.WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
\$MAMS3:	.BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
\$MTYP3:	.BYTE	AMTYP3	:: MEM. TYPE, BLK#3
\$MADR3:	.WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
\$MAMS4:	.BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
\$MTYP4:	.BYTE	AMTYP4	:: MEM. TYPE, BLK#4
\$MADR4:	.WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
\$VECT1:	.WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
\$VECT2:	.WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
\$BASE:	.WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
\$DEVN:	.WORD	ADEVN	:: DEVICE MAP
\$CDW1:	.WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
\$CDW2:	.WORD	ACDW2	:: CONTROLLER DESCRIPTION WORD#2
\$DDW0:	.WORD	ADDW0	:: DEVICE DESCRIPTOR WORD#0
\$DDW1:	.WORD	ADDW1	:: DEVICE DESCRIPTOR WORD#1
\$DDW2:	.WORD	ADDW2	:: DEVICE DESCRIPTOR WORD#2
\$DDW3:	.WORD	ADDW3	:: DEVICE DESCRIPTOR WORD#3
\$DDW4:	.WORD	ADDW4	:: DEVICE DESCRIPTOR WORD#4
\$DDW5:	.WORD	ADDW5	:: DEVICE DESCRIPTOR WORD#5
\$DDW6:	.WORD	ADDW6	:: DEVICE DESCRIPTOR WORD#6
\$DDW7:	.WORD	ADDW7	:: DEVICE DESCRIPTOR WORD#7
\$DDW8:	.WORD	ADDW8	:: DEVICE DESCRIPTOR WORD#8
\$DDW9:	.WORD	ADDW9	:: DEVICE DESCRIPTOR WORD#9
\$DDW10:	.WORD	ADDW10	:: DEVICE DESCRIPTOR WORD#10
\$DDW11:	.WORD	ADDW11	:: DEVICE DESCRIPTOR WORD#11
\$DDW12:	.WORD	ADDW12	:: DEVICE DESCRIPTOR WORD#12
\$DDW13:	.WORD	ADDW13	:: DEVICE DESCRIPTOR WORD#13

F03

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
CZR6JC.F11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 31
APT MAILBOX-ETABLE

SEG 003:

1548	001330	000000	\$DDW14: .WORD	ADDW14	::DEVICE DESCRIPTOR WORD#14
1549	001332	000000	\$DDW15: .WORD	ADDW15	::DEVICE DESCRIPTOR WORD#15
1550					
1551					
1552	001334		SETENC:		
1553					
1554		177440	ABASE=	177440	:DEFAULT BUSS ADDRESS
1555	001334	000210	RKVEC:	210	:DEFAULT CONTROLLER INTERRUPT VECTOR
1556	001336	000240	RKPRI:	PR5	:PRIORITY
1557	001340	172540	PKS:	172540	:P-CLOCK STATUS REG
1558	001342	172542	PKSB:	172542	:P-CLOCK SET BUFFER
1559	001344	172544	PKRB:	172544	:P-CLOCK READ BUFFER
1560	001346	177546	LKS:	177546	:L-CLOCK STATUS REG.
1561					
1562	001350	000100	LCVEC:	100	:L-CLOCK INTERRUPT VECTOR
1563	001352	000104	PCVEC:	104	:P-CLOCK INTERRUPT VECTOR.
1564					
1565		000114	MEMVEC=	114	:MEMORY PARITY VECTOR
1566		172100	MEMBAS=	172100	:MEMORY PARITY OPTION CSR START ADDR
1567					
1568	001354	000000	TRAPP:	0	:PC FOR MEMORY PARITY ERROR TRAP
1569					
1570	001356	000000	PARAM:	0	:1 FOR 220 START, NO DEFAULT
1571	001360	000000	FTITLE:	0	:FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
1572					
1573	001362	000000	DRVPTR:	0	:CONTAINS THE POINTER TO THE DRIVE FLAG
1574					: (DRIVO-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
1575					
1576		000040	SPBAR=	40	:SPACE BAR
1577	001364	000000	FRCYL:	0	:FROM CYLINDER
1578	001366	000000	TOCYL:	0	:TO CYLINDER
1579	001370	000000	CCYL:	0	:CURRENT CYL, USED IN N SQUARE TEST
1580	001372	000000	PCYL:	0	:PREV CYL, USED IN N SQUARE TEST
1581	001374	000000	CALDIF:	0	:CALC CYL DIFF USED IN N SQUARE TEST
1582	001376	000000	CYLDIF:	0	:CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
1583	001400	000000	CYLADD:	0	:CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
1584	001402	000000	CALADD:	0	:CYL ADDR USED IN FHD TAB ROUTINE
1585					
1586	001404	000074	HZ:	60.	:60 FOR 60 CPS
1587					:50 FOR 50 CPS
1588	001406	000000	COUNT:	0	:LOADED TO 50 OR 60 TO COUNT TO 1 SEC
1589					:OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
1590	001410	000000	SEC:	0	:SECOND COUNTER
1591	001412	000000	TIMUP:	0	:FLAG TO INDICATE TIME IS UP
1592	001414	000000	SECNT:	0	:SECTOR COUNT
1593	001416	000000	PSEC:	0	:PREVIOUS SECTOR
1594	001420	000000	ESEC:	0	:EXPECTED SECTOR
1595	001422	000000	SECTOR:	0	:SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
1596					
1597	001424	000001	T1:	1	:TIMEOUT CONSTANTS
1598	001426	000012	T10:	10.	
1599	001430	000062	T50:	50.	
1600	001432	000764	T500:	500.	
1601	001434	000144	T100:	100.	
1602	001436	011610	T5000:	5000.	
1603	001440	141520	T50000:	50000.	

1604				
1605	001442	000077	CYL:	63.
1606	001444	000177		127.
1607	001446	000277		191.
1608	001450	000377		255.
1609	001452	000477		319.
1610	001454	000577		383.
1611				
1612	001456	000000	TIM1:	0
1613	001460	000000	TIM2:	0
1614	001462	000000	TIM3:	0
1615	001464	000000	TIM4:	0
1616				
1617	001466	000000	LPCNT:	0
1618	001470	000000	LPTIM:	0
1619				
1620	001472	000000	SUM:	0
1621	001474	000000		0
1622	001476	000000	SUM1:	0
1623	001500	000000		0
1624				
1625	001502	000000	WD1:	0
1626	001504	000000	WD2:	0
1627				
1628	001506	000000	OFFERR:	0
1629				
1630				
1631	001510	000000	HEAD:	0
1632	001512	000000	HEADA:	0
1633	001514	000000	WD1:	0
1634	001516	000000	FCOMGT:	0
1635	001520	000000	FMT1:	0
1636	001522	000000	WDCNT:	0
1637				
1638	001524	000000	DATA0:	0
1639	001526	052525	DATA01:	52525
1640	001530	177777	DATA1:	177777
1641	001532	133467	DPAT1:	133467
1642	001534	070627	DPAT2:	70627
1643				
1644	001536	000000	WORD:	0
1645	001540	000000	HOWD:	0
1646				
1647	001542	000000	BSERR:	0
1648	001544	000000	LIMERR:	0
1649	001546	000000	MDSERR:	0
1650	001550	000000	BYPERR:	0
1651	001552	000000	CHKFLG:	0
1652				
1653	001554	000102	HDTAB:	.BLKW 66.
1654	001760	000102	RHTAB:	.BLKW 66.
1655	002164	000102	SRTTAB:	.BLKW 66.
1656				
1657	002370	000400	BSE22H:	.BLKW 256.
1658	002370	000400	BSE22S:	.BLKW 256.
1659	004370	000400	RDTAB:	.BLKW 256.

:CYLINDER NUMBERS USEC IN
 :CURRENT CROSSOVER TEST

:USED IN TIMING TESTS

:LOOP CTR USED IN CALCLK
 :LOOP TIME IN USEC

:LO ORDER FOR TIMING TESTS
 :HI ORDER FOR TIMING TESTS
 :LO ORDER FOR TIMING TESTS
 :HI ORDER FOR TIMING TESTS

:ACTUAL HEADER/DATA WORD
 :EXPECTED DATA WORD

:SET WHEN WRITE CHECK ERROR ON OFFSET

:HEAD NUMBER
 :HEAD # FROM H.B3, RT. JUSTIFIED
 :SHIFTED HEAD# FOR FORMATTER ROUTINE
 :FORMAT TYPE
 :SHIFTED FORMAT FOR FORMATTER ROUTINE
 :WORD COUNT

:ALL 0'S
 :0101 PATT
 :ALL 1'S

:HEADER/DATA WORD
 :HEADER WORD FROM RKDB

:CANNOT READ BSE INFO WHEN SET
 :LIMIT DETECT ERROR FLAG
 :MULT DRIVE SEL ERROR FLAG
 :SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
 :WORDS TO BE TESTED

:CALCULATED HEADER WORD TABLE
 :FILLED AFTER READ HEADER CMD
 :ABOVE RHTAB SORTED STARTING FORM
 :SECTOR 0 BY SORT ROUTINE
 :22 SECTOR HARDWARE BSE INFO.
 :22 SECTOR SOFTWARE BSE INFO.
 :FILLED AFTER READ DATA CMD

H03

JNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
DZR6JC.P:1 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 33
APT MAILBOX-ETABLE

SEG 0033

```

1660
1661 005370 000000 JNLD: 0 ;SET TO 0 IF HEADS ARE LOADED
1662 ;SET TO 1 IF HEADS UNLOADED
1663 005372 000000 BADMDR: 0 ;SET TO C IF FORMATTING OK
1664 ;SET TO 1 IF FORMATTING ALTERED
1665 005374 000000 HPEND: 0 ;SET TO 0 IF HALT NOT PENDING
1666 ;SET TO 1 IF HALT PENDING
1667
1668 ;THE ABOVE 3 FLAGS ARE USED
1669 ;BY 'STOP' ROUTINE TO BRING
1670 ;THE CPU TO A VALID HALT.
1671

```

```

1672
1673 005376 001 002 004 ATTN: .BYTE 1,2,4,10,20,40,100,200 ;ATN 0-7 RESP.
1674 005401 010 020 040
1675 005404 100 200
1676 .EVEN

```

```

1677 ;
1678 ;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
1679 ;THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
1680 ;

```

```

1681
1682 005406 000000 HCS1: 0 ;HOLD RKCS1
1683 005410 000000 HCS2: 00 ;HOLD RKCS2
1684 005412 000000 HWC: 00 ;HOLD RKWC
1685 005414 000000 HBA: 00 ;ETC.
1686 005416 000000 HDA: 00
1687 005420 000000 HDS: 00
1688 005422 000000 HER: 00
1689 005424 000000 HASOF: 00
1690 005426 000000 HDC: 00
1691 005430 000000 HDB: 00
1692 005432 000000 HMR1: 00
1693 005434 000000 HMR2: 00
1694 005436 000000 HMR3: 00
1695 005440 000000 HPCS: 00
1696 005442 000000 HPAT: 0

```

```

1697
1698 005444 000000 TEMP1: 0 ;TEMPORARY STORAGE.
1699 005446 000000 TEMP2: 00
1700 005450 000000 TEMP3: 00
1701 005452 000000 TEMP4: 00
1702 005454 000000 TEMPS: 0

```

```

1703 ;
1704 ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
1705 ;

```

```

1706 005456 000000 H.A0: 0
1707 005460 000000 H.B0: 00
1708 005462 000000 H.A1: 00
1709 005464 000000 H.B1: 00
1710 005466 000000 H.A2: 00
1711 005470 000000 H.B2: 00
1712 005472 000000 H.A3: 00
1713 005474 000000 H.B3: 0

```

```

1714 ;
1715 ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.

```


1716				
1717	005476	000000	.A0:	0
1718	005500	000000	.B0:	0
1719	005502	000000	.A1:	0
1720	005504	000000	.B1:	0
1721	005506	000000	.A2:	0
1722	005510	000000	.B2:	0
1723	005512	000000	.A3:	0
1724	005514	000000	.B3:	0
1725				
1726			;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.	
1727				
1728		000001	T.A2=BIT0	;TEST MSG A2 IF SET
1729		000002	T.B2=BIT1	
1730		000004	T.B3=BIT2	
1731				
1732				
1733			;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.	
1734				
1735				
1736	005516	000000	DDUMP:	0 ;FLAG - SET WHEN IN DDP DUMP MODE
1737	005520	000000	DDPCH:	0 ;FLAG - SET WHEN IN DDP CHAIN MODE
1738	005522	000000	ACT11:	0 ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION
1739	005524	000000	PPTP:	0 ;FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
1740	005526	000000	DRIVS:	0 ;CONTAINS THE NUMBER OF DRIVES PRESENT
1741				
1742			;THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE	
1743			;IS PRESENT AND IS TO BE TESTED.	
1744				
1745	005530	000000	DRIV0:	0 ;FLAG SET TO 1 WHEN DRIVE 0 PRESENT
1746	005532	000000	DRIV1:	0 ;FOR DRIVE 1
1747	005534	000000	DRIV2:	0 ;FOR DRIVE 2
1748	005536	000000	DRIV3:	0 ;FOR DRIVE 3
1749	005540	000000	DRIV4:	0 ;FOR DRIVE 4
1750	005542	000000	DRIV5:	0 ;FOR DRIVE 5
1751	005544	000000	DRIV6:	0 ;FOR DRIVE 6
1752	005546	000000	DRIV7:	0 ;FOR DRIVE 7
1753				
1754	005550	000000	LCLKF:	0 ;L-CLOCK FLAG PRESENT FLAG
1755	005552	000000	PCLKF:	0 ;P-CLOCK FLAG PRESENT FLAG
1756	005554	000000	DOTIM:	0 ;SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
1757	005556	000000	SIZFLG:	0 ;SET IF DEFAULT DO SIZING IN TEST 1

1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813

005560

005560 045754
005562 054012
005564 060506
005566 061102

005570 046173
005572 054012
005574 060506
005576 061102

005600 046214
005602 054012
005604 060506
005606 061102

005610 046235
005612 054012
005614 060506
005616 061102

005620 046324
005622 054012
005624 060506
005626 061102

005630 046400
005632 054012
005634 060506
005636 061102

005640 046454
005642 054012
005644 060506
005646 061102

.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
    
```

1814				
1815			:ERROR 10	
1816	005650	046537	EM11	:DRA & NED BOTH SET
1817	005652	054012	DH1	
1818	005654	060506	DT1	
1819	005656	061102	DF1	
1820			:ERR 11	
1821	005660	046603	EM12	:NO RDY
1822	005662	055407	DH27	:AFTER WRITE DATA CMD
1823	005664	060506	DT1	
1824	005666	061172	DF10	
1825			:ERR 12	
1826	005670	047073	EM21	:CERR SET
1827	005672	055407	DH27	
1828	005674	060506	DT1	
1829	005676	061172	DF10	
1830			:ERR 13	
1831	005700	046603	EM12	:NO RDY
1832	005702	055357	DH26	:AFTER READ DATA CMD
1833	005704	060506	DT1	
1834	005706	061172	DF10	
1835			:ERR 14	
1836	005710	047073	EM21	:CERR SET
1837	005712	055357	DH26	
1838	005714	060506	DT1	
1839	005716	061172	DF10	
1840			:ERR 15	
1841	005720	046603	EM12	:NO RDY
1842	005722	055616	DH32	:AFTER WRITE CHECK CMD
1843	005724	060506	DT1	
1844	005726	061172	DF10	
1845			:ERR 16	
1846	005730	052565	EM80	:WRITE CHECK ERROR SET
1847	005732	055616	DH32	:AFTER WRITE CHECK CMD
1848	005734	060506	DT1	
1849	005736	061172	DF10	
1850			:ERR 17	
1851	005740	046603	EM12	:CONTR NOT RDY
1852	005742	055072	DH18	:AFTER JNLD CMD
1853	005744	060506	DT1	
1854	005746	061172	DF10	
1855			:ERR 20	
1856	005750	046641	EM13	:NO ATTN
1857	005752	055072	DH18	
1858	005754	060506	DT1	
1859	005756	061172	DF10	
1860			:ERR 21	
1861	005760	052742	EM83	:DATA CHECK ERROR
1862	005762	055357	DH26	:AFTER READ DATA CMD
1863	005764	060506	DT1	
1864	005766	061172	DF10	
1865			:ERR 22	
1866	005770	047073	EM21	:CERR SET
1867	005772	055616	DH32	:AFTER WRITE CHECK CMD
1868	005774	060506	DT1	
1869	005776	061172	DF10	

1870		
1871	006000	047010
1872	006002	055407
1873	006004	060716
1874	006006	061306
1875		
1876	006010	047073
1877	006012	055175
1878	006014	060506
1879	006016	061172
1880		
1881	006020	047052
1882		
1883		
1884		
1885		

:ERR 23

EM18
DH27
DT13
DF21

:MSG B0 ERROR
:AFTER WRITE DATA CMD

:ERROR 24

EM21
DH21
DT1
DF10

:CERR SET
:AFTER SCLR

:ERR 25

EM20

:MSG B1 ERROR

1886				
1887	006022	055407	DH27	
1888	006024	060716	DT13	
1889	006026	061306	DF21	
1890				;ERR 26
1891	006030	047010	EM18	
1892	006032	055357	DH26	;AFTER READ DATA CMD
1893	006034	060716	DT13	
1894	006036	061306	DF21	
1895				
1896				;ERROR 27
1897	006040	047206	EM24	;VOL VALID NOT SET
1898	006042	055117	DH19	;AFTER PACK CMD
1899	006044	060506	DT1	
1900	006046	061172	DF10	
1901				;ERR 30
1902	006050	047052	EM20	;MSG B1 ERROR
1903	006052	055357	DH26	;AFTER READ DATA CMD.
1904	006054	060716	DT13	
1905	006056	061306	DF21	
1906				;ERR 31
1907	006060	047010	EM18	;MSG B0 ERROR
1908	006062	055616	DH32	;AFTER WRITE CHECK CMD
1909	006064	060716	DT13	
1910	006066	061306	DF21	
1911				;ERR 32
1912	006070	047052	EM20	;MSG B1 ERROR
1913	006072	055616	DH32	
1914	006074	060716	DT13	
1915	006076	061306	DF21	
1916				;ERR 33
1917	006100	050316	EM44	;VV NOT CLEARED
1918	006102	054126	DH4	;AFTER PACK RE-INSERTED
1919	006104	060506	DT1	
1920	006106	061172	DF10	
1921				;ERR 34
1922	006110	052462	EM76	;NO DRIVES FOUND IN DEVICE MAP
1923	006112	054012	DH1	
1924	006114	060506	DT1	
1925	006116	061102	DF1	
1926				;ERR 35
1927	006120	046767	EM17	;MSG A0 ERROR
1928	006122	054100	DH3	;AFTER AC SW OFF
1929	006124	060716	DT13	
1930	006126	061306	DF21	
1931				;ERR 36
1932	006130	047031	EM19	;MSG A1 ERROR
1933	006132	054100	DH3	
1934	006134	060716	DT13	
1935	006136	061306	DF21	
1936				;ERR 37
1937	006140	047052	EM20	;MSG A1 ERROR
1938	006142	054100	DH3	;AFTER OFFSET CMD
1939	006144	060716	DT13	
1940	006146	061306	DF21	
1941				;ERR 40

1942	006150	052113	EM71	:DETECTED 10 BAD SECTORS
1943	006152	055407	DH27	:AFTER WRITE DATA CMD
1944	006154	060506	DT1	
1945	006156	061172	DF10	
1946			:ERR 41	
1947	006160	046663	EM14	:WRONG ATTN
1948	006162	055072	DH18	:AFTER UNLOAD CMD
1949				
1950	006164	060506	DT1	
1951	006166	061172	DF10	
1952			:ERR 42	
1953	006170	046710	EM15	:DRDY NOT CLEARED
1954	006172	055072	DH18	
1955				
1956	006174	060506	DT1	
1957	006176	061172	DF10	
1958			:ERR 43	
1959	006200	046742	EM16	:DSC NOT SET
1960	006202	055072	DH18	
1961	006204	060506	DT1	
1962	006206	061172	DF10	
1963			:ERR 44	
1964	006210	047115	EM22	:DOOR NOT CLEARED
1965	006212	054374	DH8	:AFTER DRIVE UNLOADED & DOOR OPENED
1966	006214	060506	DT1	
1967	006216	061172	DF10	
1968			:ERR 45	
1969	006220	046767	EM17	:MSG A0 ERROR
1970	006222	054374	DH8	
1971	006224	060716	DT13	
1972	006226	061306	DF21	
1973			:ERR 46	
1974	006230	047010	EM18	:MSG B0 ERROR
1975	006232	054374	DH8	
1976	006234	060716	DT13	
1977	006236	061306	DF21	
1978			:ERR 47	
1979	006240	052163	EM72	:BSE ERROR IN WRITE CMD NOT ON BSE TABLE
1980	006242	055407	DH27	:AFTER WRITE DATA CMD
1981	006244	060506	DT1	
1982	006246	061172	DF10	
1983				
1984			:ERR 50	
1985	006250	052242	EM73	:DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1986	006252	054012	DH1	
1987	006254	060506	DT1	
1988	006256	061102	DF1	
1989			:ERR 51	
1990	006260	053405	EM93	:WRONG CYL# IN HEADER WORD
1991	006262	055334	DH25	:AFTER SEEK CMD
1992	006264	060660	DT9	
1993	006266	061272	DF20	
1994			:ERR 52	
1995	006270	046767	EM17	:MSG A0 ERROR
1996	006272	055407	DH27	:AFTER WRITE DATA CMD
1997	006274	060716	DT13	

2000	006276	061306		DF21	
2001	006300	047031	:ERR 53	EM19	:MSG A1 ERROR
2002	006302	055407		DH27	
2003	006304	060716		DT13	
2004	006306	061306	:ERR 54	DF21	
2005	006310	046767		EM17	:MSG AC ERROR
2006	006312	055357		DH26	:AFTER READ DATA CMD
2007	006314	060716		DT13	
2008	006316	061306	:ERROR 55	DF21	
2009	006320	046641		EM13	:NO ATTN
2010	006322	055046		DH17	:AFTER RECAL CMD
2011	006324	060506		DT1	
2012	006326	061172	:ERR 56	DF10	
2013	006330	047031		EM19	:MSG A1 ERROR
2014	006332	055357		DH26	
2015	006334	060716		DT13	
2016	006336	061306	:ERR 57	DF21	
2017	006340	046767		EM17	:MSG AD ERROR
2018	006342	055616		DH32	:AFTER WRITE CHECK CMD
2019	006344	060716		DT13	
2020	006346	061306	:ERR 60	DF21	
2021	006350	047031		EM19	:MSG A1 ERROR
2022	006352	055616		DH32	
2023	006354	060716		DT13	
2024	006356	061306	:ERR 61	DF21	
2025	006360	051724		EM69	:NO DRIVES PRESENT
2026	006362	054012		DH1	
2027	006364	060506		DT1	
2028	006366	061102	:ERR 62	DF1	
2029	006370	052406		EM75	:FOUND 10 BAD CYL
2030	006372	055407		DH27	:AFTER WRITE DATA CMD
2031	006374	060506		DT1	
2032	006376	061172	:ERR 63	DF10	
2033	006400	047031		EM19	:MSG A1 ERROR
2034	006402	054374		DH8	:AFTER DRIVE UNLOADED & DOOR OPENED
2035	006404	060716		DT13	
2036	006406	061306	:ERR 64	DF21	
2037	006410	047052		EM20	:MSG B1 ERROR
2038	006412	054374		DH8	
2039	006414	060716		DT13	
2040	006416	061306	:ERR 65	DF21	
2041	006420	047156		EM23	:SPIN SET
2042	006422	054541		DH11	:AFTER LOADING HEADS WITH DOOR OPEN
2043	006424	060506		DT1	
2044	006426	061172		DF10	

2054			:ERR 66		
2055	006430	0476767		EM17	:MSG A0 ERROR
2056	006432	054541		DH11	
2057	006434	060716		DT13	
2058	006436	061306		DF21	
2059			:ERR 67		
2060	006440	047010		EM18	:MSG B0 ERROR
2061	006442	054541		DH11	
2062	006444	060716		DT13	
2063	006446	061306		DF21	
2064			:ERR 70		
2065	006450	047031		EM19	:MSG A1 ERROR
2066	006452	054541		DH11	
2067	006454	060716		DT13	
2068	006456	061306		DF21	
2069			:ERR 71		
2070	006460	047052		EM20	:MSG B1 ERROR
2071	006462	054541		DH11	
2072	006464	060716		DT13	
2073	006466	061306		DF21	
2074			:ERR 72		
2075	006470	047244		EM25	:CARTRIDGE NOT CLEARED
2076	006472	054615		DH12	:AFTER DISK PACK REMOVED
2077	006474	060506		DT1	
2078	006476	061172		DF10	
2079			:ERR 73		
2080	006500	000000		0	
2081	006502	000000		0	
2082	006504	000000		0	
2083	006506	000000		0	
2084			:ERR 74		
2085	006510	046767		EM17	:MSG A0 ERROR
2086	006512	054126		DH4	:AFTER PACK RE-INSERTED & HDS LOADED
2087	006514	060716		DT13	
2088	006516	061306		DF21	
2089			:ERR 75		
2090	006520	047010		EM18	:MSG B0 ERROR
2091	006522	054126		DH4	
2092	006524	060716		DT13	
2093	006526	061306		DF21	
2094			:ERR 76		
2095	006530	047031		EM19	:MSG A1 ERROR
2096	006532	054126		DH4	
2097	006534	060716		DT13	
2098	006536	061306		DF21	
2099			:ERR 77		
2100	006540	047052		EM20	:MSG B1 ERROR
2101	006542	054126		DH4	
2102	006544	060716		DT13	
2103	006546	061306		DF21	
2104			:ERR 100		
2105	006550	047156		EM23	:SPIN SET
2106	006552	054645		DH13	:AFTER LOADING HEADS WITH CART. OUT
2107	006554	060506		DT1	
2108	006556	061172		DF10	
2109			:ERR 101		

2110	006560	000000		
2111	006562	000000		
2112	006564	000000		
2113	006566	000000		
2114			:ERR 102	
2115	006570	000000		
2116	006572	000000		
2117	006574	000000		
2118	006576	000000		
2119			:ERR 103	
2120	006600	000000		
2121	006602	000000		
2122	006604	000000		
2123	006606	000000		
2124			:ERR 104	
2125	006610	000000		
2126	006612	000000		
2127	006614	000000		
2128	006616	000000		
2129			:ERR 105	
2130	006620	050765		EMS2
2131	006622	057670		DH64
2132	006624	060506		DT1
2133	006626	061172		DF10
2134			:ERR 106	
2135	006630	047401		EM28
2136	006632	054762		DH15
2137	006634	060506		DT1
2138	006636	061172		DF10
2139			:ERR 107	
2140	006640	047433		EM29
2141	006642	057250		DH59
2142	006644	060506		DT1
2143	006646	061172		DF10
2144			:ERR 110	
2145	006650	050554		EM48
2146	006652	056555		DH48
2147	006654	060506		DT1
2148	006656	061172		DF10
2149			:ERR 111	
2150	006660	047460		EM30
2151	006662	055007		DH16
2152	006664	060506		DT1
2153	006666	061172		DF10
2154			:ERR 112	
2155	006670	047354		EM27
2156	006672	055007		DH16
2157	006674	060506		DT1
2158	006676	061172		DF10
2159			:ERR 113	
2160	006700	046641		EM13
2161	006702	057250		DH59
2162	006704	060506		DT1
2163	006706	061172		DF10
2164			:ERROR 114	
2165	006710	051237		EMS8

:UNS NOT SET
:AFTER MDS FOUND

:VV SET
:WITHOUT PACK CMD

:DSC NOT SET
:AFTER EVEN PARITY ISSUED

:WRL NOT CLEARED
:AFTER WRITE LOCK SWITCH DISABLED

:ATTN NOT CLEARED
:AFTER UNIT SELECT PLUG REMOVED

:NED NOT SET

:ATTN NOT SET
:AFTER EVEN PARITY ISSUED

:PARITY NOT SET

2166	006712	057250	DH59	
2167	006714	060506	DT1	
2168	006716	061172	DF10	
2169			:ERP 115	
2170	006720	050614	EM49	:WRL NOT SET
2171	006722	056616	DH49	:AFTER WRITE LOCK SW ENABLED
2172	006724	060506	DT1	
2173	006726	061172	DF10	
2174			:ERROR 116	
2175	006730	046603	EM12	:CONT NOT RDY
2176	006732	055177	DH19	:AFTER PACK CMD
2177	006734	060506	DT1	
2178	006736	061172	DF10	
2179			:ERROR 117	
2180	006740	046603	EM12	:CONT NOT RDY
2181	006742	055142	DH20	:AFTER SEL DR CMD
2182	006744	060506	DT1	
2183	006746	061172	DF10	
2184			:ERROR 120	
2185	006750	046603	EM12	
2186	006752	055175	DH21	:AFTER SUBSYS CLEAR
2187	006754	060506	DT1	
2188	006756	061172	DF10	
2189			:ERR 121	
2190	006760	050650	EM50	:WLE NOT SET
2191	006762	056656	DH50	:AFTER WRITING WITH WRITE LOCK SET
2192	006764	060506	DT1	
2193	006766	061172	DF10	
2194			:ERR 122	
2195	006770	047354	EM27	:NED NOT SET
2196	006772	055255	DH23	:AFTER WRONG PORT SELECTED
2197	006774	060506	DT1	
2198	006776	061172	DF10	
2199			:ERR 123	
2200	007000	046767	EM17	:MSG A0 ERROR
2201	007002	056656	DH50	:AFTER WRITING WITH WRITE LOCK ENABLED
2202	007004	060716	DT13	
2203	007006	061306	DF21	
2204			:ERROR 124	
2205	007010	046603	EM12	
2206	007012	055046	DH17	:AFTER RECAL CMD
2207	007014	060506	DT1	
2208	007016	061172	DF10	
2209			:ERR 125	
2210	007020	047010	EM18	:MSG B0 ERROR
2211	007022	056656	DH50	:AFTER WITING WITH WRL ENABLED
2212	007024	060716	DT13	
2213	007026	061306	DF21	
2214			:ERR 126	
2215	007030	047031	EM19	:MSG A1 ERROR
2216	007032	056656	DH50	
2217	007034	060716	DT13	
2218	007036	061306	DF21	
2219			:ERR 127	
2220	007040	047052	EM20	:MSG B1 ERROR
2221	007042	056656	DH50	

2222	007044	060716	DT13	
2223	007046	061306	DF21	
2224			:ERR 130	
2225	007050	047511	EM31	:NED NOT CLEARED
2226	007052	055474	DH29	:AFTER CORRECT PCRT SELECTED
2227	007054	060506	DT1	
2228	007056	061172	DF10	
2229			:ERROR 131	
2230	007060	046603	EM12	:NO RDY
2231	007062	055334	DH25	:AFTER SEEK CMD
2232	007064	060506	DT1	
2233	007066	061172	DF10	
2234			:ERROR 132	
2235	007070	046641	EM13	:NC ATTN
2236	007072	055334	DH25	
2237	007074	060506	DT1	
2238	007076	061172	DF10	
2239			:ERR 133	
2240	007100	047354	EM27	:NED NOT SET
2241	007102	055440	DH28	:AFTER BOTH PORTS DESELECTED
2242	007104	060506	DT1	
2243	007106	061172	DF10	
2244			:ERR 134	
2245	007110	050712	EM51	:WRITE LOCK NOT SET SECTOR BOUNDRY
2246	007112	056656	DH50	:AFTER WRITING WITH WRL ENABLED
2247	007114	060530	DT3	
2248	007116	061112	DF3	
2249			:ERR 135	
2250	007120	050712	EM51	
2251	007122	057324	DH60	:AFTER WRITE LOCK ENABLED WHILE WRITING
2252	007124	060530	DT3	
2253	007126	061126	DF4	
2254			:ERR 136	
2255	007130	050712	EM51	
2256	007132	057617	DH63	:AFTER WRITE LOCK ENABLED FROM AC OFF
2257	007134	060530	DT3	
2258	007136	061112	DF3	
2259			:ERR 137	
2260	007140	050712	EM51	
2261	007142	057617	DH63	
2262	007144	060530	DT3	
2263	007146	061126	DF4	
2264			:ERR 140	
2265	007150	047542	EM32	:SPINDLE ON NOT SET
2266	007152	055562	DH31	:AFTER DRIVE MANUALLY LOADED
2267	007154	060506	DT1	
2268	007156	061172	DF10	
2269			:ERR 141	
2270	007160	047576	EM33	:DRIVE NOT READY
2271	007162	056077	DH38	:AFTER AC POWERED UP
2272	007164	060506	DT1	
2273	007166	061172	DF10	
2274			:ERR 142	
2275	007170	047627	EM34	
2276	007172	055562	DH31	
2277	007174	060506	DT1	

2278	007176	061172	DF10	
2279			:ERR 143	
2280	007200	046603	EM12	:CONT NOT READY
2281	007202	055674	DH34	:AFTER ST SPIN. CMD
2282	007204	060506	DT1	
2283	007206	061172	DF10	
2284			:ERR 144	
2285	007210	046641	EM13	:NO ATTN
2286	007212	055674	DH34	
2287	007214	060506	DT1	
2288	007216	061172	DF10	
2289			:ERR 145	
2290	007220	047670	EM35	:HEADS NOT HOME
2291	007222	055730	DH35	:AFTER MANUAL UNLOAD
2292	007224	060506	DT1	
2293	007226	061172	DF10	
2294			:ERR 146	
2295	007230	051211	EM57	:CERR NOT SET
2296	007232	056043	DH37	:AFTER TIMEOUT TO POWER DOWN
2297	007234	060506	DT1	
2298	007236	061172	DF10	
2299			:ERR 147	
2300	007240	047767	EM37	:AC LOW NOT SET
2301	007242	056043	DH37	
2302	007244	060506	DT1	
2303	007246	061172	DF10	
2304			:ERR 150	
2305	007250	050237	EM42	:NED NOT SET
2306	007252	056043	DH37	
2307	007254	060506	DT1	
2308	007256	061172	DF10	
2309			:ERROR 151	
2310	007260	046603	EM12	:NO RDY
2311	007262	055223	DH22	:AFTER CLEAR CMD
2312	007264	060506	DT1	
2313	007266	061172	DF10	
2314			:ERR 152	
2315	007270	050264	EM43	:AC LO NOT CLEARED
2316	007272	056077	DH38	:AFTER AC POWERED UP
2317	007274	060506	DT1	
2318	007276	061172	DF10	
2319			:ERR 153	
2320	007300	050316	EM44	:VV NOT CLEARED
2321	007302	056077	DH38	
2322	007304	060506	DT1	
2323	007306	061172	DF10	
2324			:ERROR 154	
2325	007310	051112	EM55	:ATTN NOT CLEARED
2326	007312	055223	DH22	
2327	007314	060506	DT1	
2328	007316	061172	DF10	
2329			:ERR 155	
2330	007320	050360	EM45	:VV SET AFTER HDS LOADED
2331	007322	054762	DH15	:WITHOUT 'PACK' CMD
2332	007324	060506	DT1	
2333	007326	061172	DF10	

5

2334			:ERR 156		
2335	007330	050435		EM46	:NXF=0
2336	007332	056410		DH45	:AFTER SEEK WITH VV=0
2337	007334	060506		DT1	
2338	007336	061172		DF10	
2339			:ERR 157		
2340	007340	050514		EM47	:CYL ADDR CHANGED FROM 0
2341	007342	056410		DH45	
2342	007344	060756		DT14	
2343	007346	061342		DF22	
2344			:ERR 160		
2345	007350	046767		EM17	:MSG A0 ERROR
2346	007352	056410		DH45	
2347	007354	060716		DT13	
2348	007356	061306		DF21	
2349			:ERR 161		
2350	007360	047010		EM18	:MSG B0 ERROR
2351	007362	056410		DH45	
2352	007364	060716		DT13	
2353	007366	061306		DF21	
2354			:ERR 162		
2355	007370	047031		EM19	:MSG A1 ERROR
2356	007372	056410		DH45	
2357	007374	060716		DT13	
2358	007376	061306		DF21	
2359			:ERR 163		
2360	007400	047052		EM20	:MSG B1 ERROR
2361	007402	056410		DH45	
2362	007404	060716		DT13	
2363	007406	061306		DF21	
2364			:ERR 164		
2365	007410	050435		EM46	:NXF NOT SET
2366	007412	056447		DH46	:AFTER WRITE DATA WITH VV=0
2367	007414	060506		DT1	
2368	007416	061172		DF10	
2369			:ERR 165		
2370	007420	051661		EM68	:CANNOT READ BSE INFO
2371	007422	056260		DH42	:ON SECTORS 0, 2, 4, 6, 8
2372	007424	060506		DT1	
2373	007426	061252		DF17	
2374			:ERR 166		
2375	007430	000000		0	
2376	007432	000000		0	
2377	007434	000000		0	
2378	007436	000000		0	
2379			:ERR 167		
2380	007440	051661		EM68	
2381	007442	060422		DH74	:ON SEC 10, 12....2C
2382	007444	060506		DT1	
2383	007446	061252		DF17	
2384			:ERR 170		
2385	007450	000000		0	
2386	007452	000000		0	
2387	007454	000000		0	
2388	007456	000000		0	
2389			:ERROR 171		

2390	007460	046603	EM12	:NO RDY
2391	007462	055530	DH30	:AFTER READ HEADER CMD
2392	007464	060506	DT1	
2393	007466	061172	DF10	
2394			:ERROR 172	
2395	007470	051346	EM61	:NXF DID NOT SET FAULT
2396	007472	056410	DH45	:AFTER SEEK WITH VV=0
2397	007474	060506	DT1	
2398	007476	061172	DF10	
2399			:ERROR 173	
2400	007500	051414	EM63	:DLT SET
2401	007502	055530	DH30	
2402	007504	060572	DT5	
2403	007506	061236	DF15	
2404			:ERROR 174	
2405	007510	047073	EM21	:CERR SET
2406	007512	055530	DH30	
2407	007514	060572	DT5	
2408	007516	061236	DF15	
2409			:ERR 175	
2410	007520	051012	EM53	:UNLD NOT SET
2411	007522	057670	DH64	:AFTER MDS FOUND
2412	007524	060506	DT1	
2413	007526	061172	DF10	
2414			:ERR 176	
2415	007530	051042	EM54	:CANNOT FIND MDS
2416	007532	057724	DH65	:AFTER SEARCHING ALL DRIVES
2417	007534	060506	DT1	
2418	007536	061172	DF10	
2419			:ERROR 177	
2420	007540	047312	EM26	:VV NOT CLEARED
2421	007542	055007	DH16	:AFTER UNIT SEL PLUG REMOVED
2422	007544	060506	DT1	
2423	007546	061172	DF10	
2424			:ERROR 200	
2425	007550	046603	EM12	:NO RDY
2426	007552	056123	DH39	:AFTER WRITE HEADER CMD
2427	007554	060572	DT5	
2428	007556	061236	DF15	
2429			:ERROR 201	
2430	007560	047073	EM21	:CERR SET
2431	007562	056123	DH39	
2432	007564	060572	DT5	
2433	007566	061236	DF15	
2434			:ERROR 202	
2435	007570	051145	EM56	:UNEXP MEMORY PARITY ERROR
2436	007572	057767	DH66	:TEST #,PRAP PC
2437	007574	060614	DT6	
2438	007576	061142	DF5	
2439			:ERROR 203	
2440	007600	047010	EM18	:MSG BO ERROR
2441	007602	054100	DH3	:AFTER AC SWITCHED OFF
2442	007604	060716	DT13	
2443	007606	061306	DF21	
2444			:ERR 204	
2445	007610	051211	EM57	:CERR NOT SET

2446	007612	060010	DH67	;AFTER TIMEOUT TO ENABLE WRL
2447	007614	060506	DT1	
2448	007616	061172	DF10	
2449			:ERR 205	
2450	007620	050650	EM50	;WRL NOT SET
2451	007622	060010	DH67	
2452	007624	060506	DT1	
2453	007626	061172	DF10	
2454			:ERROR 206	
2455	007630	051435	EM64	;WCE AT CYL 411,TRK 2. SEC 21
2456	007632	054012	DH1	
2457	007634	060676	DT10	
2458	007636	061162	DF7	
2459			:ERROR 207	
2460	007640	051211	EM57	;CERR NOT SET
2461	007642	056656	DH50	;AFTER WRITING WITH WRL ENABLED
2462	007644	060506	DT1	
2463	007646	061172	DF10	
2464			:ERROR 210	
2465	007650	047073	EM21	;CERR SET
2466	007652	055334	DH25	
2467	007654	060506	DT1	
2468	007656	061172	DF10	
2469			:ERR 211	
2470	007660	051267	EM59	;CTO SET
2471	007662	052022	EM70	;WHILE WAITING FOR OR REC'D CONTR RDY MSG A & B BAD
2472	007664	060506	DT1	
2473	007666	061206	DF12	
2474			:ERR 212	
2475	007670	051640	EM67	;NED SET
2476	007672	052022	EM70	
2477	007674	060506	DT1	
2478	007676	061206	DF12	
2479			:ERR 213	
2480	007700	046173	EM5	;MDS SET
2481	007702	052022	EM70	
2482	007704	060506	DT1	
2483	007706	061206	DF12	
2484			:ERR 214	
2485	007710	052361	EM74	;RTZ NOT SET
2486	007712	056233	DH41	;DURING RECD CMD
2487	007714	060506	DT1	
2488	007716	061172	DF10	
2489			:ERR 215	
2490	007720	046767	EM17	;MSG AD ERROR
2491	007722	056123	DH39	;AFTER WRITE HEADER CMD.
2492	007724	060716	DT13	
2493	007726	061306	DF21	
2494			:ERR 216	
2495	007730	047010	EM18	;BO ERROR
2496	007732	056123	DH39	
2497	007734	060716	DT13	
2498	007736	061306	DF21	
2499			:ERR 217	
2500	007740	047031	EM19	;A1 ERROR
2501	007742	056123	DH39	

2502	007744	060716	DT13	
2503	007746	061306	DF21	
2504			:ERR 220	
2505	007750	047052	EM20	:B1 ERROR
2506	007752	056123	DH39	
2507	007754	060716	DT13	
2508	007756	061306	DF21	
2509			:ERROR 221	
2510	007760	000000	0	
2511	007762	000000	0	
2512	007764	000000	0	
2513	007766	000000	0	
2514			:ERROR 222	
2515	007770	000000	0	
2516	007772	000000	0	
2517	007774	000000	0	
2518	007776	000000	0	
2519			:ERROR 223	
2520	010000	000000	0	
2521	010002	000000	0	
2522	010004	000000	0	
2523	010006	000000	0	
2524			:ERROR 224	
2525	010010	000000	0	
2526	010012	000000	0	
2527	010014	000000	0	
2528	010016	000000	0	
2529			:ERROR 225	
2530	010020	000000	0	
2531	010022	000000	0	
2532	010024	000000	0	
2533	010026	000000	0	
2534			:ERROR 226	
2535	010030	046603	EM12	:NO RDY
2536	010032	055357	DH26	:AFTER READ DATA CMD
2537	010034	060506	DT1	
2538	010036	061172	DF10	
2539			:ERROR 227	
2540	010040	047073	EM21	:CERR SET
2541	010042	055357	DH26	
2542	010044	060572	DT5	
2543	010046	061236	DF15	

2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599

.SBTTL PROGRAM SETUP

```

PARSRT: MOV #1,PARAM ;SET FLAG FOR 220 START
          BR PRGSRT ;START PROGRAM

START: CLR PARAM ;CLEAR FOR 200 START
PRGSRT: RESET ;CLEAR ALL INT ENABLE & INIT
        MOV #STACK,SP ;SETUP STACK POINTER
        MOV #PRD,-(SP) ;PSW LOADED TO BE
        MOV #IS,-(SP) ;LSI-11 COMPATABLE
        RTI ;ENABLE ALL INTERRUPTS

IS: JSR PC,$TKINT ;SETUP KB VECTOR ADDR, PRIORITY 4
     ;& TURN ON KB INTERRUPT
    
```

```

;*** CPU PRIORITY LEVEL NOW AT 0 ***
;*** ANY DEVICE WHICH SETS ITS ***
;*** INTERRUPT ENABLE BIT WILL ***
;*** SERVICED. ***
    
```

```

;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'STS')
;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
    
```

```

;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
    
```

;SYSMAC 'SETUP'

.SBTTL INITIALIZE THE COMMON TAGS

```

;;CLEAR THE COMMON TAGS ($CMTAG) AREA
MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
CLR (R6)+ ;;CLEAR MEMORY LOCATION
CMP #SWR,R6 ;;DONE?
BNE -6 ;;LOOP BACK IF NO
MOV #STACK,SP ;;SETUP THE STACK POINTER

;;INITIALIZE A FEW VECTORS
MOV #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
MOV #340,@IOTVEC+2 ;;LEVEL 7
MOV #ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
MOV #340,@EMTVEC+2 ;;LEVEL 7
MOV #STRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
MOV #340,@TRAPVEC+2 ;;LEVEL 7
MOV #SPWRDN,@PWRVEC ;;POWER FAILURE VECTOR
MOV #340,@PWRVEC+2 ;;LEVEL 7
MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
MOV #1,$ERMAX ;;ALLOW ONE ERROR PER TEST
MOV #,$SLPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
MOV #,$SLPERR ;;SETUP THE ERROR LOOP ADDRESS

;;SIZE FOR A HARDWARE SWITCH REGISTER, IF NOT FOUND OR IT IS
;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
MOV @ERRVEC,-(SP) ;;SAVE ERROR VECTOR
MOV #64,$ERRVEC ;;SET UP ERROR VECTOR
MOV #DSWR,$SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
    
```

M04

UNIVERS R#6 DRIVE DIAGNOSTIC PART 3
 CZR6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 51
 INITIALIZE THE COMMON TAGS

SEQ 0051

```

2600 010270 012737 177570 001142      MOV      #DDISP,DISPLAY      ;;AND A HARDWARE DISPLAY REGISTER
2601 010276 022777 177777 170634      CMP      #-1,DSWR          ;;TRY TO REFERENCE HARDWARE SWP
2602 010304 001012                    BNE      66$              ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2603                                ;;AND THE HARDWARE SWR IS NOT = -1
2604 010306 000403                    BR       65$              ;;BRANCH IF NO TIMEOUT
2605 010310 012716 010316 64$:      MOV      #65$, (SP)        ;;SET UP FOR TRAP RETURN
2606 010314 000002                    RTI
2607 010316 012737 000176 001140 65$:      MOV      #SWREG,SWR        ;;POINT TO SOFTWARE SWR
2608 010324 012737 000174 001142      MOV      #DISPREG,DISPLAY
2609 010332 012637 000004 66$:      MOV      (SP)+, @#ERRVEC   ;;RESTORE ERROR VECTOR
2610
2611 010336 005037 001216                    CLR      $PASS            ;;CLEAR PASS COUNT
2612 010342 132737 000200 001231      BITB    #APTSIZE,$ENVM    ;;TEST USER SIZE UNDER APT
2613 010350 001403                    BEQ      67$              ;;YES, USE NON-APT SWITCH
2614 010352 012737 001232 001140      MOV      #SSWREG,SWR     ;;NO, USE APT SWITCH REGISTER
2615 010360
2616
2617 010360 012737 010424 000004 MEMPAR: MOV    #1$,ERRVEC        ;TIMEOUT VECTOR
2618 010366 012737 000340 000006      MOV      #PR7,ERRVEC+2
2619
2620 010374 012701 172100                    MOV      #MEMBAS,R1      ;ADDR OF MEM CSR
2621 010400 005011 3$:      CLR      (R1)            ;SEE IF CAN REFERENCE
2622 010402 012711 000001      MOV      #1,(R1)        ;SET ENABLE BIT IF YES
2623 010406 012737 031456 000114      MOV      #MEMERR,MEMVEC ;LD MEMORY CHK VECTOR IF DONT TIMEOUT
2624 010414 012737 000340 000116      MOV      #PR7,MEMVEC+2
2625 010422 000401      BR       2$
2626
2627 010424 022626 1$:      CMP      (SP)+,(SP)+     ;ADJ STACK
2628 010426 062701 000002 2$:      ADD      #2,R1           ;TRY NEXT CSR
2629 010432 020127 172140      CMP      R1,#MEMBAS+40  ;ALL TRIED?
2630 010436 001360      BNE      3$              ;BR IN NO
2631 010440 012737 000006 000004      MOV      #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
2632 010446 005037 000006      CLR      ERRVEC+2
2633
2634 010452 004737 024524      JSR      PC,CLRFLG      ;CLEAR DDUMP THRU SIZFLG
2635 010456 005037 001220      CLR      $DEVCT
2636 010462 005037 001222      CLR      $UNIT
2637
2638
2639 ;FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
2640 ;
2641
2642 010466 005737 000042      START1: TST      42
2643 010472 001014      BNE      1$              ;BR IF AUTO
2644 010474 004737 024544      JSR      PC,TITLE      ;MANUAL, TYPE PROG ID
2645 010500 123727 000041 000013      CMPB    41,#13        ;13=LOADED BY XXDP
2646 010506 001010      BNE      2$              ;
2647 010510 005237 005516      INC      DDUMP          ;SET RK06 DUMP MODE FLAG
2648 010514 104401 037123      TYPE    MSG2           ;REPLACE DR0 PACK W/SCRATCH & DO<CR>
2649 010520 000137 010534      JMP      ST2
2650 010524 000137 010600      JMP      ST3
2651 010530 005237 005524 1$:      JMP      ST3
2652 2$:      INC      PPTP          ;SET ACT/APT/PTP DUMP MODE FLAG
2653
2654 ;CHECK IF ALL PARAMETERS DEFAULTED. IF NOT, BEGIN INPUT DIALOGUE
2655 ;WITH OPERATOR. THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE

```

```

2656 ;DRIVE NOS. SEPERATED BY COMMAS & TERMINATED BY <CR>
2657 ; EX: DRIVES TO BE TESTED: 1,2,4<CR>
2658 ;
2659 ;
2660 010534 005737 001356 ST2: TST PARAM
2661 010540 001002 BNE 1$ ;BR IF 220 START
2662 010542 000137 010632 JMP ST4 ;200 START, DEFAULT & SIZE THE BUSS
2663 010546 104401 037174 1$: TYPE MSG3 ;DRIVES TO BE TESTED
2664 010552 004737 024624 JSR PC,GDRVS ;GET DR NOS.
2665 010556 104401 037226 TYPE MSG4 ;BUSS ADDR
2666 010562 004737 024764 JSR PC,GBA ;GET BA
2667 010566 104401 037273 TYPE MSG5 ;CONT INT VECTOR
2668 010572 004737 025012 JSR PC,GINT ;GET INT VECTOR
2669 010576 000427 BR ST5
2670 ;
2671 ;
2672 ;
2673 ;AUTO MODE
2674 ;CHECK IF LOADED BY XXDP OR OTHER. SET FLAGS & NO INPUT DIALOGUE.
2675 ;DEFAULT ALL PARAMETERS. TEST ONLY THOSE DRIVES THAT ARE READY
2676 ;ON THE BUSS
2677 ;
2678 010600 123727 000041 000013 ST3: CMPB 41,#13 ;13=LOADED BY XXDP
2679 010606 001007 BNE 1$
2680 010610 005237 005520 INC DDPCH ;SET RK06 CHAIN MODE FLAG
2681 010614 004737 024544 JSR PC,TITLE
2682 010620 104401 037410 TYPE MSG7 ;DRO NOT TSTD
2683 010624 000402 BR ST4
2684 010626 005237 005522 1$: INC ACT11 ;SET ACT AUTO FLAG.
2685 ;
2686 010632 012737 177440 001264 ST4: MOV #177440,$BASE ;DEFAULT VALUE
2687 010640 012737 000210 001334 MOV #210,RKVEC ;DEFAULT VALUE
2688 010646 004737 025044 JSR PC,SETINT
2689 010652 005237 005556 INC SIZEFLG ;DO "SIZE THE BUSS" TEST
2690 ;
2691 010656 005037 005370 ST5: CLR UNLD ;INITIALIZE FLAGS
2692 010662 005037 005372 CLR BADHDR ;USED IN 'STOP ROUTINE
2693 010666 005037 005374 CLR HPEND ;FOR VALID PROGRAM HALTS
2694 010672 012737 005530 001362 MOV #DRIVO,DRVPTD ;SETUP
2695 010700 005037 001220 CLR $DEVCT ;NO. OF DRVS DONE
2696 010704 005037 001222 CLR $UNIT ;CURRENT DRV UNDER TEST
2697 010710 012737 010756 000004 MOV #1$,ERRVEC ;SETUP TIMEOUT ERROR VECTOR
2698 010716 005777 170424 TST @LKS ;SEE IF L-CLOCK THERE
2699 010722 005237 005550 INC LCLKF ;PRESENT, SET FLAG.
2700 010726 013700 001350 MOV LCVEC,RO ;VECTOR ADDR
2701 010732 012737 011020 000004 MOV #2$,ERRVEC
2702 010740 005777 170374 TST @PKS ;SEE IF P-CLOCK THERE
2703 010744 005237 005552 INC PCLKF ;PRESENT, SET FLAG
2704 010750 013700 001352 MOV PCVEC,RO ;VECTOR ADDR
2705 010754 000412 BR 3$
2706 ;
2707 010756 022626 1$: CMP (SP)+,(SP)+ ;L-CLOCK NOT THERE, CLEAR STACK
2708 010760 012737 011024 000004 MOV #4$,ERRVEC
2709 010766 005777 170346 TST @PKS ;SEE IF P-CLOCK THERE
2710 010772 005237 005552 INC PCLKF ;PRESENT, SET FLAG
2711 010776 013700 001352 MOV PCVEC,RO ;VECTOR ADDR

```

011002	005237	005554
011006	012720	030564
011012	012710	000300
011016	000407	
011020	022626	
011022	000767	
011024	022626	
011026	005037	005554
011032	104401	037617

```

35:   INC   DOTIM      :INDICATES TIMING TESTS CAN BE DONE
      MOV   BCLOCK,(R0)+ :SERVICE ROUTINE FOR CLOCKS
      MOV   BPRS,(R0)
      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 #PRD,-(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
:*

2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780

011036 000004
011040 012737 000001 001174
011046 012706 001100

011052 012746 000000
011056 012746 011064
011062 000002
011064

011064 012737 011202 000004
011072 013705 001264
011076 005765 000000
011102 005765 000010
011106 005765 000002
011112 005765 000004
011116 005765 000006
011122 005765 000012
011126 005765 000014
011132 005765 000016
011136 005765 000020
011142 005765 000024
011146 005765 000026
011152 005765 000034
011156 005765 000036
011162 005765 000030
011166 005765 000032

011172 012737 031370 000004
011200 000404

011202 022626
011204 104007
011206 000137 024402

2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836

011212 000004
011214 012737 000001 001174
011222 012706 001100
011226 005237 001550
011232 132737 000200 001231
011240 001002
011242 000137 011356
011246 104401 037523
011252 005037 005526
011256 005000
011260 012701 005530
011264 013702 001266
011270 032702 000000
011274 001410
011276 005237 005526
011302 005211
011304 104401 001205
011310 010046
011312 104403
011314 001
011315 000
011316 005721
011320 005200
011322 022700 000010
011326 001402
011330 006002
011332 000756
011334 005737 005526
011340 001402
011342 000137 012276
011346 104034
011350 000000
011352 000137 010656
011356 012765 000040 000010
011364 013737 001426 005444
011372 004737 025062

*** TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
*** MOS LIFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
*** DICATING THE OTHER PORT IS ACCESSED.

152: SCOPE
MOV #1,STIMES ;:DO 1 ITERATION
MOV #STACK,SP ;:RESTORE STK PTR
INC BYPCERR ;:DO NOT TEST CERR IN FRDY
BITB #BIT7,SEVM ;:SEE IF USE APT SELECTED DRIVES
BNE 145 ;:BR IF YES
JMP 125 ;:ELSE DO NORM SIZING OR VERIFY
145: TYPE MSG10 ;:WILL TEST DRIVES
CLR DRIVS ;:# OF DRIVES PRESENT
CLR RD ;:DRV ADDR
MOV #DRIV0,R1 ;:DRV FLAG
MOV \$DEVN,R2 ;:APT DEVICE MAP
155: BIT #BIT0,R2 ;:SEE IF DRV IN DEVICE MAP
BEQ 165 ;:BR IF NO
INC DRIVS ;:ELSE INCR DRIVE COUNT
INC (R1) ;:& SET DRIVE PRESENT FLAG
TYPE \$CALF
MOV RD,-(SP) ;:SAVE RD FOR TYPEOUT
;:TYPE DRIVE #
;:GO TYPE--OCTAL ASCII
;:TYPE 1 DIGIT(S)
;:SUPPRESS LEADING ZEROS
165: TST (R1)+ ;:ADV POINTER TO NEXT FLAG
INC RD ;:INC DRIVE #
CMP #8.,RD ;:ALL 8 TESTED?
BEQ 175 ;:BR IF YES
ROR R2 ;:ELSE GET NEXT BIT OFF DEVICE MAP
BR 155 ;:& TRY AGAIN
175: TST DRIVS ;:SEE IF MORE DRIVES PRESENT
BEQ 185 ;:BR IF NO
JMP NUDRV ;:ELSE EXIT TEST
185: ERROR 34 ;:NO DRIVES FOUND IN \$DEVN
HALT ;:SETUP CORRECTLY & PRESS 'CONTINUE'
JMP ST5 ;:TO TRY AGAIN
125: MOV #SCLR,RKCS2(R5) ;:SUBSYSTEM CLEAR
MOV T10,TEMP1 ;:SET TIMEOUT
JSR PC,FRDY ;:FIND RDY

E05

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
 CTR6JC.F11 06-007-76 09:44

MACY11 27(1006) 06-007-76 09:54 PAGE 56
 T2 SIZE THE BUSS

SEQ 0056

```

2837 011376 104120          ERROR 120          ;RDY NOT SET BY END OF SCLR
2838
2839
2840 011400 005737 005556      TST    SIZEFLG
2841 011404 001552          BEQ    *ST3          ;; DO NOT SIZE, GOTO NEXT TEST
2842 011406 104401 037523      TYPE   ,MSG10       ;WILL TEST DRIVES
2843 011412 005037 005526      CLR    DRIVS        ;# OF DRIVES PRESENT
2844 011416 005000          CLR    RD           ;DRV ADDR
2845 011420 012701 005530      MOV    #DRIVO,R1   ;DRV FLAG
2846
2847 011424 104415          15:   SCOP1
2848 011426 012706 001100      MOV    #STACK,SP   ;RESTORE STK PTR
2849
2850 011432 012765 000040 000010  MOV    #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
2851 011440 013737 001426 005444  MOV    T10,TEMP1    ;SET TIMEOUT
2852 011446 004737 025062          JSR    PC,FRDY      ;FIND RDY
2853 011452 104120          ERROR 120          ;RDY NOT SET BY END OF SCLR
2854 011454 010065 000010      MOV    RD,RKCS2(R5) ;SELECT THE DRIVE ADDR
2855 011460 012765 000001 000000  MOV    #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
2856 011466 013737 001426 005444  MOV    T10,TEMP1
2857 011474 004737 025062          JSR    PC,FRDY      ;FIND RDY
2858 011500 104117          ERROR 117          ;NO RDY AFTER SELECT DRIVE CMD.
2859 011502 032737 100000 005406  BIT    #CERR,HCS1
2860 011510 001046          BNE   2$
2861 011512 013737 005434 005444  MOV    HMR2,TEMP1
2862 011520 042737 177770 005444  BIC    #1<DRVMSK>,TEMP1
2863 011526 020037 005444          CMP    RD,TEMP1    ;S/B SAME
2864 011532 001016          BNE   3$
2865 011534 005700          TST   RD
2866 011536 001003          BNE   4$
2867 011540 005737 005520      TST    DDPCH        ;SEE IF XXDP CHAIN MODE
2868 011544 001014          BNE   5$
2869 011546 005237 005526          4$:   INC    DRIVS        ;INC DRIVE COUNT.
2870 011552 005211          INC    (R1)         ;SET DRIVE PRESENT FLAG
2871 011554 104401 001205      TYPE   ,SCLF
2872 011560 010046          MOV    RD,-(SP)     ;; SAVE RD FOR TYPEOUT
2873
2874 011562 104403          TYPOS
2875 011564 001          .BYTE 1           ;; TYPE DR #
2876 011565 000          .BYTE 0           ;; GO TYPE--OCTAL ASCII
2877 011566 000403          BR    5$           ;; TYPE 1 DIGIT(S)
2878
2879 011570 004737 025570          3$:   JSR    PC,BYP      ;TYPE BYPASS DR #
2880 011574 104001          ERROR 1            ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #
2881
2882 011576 005721          5$:   TST    (R1)+        ;SHIFT PTR TO NEXT DR. FLAG
2883 011600 005200          INC    RD           ;INC DR #
2884 011602 022700 000010      CMP    #8.,RD
2885 011606 001306          BNE   1$           ;MOF_...FT.
2886 011610 005737 005526      TST    DRIVS
2887 011614 001054          BNE   10$
2888 011616 104061          ERROR 61          ;NO DRIVES SEEN
2889 011620 000000          HALT
2890 011622 000137 010656          JMP    STS         ;SETUP & PRESS 'CONTINUE'
2891
2892 011626 032737 001000 005410  2$:   BIT    #MDS,HCS2
  
```


F05

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
D2R6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 57
T2 SIZE THE BUSS

SEQ 0057

2893	011634	001015			BNE	6\$		
2894	011636	032737	000400	005410	BIT	#UFE,HCS2		
2895	011644	001015			BNE	7\$		
2896	011646	032737	000001	005420	BIT	#DRA,MDS		
2897	011654	001015			BNE	8\$		
2898	011656	032737	010000	005410	BIT	#NED,HCS2		
2899	011664	001424			BEQ	9\$		
2900	011666	000743			BR	5\$		
2901								
2902	011670	004737	025570		6\$: JSR	PC,BYP		;TYPE BYP DR #
2903	011674	104002			ERROR	2		;MDS DETECTED
2904	011676	000737			BR	5\$		
2905								
2906	011700	004737	025570		7\$: JSR	PC,BYP		
2907	011704	104003			ERROR	3		;UFE DETECTED
2908	011706	000733			BR	5\$		
2909								
2910	011710	032737	010000	005410	8\$: BIT	#NED,HCS2		
2911	011716	001713			BEQ	4\$		
2912	011720	104401	037730		TYPE	MSG15		;DRV#
2913	011724	010046			MOV	RC,-(SP)		::SAVE R0 FOR TYPEOUT
2914								::TYPE DR#
2915	011726	104403			TYPOS			::GO TYPE--OCTAL ASCII
2916	011730	001			.BYTE	1		::TYPE 1 DIGIT(S)
2917	011731	000			.BYTE	0		::SUPPRESS LEADING ZEROS
2918	011732	104010			ERROR	10		;DRA & NED BOTH SET
2919	011734	000720			BR	5\$		
2920								
2921	011736	004737	025570		9\$: JSR	PC,BYP		
2922	011742	104004			ERROR	4		;NO DRA & NO NED = OTHER PORT SELECTED
2923	011744	000714			BR	5\$		
2924	011746	000137	012276		10\$: JMP	NUDRV		
2925								

```

*****
*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 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.
*****

```

2941	011752	000004			11\$: SCOPE			
2942	011754	012737	000001	001174	MOV	#1,STIMES		::DO 1 ITERATION
2943	011762	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
2944	011766	005000			CLR	R0		;DRIVE ADDR
2945	011770	012701	005530		MOV	#DRIVO,R1		;DRIVE FLAG
2946	011774							
2947	011774	104415			SCOPE1			
2948	011776	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR

G05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZRGJC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 58
T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 0058

2949
2950 012002 012765 000040 000010

MOV @SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR

H05

JNIBUS RK6 DRIVE DIAGNOSTIC PART 3
 0296JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 59
 T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 3059

2951	012010	013737	001426	005444	MOV	T10,TEMP1	:SET TIME OUT
2952	012016	004737	025062		JSR	PC,FRDY	:FIND RDY
2953	012022	104120			ERROR	120	:NO RDY AFTER SCLR
2954	012024	010065	000010		MOV	RO,RKCS2(R5)	:DRV ADDR
2955	012030	012765	000001	000000	MOV	#SELDRV,RKCS1(R5)	:SELECT DRIVE CMD
2956	012036	013737	001426	005444	MOV	T10,TEMP1	
2957	012044	004737	025062		JSR	PC,FRDY	:FIND RDY
2958	012050	104117			ERROR	117	:NO RDY AFTER SELECT DRIVE CMD.
2959	012052	032737	100000	005436	BIT	#CERR,HCS1	
2960	012060	001036			BNE	2\$	
2961	012062	013737	005434	005444	MOV	HMR2,TEMP1	
2962	012070	042737	177770	005444	BIC	#IC<DRVMSK>,TEMP1	
2963	012076	020037	005444		CMP	RO,TEMP1	:S/B SAME
2964	012102	001010			BNE	3\$	
2965	012104	005711			TST	(R1)	
2966	012106	001417			BEQ	5\$	
2967	012110	005721			TST	(R1)+	:SHIFT PTR TO NEXT DR FLAG
2968	012112	005200			INC	RO	:INC DR#
2969	012114	022700	000010		CMP	#8.,RO	
2970	012120	001325			BNE	1\$:MORE LEFT
2971	012122	000467			BR	TST4	:GO TO NEXT TEST
2972							
2973	012124	004737	025570		JSR	PC,BYP	:TRY BYPASS DRIVE#
2974	012130	104001			ERROR	1	:WRITTEN DR# DOES NOT MATCH RKMR2 DR#
2975	012132	005711			TST	(R1)	
2976	012134	001765			BEQ	4\$:BRANCH IF NOT SPEC BY INPLT
2977	012136	005337	005526		DEC	DRVS	:DECREMENT TOTAL DRVS
2978	012142	005011			CLR	(R1)	:CLEAR DRIVE FLAG
2979	012144	000761			BR	4\$	
2980							
2981	012146	004737	025570		JSR	PC,BYP	
2982	012152	104005			ERROR	5	:DR PRESENT BUT NOT SPECIFIED BY OPERATOR
2983	012154	000755			BR	4\$	
2984							
2985	012156	032737	001000	005410	2\$: BIT	#MDS,HCS2	
2986	012164	001027			BNE	6\$	
2987	012166	032737	000400	005410	BIT	#UFE,HCS2	
2988	012174	001027			BNE	7\$	
2989	012176	032737	000001	005420	BIT	#DRA,HDS	
2990	012204	001005			BNE	8\$	
2991	012206	032737	010000	005410	BIT	#NED,HCS2	
2992	012214	001423			BEQ	9\$	
2993	012216	000404			BR	10\$	
2994	012220	032737	010000	005410	8\$: BIT	#NED,HCS2	
2995	012226	001726			BEQ	11\$	
2996	012230	005711			10\$: TST	(R1)	
2997	012232	001726			BEQ	4\$	
2998							
2999	012234	004737	025570		JSR	PC,BYP	:TYPE BYPASS DRIVE#
3000	012240	104006			ERROR	6	
3001	012242	000735			BR	12\$	
3002							
3003	012244	004737	025570		6\$: JSR	PC,BYP	:TYPE BYPASS DRIVE#
3004	012250	104002			ERROR	2	:MDS DETECTED
3005	012252	000762			BR	8\$	
3006							

I05

JNIBJS RKE DRIVE DIAGNOSTIC PART 3
C2R6JC.F11 DE-007-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 60
T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 0060

3007 012254 004737 025570
3008 012260 104003
3009 012262 000756
3010
3011 012264 004737 025570
3012 012270 104004
3013 012272 000752
3014
3015 012274 001237
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025 012276 005037 001550
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037 012302 000004
3038 012304 012737 000001 001174
3039 012312 012706 001100
3040 012316 012737 000004 001214
3041 012324 012737 000004 001102
3042
3043 012332 005737 005526
3044 012336 001004
3045 012340 104401 040432
3046 012344 000137 024402
3047
3048 012350 013701 001362
3049 012354 005737 001220
3050 012360 001402
3051 012362 005237 001222
3052 012366 005721
3053 012370 001774
3054 012372 005737 005520
3055 012376 001403
3056 012400 005737 001222
3057 012404 001766
3058 012406 010137 001362
3059 012412 104401 037730
3060 012416 013700 001222
3061 012422 010046
3062

7\$: JSR PC,BYP
ERROR 3 ;UFE DETECTED
BR 8\$
9\$: JSR PC,BYP
ERROR 4 ;DRA & NED RESET - OTHER PORT SELECTED
BR 8\$
BNE 1\$;BRANCH IF MORE LEFT.
: THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
: DRIVE PRESENT
: '\$SUNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
: UNDER TEST
NUDRV: CLR BYPCERR ;ENTER HERE FROM LAST TEST
;TEST CERR IN 'FRDY'
:*****
:TEST 4 FIND NEXT DRIVE TO BE TESTED
: THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
: ADDRESS IN '\$SUNIT'.
: THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
: THE DRIVE WHOSE ADDRESS IS IN '\$SUNIT'.
:*****
TST4: SCOPE
MOV #1,\$TIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
MOV #STN-1,\$TESTN
MOV #STN-1,\$STNM
TST DRVS ;ANY DRIVES PRESENT?
BNE 4\$;YES BRANCH
TYPE ,MSG27 ;ALL DRIVES TESTED
JMP \$EOP ;NO, GO TO END
4\$: MOV DRVPTR,R1 ;ADDR OF NEXT DRIVE FLAG
TST \$DEVCT ;IS FIRST DRIVE BEING CHECKED
BEQ 2\$;YES, BRANCH
1\$: INC \$SUNIT ;INCR DRIVE ADDR TO NEXT DRIVE
2\$: TST (R1)+ ;IS DRIVE PRESENT?
BEQ 1\$;NO, FIND NEXT DRIVE PRESENT
TST DDPCH ;DDP CHAIN MODE?
BEQ 3\$;NO, BRANCH
TST \$SUNIT ;YES, IS IT DRIVE 0?
1\$;IF YES, DON'T TEST DR 0
3\$: MOV R1,DRVPTR ;STORE POINTER TO THE NEXT DR. FLAG
TYPE ,MSG15 ;"DRIVE"
MOV \$SUNIT,R0 ;SAVE R0 FOR TYPEOUT
MOV R0,-(SP) ;DRIVE #

J05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 61
T4 FIND NEXT DRIVE TO BE TESTED

SEQ 0051

3063	012424	104403		
3064	012426	001		
3065	012427	000		
3066				
3067	012430	104401	001205	
3068				
3069	012434			
3070				
3071				
3072				
3073				
3074				
3075				
3076				
3077	012434	000004		
3078	012436	012737	000001	001174
3079	012444	012706	001100	
3080				
3081	012450	005737	001216	
3082	012454	001046		
3083	012456	004737	026772	
3084	012462	104024		
3085				
3086	012464	104401	037742	
3087	012470	012765	000003	000026
3088	012476	004737	026420	
3089	012502	013701	005434	
3090	012506	012704	035414	
3091	012512	010446		
3092	012514	012703	000003	
3093	012520	006101		
3094	012522	006101		
3095	012524	006101		
3096	012526	006101		
3097	012530	006101		
3098	012532	006101		
3099	012534	010100		
3100	012536	042700	177760	
3101	012542	052700	000060	
3102	012546	110024		
3103	012550	005303		
3104	012552	001364		
3105	012554	105014		
3106				
3107	012556	004737	035662	
3108	012562	104401	001205	
3109	012566	104401	001205	
3110				
3111				
3112				
3113				
3114				
3115				
3116				
3117	012572	000004		
3118	012574	012737	000001	001174

```

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,$TIMES        ;;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,RKMR1(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 $SUPRS
MOV #3,R3           ;SETUP CHAR COOUNT
ROL R1              ;INITIALIZE BIT POSITIONS
1$: ROL R1           ;GET NEXT 4 BITS
ROL R1
ROL R1
MOV R1,R0           ;GET WORKING COPY
BIC #177760,R0      ;CLEAR ALL BUT LOW 4 BITS
BIS #60,R0          ;CONVERT TO ASCII DIGIT
MOVB R0,(R4)+       ;PUT ASCII DIGIT INTO CHAR BUFF
DEC R3
BNE 1$              ;BR IF ALL 3 CHARS NOT DONE
CLRB (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.
;*****
TST6: SCOPE
MOV #1,$TIMES        ;;DO 1 ITERATION

```

K05

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZB6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 62
T6 SET VV WITH PACK COMMAND

SEG 0062

```

3119 012602 012706 001100      MOV      #STACK,SP      ;RESTORE STK PTR
3120
3121 012606 004737 026772      JSR      PC,SUBCLR
3122 012612 104024                ERROR    24              ;CERR AFTER SCLR
3123
3124 012614 032737 000100 005434  BIT      #D.VV,HMR2
3125 012622 001024                BNE     TST7             ;;GO TO NEXT TEST IF VV SET
3126
3127 012624 104415                SCOP1
3128 012626 012706 001100      MOV      #STACK,SP      ;RESTORE STK PTR
3129
3130 012632 004737 026772      JSR      PC,SUBCLR
3131 012636 104024                ERROR    24              ;CERR AFTER SCLR
3132
3133 012640 012765 000003 000000  MOV      #PACK,RKCS1(R5) ;CMD TO SET VV
3134 012646 012737 000010 005444  MOV      #10,TEMP1
3135 012654 004737 025062      JSR      PC,FRDY        ;FIND RDY
3136 012660 104116                ERROR    116            ;RDY NOT SET AFTER PACK CMD
3137
3138 012662 032737 000100 005434  BIT      #D.VV,HMR2
3139 012670 001001                BNE     TST7             ;;GO TO NEXT TEST IF VV NOW SET
3140 012672 104027                ERROR    27              ;PACK DID NOT SET VV
3141

```

```

*****
;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 DPIPE.
;
*****

```

```

3142
3143
3144
3145
3146
3147
3148
3149 012674 000004      TST7:  SCOPE
3150 012676 012737 000001 001174  MOV      #1,$TIMES      ;;DO 1 ITERATION
3151 012704 012706 001100      MOV      #STACK,SP
3152
3153 012710 005237 005370      INC     UNLD            ;USED TO CHECK FOR VALID HALT
3154
3155 012714 004737 026772      JSR      PC,SUBCLR
3156 012720 104024                ERROR    24              ;CERR AFTER SCLR
3157
3158 012722 012765 000007 000000  MOV      #UNLOAD,RKCS1(R5) ;UNLOAD CMD
3159 012730 013737 001426 005444  MOV      T10,TEMP1
3160 012736 004737 025062      JSR      PC,FRDY        ;FIND CONTR RDY
3161 012742 104017                ERROR    17              ;NO RDY AFTER UNLD CMD
3162 012744 004737 025344      JSR      PC,TSTATN
3163 012750 104020                ERROR    20              ;NO ATTN AFTER UNLOAD CMD
3164

```

.SBTTL OPERATOR INTERVENTION TESTS

```

*****
;TEST 10     INTERLOCKS TESTS
;
;      THIS TEST VERIFIES THAT THE DOOR & CARTRIDGE STATUS BITS
;      ARE OPERATING PROPERLY IN MESSAGE AD & THAT THE REMOVAL
;      OF THE CARTRIDGE CLEARS VOLUME VALID.
;      IT FURTHER VERIFIES ALL REMAINING STATUS BITS OF

```

```

3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174

```

```

3175
3176
3177
3178
3179
3180
3181
3182 012752 000004
3183 012754 012737 000001 001174
3184 012762 012706 001100
3185
3186 012766 034737 026772
3187 012772 104024
3188
3189 012774 005237 001550
3190
3191 013000 104401 037445
3192 013004 104401 043041
3193 013010 004737 030746
3194 013014 000137 013630
3195
3196 013020 104401 042336
3197 013024 104401 041265
3198 013030 004737 031006
3199
3200 013034 104401 044637
3201 013040 104401 040570
3202 013044 104401 041265
3203 013050 004737 031006
3204 013054 012765 000001 000026
3205 013062 004737 026420
3206 013066 032737 000200 005434
3207 013074 001401
3208 013076 104044
3209
3210 013100 012737 000140 005476 15:
3211 013106 005037 005500
3212 013112 012737 000540 005502
3213 013120 012737 000001 005504
3214
3215 013126 004737 025604
3216 013132 000000
3217 013134 104045
3218 013136 104046
3219 013140 104063
3220 013142 104064
3221
3222 013144 004737 026772
3223 013150 104024
3224
3225 013152 104401 040720
3226 013156 104401 041005
3227 013162 104401 041265
3228 013166 004737 031006
3229
3230 013172 004737 026420

```

```

: * 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
: *
: *****
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(R5) ;SELECT WORD 1
JSR PC,GSTAT
BIT #D.DOOR,HMR2
BEQ 15
ERROR 44 ;DOOR STATUS BIT NOT CLEARED
MOV #<D.DRA!D.VV>,E.A0 ;EXPECTED A0
CLR E.B0
MOV #<D.HDHM!D.BPHM!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
JSR PC,GSTAT

```

M05

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
DZRBJC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 64
T10 INTERLOCKS TESTS

SEG 0064

3231	013176	032737	010000	005434	BIT	#D.SPIN,HMR2	
3232	013204	001401			BEQ	2\$	
3233	013206	104065			ERROR	65	;SPIN SET IN MSGAO
3234							
3235	013210	012737	000140	005476	2\$: MOV	#<D.DRA!D.VV>,E.A0	;EXPECTED MSG AO
3236	013216	005037	005500		CLR	E.B0	
3237	013222	012737	000540	005502	MOV	#<D.HDHM!D.BRHM!D.CART>,E.A1	
3238	013230	012737	000001	005504	MOV	#1,E.B1	
3239							
3240	013236	004737	025604		JSR	PC,CHKMSG	;CHECK MSGS AO, BO, A1, B1
3241	013242	000000			.WORD	0!0!0	; & MSGS SPECIFIED HERE
3242	013244	104066			ERROR	66	;MSG AO ERROR AFTER ATTEMPT TO START SPIN & DOOR OPEN
3243	013246	104067			ERROR	67	;MSG BO ERROR
3244	013250	104070			ERROR	70	;MSG A1 ERROR
3245	013252	104071			ERROR	71	;MSG B1 ERROR
3246							
3247	013254	004737	026772		JSR	PC,SUBCLR	
3248	013260	104024			ERROR	24	;CERR AFTER SCLR
3249							
3250	013262	104401	044637		TYPE	,MSG65	;DEPRESS 'RUN-STOP' SW TO STOP
3251	013266	104401	040660		TYPE	,MSG32	;REMOVE PACK & CLOSE DOOR
3252	013272	104401	041265		TYPE	,MSG37	
3253	013276	004737	031006		JSR	PC,GETSP	;GET SPACE FROM TTY
3254							
3255	013302	012765	000001	000026	MOV	#1,RKMR1(R5)	;SELECT WORD 1
3256	013310	004737	026420		JSR	PC,GSTAT	
3257	013314	032737	000400	005434	BIT	#D.CART,HMR2	
3258	013322	001401			BEQ	3\$	
3259	013324	104072			ERROR	72	;CARTRIDGE STATUS BIT NOT RESET
3260							
3261							
3262	013326	004737	026772		3\$: JSR	PC,SUBCLR	
3263	013332	104024			ERROR	24	;CERR AFTER SCLR
3264							
3265	013334	104401	041073		TYPE	,MSG35	;PRESS 'RUN-STOP' TO 'RUN'
3266	013340	104401	041005		TYPE	,MSG34	;VERIFY DOES NOT START
3267	013344	104401	041265		TYPE	,MSG37	
3268	013350	004737	031006		JSR	PC,GETSP	;GET SPACE FROM TTY
3269							
3270	013354	004737	026420		JSR	PC,GSTAT	
3271	013360	032737	010000	005434	BIT	#D.SPIN,HMR2	
3272	013366	001401			BEQ	5\$	
3273	013370	104100			ERROR	100	;SPIN SET IN MSG AO
3274							
3275	013372	104401	044637		5\$: TYPE	,MSG65	;PRESS 'RUN-STOP' TO 'STOP'
3276	013376	104401	041170		TYPE	,MSG36	;INSERT CARTRIDGE & CLOSE DOOR
3277	013402	104401	045222		TYPE	,MSG70	;VERIFY HEADS LOAD
3278	013406	104401	045145		TYPE	,MSG69	;DEPRESS SPACE WHEN READY GOES ON
3279	013412	004737	031006		JSR	PC,GETSP	;GET SPACE FROM TTY
3280	013416	005037	005370		CLR	UNLD	;FOR VALID HALT
3281							
3282	013422	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	
3283	013430	005065	000026		CLR	RKMR1(R5)	;SELECT WORD 0
3284	013434	004737	026420		JSR	PC,GSTAT	
3285	013440	032737	000100	005434	BIT	#D.VV,HMR2	
3286	013446	001401			BEQ	7\$	


```

3287 013450 104033          ERROR 33          ;VV NOT CLEARED AFTER PACK RE-INSERTED
3288
3289 013452 012737 050240 005476 75:  MOV    #<D.DSC!D.SPIN!D.DRDY!D.DRA>,E.A0      ;EXPECTED A0
3290 013460 005037 005500          CLR    E.B0
3291 013464 012737 001720 005502          MOV    #<D.SPOK!D.CART!D.SSP!D.BRHM!D.DOOR>,E.A1
3292 013472 012737 000001 005504          MOV    #1,E.B1
3293
3294 013500 004737 025604          JSR    PC,CHKMSG          ;CHECK MSGS A0, B0, A1, B1
3295 013504 000000          .WORD 0!0!0          ; & MSGS SPECIFIED HERE
3296 013506 104074          ERROR 74          ;MSG A0 ERROR AFTER PACK REINSERTED & HEADS LOADED
3297 013510 104075          ERROR 75          ;MSH B0 ERROR
3298 013512 104076          ERROR 76          ;MSG A1 ERROR
3299 013514 104077          ERROR 77          ;MSG B1 ERROR
3300
3301 013516 012765 100000 000000          MOV    #CCLR,RKCS1(R5)
3302 013524 013765 001222 000010          MOV    $UNIT,RKCS2(R5) ;DRIVE #
3303 013532 012765 000003 000000          MOV    #PACK,RKCS1(R5) ;PACK CMD
3304 013540 013737 001426 005444          MOV    T10,TEMP1
3305 013546 004737 025062          JSR    PC,FRDY          ;FIND CONTR RDY
3306 013552 104116          ERROR 116          ;CONTR NOT RDY
3307
3308 013554 032737 000100 005434          BIT    #D.VV,HMR2
3309 013562 001001          BNE   64$
3310 013564 104027          ERROR 27          ;VOLUME VALID NOT SET AFTER PACK CMD
3311 013566          64$:
3312 013566 005237 005370          INC    UNLD          ;FOR VALID HALT
3313
3314 013572 004737 026772          JSR    PC,SUBCLR
3315 013576 104024          ERROR 24          ;CERR AFTER SCLR
3316
3317 013600 012765 000007 000000          MOV    #UNLOAD,RKCS1(R5) ;UNLOAD CMD
3318 013606 013737 001426 005444          MOV    T10,TEMP1
3319 013614 004737 025062          JSR    PC,FRDY          ;FIND CONTR RDY
3320 013620 104017          ERROR 17          ;NO RDY AFTER UNLD CMD
3321 013622 004737 025344          JSR    PC,TSTATN
3322 013626 104020          ERROR 20          ;NO ATTN AFTER UNLOAD CMD
3323 013630          8$:
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337 013630 000004          TST11: SCOPE
3338 013632 012737 000001 001174          MOV    #1,$TIMES          ;;DO 1 ITERATION
3339 013640 012706 001100          MOV    #STACK,SP          ;RESTORE STACK PTR
3340
3341 013644 004737 026772          JSR    PC,SUBCLR
3342 013650 104024          ERROR 24          ;CERR AFTER SCLR

```

```

*****
;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 CONT-E
*****

```

```

*****
TST11: SCOPE
MOV    #1,$TIMES          ;;DO 1 ITERATION
MOV    #STACK,SP          ;RESTORE STACK PTR
;
JSR    PC,SUBCLR
ERROR 24          ;CERR AFTER SCLR

```

3394	013652	104401	040016		TYPE	.MSG18	:UNIT SELECT PLUG TEST
3395	013656	104401	043041		TYPE	.MSG52	:CONT-E TO EXIT OR SPACE TO CONTINUE
3396	013662	004737	030746		JSR	PC,CCSP	:INPUT CCNT-E OR SPACE
3397	013666	000137	014500		JMP	12\$:RETURN HERE FOR CONT-E
3398	013672	012765	000020	000026	MOV	#PAT,RKMR1(R5)	:EVEN PARITY TO GET DSC & ATTN
3399	013700	004737	026420		JSR	PC,GSTAT	
3400	013704	032737	001000	005436	BIT	#D.PAR,HMR3	:SEE IF PARITY SET
3401	013712	001001			BNE	2\$:BR IF YES
3402	013714	104114			ERROR	114	:PARITY BIT NOT SET AFTER SEL DRIVE CMD
3403							:WITH EVEN PARITY ISSUED.
3404	013716	032737	040000	005434	2\$: BIT	#D.DSC,HMR2	:SEE IF DSC SET
3405	013724	001001			BNE	1\$	
3406	013726	104107			ERROR	107	:DSC NOT SET AFTER SEL DRV WITH EVEN PARITY
3407	013730	012765	100000	000000	1\$: MOV	#CCLR,RKCS1(R5)	
3408	013736	004737	026420		JSR	PC,GSTAT	
3409	013742	004737	025344		JSR	PC,TSTATN	
3410	013746	104113			ERROR	113	:NO ATTN AFTER SEL DRV WITH EVEN PARITY
3411	013750	104401	037471		TYPE	.MSG9	:REMOVE UNIT SELECT PLUG
3412	013754	104401	041265		TYPE	.MSG37	:DEPRESS SPACE WHEN DONE
3413	013760	004737	031006		JSR	PC,GETSP	:GET SPACE
3414	013764	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	
3415	013772	004737	026420		JSR	PC,GSTAT	
3416	013776	004737	025344		JSR	PC,TSTATN	:RETURN HERE FOR SPACE
3417	014002	000401			BR	3\$:NO ATTN
3418	014004	104111			ERROR	111	:REMOVING UNIT SEL PLUG, DID NOT
3419							:DISABLE ATTN.
3420	014006	012765	100000	000000	3\$: MOV	#CCLR,RKCS1(R5)	
3421	014014	004737	026420		JSR	PC,GSTAT	
3422	014020	032765	010000	000010	BIT	#NED,RKCS2(R5)	
3423	014026	001001			BNE	4\$	
3424	014030	104112			ERROR	112	:REMOVING UNIT SEL PLUG DID NOT
3425							:ASSERT NON-EXISTENT DRIVE
3426	014032	104401	045030		4\$: TYPE	.MSG67	:DESELECT ALL OTHER PORTS
3427	014036	104401	041265		TYPE	.MSG37	:TYPE SPACE WHEN DONE
3428	014042	004737	031006		JSR	PC,GETSP	:GET SPACE
3429	014046	104401	041331		TYPE	.MSG38	:INSERT UNIT SELECT PLUGS
3430	014052	104401	041467		TYPE	.MSG39	:DEPRESS CONTROL-E WHEN FINISHED
3431	014056	104401	043322		TYPE	.MSG55	:EXIT WITH CORRECT UNIT SELECT PLUG
3432	014062	013746	001222		MOV	\$UNIT,-(SP)	:SAVE \$UNIT FOR TYPEOUT
3433							:TYPE CORRECT
3434	014066	104403			TYPOS		:GO TYPE--OCTAL ASCII
3435	014070	001			.BYTE	1	:TYPE 1 DIGIT(S)
3436	014071	000			.BYTE	0	:SUPPRESS LEADING ZEROS
3437	014072	104401	001205		TYPE	,SCLF	
3438	014076	105037	033710		CLRB	\$TKQSRT	:CLEAR PREVIOUS INFO
3439	014102	005037	001160		CLR	\$TMPD	
3440	014106	113737	033710	001160	5\$: MOVB	\$TKQSRT,\$TMPD	:GET CHAR IF THERE

```

014114 042737 177600 001160 BIC #C(177),STMPD ;GET RID OF JUNK, IF ANY
014120 009737 001160 TS# STMPD
014122 001422 BEQ #S ;BR IF NOTHING TYPED YET
014128 023727 001160 000005 CMP STMPD,#5 ;SEE IF CONT-E
014136 001426 BEQ #S ;BR IF YES
014140 023727 001160 000003 CMP STMPD,#3 ;SEE IF CONT-C
014146 001004 BNE #S ;BR IF NO
014150 004737 033712 JSR PC,STKINT ;ENABLE KB INT
014154 000137 031036 JMP STOP1 ;ELSE DO VALID HALT
014160 104401 040652 135: TYPE MSG3!
014164 105037 033710 CLR #RKCSRT
014170 004737 033712 JSR PC,STKINT

014174 005000 65: CLR RD
014176 012765 100000 000000 75: MOV #CCLR,RKCS1(R5)
014204 010065 000010 MOV RD,RKCS2(R5) ;DRIVE NO.
014210 012765 000001 000000 MOV #SELDV,RKCS1(R5) ;SELECT DRIVE CMD
014216 013737 001426 005444 MOV T10,TEMP1
014224 004737 025062 JSR PC,FRDY ;FIND CONTR RDY
014230 104117 ERROR 117 ;CONTR RDY NOT SET

014232 032765 010000 000010 BIT #NED,RKCS2(R5) ;SEE IF UNIT SELECT PLUG INSERTED
014240 001405 BEQ #S ;S IF MATCH DRIVE NO IN CS2. BR IF YES
014242 005200 INC RD
014244 020027 000010 CMP RD,#8. ;ALL 8 DRIVE NOS TRIED?
014250 001352 BNE #S ;BR IF NO
014252 000715 BR #S ;ELSE RETURN

014254 032737 000100 005434 85: BIT #D.VV,HMR2
014262 001311 BNE #S ;TYPE SAME UNIT SELECT PLUG RDY ONCE
014264 012765 100000 000000 MOV #CCLR,RKCS1(R5)
014272 010065 000010 MOV RD,RKCS2(R5) ;DRIVE ADDR
014276 012765 000003 000000 MOV #PACK,RKCS1(R5) ;PACK CMD
014304 013737 001426 005444 MOV T10,TEMP1
014312 004737 025062 JSR PC,FRDY ;FIND CONTR RDY
014316 104116 ERROR 116 ;CONTR NOT RDY
014320 010046 MOV RD,-(SP) ;SAVE RD FOR TYPEOUT
;TYPE UNIT SEL PLUG NO.
;GO TYPE--OCTAL ASCII
;TYPE 1 DIGIT(S)
;SUPPRESS LEADING ZEROS

014322 104403 TYPOS
014324 001 .BYTE 1
014325 000 .BYTE 0
014326 104401 001205 TYPE #SCLF
014332 000665 BR #S

014334 004737 033712 95: JSR PC,STKINT ;ENABLE KB INT
014340 012765 100000 000000 MOV #CCLR,RKCS1(R5)
014346 013765 001222 000010 MOV #UNIT,RKCS2(R5)
014354 012765 000001 000000 MOV #SELDV,RKCS1(R5)
014362 013737 001426 005444 MOV T10,TEMP1
014370 004737 025062 JSR PC,FRDY
014374 104117 ERROR 117 ;CONT NOT RDY AFTER SEL DR CMD

014376 032765 010000 000010 BIT #NED,RKCS2(R5) ;SEE IF CORRECT PLUG FOR EXIT
014404 001411 BEQ #S ;BR IF YES
014406 104401 043322 TYPE MSG55 ;EXIT WITH CORRECT UNIT SELECT PLUG NO.
014412 013746 001222 MOV #UNIT,-(SP) ;SAVE #UNIT FOR TYPEOUT

```

014400
014401
014402
014403
014404
014405
014406
014407
014408
014409
014410
014411
014412
014413
014414
014415
014416
014417
014418
014419
014420
014421
014422
014423
014424
014425
014426
014427
014428
014429
014430
014431
014432
014433
014434
014435
014436
014437
014438
014439
014440
014441
014442
014443
014444
014445
014446
014447
014448
014449
014450
014451
014452
014453
014454
014455
014456
014457
014458
014459
014460
014461
014462
014463
014464
014465
014466
014467
014468
014469
014470
014471
014472
014473
014474
014475
014476
014477
014478
014479
014480
014481
014482
014483
014484
014485
014486
014487
014488
014489
014490
014491
014492
014493
014494
014495
014496
014497
014498
014499
014500
014501
014502
014503
014504
014505
014506
014507
014508
014509
014510
014511
014512
014513
014514
014515
014516
014517
014518
014519
014520
014521
014522
014523
014524
014525
014526
014527
014528
014529
014530
014531
014532
014533
014534
014535
014536
014537
014538
014539
014540
014541
014542
014543
014544
014545
014546
014547
014548
014549
014550
014551
014552
014553
014554
014555
014556
014557
014558
014559
014560
014561
014562
014563
014564
014565
014566
014567
014568
014569
014570
014571
014572
014573
014574
014575
014576
014577
014578
014579
014580
014581
014582
014583
014584
014585
014586
014587
014588
014589
014590
014591
014592
014593
014594
014595
014596
014597
014598
014599
014600
014601
014602
014603
014604

104403
0001
104401 041467
000627
004737 026772
104024
004737 026420
032737 000100 005434
001005
104401 041532
104401 041467
000611
104401 045250
104401 041265
004737 031006
000004
012737 000001 001174
012706 001100
004737 026772
104024
104401 040050
104401 043041
004737 030746
000137 014716
104401 041644
104401 041265
004737 031006
004737 026420
032765 010000 000010
001001
104122
104401 041724
104401 041265
004737 031006

TYPJS
.BYTE 1
.BYTE 0
TYPE MSG39
BR 55
105: JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
JSR PC,GSTAT
BIT #D,VV,HMR2
BNE 15
TYPE MSG40 ;VV NOT SET, INSERT UNIT SELECT PLUG
TYPE MSG39 ;DEPRESS CONT-E TO EXIT TEST
BR 55
115: TYPE MSG71 ;SELECT CORRECT PORT ON OTHER DRIVES
TYPE MSG37 ;TYPE SPACE WHEN DONE
JSR PC,GETSP ;GET SPACE
125:

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
*

ST12: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STACK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
TYPE MSG19 ;PORT SELECTION TESTS
TYPE MSG52 ;DEPRESS CONT-E TO EXIT OR SPACE BAR TO CONTINUE
JSR PC,CCSP ;INPUT CONT-E OR SPACE
JMP 45 ;RETURN HERE FOR CONT-E
TYPE MSG41 ;SWITCH TO OTHER PORT
TYPE MSG37 ;DEPRESS SPACE WHEN DONE
JSR PC,GETSP ;GET SPACE
JSR PC,GSTAT
BIT #NED,RKCS2(R5)
BNE 15
ERROR 122 ;NED NOT SET AFTER WRONG PORT SELECTED
15: TYPE MSG42 ;SELECT CORRECT PORT
TYPE MSG37 ;DEPRESS SPACE BAR WHEN DONE
JSR PC,GETSP ;GET SPACE

```

3511 014610 004737 026772 JSR PC,SUBCLR
3512 014614 104024 ERROR 24 ;CERR AFTER SCLR
3513
3514
3515 014616 032765 010000 000010 BIT #NED,RKCS2(R5)
3516 014624 001401 BEQ 25
3517 014626 104130 ERROR 130 ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
3518
3519 014630 104401 042002 25: TYPE ,MSG43 ;DESELECT BOTH PORTS
3520 014634 104401 041265 TYPE ,MSG37 ;DEPRESS SPACE BAR WHEN DONE
3521 014640 004737 031006 JSR PC,GETSP
3522 014644 004737 026420 JSR PC,GSTAT
3523 014650 032765 010000 000010 BIT #NED,RKCS2(R5)
3524 014656 001001 BNE 35
3525 014660 104133 ERROR 133 ;NED NOT SET AFTER BOTH PORTS DESELECTED
3526
3527
3528 014662 104401 042032 35: TYPE ,MSG44 ;SELECT CORRECT PORT
3529 014666 104401 041265 TYPE ,MSG37 ;DEPRESS SPACE BAR WHEN DONE
3530 014672 004737 031006 JSR PC,GETSP
3531
3532 014676 004737 026772 JSR PC,SUBCLR
3533 014702 104024 ERROR 24 ;CERR AFTER SCLR
3534
3535 014704 032765 010000 000010 BIT #NED,RKCS2(R5)
3536 014712 001421 BEQ TST13 ;GOTO NEXT TEST
3537 014714 104130 ERROR 130 ;NED NOT CLEARED AFTER CORRECT PORT SELECTED
3538
3539 014716 012765 100000 000000 45: MOV #CCLR,RKCS1(R5)
3540 014724 004737 026420 JSR PC,GSTAT
3541 014730 032765 010000 000010 BIT #NED,RKCS2(R5)
3542 014736 001407 BEQ TST13 ;GOTO NEXT TEST
3543 014740 104401 042062 TYPE ,MSG45 ;CORRECT PORT NOT SELECTED, TRY AGAIN
3544 014744 104401 041265 TYPE ,MSG37 ;DEPRESS SPACE BAR WHEN DONE
3545 014750 004737 031006 JSR PC,GETSP
3546 014754 000760 BR 45
3547
3548
3549 *****
3550 *TEST 13 FRONT PANEL RUN/STOP SWITCH TEST
3551 *
3552 * THIS TEST ALLOWS THE HEADS TO LOAD. THE OPERATOR IS
3553 * ASKED TO VISUALLY VERIFY THE SEQUENCE OF HEADS LOADING &
3554 * UNLOADING BOTH MECHANICALLY & BY THE LIGHTS ON THE FRONT PANEL
3555 *
3556 *****
3557 TST13: SCOPE
3558 014756 000004 MOV #1,STIMES ;DO 1 ITERATION
3559 014760 012737 MOV #STACK,SP ;RESET STACK POINTER.
3560 014766 012706
3561
3562 014772 004737 026772 JSR PC,SUBCLR
3563 014776 104024 ERROR 24 ;CERR AFTER SCLR
3564
3565 015000 104401 040101 TYPE ,MSG20 ;RUN/STOP SWITCH TEST
3566 015004 104401 043041 TYPE ,MSG52 ;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
3567 015010 004737 030746 JSR PC,CCSP ;INPUT CONT-E OR SPACE
3568 015014 000137 015262 JMP 65 ;RETURN HERE IF CONT-E

```

F06

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
DZR6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 70
T13 FRONT PANEL RUN/STOP SWITCH TEST

SEQ 0070

```

3567
3568 015020 104401 042133 TYPE ,MSG46 ;DO MANUAL DRIVE LOAD
3569 015024 104401 045145 TYPE ,MSG69 ;DEPRESS SPACE WHEN DRIVE 'READY' ON
3570 015030 004737 031006 JSR PC,GETSP ;GET SPACE
3571
3572 015034 004737 026420 JSR PC,GSTAT
3573 015040 032737 010000 005434 BIT #D.SPIN,HMR2
3574 015046 001001 SNE 1$
3575 015050 104140 ERROR 140 ;SPINDLE NOT ON AFTER DRIVE MANUALLY LOADED
3576
3577 015052 032737 000200 005434 1$: BIT #D.DRDY,HMR2 ;SEE IF DRIVE READY (LOADED)
3578 015060 001001 BNE 3$ ;BR IF YES
3579 015062 104141 ERROR 141 ;DRIVE NOT READY AFTER MANUAL LOADING
3580
3581 015064 005037 005370 3$: CLR UNLD ;FOR VALID HALT
3582 015070 104401 042336 TYPE ,MSG47 ;ATTEMPT TO OPEN DOOR
3583 015074 104401 041265 TYPE ,MSG37 ;DEPRESS SPACE WHEN FINISHED
3584 015100 004737 031006 JSR PC,GETSP ;GET SPACE
3585
3586 015104 012765 100000 000000 MOV #CCLR,RKCS1(R5)
3587 015112 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
3588 015120 004737 026420 JSR PC,GSTAT
3589 015124 032737 000200 005434 BIT #D.DOOR,HMR2
3590 015132 001001 BNE 4$
3591 015134 104142 ERROR 142 ;DOOR STATUS BIT NOT SET
3592
3593 015136 104401 042416 4$: TYPE ,MSG48 ;UNLOAD HEADS MANUALLY
3594 015142 104401 041265 TYPE ,MSG37 ;SPACE BAR WHEN DONE
3595 015146 004737 031006 JSR PC,GETSP ;GET SPACE
3596
3597 015152 005237 005370 INC UNLD ;FOR VALID HALT
3598 015156 012765 100000 000000 MOV #CCLR,RKCS1(R5)
3599 015164 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
3600 015172 004737 026420 JSR PC,GSTAT
3601 015176 032737 000040 005434 BIT #D.HDHM,HMR2 ;CHECK HEAD HOME
3602 015204 001001 BNE 5$
3603 015206 104145 ERROR 145 ;HEADS HOME NOT SET AFTER MANUAL UNLD
3604
3605 015210 104401 042133 5$: TYPE ,MSG46 ;DO MANUAL LOAD
3606 015214 104401 045145 TYPE ,MSG69 ;PRESS SPACE AFTER READY ON
3607 015220 004737 031006 JSR PC,GETSP
3608
3609 015224 004737 026772 JSR PC,SUBCLR
3610 015230 104024 ERROR 24 ;CERR AFTER SCLR
3611
3612 015232 012765 000007 000000 MOV #UNLOAD,RKCS1(R5) ;UNLOAD CMD
3613 015240 013737 001426 005444 MOV T10,TEMP1
3614 015246 004737 025062 JSR PC,FRDY ;FIND CONTR RDY
3615 015252 104017 ERROR 17 ;NO RDY AFTER UNLD CMD
3616 015254 004737 025344 JSR PC,TSTATN
3617 015260 104020 ERROR 20 ;NO ATTN AFTER UNLOAD CMD
3618
3619
3620
3621
3622

```

```

*****
*TEST 14 AC LOW DETECTION PART 1
*

```

```

3623
3624
3625
3626
3627
3628
3629
3630
3631
3632 015262 000004
3633 015264 012737 000001 001174
3634 015272 012706 001100
3635
3636 015276 004737 026772
3637 015302 104024
3638
3639 015304 104401 040132
3640 015310 104401 043041
3641
3642 015314 004737 030746
3643 015320 000137 015666
3644 015324 104401 042613
3645
3646 015330 005037 005446
3647 015334 012737 177777 005444 12$:
3648 015342 004737 026420 1$:
3649 015346 032737 100000 005406
3650 015354 001012
3651 015356 005337 005444
3652 015362 001367
3653 015364 005237 005446
3654 015370 023727 005446 000002
3655 015376 001356
3656 015400 104146
3657
3658
3659 015402 000137 015466
3660
3661
3662
3663
3664
3665 015406 032737 000100 005436
3666 015414 001001
3667 015416 104147
3668
3669 015420 012737 040100 005476 2$:
3670 015426 012737 000300 005500
3671 015434 012737 000740 005476
3672 015442 012737 000001 005504
3673
3674 015450 004737 025604
3675 015454 000000
3676 015456 104035
3677 015460 104203
3678 015462 104036

: * A PRELIMINARY AC LOW TEST IS PERFORMED HERE WHILE HEADS ARE UNLOADED.
: * BATTERY RETRACT WILL BE TESTED LATER.
: * THE PROGRAM WAITS FOR AC LOW TO ASSERT IN RKMR3.
: * FROM THIS POINT, THERE IS APPROX 4 MS AVAILABLE BEFORE DC LOW ASSERTS
: * & THE INTERFACE SHUTS DOWN.
: * THE INDICATION OF DC LOW WILL BE NON-EXISTENT DRIVE ASSERTING IN PKCS2.
: * AFTER POWER UP, VOLUME VALID IS CHECKED TO BE CLEARED.
: *
: *****
15T14: SCOPE
MOV #1,STIMES ;:DO 1 ITERATION
MOV #STACK,SP ;:RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;:CERR AFTER SCLR
TYPE ,MSG21 ;:AC LOW TEST
TYPE ,MSG52 ;:CONT-E TO BYPASS TEST
;:OR SPACE TO CONTINUE
JSR PC,CCSP ;:INPUT CONT-E OR SPACE
JMP 10$ ;:RETURN HERE IF CONT-E
TYPE ,MSG49 ;:TURN OFF AC(RET HERE IF SPACE)
CLR TEMP2
MOV #-1,TEMP1 ;:SETUP TIMEOUT
JSR PC,GSTAT
BIT #CERR,HCS1
BNE 9$
DEC TEMP1
BNE 1$
INC TEMP2
CMP TEMP2,#2 ;:SEE IF GONE THRU 2 TIMES
BNE 12$ ;:BR IF NO
ERROR 146 ;:CERR NOT DETECTED BEFORE TIMEOUT

9$: JMP 3$ ;***THIS IS A TEMP JUMP***
;:ACLO REQUIRES SECTOR PULSE AND SINCE HEADS ARE UNLOADED
;:ACLO WILL NOT ASSERT & CERR WILL BE DUE TO NED
;:THE PROBLEM IS BEING LOOKED INTO
;:THIS TEST WILL EITHER BE ELIMINATED OR AC LO WILL BE MODIFIED TO ASSERT

BIT #D.ACLO,HMR3
BNE 2$
ERROR 147 ;:AC LOW NOT SET

2$: MOV #<D.DSC!D.VV>,E.A0 ;:EXPECTED MSG A0
MOV #<D.ACLO!D.FLT>,E.B0
MOV #<D.CART!D.DOOR!D.BRHM!D.HDHM>,E.A0
MOV #1,E.B1

JSR PC,CHKMSG ;:CHECK MSGS A0, B0, A1, B1
;: & MSGS SPECIFIED HERE
WORD 0!0!0
ERROR 35 ;:MSG A0 ERROR AFTER AC SWITCH OFF FROM HEADS HOME
ERROR 203 ;:MSG B0 ERROR
ERROR 36 ;:MSG A1 ERROR

```

H06

```

3679 015464 104037          ERROR 37          :MSG B1 ERROR
3680
3681 015466 013737 001440 005444 3$: MOV T5000,TEMP1
3682 015474 012765 100000 000000 4$: MOV #CCLR,RKCS1(R5)
3683 015502 004737 026420          JSR PC,GSTAT
3684 015506 032765 010000 000010 BIT #NED,RKCS2(R5)
3685 015514 001004          BNE 5$
3686 015516 005337 005444          DEC TEMP1
3687 015522 001364          BNE 4$
3688 015524 104150          ERROR 150          :NED NOT SET BEFORE TIMEOUT
3689
3690 015526 104401 042666          5$: TYPE .MSG50          :SWITCH AC BACK ON
3691 015532 104401 045222          TYPE .MSG70          :VERIFY HEADS LOAD
3692 015536 104401 045145          TYPE .MSG69          :PRESS SPACE AFTER READY ON
3693 015542 004737 031006          JSR PC,GETSP          :GET SPACE
3694
3695 015546 004737 026772          JSR PC,SUBCLR
3696 015552 104024          ERROR 24          :CERR AFTER SCLR
3697
3698 015554 032737 000200 005434 BIT #D.DRDY,HMR2          :SEE IF LOADED
3699 015562 001001          BNE 6$          :BR IN YES
3700 015564 104141          ERROR 141          :DRY NOT RDY AFTER AC UP
3701
3702 015566 005037 005370          6$: CLR UNLD          :FOR VALID HALT
3703 015572 032737 000100 005436 BIT #D.ACLO,HMR3
3704 015600 001401          BEQ 7$
3705 015602 104152          ERROR 152          :ACLO NOT RESET AFTER POWER UP
3706
3707 015604 032737 000100 005434 7$: BIT #D.VV,HMR2
3708 015612 001401          BEQ 8$
3709 015614 104153          ERROR 153          :VV NOT RESET AFTER POWER UP
3710
3711 015616          8$:
3712 015616 012765 100000 000000 MOV #CCLR,RKCS1(R5)
3713 015624 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE #
3714 015632 012765 000003 000000 MOV #PACK,RKCS1(R5) ;PACK CMD
3715 015640 013737 001426 005444 MOV T10,TEMP1
3716 015646 004737 025062          JSR PC,FRDY          :FIND CONTR RDY
3717 015652 104116          ERROR 116          :CONTR NOT RDY
3718
3719 015654 032737 000100 005434 BIT #D.VV,HMR2
3720 015662 001001          BNE 64$
3721 015664 104027          ERROR 27          :VOLUME VALID NOT SET AFTER PACK CMD
3722 015666          64$:
3723 015666          10$:
3724
3725 *****
3726 *TEST 15 CHECK NXF LOGIC
3727 *
3728 * THIS TEST VERIFIES NON-EXECUTABLE FUNCTION (NXF) IS DETECTED
3729 * AS A RESULT OF DOING A SEEK WITH VOLUME VALID RESET.
3730 *
3731 *****
3732 015666 000004          TST15: SCOPE
3733 015670 012737 000001 001174 MOV #1,$TIMES          ;;DO 1 ITERATION
3734 015676 012706 001100          MOV #STACK,SP          ;RESTORE STACK PTR

```


3735									
3736	015702	004737	026772			JSR	PC, SUBCLR		
3737	015706	104024				ERROR	24		;CERR AFTER SCLR
3738									
3739	015710	032737	010000	005434		BIT	#D.SPIN,HMR2		;SEE IF SPINDLE ALREADY ON
3740	015716	001021				BNE	64\$;BR IF YES
3741	015720	104401	040524			TYPE	,MSG29		;PLEASE WAIT. HEADS BEING LOADED
3742									
3743	015724	012765	000011	003000		MOV	#SRTSPL,RKCS1(R5)		;START SPINDLE CMD
3744	015732	013737	001426	005444		MOV	T10,TEMP1		
3745	015740	004737	025062			JSR	PC,FRDY		;FIND CONTR RDY
3746	015744	104143				ERROR	143		;CONTR RDY NOT SET AFTER CMD
3747									
3748	015746	013737	001434	005446		MOV	T100,TEMP2		
3749	015754	004737	025376			JSR	PC,FATT1		;FIND ATTN
3750	015760	104144				ERROR	144		;NO ATTN AFTER CMD
3751	015762				64\$:				
3752	015762	005037	005370			CLR	UNLD		;FOR VALID HALT
3753									
3754	015766	004737	026772			JSR	PC, SUBCLR		
3755	015772	104024				ERROR	24		;CERR AFTER SCLR
3756									
3757	015774	104401	040173			TYPE	,MSG22		;NXF TEST
3758	016000	104401	043041			TYPE	,MSG52		;PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
3759	016004	004737	030746			JSR	PC,CCSP		;INPUT CONT-E OR SPACE
3760	016010	000137	016332			JMP	7\$;RETURN HERE FOR CONT-E
3761									
3762	016014	104401	042720			TYPE	,MSG51		;REMOVE UNIT SELECT PLUG TO CLEAR VV
3763	016020	104401	041265			TYPE	,MSG37		;PRESS SPACE WHEN DONE
3764	016024	004737	031006			JSR	PC,GETSP		;GET SPACE
3765									
3766	016030	004737	026420			JSR	PC,GSTAT		
3767	016034	032737	000100	005434		BIT	#D.VV,HMR2		
3768	016042	001403				BEQ	1\$		
3769	016044	104177				ERROR	177		;VV NOT CLEARED AFTER UNIT SEL PLUG
3770	016046	000137	016332			JMP	7\$;EXIT TEST
3771									
3772	016052	032737	000100	005434	1\$:	BIT	#D.VV,HMR2		
3773	016060	001406				BEQ	2\$		
3774	016062	104155				ERROR	155		;VV SET AFTER HEADS LOADED
3775	016064	000137	016332			JMP	7\$;EXIT TEST
3776									
3777	016070	004737	026772			JSR	PC, SUBCLR		
3778	016074	104024				ERROR	24		;CERR AFTER SCLR
3779									
3780	016076	012765	000001	000020	2\$:	MOV	#1,RKDC(R5)		
3781	016104	012765	000017	000000		MOV	#SEEK,RKCS1(R5)		;SEEK CMD
3782	016112	013737	001426	005444		MOV	T10,TEMP1		
3783	016120	004737	025062			JSR	PC,FRDY		;FIND RDY
3784	016124	104131				ERROR	131		;NO RDY AFTER SEEK CMD
3785									
3786	016126	013737	001440	005444		MOV	T50000,TEMP1		
3787	016134	004737	025472			JSR	PC,FATT2		;FIND ATTN
3788	016140	104132				ERROR	132		;NO ATTN AFTER SEEK CMD
3789									
3790	016142	032737	000400	005436		BIT	#D.NXF,HMR3		

```

3791 016150 001003          BNE      3$
3792 016152 104156          ERROR   156      ;NXF NOT SET AFTER SEEK WITH VV=0
3793 016154 000137 016332   JMP      7$      ;EXIT TEST
3794
3795 016160 032737 000200 005436 3$:   BIT      #D.FLT,HMR3
3796 016166 001003          BNE      4$
3797 016170 104172          ERROR   172      ;NXF DID NOT SET FAULT
3798 016172 000137 016332   JMP      7$      ;EXIT TEST
3799
3800 016176 012737 050240 005476 4$:   MOV      #<D.DSC!D.SPIN!D.DRDY!D.DRA>,E.A0 ;EXPECTED A0
3801 016204 012737 000600 005500   MOV      #<D.NXF!D.FLT>,E.B0
3802 016212 012737 001720 005502   MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
3803 016220 012737 000001 005504   MOV      #1,E.B1
3804
3805 016226 004737 025604          JSR      PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
3806 016232 000003          .WORD   T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
3807 016234 104160          ERROR   160      ;MSG A0 ERROR AFTER SEEK WITH VV=0
3808 016236 104161          ERROR   161      ;MSG B0 ERROR
3809 016240 104162          ERROR   162      ;MSG A1 ERROR
3810 016242 104163          ERROR   163      ;MSG B1 ERROR
3811 016244 005737 001400          TST      CYLADD
3812 016250 001401          BEQ      5$
3813 016252 104157          ERROR   157      ;HEADS MOVED WITH SEEK & DXF
3814
3815 016254 004737 026772          JSR      PC,SUBCLR 5$:
3816 016260 104024          ERROR   24      ;CERR AFTER SCLR
3817
3818 016262 012765 100000 000000   MOV      #CCLR,RKCS1(R5)
3819 016270 013765 001222 000010   MOV      $UNIT,RKCS2(R5) ;DRIVE #
3820 016276 012765 000003 000000   MOV      #PACK,RKCS1(R5) ;PACK CMD
3821 016304 013737 001426 005444   MOV      T10,TEMP1
3822 016312 004737 025062          JSR      PC,FRDY ;FIND CONTR RDY
3823 016316 104116          ERROR   116      ;CONTR NOT RDY
3824
3825 016320 032737 000100 005434   BIT      #D.VV,HMR2
3826 016326 001001          BNE      65$
3827 016330 104027          ERROR   27      ;VOLUME VALID NOT SET AFTER PACK CMD
3828
3829 016332          65$:
3830          7$:

```

```

3831
3832 ;*****
3833 ;TEST 16 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL *
3834 ;
3835 ; THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
3836 ; THIS AREA CONTAINS BAD SECTOR INFOR WHICH IS WRITTEN BY THE
3837 ; FACTORY DURING MANF. ALL BAD SECTOR INFOR (BSE) WILL BE STORED
3838 ; AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
3839 ;
3840 ; SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
3841 ; SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
3842 ;
3843 ; IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
3844 ; IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
3845 ; A MESSAGE WILL BE TYPED INDICATING THAT ALL
3846 ; FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
3847 ; THIS IS DONE SO AS NOT TO DESTROY BSE INFOR OR AN ALIGNMENT PACK BY WRITING

```

K06

```

3847
3848
3849
3850
3851 016332 000004
3852 016334 012737 000001 001174
3853 016342 012706 001100
3854
3855 016346 004737 026772
3856 016352 104024
3857 016354 005037 005446
3858 016360 005037 005450
3859
3860
3861 016364 012737 002370 005452
3862 016372 013765 005452 000004
3863 016400 012737 001000 005454
3864 016406 013765 005454 000006
3865
3866 016414 012765 000632 000020 15:
3867 016422 012765 177400 000002
3868 016430 012765 000021 000000
3869 016436 013737 001440 005444
3870 016444 004737 025062
3871 016450 104226
3872 016452 004737 026420
3873 016456 032737 100000 005406
3874 016464 001470
3875 016466 104227
3876
3877 016470 012737 010340 005476
3878 016476 005037 005500
3879 016502 012737 001720 005502
3880 016510 012737 000001 005504
3881 016516 005037 005506
3882 016522 012737 000002 005510
3883 016530 012737 000003 005514
3884
3885 016536 004737 025604
3886 016542 000000
3887 016544 104054
3888 016546 104026
3889 016550 104056
3890 016552 104030
3891 016554 004737 026772
3892 016560 104024
3893 016562 005237 005446
3894 016566 023727 005446 000005
3895 016574 001007
3896 016576 005737 005450
3897 016602 001002
3898 016604 104165
3899 016606 000414
3900 016610 104167
3901 016612 000412
3902

; *
; * THE PACK SERIAL * IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.
; *
; *****
15: SCOPE
MOV #1,STIMES ;:DO 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)

15: MOV #410.,RKDC(R5) ;:CYL 410
MOV #-256.,RKWC(R5) ;:LOAD WORD CT
MOV #RDDATA,RKCS1(R5) ;:READ DATA COMMAND
MOV T5000,TEMP1 ;:SETUP TIMEOUT
JSR PC,FRDY ;:FIND RDY
ERROR 226 ;:NO RDY AFTER READ DATA CMD
JSR PC,GSTAT ;:GET FRESH DATA
BIT #CERR,HCS1
BEQ 85
ERROR 227 ;:CERR AFTER READ DATA CMD

MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;:EXPECTED MSG A0
CLR E.B0 ;:EXPECTED MSG B0
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;:EXPECTED A1
MOV #1,E.B1 ;:MSG ID FOR EXPECTED MSG B1
CLR E.A2 ;:EXPECTED MSG A2
MOV #2,E.B2 ;:MSG ID FOR EXPECTED MSG B2
MOV #3,E.B3 ;:MSG ID FOR EXPECTED MSG B3

JSR PC,CHKMSG ;:CHECK MSGS A0, B0, A1, B1
;: & MSGS SPECIFIED HERE
;WORD 0!0!0
ERROR 54 ;:MSG A0 ERROR AFTER READ DATA CMD
ERROR 26 ;:MSG B0 ERROR
ERROR 56 ;:MSG A1 ERROR
ERROR 30 ;:MSG B1 ERROR

JSR PC,SUBCLR
ERROR 24 ;:CERR AFTER SUBCLR
INC TEMP2
CMP TEMP2,#5 ;:READ ALL 5 SECTORS?
BNE 55
TST TEMP3
BNE 25
ERROR 165 ;:CANT READ SECTORS 0,2,4,6,8
BR 35
25: ERROR 167 ;:CANT READ SECTORS 10,12,14,16,18,20
BR 35

```

```

3903 016614 013765 005452 000004 5$: MOV TEMP4,RKBA(R5) ;RESTORE TABLE ADDR
3904 016622 062737 000002 005454 ADD #2,TEMP5 ;SETUP TO READ 2 SECTORS FROM LAST
3905 016630 013765 005454 000006 MOV TEMP5,RKDA(R5)
3906 016636 000666 BR 1$
3907
3908 016640 005237 001542 3$: INC BSERR ;SET BSE FLAG
3909 016644 000504 BR TST17 ;GO TO NEXT TEST
3910 016646 005737 002376 8$: TST BSE22H+6 ;TEST CARTRIDGE TYPE
3911 016652 001405 BEQ 9$ ;BRANCH IF DATA CARTRIDGE
3912 016654 104401 043402 TYPE ,MSG56 ;ALIGNMENT CARTRIDGE USED
3913 016660 005237 001542 INC BSERR ;SET BSE ERROR FLAG
3914 016664 000426 BR 10$
3915
3916 016666 005237 005450 9$: INC TEMP3
3917 016672 023727 005450 000001 CMP TEMP3,#1
3918 016700 001020 BNE 10$
3919 016702 005037 005446 CLR TEMP2
3920 016706 012737 003370 005452 MOV #BSE22S,TEMP4 ;STORE 22 SECTOR SOFTWARE BSE ADDR
3921 016714 013765 005452 000004 MOV TEMP4,RKBA(R5)
3922 016722 012737 001012 005454 MOV #1012,TEMP5 ;TRACK 2, SECTOR 12
3923 016730 013765 005454 000006 MOV TEMP5,RKDA(R5)
3924 016736 000137 016414 JMP 1$ ;REPEAT
3925
3926 016742 005737 001216 10$: TST $PASS
3927
3928
3929 016746 001014 BNE 13$ ;TYPE CART # ONLY ON 1'ST PASS
3930 016750 104401 037766 TYPE ,MSG17 ;CART SERIAL #
3931 016754 012746 002370 MOV #BSE22H,-(SP)
3932 016760 004737 035312 JSR PC,$DB20 ;CONVERT DBL BINARY WORD TO OCTAL
3933 016764 004737 035662 JSR PC,$SUPRS ;TYPE SERIAL #
3934 016770 104401 001205 TYPE ,SCLRF
3935 016774 104401 001205 TYPE ,SCLRF
3936
3937 017000 004737 026772 13$: JSR PC,SUBCLR
3938 017004 104024 ERROR 24 ;CERR AFTER SCLR
3939 ;RETURN TO CYL 0
3940
3941 017006 012765 000017 000000 MOV #SEEK,RKCSI(R5) ;SEEK CMD
3942 017014 013737 001430 005444 MOV T50,TEMP1 ;SETUP TIMEOUT
3943 017022 004737 025062 JSR PC,FRDY ;FIND RDY
3944 017026 104131 ERROR 131 ;NO RDY AFTER SEEK CMD
3945
3946 017030 013737 001440 005444 MOV T50000,TEMP1 ;SETUP TIMEOUT
3947 017036 004737 025472 JSR PC,FATT2 ;FIND ATTN
3948 017042 104132 ERROR 132 ;NO ATTN AFTER SEEK CMD
3949
3950 017044 032737 100000 005406 BIT #CERR,HCS1
3951 017052 001401 BEQ 64$
3952 017054 104210 ERROR 210 ;CERR AFTER SEEK CMD
3953
3954 017056 64$:
3955
3956
3957
3958

```

; *TEST 17 WRITE LOCK TEST

3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014

017056 000004
017060 012737 000001 001174
017066 012706 001100
017072 005737 001542
017076 001402
017100 000137 022016
017104 004737 026772
017110 104024
017112 104401 040341
017116 104401 043041
017122 004737 030746
017126 000137 022016
017132 004737 026420
017136 032737 004000 005434
017144 001416
017146 104401 043142
017152 104401 041265
017156 004737 031006
017162 004737 026420
017166 032737 004000 005434
017174 001402
017176 104110
017200 000762
017202 005037 001366
017206 005037 001402
017212 005037 001510
017216 005037 001516
017222 004737 027754
017226 012765 001554 000004
017234 012765 177676 000002
017242 000337 001510
017246 013765 001510 000006
017254 000337 001510
017260 012765 000027 000000
017266 013737 001440 005444
017274 004737 025062
017300 104200
017302 004737 026420
017306 032737 100000 005406
017314 001401

```

: *
: * 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
: *
: *****
TST17: SCOPE
MOV #1, $TIMES ; DO 1 ITERATION
MOV #STACK, SP ; RESTORE STACK PTR
TST BSERR ; SEE IF ALIGN CRT
BEQ 15$ ; BR IF NO
JMP 12$ ; ELSE EXIT TEST

15$: JSR PC, SUBCLR ; CERR AFTER SCLR
ERROR 24

TYPE ,MSG24 ; WRITE LOCK TEST
TYPE ,MSG52 ; PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
JSR PC, CCSP ; INPUT CONT-E OR SPACE
JMP 12$ ; RETURN HERE IF CONT-E

JSR PC, GSTAT
BIT #D.WRL, HMR2 ; SEE IF WRITE LOCK IS ON
BEQ 2$ ; BR IF NO
1$: TYPE ,MSG53 ; DISABLE WRITE LOCK
TYPE ,MSG37 ; TYPE SPACE WHEN DONE
JSR PC, GETSP ; GET SPACE

JSR PC, GSTAT
BIT #D.WRL, HMR2 ; SEE IF WRITE LOCK IS OFF
BEQ 2$
ERROR 110 ; WRITE LOCK NOT DISABLED
BR 1$

2$: CLR TOCYL ; SETUP
CLR CALADD ; FOR
CLR HEAD ; FILL HEADER TABLE
CLR FORMAT ; ROUTINE

16$: JSR PC, FHDTAB ; BUILD STD 22 SECTOR HEADER TABLE

MOV #HDTAB, RKBA(R5)
MOV #-66., RKWC(R5)
SWAB HEAD ; HEAD ADDR
MOV HEAD, RKDA(R5)
SWAB HEAD

MOV #<WRHEAD>, RKCS1(R5) ; WRITE HEADER CMD
MOV T5000, TEMP1 ; SETUP TIMEOUT
JSR PC, FRDY ; FIND RDY
ERROR 200 ; NO RDY AFTER WRITE HEADER CMD
JSR PC, GSTAT ; GET FRESH STATUS
BIT #CERR, HCS1
BEQ 64$

```

```

4015 017316 104201          ERROR 201          ;CERR AFTER WRITE HEADER CMD
4016 017320          64$:
4017
4018
4019 017320 005237 001510          INC HEAD
4020 017324 023727 001510 000003    CMP HEAD,#3          ;SEE IF ALL HEADS DONE
4021 017332 001333          BNE 16$             ;BR IF NO
4022
4023 017334 004737 026772          JSR PC,SUBCLR
4024 017340 104024          ERROR 24           ;CERR AFTER SCLR
4025
4026 017342 005037 001422          CLR SECTOR
4027 017346 013765 001422 000006    14$: MOV SECTOR,RKDA(R5)
4028
4029 017354 012765 001524 000004    MOV #DATA0,RKBA(R5) ;WRITE ALL 0'S
4030 017362 052765 000020 000010    BIS #BAI,RKCS2(R5)
4031 017370 012765 177400 000002    MOV #-256.,RKWC(R5) ;CYL 0, TRK 0, SECTOR 0
4032
4033 017376 012765 000023 000000    MOV #<WRDATA>,RKCS1(R5) ;WRITE DATA CMD
4034 017404 013737 001440 005444    MOV T5000,TEMP1     ;SETUP TIMEOUT
4035 017412 004737 025062          JSR PC,FRDY         ;FIND RDY
4036 017416 104011          ERROR 11           ;NO RDY AFTER WRITE DATA CMD
4037 017420 004737 026420          JSR PC,GSTAT        ;GET FRESH STATUS
4038 017424 032737 100000 005406    BIT #CERR,HCS1
4039 017432 001465          BEQ 68$             ;BR IF NO ERRORS
4040
4041 017434 032737 000200 005422    BIT #BSE,HER        ;SEE IF BAD SECTOR FLAG
4042 017442 001421          BEQ 66$             ;BR IF NO
4043 017444 004737 030412          JSR PC,TRUERR       ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4044 017450 000455          BR 67$              ;RETURN HERE IF NO
4045
4046 017452 005237 001422          INC SECTOR          ;RETURN HERE IF YES
4047 017456 023727 001422 000012    CMP SECTOR,#10.     ;ARE 10 CONSEC. SECTORS BAD
4048 017464 001003          BNE 65$             ;BR IF NO
4049 017466 104040          ERROR 40           ;ABORTING TEST DETECTED 10 BAD SECTORS
4050 017470 000137 022016          JMP 12$             ;BYPASS TEST
4051
4052 017474 012765 100000 000000    65$: MOV #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
4053 017502 000137 017346          JMP 14$
4054
4055 017506 104012          66$: ERROR 12           ;CERR WITH WRITE DATA CMD
4056
4057 017510 012737 010340 005476    MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4058 017516 005037 005500          CLR E.B0           ;EXPECTED MSG B0
4059 017522 012737 001720 005502    MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4060 017530 012737 000001 005504    MOV #1,E.B1        ;MSG ID FOR EXPECTED MSG B1
4061 017536 005037 005506          CLR E.A2           ;EXPECTED MSG A2
4062 017542 012737 000002 005510    MOV #2,E.B2        ;MSG ID FOR EXPECTED MSG B2
4063 017550 012737 000003 005514    MOV #3,E.B3        ;MSG ID FOR EXPECTED MSG B3
4064
4065 017556 004737 025604          JSR PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4066 017562 000003          .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4067 017564 104052          ERROR 52           ;MSG A0 ERROR AFTER WRITE DATA CMD
4068 017566 104023          ERROR 23           ;MSG B0 ERROR
4069 017570 104053          ERROR 53           ;MSG A1 ERROR
4070 017572 104025          ERROR 25           ;MSG B1 ERROR

```

4071	017574	104401	045333		TYPE	MSG72	:ABORTING BALANCE OF TESTS
4072	017600	000137	024402		JMP	\$EOP	
4073	017604	104047		678:	ERROR	47	:BAD SECTOR NOT LISTED IN TABLE
4074	017606			688:			
4076	017606	004737	026772		JSR	PC,SUBCLR	
4077	017612	104024			ERROR	24	:CERR AFTER SCLR
4078							
4079	017614	012765	001524	000004	MOV	#DATA0,RKBA(R5)	
4080	017622	052765	000020	000010	BIS	#BA1,RKCS2(R5)	
4081	017630	012765	177400	000002	MOV	#-256,RKWC(R5)	
4082	017636	013737	001524	001504	MOV	DATA0,W02	:EXPECTED WORD FOR TRUEXP TYPEOUT
4083	017644	013765	001422	000006	MOV	SECTOR,RKDA(R5)	
4084							
4085	017652	012765	000031	000000	MOV	#(WRTCHK),RKCS1(R5)	:WRITE CHECK CMD
4086	017660	013737	001440	005444	MOV	T5000,TEMP1	:SETUP TIMEOUT
4087	017666	004737	025062		JSR	PC,FRDY	:FIND RDY
4088	017672	104015			ERROR	15	:NO RDY AFTER WRITE CHECK CMD
4089	017674	004737	026420		JSR	PC,GSTAT	:GET FRESH STATUS
4090	017700	032737	100000	005406	BIT	#CERR,HCS1	
4091	017706	001450			BEQ	705	
4092	017710	032737	040000	005410	BIT	#WCE,HCS2	:SEE IF WRITE CHECK ERROR
4093	017716	001405			BEQ	695	
4094	017720	016537	000024	001502	MOV	RKDB(R5),W01	:ACTUAL WORD FOR PRINTOUT
4095	017726	104016			ERROR	16	:WCE AFTER WRITE CMD
4096	017730	000437			BR	705	
4097							
4098	017732	104022		695:	ERROR	22	:CERR AFTER WRITE CHECK CMD
4099							
4100	017734	012737	010340	005476	MOV	#(D.D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	:EXPECTED MSG A0
4101	017742	005037	005500		CLR	E.B0	:EXPECTED MSG B0
4102	017746	012737	001720	005502	MOV	#(D.SP0K!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	:EXPECTED A1
4103	017754	012737	000001	005504	MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1
4104	017762	005037	005506		CLR	E.A2	:EXPECTED MSG A2
4105	017766	012737	000002	005510	MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2
4106	017774	012737	000003	005514	MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3
4107							
4108	020002	004737	025604		JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1
4109	020006	000003			.WORD	T.A2!T.B2!0	:& MSGS SPECIFIED HERE
4110	020010	104057			ERROR	57	:MSG A0 ERROR AFTER WRITE CHECK CMD
4111	020012	104031			ERROR	31	:MSG B0 ERROR
4112	020014	104060			ERROR	60	:MSG A1 ERROR
4113	020016	104032			ERROR	32	:MSG B1 ERROR
4114	020020	104401	045333		TYPE	MSG72	:ABORTING BALANCE OF TESTS
4115	020024	000137	024006		JMP	ENDRV	
4116							
4117	020030			705:			
4118							
4119	020030	004737	026772	35:	JSR	PC,SUBCLR	
4120	020034	104024			ERROR	24	:CERR AFTER SCLR
4121							
4122	020036	104401	043233		TYPE	MSG54	:ENABLE WRITE LOCK
4123	020042	104401	041265		TYPE	MSG37	:PRESS SPACE WHEN DONE
4124	020046	004737	031006		JSR	PC,GETSP	:GET SPACE
4125							
4126	020052	012765	100000	000000	MOV	#CLR,RKCS1(R5)	

```

4:02 020060 004737 026420 JSR PC,GSTAT
4:03 020064 032737 004000 005434 BIT #D.WAL,HMR2 ;SEE IF WRITE LOCK IS ON
4:04 020072 001002 BNE 45
4:05 020074 104115 ERROR 115 ;WRITE LOCK NOT ENABLED
4:06 020076 000754 BR 35
4:07 020100 012765 001530 000004 45: MOV #DATA1,RKBA(R5) ;ATTEMPT TO WRITE ALL 1'S WITH WRITE LOCK SET
4:08 020106 052765 000020 000010 BIS #BA1,RKCS2(R5)
4:09 020114 012765 177400 000002 MOV #256,RKWC(R5) ;CYLINDER 0, HEAD 0
4:10 020122 013765 001422 000006 MOV SECTOR,RKDA(R5)
4:11 020130 012765 000023 000000 MOV #WRDATA,RKCS1(R5)
4:12 020136 013737 001440 005444 MOV T5000,TEMP1 ;SETUP TIMEOUT
4:13 020144 004737 025062 JSR PC,FRDY ;FIND RDY
4:14 020150 104116 ERROR 116 ;CONTR NOT RDY
4:15 020152 004737 026420 JSR PC,GSTAT ;GET FRESH STATUS
4:16 020156 032737 100000 005406 BIT #CERR,HCS1
4:17 020164 001001 BNE 175
4:18 020166 104207 ERROR 207 ;CERR NOT SET AFTER WRITE WITH WAL SET
4:19 020170 032737 004000 005436 175: BIT #D.WLE,HMR3 ;CHECK WRITE LOCK ERROR SET
4:20 020176 001001 BNE 55 ;BR IF SET
4:21 020200 104121 ERROR 121 ;WRITE LOCK ERROR NOT SET
4:22 020202 012737 054340 005476 55: MOV #<D.DSC!D.SPIN!D.WAL!D.DRDY!D.VV!D.DRA>,E.A0 ;EXP A0
4:23 020210 012737 004200 005500 MOV #<D.WLE!D.FLT>,E.B0
4:24 020216 012737 001720 005502 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4:25 020224 012737 000301 005504 MOV #1,E.B1
4:26 020232 004737 025604 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4:27 020236 000000 .WORD 0:0:0 ;& MSGS SPECIFIED HERE
4:28 020240 104123 ERROR 123 ;MSG A0 ERROR AFTER WRITE WITH WRITE LOCK SET
4:29 020242 104125 ERROR 125 ;MSG B0 ERROR
4:30 020244 104126 ERROR 126 ;MSG A1 ERROR
4:31 020246 104127 ERROR 127 ;MSG B1 ERROR
4:32 020250 004737 026772 JSR PC,SUBCLR
4:33 020254 104024 ERROR 24 ;CERR AFTER SCLR
4:34 020256 012765 001524 000004 MOV #DATA0,RKBA(R5) ;CHECK THAT NONE OF ORIG DATA
4:35 020264 052765 000020 000010 BIS #BA1,RKCS2(R5) ;HAS CHANGED
4:36 020272 012765 177400 000002 MOV #256,RKWC(R5) ;AS RESULT OF WRITE WITH WRITE LOCK SET
4:37 020300 013737 001524 001504 MOV DATA0,WD2 ;EXPECTED WORD FOR TRJERR TYPEOUT

```



```

4169 020306 013765 001422 000006      MOV      SECTOR,RKDA(R5)
4170
4171 020314 012765 000031 000000      MOV      #(<WRKCHK>,RKCS1(R5))      ;WRITE CHECK CMD
4172 020322 013737 001440 005444      MOV      T5000,TEMP!      ;SETUP TIMEOUT
4173 020330 004737 025062      JSR      PC,FRDY      ;FIND RDY
4174 020334 104015      ERROR   15      ;NO RDY AFTER WRITE CHECK CMD
4175 020336 004737 026420      JSR      PC,GSTAT      ;GET FRESH STATUS
4176 020342 032737 100000 005406      BIT      #CERR,HCS1
4177 020350 001450      BEQ     72$
4178 020352 032737 040000 005410      BIT      #WCE,HCS2      ;SEE IF WRITE CHECK ERROR
4179 020360 001405      BEQ     71$
4180 020362 016537 000024 001502      MOV      RKDB(R5),WD1      ;ACTUAL WORD FOR PRINTOUT
4181 020370 104016      ERROR   16      ;WCE AFTER WRITE CMD
4182 020372 000437      BR      72$
4183
4184 020374 104022      71$:  ERROR   22      ;CERR AFTER WRITE CHECK CMD
4185
4186 020376 012737 010340 005476      MOV      #(<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0) ;EXPECTED MSG A0
4187 020404 005037 005500      CLR     E.B0      ;EXPECTED MSG B0
4188 020410 012737 001720 005502      MOV      #(<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1) ;EXPECTED A1
4189 020416 012737 000001 005504      MOV      #1,E.B1      ;MSG ID FOR EXPECTED MSG B1
4190 020424 005037 005506      CLR     E.A2      ;EXPECTED MSG A2
4191 020430 012737 000002 005510      MOV      #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
4192 020436 012737 000003 005514      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
4193
4194 020444 004737 025604      JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
4195 020450 000003      .WORD   T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
4196 020452 104057      ERROR   57      ;MSG A0 ERROR AFTER WRITE CHECK CMD
4197 020454 104031      ERROR   31      ;MSG B0 ERROR
4198 020456 104060      ERROR   60      ;MSG A1 ERROR
4199 020460 104032      ERROR   32      ;MSG B1 ERROR
4200 020462 104401 045333      TYPE    MSG72      ;ABORTING BALANCE OF TESTS
4201 020466 000137 024006      JMP     ENDRV
4202
4203 020472      72$:
4204 020472 104401 044506      TYPE    ,MSG64      ;FOLLOWING TEST TEMPORARY BYPASSED
4205
4206      ;THE CHANGE OF WRL LOGIC RESULTS IN A DSC AND ATTN TO THE CONTROLLER
4207      ;WHICH RECOGNIZES IT AS AN AN ERROR
4208      ;THEREFORE, IN CONTINUOUS WRITING RKDA WILL ALWAYS = 1 WHEN THE
4209      ;WRL SWITCH IS ENABLED.
4210      ;THE FOLLOWING CODE WILL BE VALID IF THE PROPOSED ECO IS APPROVED TO
4211      ;ELIMINATE THE CHANGE OF WRL LOGIC
4212
4213 020476 000137 022016      JMP     12$      ;EXIT TEST
4214
4215 020502 004737 026772      6$:  JSR      PC,SUBCLR
4216 020506 104024      ERROR   24      ;CERR AFTER SCLR
4217
4218 020510 104401 043142      TYPE    ,MSG53      ;DISABLE WRITE LOCK
4219 020514 104401 041265      TYPE    MSG37      ;SPACE BAR WHEN DONE
4220 020520 004737 031006      JSR      PC,GETSP      ;GET SPACE
4221 020524 004737 026420      JSR      PC,GSTAT
4222 020530 032737 004000 005434      BIT      #D.WRL,HMR2      ;SEE IF WRITE LOCK OFF
4223 020536 001361      BNE     6$
4224
;TEST FOR WRITE LOCK AT SECTOR BOUNDRIES

```

E07

JNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
0296JC.F11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 82
T17 WRITE LOCK TEST

SEQ 0092

```

4225          ;FOR CONTINUOUS WRITING ON CYL 0, TRACK 0,1,2
4226 020540 104401 043233          TYPE      MSG54          ;SET WRITE LOCK
4227 020544 013737 001436 001160      MOV      T5000,STMP0
4228 020552 005037 001162          CLR      STMP1          ;BIT0=0:WRITE 0'S; BIT0=1:WRITE 1'S
4229 020556 005037 001366          CLR      TOCYL
4230
4231 020562 004737 026772          JSR      PC,SUBCLR      7$:
4232 020566 104024 24          ERROR    24          ;CERR AFTER SCLR
4233
4234 020570 032737 000001 001162      BIT      #BIT0,STMP1
4235 020576 001004 85          BNE      85          ;BR IF WRITING 1'S
4236 020600 012765 001524 000004      MOV      #DATA0,RKBA(R5) ;SETUP ALL 0'S
4237 020606 000403 95          BR
4238
4239 020610 012765 001530 000004 85:      MOV      #DATA1,RKBA(R5) ;SETUP ALL 1'S
4240 020616 052765 000020 000010 95:      BIS      #BA1,RKCS2(R5)
4241 020624 012765 140400 000002      MOV      #-63,*256.,RKWC(R5) ;WRITE TRACK 0,1,2 63 SECTORS
4242 020632 005065 000006          CLR      RKDA(R5)      ;BEGIN AT TRACK AND SEC 0
4243 020636 013765 001366 000020      MOV      TOCYL,RKDC(R5)
4244 020644 012765 000023 000000      MOV      #WRDATA,RKCS1(R5)
4245 020652 013737 001440 005444      MOV      T5000,TEMP1
4246 020660 004737 025062          JSR      PC,FRDY
4247 020664 104011 11          ERROR    11          ;FIND CONTR RDY
4248          ;CONTR NOT RDY
4249 020666 032737 100000 005406      BIT      #CERR,HCS1
4250 020674 001017 135          BNE      135
4251 020676 005337 001160          DEC      STMP0
4252 020702 001011 105          BNE      105
4253 020704 104204 204          ERROR    204          ;CERR NOT SET BY TIMEOUT
4254 020706 104401 043142          TYPE      ,MSG53          ;DISABLE WRITE LOCK
4255 020712 104401 041265          TYPE      ,MSG37          ;PRESS SPACE WHEN DONE
4256 020716 004737 031006          JSR      PC,GETSP      ;INPUT SPACE
4257 020722 000137 022016 125          JMP      125          ;EXIT TEST
4258
4259 020726 005237 001162          10$:      INC      STMP1
4260 020732 000713 75          BR
4261 020734 032737 000200 005422 13$:      BIT      #BSE,HER          ;GC WRITE OPPOSITE DATA
4262 020742 001411 215          BEQ      215          ;SEE IF BAD SECTOR
4263 020744 005237 001366          INC      TOCYL          ;BR IF NO
4264 020750 023727 001366 000012      CMP      TOCYL,#10.      ;ELSE TRY ANOTHER CYL
4265 020756 001301 75          BNE      75          ;SEE IF 10 CONSEC CYL BAD
4266 020760 104062 62          ERROR    62          ;BR IF NO & DO AGAIN
4267 020762 000137 022016 125          JMP      125          ;10 BAD CONSEC CYLINDERS
4268          ;EXIT TEST
4269 020766 032737 004000 005436 21$:      BIT      #D.WLE,HMR3
4270 020774 001001 115          BNE      115
4271 020776 104205 205          ERROR    205          ;NO WRL BY TIMEOUT
4272
4273 021000 104401 043142          11$:      TYPE      ,MSG53          ;DISABLE WRITE LOCK
4274 021004 104401 041265          TYPE      ,MSG37          ;TYPE SPACE WHEN DONE
4275 021010 004737 031006          JSR      PC,GETSP
4276
4277 021014 013737 005416 001164      MOV      HDA,STMP2      ;STORE RKDA OF ERROR
4278 021022 013737 005416 001166      MOV      HDA,STMP3      ;STORE FOR ERROR TYPEOUT
4279
4280 021030 023727 001164 000001      CMP      STMP2,#1          ;SEE IF WRL ON TRK 0, SEC 1

```

F07

JNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 83
T17 WRITE LOCK TEST

SEG 0093

```

4281 021036 001003 BNE 22$ ;BR IF NO
4282 021040 104401 045531 TYPE MSG100 ;ELSE WRL OCCURRED ON TRK 2, SEC 2:
4283 021044 000616 BR 6$ ;& NO NEW DATA XFER TOOK PLACE
4284
4285 021046 005737 001164 22$: TST $TMP2 ;SEE IF WRL ON TRK 0, SEC 0
4286 021052 001004 BNE 23$ ;BR IF NO
4287 021054 104401 045531 TYPE MSG100 ;ELSE WRL ON TRK 2, SEC 20
4288 021060 000137 020502 JMP 6$ ;& NO NEW DATA XFER TOOK PLACE
4289
4290 021064 023727 001164 001025 23$: CMP $TMP2,#1025 ;SEE IF WRL ON TRK 2, SEC 2:
4291 021072 001004 BNE 24$ ;BR IF NO
4292 021074 104401 045531 TYPE MSG100 ;ELSE WRL ON TRK 2, SEC 19
4293 021100 000137 020502 JMP 6$ ;& NO NEW DATA XFER TOOK PLACE
4294
4295 021104 023727 001164 001024 24$: CMP $TMP2,#1024 ;SEE IF WRL ON TRK 2, SEC 20
4296 021112 001004 BNE 25$ ;BR IF NO
4297 021114 104401 045531 TYPE MSG100 ;ELSE WRL ON TRK 2, SEC 18
4298 021120 000137 020502 JMP 6$ ;& NO OLD DATA TO CHECK AGAINST
4299
4300 021124 023727 001164 000400 25$: CMP $TMP2,#400 ;SEE IF WRL AT TRK 1, SEC 0
4301 021132 001004 BNE 18$ ;BR IF NO
4302 021134 012765 000025 000006 MOV #21.,RKDA(R5) ;ELSE SECTOR AT WRL IS TRK 0, SEC 2:
4303 021142 000415 BR 20$
4304
4305 021144 023727 001164 001000 18$: CMP $TMP2,#1000 ;SEE IF WRL AT TRK 2, SEC 0
4306 021152 001004 BNE 19$ ;BR IF NO
4307 021154 012765 000425 000006 MOV #425,RKDA(R5) ;ELSE SECTOR AT WRL IS TRK 1, SEC 2:
4308 021162 000405 BR 20$
4309
4310 021164 005337 001164 19$: DEC $TMP2 ;GET SECTOR AT WRL
4311 021170 013765 001164 000006 MOV $TMP2,RKDA(R5)
4312
4313 021176 016537 000006 001166 20$: MOV RKDA(R5),$TMP3 ;FOR ERROR PRINTOUT
4314
4315
4316
4317 021204 004737 026772 JSR PC,SUBCLR
4318 021210 104024 ERROR 24 ;CERR AFTER SCLR
4319
4320 021212 005737 001164 TST $TMP2 ;SEE IF TRK/SECTOR 0
4321 021216 001414 BEQ 80$ ;REPEAT,NO NEW DATA XFER TOOK PLACE
4322 021220 023727 001164 001023 CMP $TMP2,#1023 ;SEE IF TRK 2,SECTOR 19
4323 021226 001410 BEQ 80$ ;REPEAT,NO OLD DATA TO CHECK AGAINST
4324 021230 032737 000001 001162 BIT #BIT0,$TMP1
4325 021236 001006 BNE 73$ ;BR IF WRITING 1'S WHEN WLE OCCURRED
4326 021240 012765 001530 000004 MOV #DATA1,RKBA(R5) ;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4327 021246 000405 BR 74$
4328
4329 021250 000137 020502 80$: JMP 6$
4330
4331 021254 012765 001524 000004 73$: MOV #DATA0,RKBA(R5) ;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4332 021262 052765 000020 000010 74$: BIS #BA1,RKCS2(R5)
4333 021270 012765 177400 000002 MOV #-256.,RKWC(R5)
4334 021276 013765 001166 000006 MOV $TMP3,RKDA(R5) ;REFRESH RKDA
4335 021304 017537 000004 001504 MOV @RKBA(R5),WD2 ;EXPECTED WORD FOR TRUERR TYPEOUT
4336 021312 013765 001366 000020 MOV TOCYL,RKDC(R5)

```

```

4337
4338 021320 012765 000031 000000 MOV      #(<WATCH>,RKCS1(R5)) :WRITE CHECK CMD
4339 021326 013737 001440 005444 MOV      T5000,TEMP1      :SETUP TIMEOUT
4340 021334 004737 025062 JSR      PC,FRDY        :FIND RDY
4341 021340 104015 ERROR    15             :NO RDY AFTER WRITE CHECK CMD
4342 021342 004737 026420 JSR      PC,GSTAT      :GET FRESH STATUS
4343 021346 032737 100000 005406 BIT      #CERR,HCS1
4344 021354 001450 BEQ      B2$
4345 021356 032737 040000 005410 BIT      #WCE,HCS2      :SEE IF WRITE CHECK ERROR
4346 021364 001450 BEQ      B1$
4347 021366 016537 000024 001502 MOV      RKDB(R5),WD1   :ACTUAL WORD FOR PRINTOUT
4348 021374 104134 ERROR    134          :WCE AFTER WRITE CMD
4349 021376 000437 BR
4350
4351 021400 104022 B1$: ERROR 22          :CERR AFTER WRITE CHECK CMD
4352
4353 021402 012737 010340 005476 MOV      #(<D.D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0) :EXPECTED MSG A0
4354 021410 005037 005500 CLR      E.B0          :EXPECTED MSG B0
4355 021414 012737 001720 005502 MOV      #(<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1) :EXPECTED A1
4356 021422 012737 000001 005504 MOV      #1,E.B1      :MSG ID FOR EXPECTED MSG B1
4357 021430 005037 005506 CLR      E.A2          :EXPECTED MSG A2
4358 021434 012737 000002 005510 MOV      #2,E.B2      :MSG ID FOR EXPECTED MSG B2
4359 021442 012737 000003 005514 MOV      #3,E.B3      :MSG ID FOR EXPECTED MSG B3
4360
4361 021450 004737 025604 JSR      PC,CHKMSG     :CHECK MSGS A0, B0, A1, B1
4362 021454 000003 .WORD    T.A2!T.B2!0   :8 MSGS SPECIFIED HERE
4363 021456 104057 ERROR    57          :MSG A0 ERROR AFTER WRITE CHECK CMD
4364 021460 104031 ERROR    31          :MSG B0 ERROR
4365 021462 104060 ERROR    60          :MSG A1 ERROR
4366 021464 104032 ERROR    32          :MSG B1 ERROR
4367 021466 104401 045333 TYPE    ,MSG72        :ABORTING BALANCE OF TESTS
4368 021472 000137 024006 JMP
4369
4370 021476 B2$:
4371 021476 000240 NOP
4372 021500 000240 NOP
4373
4374 021502 023727 001164 000400 CMP      $TMP2,#400    :SEE IF WRL AT TRK 1, SECTOR 0
4375 021510 001004 BNE
4376 021512 012765 000025 000006 MOV      #21.,RKDA(R5) :ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21
4377 021520 000415 BR
4378 021522 023727 001164 001000 75$: CMP      $TMP2,#1000  :SEE IF WRL AT TRK 2,SECTOR 0
4379 021530 001004 BNE
4380 021532 012765 000425 000006 MOV      #425.,RKDA(R5) :ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21
4381 021540 000405 BR
4382
4383 021542 005337 001164 76$: DEC      $TMP2        :GET SECTOR BEFORE WRL
4384 021546 013765 001164 000006 MOV      $TMP2,RKDA(R5)
4385 021554 016537 000006 001166 77$: MOV      RKDA(R5),$TMP3 :FOR ERROR PRINTOUT
4386 021562 032737 000001 001162 BIT      #BIT0,$TMP1
4387 021570 001004 BNE      78$
4388 021572 012765 001524 000004 MOV      #DATA0,RKBA(R5) :WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S
4389 021600 000403 BR
4390
4391 021602 012765 001530 000004 78$: MOV      #DATA1,RKBA(R5) :WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S
4392 021610 052765 000020 000010 79$: BIS      #BA1,RKCS2(R5)

```

```

4393 021616 012765 177400 000002      MOV      #256,RKWC(R5)
4394 021624 017537 000004 001504      MOV      @RKBA(R5),WC2 ;EXPECTED WORD FOR TRUERR TYPEOUT
4395 021632 013765 001366 000020      MOV
4396
4397 021640 012765 000031 000000      MOV      @<WRCHK>,RKCS1(R5) ;WRITE CHECK CMD
4398 021646 013737 001440 005444      MOV      T50000,TEMP1 ;SETUP TIMEOUT
4399 021654 004737 025062      JSR      PC,FRDY ;FIND RDY
4400 021660 104015      ERROR    15 ;NO RDY AFTER WRITE CHECK CMD
4401 021662 004737 026420      JSR      PC,GSTAT ;GET FRESH STATUS
4402 021666 032737 100000 005406      BIT      @CERR,HCS1
4403 021674 001450      BEQ     B4$
4404 021676 032737 040000 005410      BIT      @WCE,HCS2 ;SEE IF WRITE CHECK ERROR
4405 021704 001405      BEQ     B3$
4406 021706 016537 000024 001502      MOV      @RKDB(R5),WD1 ;ACTUAL WORD FOR PRINTOUT
4407 021714 104135      ERROR    135 ;WCE AFTER WRITE CMD
4408 021716 000437      BR      B4$
4409
4410 021720 104022      B3$:    ERROR    22 ;CERR AFTER WRITE CHECK CMD
4411
4412 021722 012737 010340 005476      MOV      @<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4413 021730 005037 005500      CLR     E.B0 ;EXPECTED MSG B0
4414 021734 012737 001720 005502      MOV      @<D.SP0K!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4415 021742 012737 000001 005504      MOV      #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4416 021750 005037 005506      CLR     E.A2 ;EXPECTED MSG A2
4417 021754 012737 000002 005510      MOV      #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4418 021762 012737 000003 005514      MOV      #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4419
4420 021770 004737 025604      JSR      PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4421 021774 000003      .WORD   T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4422 021776 104057      ERROR    57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
4423 022000 104031      ERROR    31 ;MSG B0 ERROR
4424 022002 104060      ERROR    60 ;MSG A1 ERROR
4425 022004 104032      ERROR    32 ;MSG B1 ERROR
4426 022006 104401 045333      TYPE    MSG72 ;ABORTING BALANCE OF TESTS
4427 022012 000137 024006      JMP     ENDRV
4428
4429 022016      B4$:
4430
4431 022016      12$:
4432
4433 ;*****
4434 ;*TEST 20 AC LOW DETECTION PART 2
4435 ;*
4436 ;* THIS TEST VERIFIES THAT WHEN AC POWER IS LOST, THAT WRITING CEASES
4437 ;* AT SECTOR BOUNDRIES & THAT THE BATTERY RETRACT IS FUNCTIONAL.
4438 ;* THERE IS APPROX 4 MS BETWEEN AC LOW ASSERTING AND NED ASSERTING
4439 ;* WHEN THE INTERFACE SHUTS DOWN.
4440 ;*
4441 ;*****
4442 022016 000004      †ST20: SCOPE
4443 022020 012737 000001 001174      MOV      #1,$TIMES ;DO 1 ITERATION
4444 022026 012706 001100      MOV      @STACK,SP ;RESTORE STK PTR
4445
4446 022032 004737 026772      JSR      PC,SUBCLR
4447 022036 104024      ERROR    24 ;CERR AFTER SCLR
4448

```

```

4449 022040 104401 040371 TYPE ,MSG26 ;AC LOW-PART 2
4450 022044 104401 043041 TYPE ,MSG52 ;CONT-E TO BYPASS TEST
4451 022050 004737 030746 JSR PC,CCSP ;OR SPACE TO CONTINUE
4452 022054 000137 024006 JMP 12$ ;INPUT CONT-E OR SPACE
4453 022060 005737 001542 TST BSERR ;RETURN HERE FOR CONT-E
4454 022064 001402 BEQ 1$ ;RETURN HERE FOR SPACE
4455 022066 000137 022524 JMP 2$ ;TEST FOR ALIGN CARTRIDGE
4456 022072 004737 026420 1$: JSR PC,GSTAT ;BR IF NOT ALIGN CART.
4457 022076 032737 004000 005434 BIT #D.WRL,HMR2 ;SEE IF WRITE LOCK
4458 022104 001417 BEQ 11$ ;BR IF NO
4459 022106 104401 043142 TYPE ,MSG53 ;DISABLE WRITE LOCK
4460 022112 104401 041265 TYPE ,MSG37 ;PRESS SPACE WHEN DONE
4461 022116 004737 031006 JSR PC,GETSP ;GET SPACE
4462 022122 004737 026420 JSR PC,GSTAT
4463 022126 032737 004000 005434 BIT #D.WRL,HMR2 ;SEE IF STILL WRITE LOCK
4464 022134 001403 BEQ 11$ ;BR IF NO
4465 022136 104110 ERROR 110 ;WRITE LOCK NOT DISABLED
4466 022140 000137 024006 JMP 12$ ;EXIT TEST
4467 022144 005037 001366 11$: CLR TOCYL ;SETUP
4468 022150 005037 001402 CLR CALADD ;FOR
4469 022154 005037 001510 CLR HEAD ;FILL HEADER TABLE
4470 022160 005037 001516 CLR FORMAT ;ROUTINE
4471 022164 004737 027754 13$: JSR PC,FHDTAB ;BUILD STD 22 SECTOR HEADER TABLE
4472 022170 012765 001554 000004 MOV #HDTAB,RKBA(R5)
4473 022176 012765 177676 000002 MOV #-66.,RKWC(R5)
4474 022204 000337 001510 SWAB HEAD
4475 022210 013765 001510 000006 MOV HEAD,RKDA(R5) ;HEAD ADDR
4476 022216 000337 001510 SWAB HEAD
4477 022222 012765 000027 000000 MOV #<WRHEAD>,RKCS1(R5) ;WRITE HEADER CMD
4478 022230 013737 001440 005444 MOV T5000,TEMP1 ;SETUP TIMEOUT
4479 022236 004737 025062 JSR PC,FRDY ;FIND RDY
4480 022242 104200 ERROR 200 ;NO RDY AFTER WRITE HEADER CMD
4481 022244 004737 026420 JSR PC,GSTAT ;GET FRESH STATUS
4482 022250 032737 100000 005406 BIT #CERR,HCS1
4483 022256 001401 BEQ 64$
4484 022260 104201 ERROR 201 ;CERR AFTER WRITE HEADER CMD
4485 022262 64$:
4486 022262 005237 001510 INC HEAD
4487 022266 023727 001510 000003 CMP HEAD,#3 ;SEE IF ALL HEADS DONE
4488 022274 001333 BNE 13$ ;BR IF NO
4489 022276 005037 001366 CLR TOCYL
4490 022302 004737 026772 16$: JSR PC,SUBCLR
4491 022306 104024 ERROR 24 ;CERR AFTER SC_LR
4492 022310 012765 001530 000004 MOV #DATA1,RKBA(R5) ;RETURN HERE FOR SPACE

```

4505	022316	052765	000020	000010	BIS	#BAI,RKCS2(R5)	;WRITE INITIAL BACKGROUND OF 1'S
4506	022324	012765	137000	000002	MOV	#-66,*256,RKWC(R5)	;TRACK 0,1,2 ALL SECTORS
4507	022332	013765	001366	000020	MOV	TOCYL,RKDC(R5)	
4508	022340	012765	000023	000000	MOV	#WRDATA,RKCS1(R5)	;WRITE DATA CMD
4509	022346	013737	001440	005444	MOV	T50000,TMP1	;SETUP TIMEOUT
4510	022354	004737	025062		JSR	PC,FRDY	;FIND RDY
4511	022360	104011			ERROR	11	;NO RDY AFTER WRITE DATA CMD
4512	022362	004737	026420		JSR	PC,GSTAT	;GET FRESH STATUS
4513	022366	032737	100000	005406	EIT	#CERR,HCS1	
4514	022374	001453			BEQ	25	
4515	022376	032737	000200	005422	BIT	#BSE,HER	;SEE IF BAD SECTOR
4516	022404	001034			BNE	175	
4518	022406	012737	010340	005476	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4519	022414	005037	005500		CLR	E.B0	;EXPECTED MSG B0
4520	022420	012737	001720	005502	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4521	022426	012737	000001	005504	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4522	022434	005037	005506		CLR	E.A2	;EXPECTED MSG A2
4523	022440	012737	000002	005510	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4524	022446	012737	000003	005514	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4525							
4526	022454	004737	025604		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4527	022460	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4528	022462	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
4529	022464	104023			ERROR	23	;MSG B0 ERROR
4530	022466	104053			ERROR	53	;MSG A1 ERROR
4531	022470	104025			ERROR	25	;MSG B1 ERROR
4532	022472	000137	024006		JMP	125	
4533							
4534	022476	005237	001366		INC	TOCYL	
4535	022502	023727	001366	000012	CMP	TOCYL,#10.	;TRIED 10 CYLINDERS?
4536	022510	001003			BNE	185	;BR IF NO
4537	022512	104062			ERROR	62	;CANNOT WRITE ON 10 CONSEC CYL
4538	022514	000137	024006		JMP	125	
4539							
4540	022520	000137	022302		JMP	165	
4541							
4542	022524	004737	026772		JSR	PC,SUBCLR	
4543	022530	104024			ERROR	24	;CERR AFTER SCLR
4544							
4545	022532	104401	042613		TYPE	,MSG49	;TURN OFF AC
4546	022536	104401	043606		TYPE	,MSG57	;VERIFY BATTERY RETRACT FUNCTIONA
4547							
4548	022542	005737	001542		TST	BSERR	;SEE IF ALIGN CART USED
4549	022546	001405			BEQ	155	;BR IF NO
4550	022550	104401	041265		TYPE	,MSG37	;PRESS SPACE WHEN DONE
4551	022554	004737	031006		JSR	PC,GETSP	;GET SPACE
4552	022560	000526			BR	95	;SKIP ALL WRITING
4553							
4554	022562	013737	001440	001160	MOV	T50000,\$TMP0	
4555	022570	005037	001162		CLR	\$TMP1	;BIT0=0;WRITE 0'S; BIT0=1:WRITE 1'S
4556							
4557	022574	004737	026772		JSR	PC,SUBCLR	
4558	022600	104024			ERROR	24	;CERR AFTER SCLR
4559							
4560	022602	032737	000001	001162	BIT	#BIT0,\$TMP1	



K07

JNIBJS RKE DRIVE DIAGNOSTIC PART 3
 022670.F11 06-007-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 88
 T20 AC LOW DETECTION PART 2

SEG 0098

4561	022610	001004				BNE	55		:BR IF WRITING 1'S
4562	022612	012765	001524	000004		MOV	#DATA0,RKBA(R5)		:SETUP ALL 0'S
4563	022620	000403				BR	65		
4564									
4565	022622	012765	001530	000004	55:	MOV	#DATA1,RKBA(R5)		:SETUP ALL 1'S
4566	022630	052765	000020	000010	65:	BIS	#BA1,RKCS2(R5)		
4567	022636	012765	140400	000002		MOV	#-63.*256.,RKWC(R5)		:TRACK 0,1,2 63 SECTORS
4568	022644	005065	000006			CLR	RKDA(R5)		:BEGIN AT TRK AND SECTOR 0
4569	022650	013765	001366	000020		MOV	TOCYL,RKDC(R5)		
4570	022656	012765	000023	000000		MOV	#WRDATA,RKCS1(R5)		
4571	022664	013737	001440	005444		MOV	T50000,TEMP1		
4572	022672	004737	025062			JSR	PC,FRDY		:FIND CONTR RDY
4573	022676	104011				ERROR	11		:CONTR NOT RDY
4574									
4575	022700	004737	026420			JSR	PC,GSTAT		
4576	022704	032737	100000	005406		BIT	#CERR,HCS1		
4577	022712	001017				BNE	35		
4578	022714	005337	001160			DEC	\$TMP0		
4579	022720	001011				BNE	75		
4580	022722	104146				ERROR	146		:CERR NOT SET BY TIMEOUT
4581	022724	104401	042666			TYPE	,MSG50		:TURN AC BACK ON
4582	022730	104401	045145			TYPE	,MSG69		:DEPRESS SPACE AFTER 'READY' LIGHT ON
4583	022734	004737	031006			JSR	PC,GETSP		:GET SPACE
4584	022740	000137	024006			JMP	125		:EXIT TEST
4585									
4586	022744	005237	001162		75:	INC	\$TMP1		
4587	022750	000711				BR	45		:GO WRITE OPPOSITE DATA
4588									
4589	022752	032737	000100	005436	35:	BIT	#D.ACLO,HMR3		
4590	022760	001001				BNE	105		
4591	022762	104147				ERROR	147		:AC LOW NOT SET
4592									
4593	022764	005237	005370		105:	INC	UNLD		:FOR VALID HALT
4594	022770	012737	070140	005476		MOV	#<D.DSC!D.PIP!D.SPIN!D.VV!D.DRA>,E.A0		:EXPECTED MSG A0
4595	022776	012737	010300	005500		MOV	#<D.SPLS!D.ACLO!D.FLT>,E.B0		
4596	023004	012737	044720	005502		MOV	#<D.UNLD!D.REV!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		
4597	023012	012737	000001	005504		MOV	#1,E.B1		
4598									
4599	023020	004737	025604			JSR	PC,CHKMSG		:CHECK MSGS A0, B0, A1, B1
4600	023024	000000				.WORD	0!0!0		:& MSGS SPECIFIED HERE
4601	023026	104035				ERROR	35		:MSG A0 ERROR AFTER AC SWITCH OFF FROM HEADS LOADED
4602	023030	104203				ERROR	203		:MSG B0 ERROR
4603	023032	104036				ERROR	36		:MSG A1 ERROR
4604	023034	104037				ERROR	37		:MSG B1 ERROR
4605									
4606	023036	013737	005416	001164	85:	MOV	HDA,\$TMP2		:SAVE RKDA
4607	023044	013737	005416	001166		MOV	HDA,\$TMP3		:SAVE FOR TYPEOUT
4608									
4609	023052	104401	042666			TYPE	,MSG50		:TURN AC BACK ON
4610	023056	104401	045145			TYPE	,MSG69		:DEPRESS SPACE AFTER 'READY' LIGHT ON
4611	023062	004737	031006			JSR	PC,GETSP		:GET SPACE
4612									
4613	023066	004737	026772			JSR	PC,SUBCLR		
4614	023072	104024				ERROR	24		:CERR AFTER SC_LR
4615									
4616	023074	032737	000100	005436		BIT	#D.ACLO,HMR3		


```

4617 023102 001401          BEQ      9$
4618 023104 104152          ERROR    152          ;ACLO NOT RESET AFTER POWER UP
4619
4620 023106 005037 005370          9$:      CLR      UNLD          ;FOR VALID HALT
4621 023112 012765 100000 000000          MOV      #CCLR,RKCS1(R5)
4622 023120 013765 001222 000010          MOV      $UNIT,RKCS2(R5) ;DRIVE #
4623 023126 012765 000003 000000          MOV      #PACK,RKCS1(R5) ;PACK CMD
4624 023134 013737 001426 005444          MOV      T10,TEMP1
4625 023142 004737 025062          JSR      PC,FRDY          ;FIND CONTR RDY
4626 023146 104116          ERROR    116          ;CONTR NOT RDY
4627
4628 023150 032737 000100 005434          BIT      #D.VV,HMR2
4629 023156 001001          BNE      65$
4630 023160 104027          ERROR    27          ;VOLUME VALID NOT SET AFTER PACK CMD
4631 023162          65$:
4632 023162 005737 001542          TST      BSERR          ;SEE IF ALIGN CART USED
4633 023166 001402          BEQ      14$          ;BR IF NO
4634 023170 000137 024006          JMP      12$          ;ELSE EXIT TEST
4635
4636 023174          14$:
4637
4638 023174 004737 026772          JSR      PC,SUBCLR
4639 023200 104024          ERROR    24          ;CERR AFTER SCLR
4640
4641 023202 005737 001164          TST      $TMP2          ;SEE IF TRK/SECTOR 0
4642 023206 001414          BEQ      73$          ;REPEAT,NO NEW DATA XFER TOOK PLACE
4643 023210 023727 001164 001023          CMP      $TMP2,#1023    ;SEE IF TRK 2,SECTOR 19
4644 023216 001410          BEQ      73$          ;REPEAT,NO OLD DATA TO CHECK AGAINST
4645 023220 032737 000001 001162          BIT      #BIT0,$TMP1
4646 023226 001006          BNE      66$          ;BR IF WRITING 1'S WHEN WLE OCCURRED
4647 023230 012765 001530 000004          MOV      #DATA1,RKBA(R5) ;WRITING 0'S:WLE SECTOR SHOULD HAVE ALL 1'S
4648 023236 000405          BR
4649
4650 023240 000137 022524          73$:      JMP      2$
4651
4652 023244 012765 001524 000004          66$:      MOV      #DATA0,RKBA(R5) ;WRITING 1'S:WLE SECTOR SHOULD HAVE ALL 0'S
4653 023252 052765 000020 000010          67$:      BIS      #BA1,RKCS2(R5)
4654 023260 012765 177400 000002          MOV      #-256,RKWC(R5)
4655 023266 013765 001166 000006          MOV      $TMP3,RKDA(R5) ;REFRESH RKDA
4656 023274 017537 000004 001504          MOV      #RKBA(R5),WD2 ;EXPECTED WORD FOR TRUERR TYPEOUT
4657 023302 013765 001366 000020          MOV      TOCYL,RKDC(R5)
4658
4659 023310 012765 000031 000000          MOV      #<WRCHK>,RKCS1(R5) ;WRITE CHECK CMD
4660 023316 013737 001440 005444          MOV      T5000,TEMP1    ;SETUP TIMEOUT
4661 023324 004737 025062          JSR      PC,FRDY          ;FIND RDY
4662 023330 104015          ERROR    15          ;NO RDY AFTER WRITE CHECK CMD
4663 023332 004737 026420          JSR      PC,GSTAT        ;GET FRESH STATUS
4664 023336 032737 100000 005406          BIT      #CERR,HCS1
4665 023344 001450          BEQ      75$
4666 023346 032737 040000 005410          BIT      #WCE,HCS2          ;SEE IF WRITE CHECK ERROR
4667 023354 001405          BEQ      74$
4668 023356 016537 000024 001502          MOV      RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
4669 023364 104136          ERROR    136          ;WCE AFTER WRITE CMD
4670 023366 000437          BR      75$
4671
4672 023370 104022          74$:      ERROR    22          ;CERR AFTER WRITE CHECK CMD

```

4673										
4674	023372	012737	010340	005476		MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	:EXPECTED MSG A0		
4675	023400	005037	005500			CLR	E.B0	:EXPECTED MSG B0		
4676	023404	012737	001720	005502		MOV	#.D.SP0K!D.CART!D.DOOR!D.BRHM!D.SSP>.E.A1	:EXPECTED A1		
4677	023412	012737	000001	005504		MOV	#1,E.B1	:MSG ID FOR EXPECTED MSG B1		
4678	023420	005037	005506			CLR	E.A2	:EXPECTED MSG A2		
4679	023424	012737	000002	005510		MOV	#2,E.B2	:MSG ID FOR EXPECTED MSG B2		
4680	023432	012737	000003	005514		MOV	#3,E.B3	:MSG ID FOR EXPECTED MSG B3		
4681										
4682	023440	004737	025604			JSR	PC,CHKMSG	:CHECK MSGS A0, B0, A1, B1		
4683	023444	000003				.WORD	T.A2!T.B2!0	:& MSGS SPECIFIED HERE		
4684	023446	104057				ERROR	57	:MSG A0 ERROR AFTER WRITE CHECK CMD		
4685	023450	104031				ERROR	31	:MSH B0 ERROR		
4686	023452	104060				ERROR	60	:MSG A1 ERROR		
4687	023454	104032				ERROR	32	:MSG B1 ERROR		
4688	023456	104401	045333			TYPE	MSG72	:ABORTING BALANCE OF TESTS		
4689	023462	000137	024006			JMP	ENDRV			
4690										
4691	023466				75\$:					
4692	023466	000240				NOP				
4693	023470	000240				NOP				
4694										
4695	023472	023727	001164	000400		CMF	\$TMP2,#400	:SEE IF WRL AT TRK 1, SECTOR 0		
4696	023500	001004				BNE	68\$:BR IF NO		
4697	023502	012765	000025	000006		MOV	#21.,RKDA(R5)	:ELSE SECTOR BEFORE WRL IS TRK 0, SEC 21		
4698	023510	000415				BR	70\$			
4699	023512	023727	001164	001000	68\$:	CMF	\$TMP2,#1000	:SEE IF WRL AT TRK 2,SECTOR 0		
4700	023520	001004				BNE	69\$:BR IF NO		
4701	023522	012765	000425	000006		MOV	#425,RKDA(R5)	:ELSE SECTOR BEFORE WRL IS TRK 1, SEC 21		
4702	023530	000405				BR	70\$			
4703										
4704	023532	005337	001164		69\$:	DEC	\$TMP2	:GET SECTOR BEFORE WRL		
4705	023536	013765	001164	000006		MOV	\$TMP2,RKDA(R5)			
4706	023544	016537	000006	001166	70\$:	MOV	RKDA(R5), \$TMP3	:FOR ERROR PRINTOUT		
4707	023552	032737	000001	001162		BIT	#BIT0,\$TMP1			
4708	023560	001004				BNE	71\$:BR IF WRITING 1'S WHEN WLE OCCURRED		
4709	023562	012765	001524	000004		MOV	#DATA0,RKBA(R5)	:WRITING 0'S:WLE SECTOR -1 SHOULD HAVE ALL 0'S		
4710	023570	000403				BR	72\$			
4711										
4712	023572	012765	001530	000004	71\$:	MOV	#DATA1,RKBA(R5)	:WRITING 1'S:WLE SECTOR -1 SHOULD HAVE ALL 1'S		
4713	023600	052765	000020	000010	72\$:	BIS	#BA1,RKCS2(R5)			
4714	023606	012765	177400	000002		MOV	#-256.,RKWC(R5)			
4715	023614	017537	000004	001504		MOV	3RKBA(R5),WD2	:EXPECTED WORD FOR TRUERR TYPEOUT		
4716	023622	013765	001366	000020		MOV	TOCYL,RKDC(R5)			
4717										
4718	023630	012765	000031	000000		MOV	#<WRTCHK>,RKCS1(R5)	:WRITE CHECK CMD		
4719	023636	013737	001440	005444		MOV	T5000,TEMP1	:SETUP TIMEOUT		
4720	023644	004737	025062			JSR	PC,FRDY	:FIND RDY		
4721	023650	104015				ERROR	15	:NO RDY AFTER WRITE CHECK CMD		
4722	023652	004737	026420			JSR	PC,GSTAT	:GET FRESH STATUS		
4723	023656	032737	100000	005406		BIT	#CERR,HCS1			
4724	023664	001450				BEQ	77\$			
4725	023666	032737	040000	005410		BIT	#WCE,HCS2	:SEE IF WRITE CHECK ERROR		
4726	023674	001405				BEQ	76\$			
4727	023676	016537	000024	001502		MOV	RKDB(R5),WD1	:ACTUAL WORD FOR PRINTOUT		
4728	023704	104137				ERROR	137	:WCE AFTER WRITE CMD		

```

4729 023706 000437
4730
4731 023710 104022
4732
4733 023712 012737 010340 005476
4734 023720 005037 005500
4735 023724 012737 001720 005502
4736 023732 012737 000001 005504
4737 023740 005037 005506
4738 023744 012737 000002 005510
4739 023752 012737 000003 005514
4740
4741 023760 004737 025604
4742 023764 000003
4743 023766 104057
4744 023770 104031
4745 023772 104060
4746 023774 104032
4747 023776 104401 045333
4748 024002 000137 024006
4749
4750 024006
4751
4752 024006
4753
4754 024006
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766 024006 000004
4767 024010 012737 000001 001174
4768 024016 012705 001100
4769
4770 024022 005237 001220
4771 024026 023737 005526 001220
4772 024034 001402
4773 024036 000137 012276
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784

```

```

BR 775
765: ERROR 22 ;CERR AFTER WRITE CHECK CMD
MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
CLR E.B0 ;EXPECTED MSG B0
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
CLR E.A2 ;EXPECTED MSG A2
MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
; & MSGS SPECIFIED HERE
.WORD T.A2!T.B2!0
ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
ERROR 31 ;MSG B0 ERROR
ERROR 60 ;MSG A1 ERROR
ERROR 32 ;MSG B1 ERROR
TYPE MSG72 ;ABORTING BALANCE OF TESTS
JMP ENDRV
775:
125:
ENDRV:
;*****
;*TEST 21 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'.
;*****
TST21: SCOPE
MOV #1,$TIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
INC $DEVCT ;INCR COUNT FOR # OF DRIVES THAT ARE CHECKED
CMP DRIVS,$DEVCT ;ARE ALL DRIVES PRESENT TESTED?
BEQ TST22 ;GO TO NEXT TEST IF YES
JMP NUDRV ;IF NOT, GO BACK & TEST NEXT DRIVE PRESENT.
;*****
;*TEST 22 MULTIPLE DRIVE DETECTION TEST
;*
;* 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 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

TST22: SCOPE

48:01	024042	003004				MOV	#1,STIMES	::DO 1 ITERATION
48:02	024044	012737	000001	001174		MOV	#STACK,SP	::RESTORE STACK PTR
48:03	024052	012706	001100					
48:04	024056	005737	001216			TST	\$PASS	
48:05	024062	001402				BEG	1\$::DO TEST ONLY IN 1ST PASS
48:06	024064	000137	024370			JMP	11\$::ELSE EXIT TEST
48:07	024070	023727	005526	000001	1\$:	CMP	DRIVS,#1	
48:08	024076	001004				BNE	2\$::BR IF MORE THAN 1 DRIVE PRESENT
48:09	024100	104401	044217			TYPE	.MSG62	::BYPASS TEST, ONLY 1 DRIVE PRESENT
48:10	024104	000137	024370			JMP	11\$::ELSE EXIT TEST
48:11	024110	104401	040462		2\$:	TYPE	.MSG28	::MULT DRV DETECTION TEST
48:12	024114	104401	043041			TYPE	.MSG52	::PRESS CONT-E TO EXIT OR SPACE TO CONTINUE
48:13	024120	004737	030746			JSR	PC,CCSP	::INPUT CONT-E OR SPACE
48:14	024124	000137	024370			JMP	11\$::RETURN HERE FOR CONT-E
48:15	024130	104401	043654			TYPE	.MSG58	::LOAD HEADS ON ALL DRIVES
48:16	024134	104401	041265			TYPE	.MSG37	::PRESS SPACE WHEN DONE
48:17	024140	004737	031006			JSR	PC,GETSP	::GET SPACE
48:18	024144	004737	026772		3\$:	JSR	PC,SUBCLR	
48:19	024150	104024				ERROR	24	::CERR AFTER SCLR
48:20	024152	104401	043736			TYPE	.MSG59	::INSERT SAME UNIT SEL PLUG # IN 2 DRIVES
48:21	024156	104401	041265			TYPE	.MSG37	::DEPRESS SPACE BAR WHEN DONE
48:22	024162	004737	031006			JSR	PC,GETSP	::GET SPACE
48:23	024166	005000				CLR	R0	::DRIVE # COUNTER
48:24	024170	012765	100000	000000	6\$:	MOV	#CLR,RKCS1(R5)	
48:25	024176	010065	000010			MOV	R0,RKCS2(R5)	::DRIVE #
48:26	024202	012765	000001	000000		MOV	#SELDRV,RKCS1(R5)	::SELECT DRIVE CMD TO GET STATUS
48:27	024210	013737	001426	005444		MOV	T10,TEMP1	
48:28	024216	004737	025062			JSR	PC,FRDY	::FIND CONTR RDY
48:29	024222	104117				ERROR	117	::NO CONTR RDY
48:30	024224	012737	011610	005444		MOV	#5000,TEMP1	
48:31	024232	004737	025552			JSR	PC,DLY	::REQ 2 MS DELAY BEFORE MDS DETECTED
48:32	024236	012765	100000	000000		MOV	#CLR,RKCS1(R5)	
48:33	024244	010065	000010			MOV	R0,RKCS2(R5)	
48:34	024250	012765	000001	000000		MOV	#SELDRV,RKCS1(R5)	
48:35	024256	013737	001426	005444		MOV	T10,TEMP1	

```

4850 JSR PC,FRDY          004737 025062          JSR PC,FRDY
4851 ERROR 117          :NO CONTR RDY
4852
4853 BIT #MDS,MOSE2      001000 005410          BIT #MDS,MOSE2
4854 BNE 75             :SEE IF THAT DRIVE HAS MDS
4855 INC RD              :BR IF YES
4856 CMP RD,#8          :ELSE TRY ANOTHER DRIVE
4857 BNE 105            :SEE IF ALL DRIVES TESTED
4858 ERROR 176          :BR IF NO
4859 BR 105             :CANNOT FIND MDS
4860
4861 MOV RD,TYPE         104401 044151          75: TYPE MSG61
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

.SETTL END OF PASS ROUTINE

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

SEOP:

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?

SEOPCT:

.WORD 1
BGT \$DOAGN ;: YES
MOV (PC)+,2(PC)+ ;: RESTORE COUNTER

SENDCT:

.WORD 1
SEOPCT

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 2(PC)+ ;: RETURN

\$RTNAD:

.WORD ST5

\$ENULL:

.BYTE -1,-1,0 ;: NULL CHARACTER STRING

\$SENDMG:

.ASCIZ (<15><12> /END PASS #/

4873 024402 000004
4874 024402 005037 001102
4875 024404 005037 001174
4881 024410 005237 001216
4882 024414 005237 001216
4883 024420 042737 100000 001216
4884 024426 005327
4885 024430 000001
4886 024432 003022
4887 024434 012737
4888 024436 000001
4889 024440 024430
4890 024442 104401 024507
4891 024446 013746 001216
4892 024452 104405
4893 024454 104401 024504
4894 024460 013700 000042
4895 024464 001405
4896 024466 000005
4897 024470 004710
4898 024472 000240
4899 024474 000240
4900 024476 000240
4901 024500
4902 024500 000137
4903 024502 010656
4904 024504 377 377 000
4905 024507 015 042412 042116
4906 024514 050040 051501 020123
4907 024522 000043

```

4908
4909
4910
4911
4912
4913 024524 012700 005516
4914 024530 012701 177757
4915 024534 005020
4916 024536 005201
4917 024540 001375
4918 024542 000207
4919
4920
4921
4922
4923
4924 024544 005737 001360
4925 024550 001024
4926 024552 005237 001360
4927 024556 104401 036220
4928
4929 024562 005737 000042
4930 024566 001012
4931 024570 123727 001230 000001
4932 024576 001406
4933 024600 023727 001140 000176
4934 024606 001005
4935 024610 104406
4936 024612 000403
4937 024614 112737 000001 001134
4938 024622
4939 024622 000207
4940
4941
4942
4943
4944
4945
4946 024624 104411
4947 024626 012600
4948 024630 012701 177770
4949 024634 112002
4950 024636 042702 177400
4951 024642 012703 005530
4952 024646 012704 000060
4953
4954 024652 020402
4955 024654 001415
4956 024656 005723
4957 024660 005204
4958 024662 020427 000070
4959 024666 001371
4960 024670 005702
4961 024672 001022
4962 024674 020127 177770
4963 024700 001426

```

```

.SBTTL SUBROUTINES
:SUBROUTINE TO CLEAR ALL FLAGS FROM DUMP THRU DCTIM
:
CLRFLG: MOV      #DUMP, R0
        MOV      #-17, R1
IS:     CLR      (R0)+
        INC      R1
        BNE     IS
        RTS     PC

:
:TYPE PROGRAM ID IF FTITLE=0
:
TITLE:  TST      FTITLE
        BNE     IS
        INC     FTITLE
        TYPE    MSG1
.SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
        TST     #42
        BNE     645 ;:ARE WE RUNNING UNDER XXDP/ACT?
        CMPB    $ENV, #1 ;:BRANCH IF YES
        BEQ     645 ;:ARE WE RUNNING UNDER APT?
        CMP     SWR, #SWREG ;:BRANCH IF YES
        BNE     655 ;:SOFTWARE SWITCH REG SELECTED?
        GTSWR   ;:BRANCH IF NO
        BR      655 ;:GET SOFT-SWR SETTINGS
        MOVB    #1, $AUTOS ;:SET AUTO-MODE INDICATOR
IS:     RTS     PC

:
:ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
:DRIVS, DRIVO-DRIV7 REGISTERS APPROPRIATELY
:
GDRVS:  RDLIN
        MOV     (SP)+, R0 ;:GET STARTING ADDR OF ASCII STRING
        MOV     #-8, R1 ;:SET UP COUNT
IS:     MOVB   (R0)+, R2 ;:GET ASCII CHAR
        BIC     #177400, R2 ;:MASK HI BYTE
        MOV     #DRIVO, R3 ;:DRIVE FLAG ADDR
        MOV     #60, R4
        CMP     R4, R2 ;:WAS TYPED CHAR 0 THRU 7?
        BEQ     3$ ;:BRANCH IF YES
        TST    (R3)+ ;:NO, INCREMENT DR FLAG ADDR
        INC     R4
        CMP     R4, #70
        BNE     2$ ;:S/B 0-7 OR TERMINATOR
        TST    R2
        BNE     4$
        CMP     R1, #-8.
        BEQ     6$ ;:DEFAULT ALL DRIVES

```

F08

IBUS Rk6 DRIVE DIAGNOSTIC PART 3
 0276JC.P11 06-OCT-76 09:44

MAY11 27(1006) 06-OCT-76 09:54 PAGE 96
 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEG 0096

4984	024702	005037	005556	75:	CLR RTS	SIZFLG PC	:BYPASS TEST 1 (SIZING) :FOUND TERMINATOR, EXIT
4985	024706	000207					
4986	024710	005213		35:	INC INC MOVB	DR3 DRIVS (R0)+,R2	:SET UP FLAG FOR THE DRIVE :INCREMENT TOTAL # DRIVES TO BE TESTED :GET NEXT ASCII CHAR.
4987	024712	005237	005526		BIC	#177400,R2	:MASK
4988	024716	112002			CMP	#54,R2	:IS IT A COMMA?
4989	024720	042702	177400		BEQ	55	:YES, GO TO NEXT WORD.
4990	024724	022702	000054		TS	R2	:NO, IS IT A TERMINATOR?
4991	024730	001407			BNE	45	:IF NOT, SOMETHING WRONG.
4992	024732	005702			BR	75	:FOUND TERMINATOR, EXIT
4993	024734	001001					
4994	024736	000761					
4995	024740	104401	045700	45:	TYPE JMP	EMI PRGSRT	:ONLY 0-7 ALLOWED. :START ALL OVER
4996	024744	000137	010064				
4997	024750	005201		55:	INC BNE BR	R1 15 45	:S/B NO MORE THAN 8 DIFF :DRIVES TYPED IN. :IF MORE, HAVE ERROR.
4998	024752	001330					
4999	024754	000771					
5000	024756	005237	005556	65:	INC RTS	SIZFLG PC	:DO TEST 1 (SIZING) :EXIT.
5001	024762	000207					
5002							:ROUTINE TO INPUT RKBAS OR DEFAULT.
5003							:
5004	024764	104412		69A:	RDOCT MOV	(SP)+,R0	:GET LOW ORDER FROM STACK
5005	024766	012600			TST	R0	
5006	024770	005700			BEQ	15	:BRANCH IF DEFAULT.
5007	024772	001403			MOV	R0,\$BASE	
5008	024774	010037	001264		RTS	PC	
5009	025000	000207		15:	MOV	#177440,\$BASE	:DEFAULT VALUE
5010	025002	012737	177440 001264		RTS	PC	
5011	025010	000207					:ROUTINE TO INPUT RKVEC OR DEFAULT
5012							:
5013	025012	104412		GINT:	RDOCT MOV	(SP)+,R0	:GET LOW ORDER FROM STACK
5014	025014	012600			TST	R0	
5015	025016	005700			BEQ	15	:BRANCH IF DEFAULT
5016	025020	001405			MOV	R0,RKVEC	
5017	025022	010037	001334	25:	JSR	PC,SETINT	
5018	025026	004737	025044		RTS	PC	
5019	025032	000207		15:	MOV	#210,RKVEC	:DEFAULT VALUE
5020	025034	012737	000210 001334		BR	25	
5021	025042	000771					:ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
5022							:
5023	025044	013700	001334	SETINT:	MOV	RKVEC,R0	
5024	025050	012720	031520		MOV	#INTER,(R0)+	:INTER ADDR TO RKVEC


```

5020 025054 013710 001336      MOV      RKPRI,(R0)      ;PRS TO RKVEC+2
5021 025060 000207      RTS      PC
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
025062 032765 000200 000000  FRDY:   BIT      #RDY,RKCS1(R5)
025070 001010      BNE     1$
025072 005337 005444      DEC     TEMP1
025076 001371      BNE     FRDY
025100 004737 025216      JSR    PC,HOLD          ;STORE ALL RK611 REGS IN HOLDING REGS.
025104 004737 026336      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR
025110 000207      RTS     PC              ;NO RDY, EXIT
025112 062716 000002      1$:    ADD     #2,(SP)    ;SKIP OVER ERROR
025116 004737 025216      JSR    PC,HOLD
025122 004737 026336      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR
025126 000207      RTS     PC

;ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
5047 025130 032765 000200 000000  FRDY1:  BIT      #RDY,RKCS1(R5)
5048 025136 001014      BNE     1$
5049 025140 005337 005444      DEC     TEMP1
5050 025144 001371      BNE     FRDY1
5051 025146 016537 000034 005434      MOV     RKMR2(R5),HMR2
5052 025154 016537 000036 005436      MOV     RKMR3(R5),HMR3
5053 025162 004737 026336      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
5054 025166 000207      RTS     PC              ;NO RDY, EXIT
5055 025170 062716 000002      1$:    ADD     #2,(SP)    ;SKIP OVER ERROR
5056 025174 016537 000034 005434      MOV     RKMR2(R5),HMR2
5057 025202 016537 000036 005436      MOV     RKMR3(R5),HMR3
5058 025210 004737 026336      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
5059 025214 000207      RTS     PC

;STORE ALL RK611 REGISTERS IN HOLDING REGS
5064 025216 016537 000000 005406  HOLD:   MOV     RKCS1(R5),HCS1
5065 025224 016537 000010 005410      MOV     RKCS2(R5),HCS2
5066 025232 016537 000002 005412      MOV     RKWC(R5),HWC
5067 025240 016537 000004 005414      MOV     RKBA(R5),HBA
5068 025246 016537 000006 005416      MOV     RKDA(R5),HDA
5069 025254 016537 000012 005420      MOV     RKDS(R5),HDS
5070 025262 016537 000014 005422      MOV     RKER(R5),HER
5071 025270 016537 000016 005424      MOV     RKASOF(R5),HASOF
5072 025276 016537 000020 005426      MOV     RKDC(R5),HDC
5073 025304 016537 000026 005432      MOV     RKMR1(R5),HMR1
5074 025312 016537 000034 005434      MOV     RKMR2(R5),HMR2
5075 025320 016537 000036 005436      MOV     RKMR3(R5),HMR3

```

```

5076 025326 016537 000030 005440
5077 025334 016537 000032 005442
5078 025342 000207
5079
5080
5081
5082
5083
5084
5085 025344 010446
5086 025346 013704 001222
5087 025352 136437 005376 005425
5088 025360 001404
5089 025362 012604
5090 025364 062716 000002
5091 025370 000207
5092 025372 012604
5093 025374 000207
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105 025376 010446
5106 025400 012737 177777 005444
5107 025406 013704 001222
5108 025412 136465 005376 000017
5109 025420 001014
5110 025422 005337 005444
5111 025426 001371
5112 025430 005337 005446
5113 025434 001361
5114 025436 005065 000026
5115 025442 004737 026420
5116 025446 012604
5117 025450 000207
5118 025452 005065 000026
5119 025456 004737 026420
5120 025462 012604
5121 025464 062716 000002
5122 025470 000207
5123
5124
5125
5126
5127
5128
5129
5130
5131

```

```

MOV RKECP5(R5),HPOS
MOV RKECPT(R5),HPAT
RTS PC

:ROUTINE TO CHECK FOR CORRECT ATTN
:RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
:RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
:
†STATN: MOV R4, -(SP) ;SAV R4
MOV $UNIT, R4
BITB ATTN(R4), HASOF+1
BEG 1$ ;BRANCH IF ATTN NOT PRESENT
MOV (SP)+, R4 ;RESTOR R4
ADD #2, (SP) ;INCR RET ADDR TO JUMP OVER ERROR.
RTS PC
1$: MOV (SP)+, R4 ;RESTOR R4
RTS PC

:ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
:ENTER WITH TIME IN SECONDS IN TEMP2
:RETURN IF NO ATTN (ERROR CONDITION)
:RETURN +2 IF ATTN FOUND
:STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
:
FATT1: MOV R4, -(SP) ;SAV R4
3$: MOV #1, TEMP1
MOV $UNIT, R4
1$: BITB ATTN(R4), RKASOF+1(R5) ;FIND CORRECT ATTN
BNE 2$
DEC TEMP1
BNE 1$
DEC TEMP2
BNE 3$
CLR RKMR1(R5) ;SELECT WORD 0
JSR PC, GSTAT ;GET LATEST STATUS
MOV (SP)+, R4 ;RESTOR R4
RTS PC

2$: CLR RKMR1(R5)
JSR PC, GSTAT ;GET STATUS AFTER ATTN SEEN
MOV (SP)+, R4 ;RESTOR R4
ADD #2, (SP) ;SKIP OVER ERROR
RTS PC

:ROUTINE TO FIND ATTN WITHIN 1 SEC
:ENTER WITH COUNT IN TEMP1
:RETURN IF NO ATTN (ERROR)
:RETURN +2 IF ATTN FOUND
:STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
:

```

```

5132 025472 010446
5133 025474 013704 001222
5134 025500 136465 005376 000017
5135 025506 001011
5136 025510 005337 005444
5137 025514 001367
5138 025516 005065 000026
5139 025522 004737 026420
5140 025526 012604
5141 025530 000207
5142 025532 005065 000026
5143 025536 004737 026420
5144 025542 012604
5145 025544 062716 000002
5146 025550 000207
5147
5148
5149
5150
5151
5152 025552 005737 005444
5153 025556 001403
5154 025560 005337 005444
5155 025564 000772
5156 025566 000207
5157
5158
5159
5160
5161 025570 104401 037705
5162 025574 010046
5163
5164 025576 104403
5165 025600 001
5166 025601 000
5167 025602 000207
5168
5169
5170
5171 025604 017637 000000 001552
5172 025612 062716 000002
5173 025616 004737 026462
5174
5175 025622 053737 001222 005476
5176 025630 053737 001222 005502
5177 025636 053737 001222 005506
5178 025644 053737 001222 005512
5179
5180 025652 013746 005444
5181
5182 025656 013737 005476 005444
5183 025664 004737 030642
5184 025670 013737 005444 005476
5185
5186 025676 013737 005502 005444
5187 025704 004737 030642

```

```

FATT2: MOV R4, -(SP) ; SAV R4
25: MOV $UNIT, R4
BITB ATTN(R4), RKASOF+1(R5) ; FIND CORRECT ATTN
BNE 1$
DEC TEMP1
BNE 2$
CLR RKMR1(R5) ; SELECT WORD 0
JSR PC, GSTAT ; GET LATEST STATUS.
MOV (SP)+, R4 ; RESTOR R4
RTS PC
1$: 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 1$ ; 2.5 US
DEC TEMP1 ; 6.8 US
BR DLY ; 2.5 US
1$: RTS PC ; 3.8 US

; THIS ROUTINE TYPES BYPASSED DRIVE#. ENTER WITH DRIVE# IN R0
BYP: TYPE MSG14 ; BYPASS DRIVE
MOV R0, -(SP) ; SAVE R0 FOR TYPEOUT
; TYPE DR#
; GO TYPE--OCTAL ASCII
; TYPE 1 DIGIT(S)
; SUPPRESS LEADING ZEROS
TYPECS 1
.BYTE 1
.BYTE 0
RTS PC

; THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.
CHKMSG: MOV #1(SP), 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

```

JOB

JTBUS RKE DRIVE DIAGNOSTIC PART 3
 02610.F11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 100
 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEG 0100

5188	025710	013737	005444	005502		MOV	TEMP1,E.A1	
5189								
5190	025716	013737	005506	005444		MOV	E.A2,TEMP1	
5191	025724	004737	030642			JSR	PC,SBPAR	;GET PARITY FOR MSG A2
5192	025730	013737	005444	005506		MOV	TEMP1,E.A2	
5193								
5194	025736	013737	005500	005444		MOV	E.B0,TEMP1	
5195	025744	004737	030642			JSR	PC,SBPAR	;GET PARITY FOR MSG B0
5196	025750	013737	005444	005500		MOV	TEMP1,E.B0	
5197								
5198	025756	013737	005504	005444		MOV	E.B1,TEMP1	
5199	025764	004737	030642			JSR	PC,SBPAR	;GET PARITY FOR MSG B1
5200	025770	013737	005444	005504		MOV	TEMP1,E.B1	
5201								
5202	025776	013737	005510	005444		MOV	E.B2,TEMP1	
5203	026004	004737	030642			JSR	PC,SBPAR	;GET PARITY FOR MSG B2
5204	026010	013737	005444	005510		MOV	TEMP1,E.B2	
5205								
5206	026016	013737	005514	005444		MOV	E.B3,TEMP1	
5207	026024	004737	030642			JSR	PC,SBPAR	;GET PARITY FOR MSG B3
5208	026030	013737	005444	005514		MOV	TEMP1,E.B3	
5209								
5210	026036	012637	005444			MOV	(SP)+,TEMP1	;RESTORE TEMP1
5211	026042	013737	001176	001172		MOV	\$ESCAPE,\$TMP5	;SAVE ESCAPE
5212								
5213	026050	023737	005456	005476		CMP	H.A0,E.A0	;TEST MSG A0
5214	026056	001411				BEQ	2\$;BR IF OK
5215	026060	012737	026072	001176		MOV	#1\$, \$ESCAPE	;ELSE SETUP ESCAPE
5216	026066	011646				MOV	(SP),-(SP)	;COPY RET ADDR.
5217	026070	000207				RTS	PC	; & RETURN TO MAINLINE ERROR
5218								
5219	026072	032777	001000	153040	1\$:	BIT	#SW9,\$SWR	;RET HERE FROM MAINLINE ERROR
5220	026100	001107				BNE	20\$; & BR IF LOOP ON ERROR
5221	026102	062716	000002		2\$:	ADD	#2,(SP)	;BUMP RET ADDR TO NEXT ERROR
5222								
5223	026106	023737	005460	005500		CMP	H.B0,E.B0	;TEST MSG B0
5224	026114	001411				BEQ	5\$;BR IF OK
5225	026116	012737	026130	001176		MOV	#4\$, \$ESCAPE	;ELSE SETUP ESCAPE
5226	026124	011646				MOV	(SP),-(SP)	;COPY RET ADDR
5227	026126	000207				RTS	PC	; & RETURN TO MAINLINE ERROR
5228								
5229	026130	032777	001000	153002	4\$:	BIT	#SW9,\$SWR	;RETURN HERE FROM MAINLINE ERROR
5230	026136	001070				BNE	20\$; & BR IF LOOP ON ERROR
5231	026140	062716	000002		5\$:	ADD	#2,(SP)	;BUMP RET ADDR TO NEXT ERROR
5232								
5233	026144	023737	005462	005502		CMP	H.A1,E.A1	;TEST MSG A1
5234	026152	001411				BEQ	8\$;BR IF OK
5235	026154	012737	026166	001176		MOV	#7\$, \$ESCAPE	;ELSE SETUP ESCAPE
5236	026162	011646				MOV	(SP),-(SP)	;COPY RET ADDR
5237	026164	000207				RTS	PC	; & RETURN TO MAINLINE ERROR
5238								
5239	026166	032777	001000	152744	7\$:	BIT	#SW9,\$SWR	;RETURN HERE FROM MAINLINE ERROR
5240	026174	001051				BNE	20\$; & BR IF LOOP ON ERROR
5241	026176	062716	000002		8\$:	ADD	#2,(SP)	;BUMP RET ADDR TO NEXT ERROR
5242								
5243	026202	023737	005464	005504		CMP	H.B1,E.B1	;TEST MSG B1

K08

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZRB6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 101
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEG 210:

```

5244 026210 001411          BEQ      11$          ;BR IF OK
5245 026212 012737 026224 001176      MOV      #10$, $ESCAPE
5246 026220 011646          MOV      (SP), -(SP)
5247 026222 000207          RTS      PC
5248
5249 026224 032777 001000 152706 10$:   BIT      #SW9, $SWR
5250 026232 001032          BNE      20$
5251 026234 062716 000002          11$:   ACD      #2, (SP)
5252
5253 026240 032737 000001 001552 12$:   BIT      #T.A2, CHKFLG ;TEST MSG A2?
5254 026246 001402          BEQ      13$          ;BR IF NO
5255 026250 004737 027256          JSR      PC, RCYLD    ;PUT INFO CYLDIF, DO NOT CHECK
5256 026254 032737 000002 001552 13$:   BIT      #T.B2, CHKFLG ;TEST MSG B2?
5257 026262 001402          BEQ      14$          ;BR IF NO
5258 026264 004737 027330          JSR      PC, RCYLA   ;PUT INFO IN CYLADD, DO NOT CHECK
5259
5260 026270 032737 000004 001552 14$:   BIT      #T.B3, CHKFLG ;TEST MSG B3?
5261 026276 001404          BEQ      15$
5262 026300 004737 027366          JSR      PC, RSEC    ;PUT INFO IN SECTOR, DO NOT CHECK
5263 026304 004737 027424          JSR      PC, RHEAD   ;PUT INFO IN HEADA, DO NOT CHECK
5264
5265 026310 013737 001172 001176 15$:   MOV      $TMP5, $ESCAPE ;RESTORE ESCAPE
5266 026316 000207          RTS      PC
5267
5268 026320 012706 001100 20$:   MOV      #STACK, SP   ;RESET STACK PTR
5269 026324 013737 001172 001176      MOV      $TMP5, $ESCAPE ;RESTORE ESCAPE
5270 026332 000177 152552          JMP      $SLPERR
5271
5272          ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
5273          ; I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
5274
5275 026336 005737 001550      CKCERR: TST      BYPCERR
5276 026342 001025          BNE      4$
5277 026344 032737 100000 005406      BIT      #CERR, HCS1
5278 026352 001001          BNE      1$          ;BR IF CERR
5279 026354 000207          RTS      PC
5280
5281 026356 032737 004000 005406 1$:   BIT      #CTO, HCS1
5282 026364 001402          BEQ      2$          ;BR IF NOT CTO
5283 026366 104211          ERROR   211         ;CTO ERROR, MSG A & B INVALID
5284 026370 000207          RTS      PC
5285
5286 026372 032737 010000 005410 2$:   BIT      #NED, HCS2
5287 026400 001401          BEQ      3$          ;BR IF NOT NED
5288 026402 104212          ERROR   212         ;NED ERROR, MSG A & B INVALID
5289
5290 026404 032737 001000 005410 3$:   BIT      #MDS, HCS2
5291 026412 001401          BEQ      4$
5292 026414 104213          ERROR   213         ;MDS ERROR, MSG A & B INVALID
5293
5294 026416 000207          4$:   RTS      PC
5295
5296          ; THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
5297          ; IT THEN WAITS FOR CONTROLLER READY
5298
5299

```

L08

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
 02R6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 102
 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0102

```

5300
5301
5302
5303 026420 013746 005444
5304 026424 013765 001222 000010
5305 026432 012765 000001 000000
5306 026440 013737 001426 005444
5307 026446 004737 025062
5308 026452 104117
5309 026454 012637 005444
5310 026460 000207
5311
5312
5313
5314
5315 026462 013746 005444
5316 026466 004737 025216
5317 026472 012765 100000 000000
5318 026500 013765 001222 000010
5319 026506 012765 000003 000026
5320 026514 012765 000001 000000
5321 026522 013737 001426 005444
5322 026530 004737 025130
5323 026534 104117
5324 026536 013737 005434 005472
5325 026544 013737 005436 005474
5326
5327 026552 012765 100000 000000
5328 026560 013765 001222 000010
5329 026566 012765 000002 000026
5330 026574 012765 000001 000000
5331 026602 013737 001426 005444
5332 026610 004737 025130
5333 026614 104117
5334 026616 013737 005434 005466
5335 026624 013737 005436 005470
5336
5337 026632 012765 100000 000000
5338 026640 013765 001222 000010
5339 026646 012765 000001 000026
5340 026654 012765 000001 000000
5341 026662 013737 001426 005444
5342 026670 004737 025130
5343 026674 104117
5344 026676 013737 005434 005462
5345 026704 013737 005436 005464
5346
5347 026712 012765 100000 000000
5348 026720 013765 001222 000010
5349 026726 012765 000001 000000
5350 026734 013737 001426 005444
5351 026742 004737 025130
5352 026746 104117
5353 026750 013737 005434 005456
5354 026756 013737 005436 005460
5355
  
```

: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, RKMR1(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, RKMR1(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, RKMR1(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
  
```

M08

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
DZ6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 103
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEG 0103

```

5356 026764 012637 005444      MOV    (SP)+,TEMP1      ;RESTORE TEMP1
5357 026770 000207      RTS    PC
5358
5359
5360
5361      ; THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
5362      ; IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
5363      ; THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
5364      ; RETURN IF CERR SET
5365      ; RETURN +2 IF CERR CLEAR
5366 026772 012765 000040 000010  SUBCLR: MOV    #SCLR,RKCS2(R5) ;SUBSYS CLEAR
5367 027000 013737 001426 005444      MOV    T10,TEMP1
5368 027006 004737 025062      JSR    PC,FRDY          ;FIND RDY
5369 027012 104120      ERROR 120              ;RDY NOT SET BY END OF SCLR
5370 027014 013765 001222 000010  MOV    $UNIT,RKCS2(R5) ;CURRENT DRIVE #
5371 027022 005065 000026      CLR    RKMR1(R5)       ;SELECT WORD 0
5372 027026 004737 026420      JSR    PC,GSTAT        ;GET STATUS
5373 027032 032737 100000 005406  BIT    #CERR,HCS1      ;CHECK FOR CONT ERROR
5374 027040 001401      BEQ    1$
5375 027042 000207      RTS    PC
5376 027044 062716 000002 1$: ADD    #2,(SP)        ;SKIP OVER ERROR
5377 027050 000207      RTS    PC
5378
5379
5380      ; READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
5381
5382 027052 012765 000003 000026  RDSEC: MOV    #3,RKMR1(R5) ;WORD 3
5383 027060 004737 026420      JSR    PC,GSTAT
5384 027064 013737 005436 001422  MOV    HMR3,SECTOR
5385 027072 042737 177017 001422  BIC    #1<M.SECT>,SECTOR
5386 027100 006237 001422      ASR    SECTOR          ;RIGHT JUSTIFY
5387 027104 006237 001422      ASR    SECTOR          ;SECTOR
5388 027110 006237 001422      ASR    SECTOR          ;INFO
5389 027114 006237 001422      ASR    SECTOR
5390 027120 000207      RTS    PC
5391
5392      ; READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
5393
5394 027122 012765 000002 000026  RDCYLD: MOV    #2,RKMR1(R5) ;WORD 2
5395 027130 004737 026420      JSR    PC,GSTAT
5396 027134 013737 005434 001376  MOV    HMR2,CYLDIF
5397
5398 027142 042737 160017 001376  BIC    #1<M.CDIF>,CYLDIF
5399 027150 006237 001376      ASR    CYLDIF          ;RIGHT JUSTIFY
5400 027154 006237 001376      ASR    CYLDIF          ;CYL DIFF/OFFSET
5401 027160 006237 001376      ASR    CYLDIF          ;INFO
5402 027164 006237 001376      ASR    CYLDIF
5403 027170 023727 001376 000777  CMP    CYLDIF,#777     ;CHK TO SEE IF RET IN COMPL. FORM
5404 027176 001002      BNE    1$              ;BR IF NOT
5405 027200 005037 001376      CLR    CYLDIF          ;CLR IF YES
5406 027204 000207 1$: RTS    PC
5407
5408      ; READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
5409
5410 027206 012765 000002 000026  RDCYLA: MOV    #2,RKMR1(R5) ;WORD 2
5411 027214 004737 026420      JSR    PC,GSTAT

```

```

5412 027220 013737 005436 001400      MOV      HMR3,CYLADD
5413 027226 042737 160017 001400      BIC      #1C<M.CADD>,CYLADD
5414 027234 006237 001400      ASR      CYLADD          ;RIGHT JUSTIFY
5415 027240 006237 001400      ASR      CYLADD          ;CYL ADDR
5416 027244 006237 001400      ASR      CYLADD          ;INFO
5417 027250 006237 001400      ASR      CYLADD
5418 027254 000207                RTS      PC
5419
5420      ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
5421
5422 027256 013737 005466 001376  RCYLD:  MOV      H.A2,CYLDIF
5423 027264 042737 160017 001376      BIC      #1C<M.CDIF>,CYLDIF ;CLEAR UNWANTED INFO
5424 027272 006237 001376      ASR      CYLDIF          ;RIGHT JUSTIFY
5425 027276 006237 001376      ASR      CYLDIF
5426 027302 006237 001376      ASR      CYLDIF
5427 027306 006237 001376      ASR      CYLDIF
5428 027312 023727 001376 000777      CMP      CYLDIF,#777      ;CHK TO SEE IF RET IN COMPL. FORM
5429 027320 001002                BNE      IS              ;BR IF NO
5430 027322 005037 001376      CLR      CYLDIF          ;ELSE CLEAR
5431 027326 000207                RTS      PC
5432
5433      ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
5434
5435 027330 013737 005470 001400  RCYLA:  MOV      H.B2,CYLADD
5436 027336 042737 160017 001400      BIC      #1C<M.CADD>,CYLADD ;CLEAR UNWANTED INFO
5437 027344 006237 001400      ASR      CYLADD          ;RIGHT JUSTIFY
5438 027350 006237 001400      ASR      CYLADD
5439 027354 006237 001400      ASR      CYLADD
5440 027360 006237 001400      ASR      CYLADD
5441 027364 000207                RTS      PC
5442
5443      ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
5444
5445 027366 013737 005474 001422  RSEC:   MOV      H.B3,SECTOR
5446 027374 042737 177017 001422      BIC      #1C<M.SECT>,SECTOR ;CLEAR UNWANTED INFO
5447 027402 006237 001422      ASR      SECTOR          ;RIGHT JUSTIFY
5448 027406 006237 001422      ASR      SECTOR
5449 027412 006237 001422      ASR      SECTOR
5450 027416 006237 001422      ASR      SECTOR
5451 027422 000207                RTS      PC
5452
5453      ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
5454
5455 027424 013737 005474 001512  RHEAD:  MOV      H.B3,HEADA
5456 027432 042737 170777 001512      BIC      #1C<M.HEAD>,HEADA ;CLEAR UNWANTED INFO
5457 027440 006237 001512      ASR      HEADA          ;RIGHT JUSTIFY IT
5458 027444 000337 001512      SWAB    HEADA
5459 027450 000207                RTS      PC
5460
5461      ;FIND SECTOR 17
5462      ;RETURN IF NOT FOUND
5463      ;RETURN +4 IF FOUND
5464
5465 027452 013737 001436 005444  FSEC17: MOV      T5000,TEMP1    ;SETUP TIMEOUT
5466 027460 004737 027052                IS:      JSR      PC,ROSEC    ;READ SECTOR
5467 027464 023727 001422 000021      CMP      SECTOR,#17.      ;TEST FOR SECTOR 17

```



```

027472 001014          BNE      25          :BR IF NOT 17
027474 004737 027052  JSR      PC,RDSEC
027500 023727 001422 000021  CMP      SECTOR,#17.
027506 001412          BEQ      35          :BR IF READ SAME TWICE
027510 004737 027052  JSR      PC,RDSEC  :ELSE TRY 1 MORE TIME
027514 023727 001422 000021  CMP      SECTOR,#17.
027522 001404          BEQ      35          :BR IF 17
027524 005337 005444  25:     DEC      TEMP1
027530 001353          BNE      15          ;TRY AGAIN
027532 000207          RTS      PC
027534 062716 000004  35:     ADD      #4,(SP)
027540 000207          RTS      PC          :SKIP OVER ERROR

:
: FIND DESIRED CYL DIFF
: RETURN IF NOT FOUND
: RETURN+6 IF FOUND
:
027542 013737 001436 005444  FCYL:   MOV      T5000,TEMP1  :SETUP TIMEOUT
027550 004737 027122  15:     JSR      PC,RDCYLD
027554 023737 001376 005446  CMP      CYLDIF,TEMP2  :TEST FOR CYL DIFF
027562 001014          BNE      25          :BR IF NOT FOUND
027564 004737 027122          JSR      PC,RDCYLD
027570 023737 001376 005446  CMP      CYLDIF,TEMP2
027576 001412          BEQ      35          :BR IF READ SAME TWICE
027600 004737 027122  JSR      PC,RDCYLD  :ELSE TRY 1 MORE TIME
027604 023737 001376 005446  CMP      CYLDIF,TEMP2
027612 001404          BEQ      35
027614 005337 005444  25:     DEC      TEMP1
027620 001353          BNE      15          ;TRY AGAIN
027622 000207          RTS      PC
027624 062716 000006  35:     ADD      #6,(SP)
027630 000207          RTS      PC          :SKIP OVER ERROR

:
: ROUTINE TO FIND HEADS HOME IN RMR2 WORD 1 BEFORE SPECIFIED DELAY
: ENTER WITH TIME IN SECONDS IN TEMP2
: RETURN IF NOT FOUND
: RETURN+2 IF FOUND - SKIP OVER ERROR
:
027632 012737 177777 005444  F4DMM:  MOV      #1,TEMP1      :ALL 1'S
027640 012765 000001 000026  MOV      #1,RMR1(R5)  :WORD 1
027646 004737 026420  15:     JSR      PC,GSTAT
027652 032737 000040 005434  BIT      #0,H0MM,HMR2
027660 001007          BNE      25
027662 005337 005444  DEC      TEMP1
027666 001367          BNE      15
027670 005337 005446  DEC      TEMP2
027674 001356          BNE      F4DMM

```

06-00-76 09:44 000002 005444 000026 005434 005444

MAY 27 (1006) 06-00-76 09:54 PAGE 106
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEG 0106

000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200
000201
000202
000203
000204
000205
000206
000207
000208
000209
000210
000211
000212
000213
000214
000215
000216
000217
000218
000219
000220
000221
000222
000223
000224
000225
000226
000227
000228
000229
000230
000231
000232
000233
000234
000235
000236
000237
000238
000239
000240
000241
000242
000243
000244
000245
000246
000247
000248
000249
000250
000251
000252
000253
000254
000255
000256
000257
000258
000259
000260
000261
000262
000263
000264
000265
000266
000267
000268
000269
000270
000271
000272
000273
000274
000275
000276
000277
000278
000279
000280
000281
000282
000283
000284
000285
000286
000287
000288
000289
000290
000291
000292
000293
000294
000295
000296
000297
000298
000299
000300
000301
000302
000303
000304
000305
000306
000307
000308
000309
000310
000311
000312
000313
000314
000315
000316
000317
000318
000319
000320
000321
000322
000323
000324
000325
000326
000327
000328
000329
000330
000331
000332
000333
000334
000335
000336
000337
000338
000339
000340
000341
000342
000343
000344
000345
000346
000347
000348
000349
000350
000351
000352
000353
000354
000355
000356
000357
000358
000359
000360
000361
000362
000363
000364
000365
000366
000367
000368
000369
000370
000371
000372
000373
000374
000375
000376
000377
000378
000379
000380
000381
000382
000383
000384
000385
000386
000387
000388
000389
000390
000391
000392
000393
000394
000395
000396
000397
000398
000399
000400
000401
000402
000403
000404
000405
000406
000407
000408
000409
000410
000411
000412
000413
000414
000415
000416
000417
000418
000419
000420
000421
000422
000423
000424
000425
000426
000427
000428
000429
000430
000431
000432
000433
000434
000435
000436
000437
000438
000439
000440
000441
000442
000443
000444
000445
000446
000447
000448
000449
000450
000451
000452
000453
000454
000455
000456
000457
000458
000459
000460
000461
000462
000463
000464
000465
000466
000467
000468
000469
000470
000471
000472
000473
000474
000475
000476
000477
000478
000479
000480
000481
000482
000483
000484
000485
000486
000487
000488
000489
000490
000491
000492
000493
000494
000495
000496
000497
000498
000499
000500
000501
000502
000503
000504
000505
000506
000507
000508
000509
000510
000511
000512
000513
000514
000515
000516
000517
000518
000519
000520
000521
000522
000523
000524
000525
000526
000527
000528
000529
000530
000531
000532
000533
000534
000535
000536
000537
000538
000539
000540
000541
000542
000543
000544
000545
000546
000547
000548
000549
000550
000551
000552
000553
000554
000555
000556
000557
000558
000559
000560
000561
000562
000563
000564
000565
000566
000567
000568
000569
000570
000571
000572
000573
000574
000575
000576
000577
000578
000579
000580
000581
000582
000583
000584
000585
000586
000587
000588
000589
000590
000591
000592
000593
000594
000595
000596
000597
000598
000599
000600
000601
000602
000603
000604
000605
000606
000607
000608
000609
000610
000611
000612
000613
000614
000615
000616
000617
000618
000619
000620
000621
000622
000623
000624
000625
000626
000627
000628
000629
000630
000631
000632
000633
000634
000635
000636
000637
000638
000639
000640
000641
000642
000643
000644
000645
000646
000647
000648
000649
000650
000651
000652
000653
000654
000655
000656
000657
000658
000659
000660
000661
000662
000663
000664
000665
000666
000667
000668
000669
000670
000671
000672
000673
000674
000675
000676
000677
000678
000679
000680
000681
000682
000683
000684
000685
000686
000687
000688
000689
000690
000691
000692
000693
000694
000695
000696
000697
000698
000699
000700
000701
000702
000703
000704
000705
000706
000707
000708
000709
000710
000711
000712
000713
000714
000715
000716
000717
000718
000719
000720
000721
000722
000723
000724
000725
000726
000727
000728
000729
000730
000731
000732
000733
000734
000735
000736
000737
000738
000739
000740
000741
000742
000743
000744
000745
000746
000747
000748
000749
000750
000751
000752
000753
000754
000755
000756
000757
000758
000759
000760
000761
000762
000763
000764
000765
000766
000767
000768
000769
000770
000771
000772
000773
000774
000775
000776
000777
000778
000779
000780
000781
000782
000783
000784
000785
000786
000787
000788
000789
000790
000791
000792
000793
000794
000795
000796
000797
000798
000799
000800
000801
000802
000803
000804
000805
000806
000807
000808
000809
000810
000811
000812
000813
000814
000815
000816
000817
000818
000819
000820
000821
000822
000823
000824
000825
000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845
000846
000847
000848
000849
000850
000851
000852
000853
000854
000855
000856
000857
000858
000859
000860
000861
000862
000863
000864
000865
000866
000867
000868
000869
000870
000871
000872
000873
000874
000875
000876
000877
000878
000879
000880
000881
000882
000883
000884
000885
000886
000887
000888
000889
000890
000891
000892
000893
000894
000895
000896
000897
000898
000899
000900
000901
000902
000903
000904
000905
000906
000907
000908
000909
000910
000911
000912
000913
000914
000915
000916
000917
000918
000919
000920
000921
000922
000923
000924
000925
000926
000927
000928
000929
000930
000931
000932
000933
000934
000935
000936
000937
000938
000939
000940
000941
000942
000943
000944
000945
000946
000947
000948
000949
000950
000951
000952
000953
000954
000955
000956
000957
000958
000959
000960
000961
000962
000963
000964
000965
000966
000967
000968
000969
000970
000971
000972
000973
000974
000975
000976
000977
000978
000979
000980
000981
000982
000983
000984
000985
000986
000987
000988
000989
000990
000991
000992
000993
000994
000995
000996
000997
000998
000999
001000

```

:
:
: ROUTINE TO FIND LOAD HEADS IN R/MR2 WORD 1 BEFORE SMS
: RETURN IF NOT FOUND
: RETURN+2 IF FOUND: SKIP OVER ERROR
:
LOAD: MOV 0250, TEMP1
      MOV 01, R/MR1(RS)      :WORD 1
18:   JSR PC, CSTAT
      BIT 00, LOAD, R/MR2
      BNE 25
      DEC TEMP1
      IS

```

000002

005444
000026
005434
005444

28: RYS PC
RCC 02, (SP) :SKIP OVER ERROR
RYS PC
: ROUTINE TO FIND LOAD HEADS IN R/MR2 WORD 1 BEFORE SMS
: RETURN IF NOT FOUND
: RETURN+2 IF FOUND: SKIP OVER ERROR
LOAD: MOV 0250, TEMP1
MOV 01, R/MR1(RS) :WORD 1
18: JSR PC, CSTAT
BIT 00, LOAD, R/MR2
BNE 25
DEC TEMP1
IS

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094

027744 000207
027746 062716 000002
027752 000207

027754 010046
027756 010146
027760 012700 001554
027764 005001
027766 013737 001510 001514
027774 006337 001514
030000 006337 001514
030004 006337 001514
030010 006337 001514
030014 006337 001514
030020 013737 001516 001520
030026 006337 001520
030032 006337 001520

030036 013720 001402
030042 010110
030044 053710 001514
030050 053710 001520
030054 004737 030134

030060 013737 001402 005444
030066 011037 005446
030072 043737 001402 005446
030100 042037 005444
030104 053737 005444 005446
030112 013720 005446

030116 005201
030120 020127 000026
030124 001344

030126 012601
030130 012600
030132 000207

030134 010246
030136 005737 001516
030142 001016
030144 012702 002400
030150 004737 030204
030154 052710 100000

030160 012702 003400
030164 004737 030204

```
25:   RTS      PC
      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)      ;SAVE RO
        MOV      RI,-(SP)      ;SAVE RI
        MOV      #FHDTAB,RO    ;HEADER WORD TABLE ADDR
        CLR      R1            ;SECTOR COUNTER
        MOV      HEAD,HD1
        ASL      HD1
        ASL      HD1
        ASL      HD1
        ASL      HD1
        ASL      HD1
        ASL      HD1
        MOV      FORMAT,FMT1   ;SETUP HEAD # FOR WORD 2 OF HEADER
        SWAB    FMT1
        ASL      FMT1         ;SETUP FORMAT FOR WORD 2 OF HEADER

15:   MOV      CALADD,(RO)+    ;HEADER WORD 1-CYL ADDR
        MOV      R1,(RO)      ;HEADER WORD 2-SECTOR NO
        BIS      HD1,(RO)     ;
        BIS      FMT1,(RO)    ;
        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     IS           ;BR IF NO

        MOV      (SP)+,R1     ;RESTOR R1
        MOV      (SP)+,RO     ;RESTOR RO
        RTS      PC

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

SECFLG: MOV      R2,-(SP)      ;SAVE R2
        TST      FORMAT
        BNE     IS           ;BR IF 20 SECTOR FORMAT
        MOV      #BSE22H+8.,R2
        JSR      PC,FLGTST    ;GET HARDWARE DETECTED FLAG
        BIS      #BIT15,(RO)  ;RETURN HERE IF GOOD SECTOR

        MOV      #BSE225+8.,R2 ;ELSE RETURN HERE
        JSR      PC,FLGTST    ;GET SOFTWARE DETECTED FLAG
```

E09

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
DZ6JC.P11 06-OCT-76 09:44

MAY11 27:1006) 06-OCT-76 09:54 PAGE 108
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEG 0108

```

5595 030170 052710 040000      BIS      #BIT14,(R0)      ;RETURN HERE IF GOOD SECTOR
5596
5597 030174 012602      MOV      (SP)+,R2      ;ELSE RETURN HERE
5598 030176 000207      RTS      PC
5600 030200 012602      15:     MOV      (SP)+,R2      ;RESTORE R2
5601 030202 000207      RTS      PC
5602
5603
5604
5605
5606
5607
5608
5609 030204 010346      FLGTS*: MOV      R3,-(SP)      ;SAVE R3
5610
5611 030206 021227 177777      15:     CMP      (R2),#-1      ;SEE IF ALL 1'S
5612 030212 001002      BNE      25            ;BR IF NC
5613 030214 012603      MOV      (SP)+,R3      ;RESTORE R3
5614 030216 000207      RTS      PC
5615
5616 030220 022237 001402      25:     CMP      (R2)+,0ALADD    ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
5617 030224 001403      BEQ      35            ;
5618 030226 062702 000002      ADD      #2,R2        ;GO TO NEXT CYL WORD IN TABLE
5619 030232 000765      BR       15
5620
5621 030234 013703 001510      35:     MOV      HEAD,R3      ;GET HEAD # FROM FHTAB ROUTINE
5622 030240 000303      SWAB     R3
5623 030242 050103      BXS     R1,R3          ;ADD SECTOR # FROM FHTAB ROUTINE
5624 030244 022203      CMP      (R2)+,R3      ;SEE IF SECTOR/HEAD COMPARE
5625
5626 030246 001401      BEQ      45            ;& INCR PTR TO NEXT CYL WORD
5627 030250 000756      BR       15            ;BR IF COMPARE
5628
5629 030252 012603      45:     MOV      (SP)+,R3      ;RESTORE R3
5630 030254 062716 000004      ADD      #4,(SP)      ;INCREMENT RET ADDR
5631 030260 000207      RTS      PC
5632
5633
5634
5635
5636
5637 030262 010046      SORT:   MOV      R0,-(SP)      ;SAVE R0
5638 030264 010146      MOV      R1,-(SP)      ;SAVE R1
5639 030266 004737 027052      JSR      PC,RDSEC
5640 030272 062737 000001 001422      ADD      #1,SECTOR
5641 030300 004737 030370      JSR      PC,MULT6      ;MULT SECTOR BY 6
5642
5643 030304 012700 000204      MOV      #132,R0
5644 030310 163700 001422      SUB     SECTOR,R0      ;R0-SECTOR TO R0 = INDEX
5645 030314 010037 001422      MOV     R0,SECTOR
5646 030320 062737 001760 001422      ADD     #RHTAB,SECTOR  ;SAVE INDEX
5647
5648 030326 062700 001760      ADD     #RHTAB,R0      ;INDEX TO BOT HALF OF RHTAB
5649 030332 012701 002164      MOV     #SRTTAB,R1     ;INDEX TO TOP HALF OF SRTTAB
5650

```

THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES.
ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE225, ETC.) IN TEMP!
RETURN IF NO COMPARE
RETURN+4 IF COMPARE

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

```

5651
5652 030336 012021 15: MOV (R0)+,R1+ ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
5653 030340 020027 002164 CMP R0,#RHTAB+132.
5654 030344 001374 BNE 15
5655
5656 030346 012700 001760 25: MOV #RHTAB,R0 ;PUT TOP OF RHTAB TO BOT OF SRTTAB
5657 030352 012021 MOV (R0)+,(R1)+
5658 030354 020037 001422 CMP R0,SECTOR
5659 030360 001374 BNE 25
5660
5661 030362 012601 MOV (SP)+,R1 ;RESTOR R1
5662 030364 012600 MOV (SP)+,R0 ;RESTOR R0
5663 030366 000207 RTS PC
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706

;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
MULT6: ASL SECTOR ;2 X SECTOR
MOV SECTOR,-(SP)
ASL SECTOR ;4 X SECTOR
ADD (SP)+,SECTOR ;(4 X 5)+(2 X 5) = 6 X SECTOR
RTS PC

;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
;CMD. IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
;CYLINDER AT CYL 410, TRACK 2.
;RETURN IF SECTOR NOT LISTED IN BSE TABLE. ERROR CONDITION.
;RETURN+2 IF LISTED, SKIP OVER ERROR
TRUERR: MOV R4,-(SP) ;SAVE R4
BIT #CFMT,HCS1 ;CHECK FORMAT
BNE 35 ;BR FOR 20 SECTOR FORMAT
;NOTE, 20 SECTOR FORMAT NOT
;DONE IN THIS PROGRAM
MOV #BSE22H+8.,R4
JSR PC,TERR1 ;SEE IF ON HARDWARE DETECTED TABLE
BR 35 ;RETURN HERE IF YES
MOV #BSE22S+8.,R4 ;ELSE RETURN HERE
JSR PC,TERR1 ;& SEE IF ON SOFTWARE DETECTED TABLE
BR 35 ;RETURN HERE IF YES
15: MOV (SP)+,R4 ;RESTORE R4
RTS PC ;RETURN WITHOUT JUMPING OVER ERROR
35: MOV (SP)+,R4 ;RESTORE R4
ADD #2,(SP) ;SKIP OVER ERROR ON RETURN
RTS PC
;THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST

```

```

5707
5708
5709
5710
5711 030464 021427 177777
5712 030470 001405
5713 030472 022437 001370
5714 030476 001405
5715 030500 005724
5716 030502 000770
5717
5718 030504 062716 000002
5719 030510 000207
5720
5721 030512 022437 005416
5722 030516 001401
5723 030520 000761
5724
5725 030522 000207
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735 030524 005037 001412
5736 030530 005737 005552
5737 030534 001004
5738 030536 012777 000100 150602
5739 030544 000207
5740 030546 012777 177777 150566
5741 030554 012777 000135 150556
5742 030562 000207
5743
5744
5745 030564 005037 001412
5746 030570 005337 001406
5747 030574 001010
5748 030576 013737 001404 001406
5749 030604 005337 001410
5750 030610 001002
5751 030612 005237 001412
5752 030616 000002
5753
5754
5755
5756 030620 005737 005552
5757 030624 001003
5758 030626 005077 150514
5759 030632 000207
5760 030634 005077 150500
5761 030640 000207
5762

```

```

:THE BSE TABLE FOR THE ABOVE SUBROUTINE.
:RETURN IF FOUND ON TABLE
:RETURN+2 IF NOT FOUND
ERR1:  CMP      (R4),#-1      :SEE IF ALL 1'S
      BEQ      1$           :BR IF YES, NOT ON TABLE
      CMP      (R4)+,CCYL    :SEE IF CYL MATCH
      BEQ      2$           :BR IF YES
      TST      (R4)+        :ELSE ADV TO NEXT CYL WORD
      BR       TERR1        :& TRY AGAIN.
1$:    ADD      #2,(SP)
      RTS      PC
2$:    CMP      (R4)+,HDA    :SEE IF SECTOR & TRACK MATCH
      BEQ      3$           :BR IF YES
      BR       TERR1        :OR TRY AGAIN
3$:    RTS      PC

:
:ROUTINE TO TURN L OR P CLOCK INTERRUPT ON
CLKON: CLR      TIMUP
      TST      PCLKF
      BNE     1$           :BRANCH IF P-CLOCK PRESENT
      MOV     #100,2LKS    :L-CLOCK, ENABLE INT
      RTS     PC
1$:    MOV     #-1,2PKSB   :P-CLOCK, ALL 1'S
      MOV     #135,2PKS   :ENABLE INT, CT JP. REP INT
      RTS     PC          :LINE FREQ & RUN

:KW11-L & KW11-P INTERRUPT HANDLER
CLOCK: CLR      TIMUP
      DEC     COUNT
      BNE     1$
      MOV     HZ,COUNT
      DEC     SEC
      BNE     1$
      INC     TIMUP       :SORRY, TIME IS UP
1$:    RTI

:ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
CLKOF: TST      PCLKF
      BNE     1$           :BRACH IF P-CLOCK PRESENT
      CLR     2LKS        :L-CLOCK, CLEAR INTERRUPT
      RTS     PC
1$:    CLR     2PKS       :P-CLOCK, CLEAR INTERRUPT
      RTS     PC

```

5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818

030642 010046
030644 010146
030646 012700 000021
030652 005001
030654 000241

030656 006137 005444
030662 103001
030664 005201
030666 005300
030670 001372

030672 032701 000001
030676 001003
030700 052737 100000 005444
030706 012601
030710 012600
030712 000207

030714 032777 001000 150216
030722 001406
030724 105737 001103
030730 001403
030732 013716 001110
030736 000002

030740 011637 001110
030744 000002

030746 005777 150174
030752 104410
030754 012600
030756 020027 000040
030762 001406

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

SBPAR: MOV R0, -(SP) ;SAVE R0
MOV R1, -(SP) ;SAVE R1
MOV #17, R0 ;SHIFT COUNTER
CLR R1 ;COUNT # OF 1'S IN TEMP1
CLC ;CLEAR CARRY

15: ROL TEMP1
BCC 25 ;BR IF CARRY CLEAR
INC R1 ;COUNT # OF 1'S
25: DEC R0 ;SHIFT COUNTER
BNE 15

35: BIT #BIT0, R1
BNE 35 ;BR IF ODD # IN R0
BIS #M.PAR, TEMP1 ;SET PARITY BIT
MOV (SP)+, R1 ;RESTORE R1
MOV (SP)+, R0 ;RESTORE R0
RTS PC

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

SCOPE15: BIT #SW9, \$SWR ;LOOP ON ERROR?
BEQ 15 ;BR IF NO
TSTB \$ERFLG ;HAD ERROR?
BEQ 15 ;BR IF NO
MOV \$LPERR, (SP)
RTI

15: MOV (SP), \$LPERR ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
RTI

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

CCSP: TST \$STKB ;CLEAR DONE FLAG
RDCHR ;READ CHAR FROM TTY
MOV (SP)+, R0 ;GET CHAR OFF STACK
CMP R0, #SPBAR ;SEE IF SPACE
BEQ 15 ;BR IF YES.

```

5819
5820 030764 020027 000005      CMP      RD,#5      ;SEE IF CONTROL-E
5821 030770 001405      BEQ      2$        ;BR IF YES
5822 030772 104401 040652      TYPE    MSG31     ;
5823 030776 000763      BR       CCSF      ;TRY AGAIN
5824
5825 031000 062716 000004      1$:     ADD      #4,(SP)
5826 031004 000207      2$:     RTS       PC
5827
5828
5829
5830
5831
5832
5833
5834 031006 005777 150134      GETSP:  TST      0$TKB      ;CLEAR DONE FLAG
5835 031012 104410      RDCHR   ;READ CHAR OFF TTY
5836 031014 012600      MCV     (SP)+,RD     ;GET CHAR OFF STACK
5837 031016 020027 000040      CMP     RD,#SPBAR   ;SEE IF SPACE
5838 031022 001403      BEQ     1$          ;EXIT IF YES
5839 031024 104401 040652      TYPE    MSG31     ;
5840 031030 000766      BR      GETSP      ;TRY AGAIN
5841 031032 000207      1$:     RTS       PC
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
031034 022626      STOP:   CMP      (SP)+,(SP)+ ;RESTORE STACK FROM INTERRUPT
031036 004737 026772      STOP1: JSR      PC,SUBCLR   ;CERR AFTER
031042 104024      ERROR  24
031044 005737 005370      TST     UNLD      ;SEE IF HEADS UNLOADED
031050 001440      BEQ     3$         ;BR IF NO
031052 005737 000042      TST     42        ;SEE IF MANUAL OR AUTO MODE
031056 001403      BEQ     1$         ;BR IF MANUAL MODE
031060 104401 045367      TYPE    MSG74     ;PGM ABORT PENDING
031064 000402      BR     2$         ;
031066 104401 045435      1$:     TYPE    .MSG75   ;HALT PENDING
031072
031072 004737 026772      JSR     PC,SUBCLR   ;CERR AFTER SCLR
031076 104024      ERROR  24
031100 032737 010000 005434      BIT     #0.SPIN,HMR2 ;SEE IF SPINDLE ALREADY ON
031106 001021      BNE    64$        ;BR IF YES
031110 104401 040524      TYPE    ,MSG29    ;PLEASE WAIT, HEADS BEING LOADED
031114 012765 000011 000000      MOV     #SRTSPL,RKCS1(R5) ;START SPINDLE CMD
031122 013737 001426 005444      MOV     T10,TEMP1
031130 004737 025062      JSR     PC,FRDY    ;FIND CONTR RDY
031134 104143      ERROR  143        ;CONTR RDY NOT SET AFTER CMD
031136 013737 001434 005446      MOV     T100,TEMP2 ;FIND ATTN
031144 004737 025376      JSR     PC,FATT1   ;NO ATTN AFTER CMD
031150 104144      ERROR  144

```



```

031:52      031:52      005737      005372      64$:      TST      BADHDR      ;SEE IF HEADERS VALID
031:56      001466      005237      005374      BEQ      4$      ;BR IF YES
031:60      012765      100000      000000      INC      HPEND
031:64      013765      001222      000010      MOV      #CCLR,RKCS1(R5)
031:68      012765      000013      000000      MOV      $UNIT,RKCS2(R5)
031:72      012765      000013      000000      MOV      #RECAL,RKCS1(R5) ;RECAL CMD
031:76      013737      001426      005444      ;RESET CYL DIFF/OFFSET & CYL ADDR REG
031:80      004737      025062      JSR      T10,TEMP1 ;IN RKMR2 & RKMR3 RESP.
031:84      104124      ERROR      124      ;SETUP TIMEOUT
031:88      013737      001426      005444      JSR      PC,FRDY ;FIND RDY
031:92      104124      ERROR      124      ;RDY NOT SET AFTER RECAL CMD
031:96      012765      000001      000026      MOV      #1,RKMR1(R5) ;SELECT WORD 1
031:100     004737      026420      JSR      PC,GSTAT
031:104     032737      020000      005434      BIT      #D,RTZ,HMR2
031:108     001001      BNE      65$
031:112     104214      ERROR      214      ;RTZ NOT SET DURING RECAL CMD
031:116     013737      001426      005446      65$:     MOV      T10,TEMP2 ;SETUP TIMEOUT
031:120     004737      025376      JSR      PC,FATT1 ;FIND ATTN
031:124     104055      ERROR      55      ;NO ATTN AFTER RECAL CMD
031:128     012765      100000      000000      MOV      #CCLR,RKCS1(R5)
031:132     013765      001222      000010      MOV      $UNIT,RKCS2(R5) ;DRIVE#
031:136     012765      000005      000000      MOV      #CLEAR,RKCS1(R5) ;DRIVE CLEAR CMD
031:140     013737      001426      005444      MOV      T10,TEMP1 ;SETUP TIMEOUT
031:144     004737      025062      JSR      PC,FRDY ;FIND RDY
031:148     104151      ERROR      151      ;NO RDY AFTER DRIVE CLEAR CMD
031:152     004737      025344      JSR      PC,TSTATN ;TEST FOR ATTN
031:156     000401      BR      66$
031:160     104154      ERROR      154      ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
031:164     031:330     000137      031:370     66$:     JMP      FORM
031:168     031:330     000137      031:370     ;WRITE VALID FORMATS
031:172     005737      000042      4$:     TST      42      ;SEE IF MANUAL OR AUTO MODE
031:176     001406      BEQ      5$      ;BR IF MANUAL MODE
031:180     104401      045472      TYPE     ,MSG76 ;PGM ABORTED
031:184     005037      024430      CLR      $EOPCT ;SET UP EOP TO EXIT TO MONITOR
031:188     000137      024402      JMP      $EOP
031:192     104401      045514      5$:     TYPE     ,MSG77 ;CPU HALTED
031:196     000000      HALT
031:200     000137      010656      JMP      ST5 ;START OVER IF CONTINUE PRESSED
031:204     031:370
FORM:
.SBTTL UNEXPECTED TIMEOUT HANDLER
; THIS ROUTINE IS ENTERED IF THERE IS
; A. NON EXISTANT MEMORY (NO SSYN)
; B. BOUNDARY ERROR
; C. STACK OVERFLOW

```

```

5931
5932 031370 011600      BADTMO: MOV      (SP),RO      ;SAVE PC WHERE TIMEOUT OCCURRED.
5933 031372 005740      TST      -(RO)      ;GET PC BEFORE UPDATE
5934 031374 032777 020000 147536  BIT      #SW13,2SWR  ;INHIBIT ERR TYP0UT?
5935 031402 001005      BNE      1$      ;YES, DON'T TYPE
5936 031404 104401 046044  TYPE     EM3      ;ABORT TESTS UNEXP T.O. @ PC=
5937 031410 010046      MOV      RO,-(SP)  ;SAVE RO FOR TYPEOUT
5938
5939 031412 104403      TYPOS    ;TYPE PC
5940 031414 006      .BYTE   6      ;GO TYPE--OCTAL ASCII
5941 031415 000      .BYTE   0      ;TYPE 6 DIGIT(S)
5942 031416 032777 001000 147514 1$:  BIT      #SW9,2SWR  ;SUPPRESS LEADING ZEROS
5943 031424 001403      BEQ      2$      ;LOOP ON ERROR?
5944 031426 022626      CMP      (SP)+,(SP)+ ;NO, BRANCH
5945 031430 000177 147452  JMP      2$LPADR  ;YES, RESTORE STACK
5946
5947 031434 032777 040000 147476 2$:  BIT      #SW14,2SWR ;GO TO STARTING ADDR OF TEST
5948 031442 001401      BEQ      3$      ;THAT GAVE BAD TIMEOUT
5949 031444 000002      RTI      ;LOOP ON TEST?
5950
5951 031446 000000      3$:  HALT      ;NO BRANCH
5952
5953
5954
5955
5956
5957 031450 022626      CMP      (SP)+,(SP)+ ;UNEXPECTED TIME OUT OCCURRED
5958 031452 000137 024402  JMP      $EOP     ;AS INDICATED, YOU CAN LOOP ON
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971 031510 012706 001100 2$:  MOV      #STACK,SP ;ERROR, LOOP ON TEST OR INHIBIT
5972 031514 000177 147370  JMP      2$LPERR ;ERROR TYPEOUT BY SETTING THOSE
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986

```

.SBTTL MEMORY CHECK ENABLE TRAP

```

5961
5962 031456 012737 031472 001176 MEMERR: MOV      #1$, $ESCAPE
5963 031464 011637 001354  MOV      (SP),TRAPPC ;STORE PC
5964 031470 104202      ERROR    202      ;UNEXP MEM PARITY ERROR
5965

```

```

5966 031472 005037 001176 1$:  CLR      $ESCAPE
5967 031476 032777 001000 147434  BIT      #SW9,2SWR  ;CHECK IF LOOP ON ERROR
5968 031504 001001      BNE      2$      ;YES, FORCE STACK AND TRY AGAIN
5969 031506 000002      RTI      ;ELSE RETURN
5970

```

```

5971 031510 012706 001100 2$:  MOV      #STACK,SP ;INIT STACK
5972 031514 000177 147370  JMP      2$LPERR
5973

```

.SBTTL RK06 INTERRUPT HANDLER

```

5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986

```

```

5976 031520 000240      INTER: NOP
5977 031522 000240      NOP
5978 031524 000240      NOP
5979 031526 011600      MOV      (SP),RO  ;SAVE PC WHERE INT OCCURRED.
5980 031530 005740      TST      -(RO)   ;GET PC BEFORE UPDATE.
5981 031532 104401 037354  TYPE     MSG6     ;INT AT PC=
5982 031536 010046      MOV      RO,-(SP) ;SAVE RO FOR TYPEOUT
5983
5984
5985
5986

```

```

5987 031544 000000      HALT
5988 031546 000240      NOP
5989 031550 000240      NOP
5990 031552 000002      RTI
5991
5992      .SBTTL  POWER DOWN AND UP ROUTINES
5993
5994      ;POWER DOWN ROUTINE
5995
5996 031554 012737 031566 000024 SPWRDN: MOV      #SPWRUP,PWRVEC ;SET UP VECTOR
5997 031562 000000      HALT
5998 031564 000776      BR      .-2      ;HANG UP.
5999
6000      ;POWER UP ROUTINE
6001
6002 031566 005037 031640 SPWRUP: CLR      $PWRCT      ;WAIT LOOP FOR TTY
6003 031572 005237 031640 15:      INC      $PWRCT      ;WAIT FOR THE INCR
6004 031576 001375      BNE      15          ;OF WORD
6005 031600 012737 031554 000024 MOV      #SPWRDN,PWRVEC ;SET POWER DOWN VECTOR
6006 031606 012737 000340 000026 MOV      #PR7,PWRVEC+2 ;PRIORITY 7
6007 031614 012737 000340 000036 MOV      #PR7,TRAPVEC+2 ;LOCKOUT ALL INTERRUPTS FOR TRAPS
6008 031622 012706 001100 MOV      #STACK,SP      ;INITIALIZE STACK
6009 031626 104401 037550 TYPE      .MSG11      ;REPORT POWER FAIL
6010 031632 000005      RESET
6011 031634 000137 012434 JMP      PFSRT
6012
6013 031640 000000      $PWRCT: 0      ;WAIT COUNT FOR TTY
6014
6015
6016      ;DIVISION UTILITY ROUTINE
6017
6018      ;R0-R1-R2-R3=DIVIDEND
6019      ;R4-R5=DIVISOR
6020      ;R0-R1=REMAINDER AFTER DIVISION
6021      ;R2-R3=QUOTIENT AFTER DIVISION
6022      ;ENTER WITH JSR PC,M.DPID
6023
6024 031642 012746 000040 M.DPID: MOV      #40,-(SP) ;COUNTER FOR DIVISION CYCLES
6025 031646 010446      MOV      R4,-(SP) ;HI ORDER
6026 031650 010546      MOV      R5,-(SP) ;LO ORDER TO THE STACK
6027 031652 005466 000002 NEG      2(SP) ;FORM NEGATIVE
6028 031656 005416      NEG      @SP ;VERSION OF DIVISOR
6029 031660 005666 000002 SBC      2(SP)
6030 031664 061601      ADD      @SP,R1
6031 031666 005500      ADC      R0 ;PERFORM INIT SUBT.
6032 031670 066600 000002 ADD      2(SP),R0
6033 031674 103445      BCS      M.DP50 ;IF CARRY THEN OVERFLOW HAS OCCURRED
6034 031676 005046      CLR      -(SP) ;THIS IS A LONGER LASTING CARRY BIT
6035 031700 006103 M.DP40: ROL      R3
6036 031702 006102      ROL      R2
6037 031704 006101      ROL      R1
6038 031706 006100      ROL      R0
6039 031710 005716      TST      @SP ;TEST CARRY INDICATOR
6040 031712 001410      BEQ      M.DP41 ;IF TO CARRY THEN ADD, ELSE SUBT.
6041 031714 005016      CLR      @SP ;CLEAR UP FOR NEXT TIME
6042 031716 066601 000002 ADD      2(SP),R1
  
```

6043	031722	005500		ADC	R0	:ADD -(DIVISOR)
6044	031724	005516		ADC	2SP	:SET CARRY
6045	031726	066600	000004	ADD	4(SP),R0	
6046	031732	000404		BR	M.DP42	
6047						
6048	031734	060501		M.DP41: ADD	R5,R1	
6049	031736	005500		ADC	R0	:ADD +(DIVISOR)
6050	031740	005516		ADC	2SP	:SET CARRY
6051	031742	060400		ADD	R4,R0	
6052	031744	005516		M.DP42: ADC	2SP	:SET CARRY
6053	031746	005716		TST	2SP	:TEST THE UPDATE INDICATOR
6054	031750	001401		BEQ	.+4	:IF 0, FORGET IT
6055	031752	005203		INC	R3	:NO CARRY POSSIBLE HERE
6056	031754	005366	000006	DEC	6(SP)	:DECREMENT CTR
6057	031760	003347		BGT	M.DP40	:BR IF MORE TO DO
6058	031762	006003		ROR	R3	
6059	031764	103404		BCS	M.DP44	
6060	031766	060501		ADD	R5,R1	
6061	031770	005500		ADC	R0	
6062	031772	060400		ADD	R4,R0	
6063	031774	000241		CLC		
6064						
6065	031776	006103		M.DP44: ROL	R3	
6066	032000	062706	000010	ADD	#10,SP	:ADJUST STACK BY 4 WORDS
6067	032004	000242		CLV		
6068	032006	000207		RTS	PC	
6069						
6070	032010	062706	000006	M.DP50: ADD	#6,SP	
6071	032014	000262		SEV		
6072	032016	000207		RTS	PC	
6073						

.SBTTL SCOPE HANDLER ROUTINE

```

*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER($STSTNM) INTO THE DISPLAY REG.(DISPLAY 7:0).
*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY 15:08
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1      LOOP ON TEST
*SW11=1      INHIBIT ITERATIONS
*SW09=1      LOOP ON ERROR
*SW08=1      LOOP ON TEST IN SWR<7:0>
*CALL
*          SCOPE          ;;SCOPE=IOT

```

```

$SCOPE:
CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
1$: BIT #BIT14,$SWR          ;;LOOP ON PRESENT TEST?
   BNE $OVER          ;;YES IF SW14=1
*****START OF CODE FOR THE XOR TESTER*****
$XSTR: BR 6$          ;;IF RUNNING ON THE "XOR" TESTER CHANGE
   MOV 2#ERRVEC,-(SP)      ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
   MOV 5$,$ERRVEC          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
   TST 2#177060          ;;SET FOR TIMEOUT
   MOV (SP)+,2#ERRVEC      ;;TIME OUT ON XOR?
   BR 5$SVLAD          ;;RESTORE THE ERROR VECTOR
   CMP (SP)+,(SP)+          ;;GO TO THE NEXT TEST
   MOV (SP)+,2#ERRVEC      ;;CLEAR THE STACK AFTER A TIME OUT
   BR 7$          ;;RESTORE THE ERROR VECTOR
6$: *****END OF CODE FOR THE XOR TESTER*****
   BIT #BIT08,$SWR          ;;LOOP ON SPEC. TEST?
   BEQ 2$          ;;BR IF NO
   CMPB 2SWR,$STSTNM          ;;ON THE RIGHT TEST? SWR<7:0>
   BEQ $OVER          ;;BR IF YES
   TSTB $ERFLG          ;;HAS AN ERROR OCCURRED?
   BEQ 3$          ;;BR IF NO
   CMPB $ERMAX,$ERFLG          ;;MAX. ERRORS FOR THIS TEST OCCURRED?
   BHI 3$          ;;BR IF NO
   BIT #BIT09,$SWR          ;;LOOP ON ERROR?
   BEQ 4$          ;;BR IF NO
7$: MOV $LPERR,$LPADR          ;;SET LOOP ADDRESS TO LAST SCOPE
   BR $OVER
4$: CLRB $ERFLG          ;;ZERO THE ERROR FLAG
   CLR $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
   BR 1$          ;;ESCAPE TO THE NEXT TEST
3$: BIT #BIT11,$SWR          ;;INHIBIT ITERATIONS?
   BNE 1$          ;;BR IF YES
   TST $PASS          ;;IF FIRST PASS OF PROGRAM
   BEQ 1$          ;;INHIBIT ITERATIONS
   INC $ICNT          ;;INCREMENT ITERATION COUNT
   CMP $TIMES,$ICNT          ;;CHECK THE NUMBER OF ITERATIONS MADE
   BGE $OVER          ;;BR IF MORE ITERATION REQUIRED
1$: MOV #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
   MOV $MXCNT,$TIMES          ;;SET NUMBER OF ITERATIONS TO DO
$SVLAD: INCB $STSTNM          ;;COUNT TEST NUMBERS
   MOVB $STSTNM,$TESTN          ;;SET TEST NUMBER IN APT MAILBOX

```

```

6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088 032020
6089 032020 104407
6090 032022 032777 040000 147110
6091 032030 001114
6092
6093 032032 000416
6094
6095 032034 013746 000004
6096 032040 012737 032060 000004
6097 032046 005737 177060
6098 032052 012637 000004
6099 032056 000463
6100 032060 022626
6101 032062 012637 000004
6102 032066 000423
6103 032070
6104 032070 032777 000400 147042
6105 032076 001404
6106 032100 127737 147034 001102
6107 032106 001465
6108 032110 105737 001103
6109 032114 001421
6110 032116 123737 001115 001103
6111 032124 101015
6112 032126 032777 001000 147004
6113 032134 001404
6114 032136 013737 001110 001106
6115 032144 000446
6116 032146 105037 001103
6117 032152 005037 001174
6118 032156 000415
6119 032160 032777 004000 146752
6120 032166 001011
6121 032170 005737 001216
6122 032174 001406
6123 032176 005237 001104
6124 032202 023737 001174 001104
6125 032210 002024
6126 032212 012737 000001 001104
6127 032220 013737 032276 001174
6128 032226 105237 001102
6129 032232 113737 001102 001214

```

B10

MAY 11 06:27:11 06-07-76 09:54 PAGE 118
PART 3

MAY 11 06:27:11 06-07-76 09:54 PAGE 118
SCOPE HANDLER ROUTINE

SEQ 0118

```

00000000 001106  

00000001 001106  

00000002 001106  

00000003 001106  

00000004 001106  

00000005 001106  

00000006 001106  

00000007 001106  

00000008 001106  

00000009 001106  

00000010 001106  

00000011 001106  

00000012 001106  

00000013 001106  

00000014 001106  

00000015 001106  

00000016 001106  

00000017 001106  

00000018 001106  

00000019 001106  

00000020 001106  

00000021 001106  

00000022 001106  

00000023 001106  

00000024 001106  

00000025 001106  

00000026 001106  

00000027 001106  

00000028 001106  

00000029 001106  

00000030 001106  

00000031 001106  

00000032 001106  

00000033 001106  

00000034 001106  

00000035 001106  

00000036 001106  

00000037 001106  

00000038 001106  

00000039 001106  

00000040 001106  

00000041 001106  

00000042 001106  

00000043 001106  

00000044 001106  

00000045 001106  

00000046 001106  

00000047 001106  

00000048 001106  

00000049 001106  

00000050 001106  

00000051 001106  

00000052 001106  

00000053 001106  

00000054 001106  

00000055 001106  

00000056 001106  

00000057 001106  

00000058 001106  

00000059 001106  

00000060 001106  

00000061 001106  

00000062 001106  

00000063 001106  

00000064 001106  

00000065 001106  

00000066 001106  

00000067 001106  

00000068 001106  

00000069 001106  

00000070 001106  

00000071 001106  

00000072 001106  

00000073 001106  

00000074 001106  

00000075 001106  

00000076 001106  

00000077 001106  

00000078 001106  

00000079 001106  

00000080 001106  

00000081 001106  

00000082 001106  

00000083 001106  

00000084 001106  

00000085 001106

```

```

00:11:5  

146652  


```

```

MOV (SP), $LPAOR ;; SAVE SCOPE LOOP ADDRESS
MOV (SP), $LPERR ;; SAVE ERROR LOOP ADDRESS
CLR $ESCAPE ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
MOVB #1, $ERMAX ;; ONLY ALLOW ONE(1) ERROR ON NEXT "ERR"
SOVER: MOV $STNM, @DISPLAY ;; DISPLAY TEST NUMBER
MOV $LPAOR, (SP) ;; FUDGE RETURN ADDRESS
RTI ;; FIXES PS
SMXCNT: 2000 ;; MAX. NUMBER OF ITERATIONS
.SBTTL ERROR HANDLER ROUTINE

```

```

*****
* THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
* SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
* AND GO TO TYPERR ON ERROR
* THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
* SW15=1 HALT ON ERROR
* SW13=1 INHIBIT ERROR TYPEOUTS
* SW10=1 BELL ON ERROR
* SW09=1 LOOP ON ERROR
* CALL
* ERROR N ;; ERROR=EMT AND N=ERROR ITEM NUMBER

```

```

ERROR:
7S: CKSWR ;; TEST FOR CHANGE IN SOFT-SWR
INCB $ERFLG ;; SET THE ERROR FLAG
BEQ 7S ;; DON'T LET THE FLAG GO TO ZERO
MOV $STNM, @DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
BIT #BIT10, @SWR ;; BELL ON ERROR?
BEQ 1S ;; NO - SKIP
TYPE $SBELL ;; RING BELL
1S: INC $ERTTL ;; COUNT THE NUMBER OF ERRORS
MOV (SP), $ERRPC ;; GET ADDRESS OF ERROR INSTRUCTION
SUB #2, $ERRPC
MOVB @ERRPC, $ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
BIT #BIT13, @SWR ;; SKIP TYPEOUT IF SET
BNE 20S ;; SKIP TYPEOUTS
JSR PC, TYPERR ;; GO TO USER ERROR ROUTINE
TYPE $SRLF

20S: CMPB @APTENV, $ENV ;; RUNNING IN APT MODE
BNE 2S ;; NO SKIP APT ERROR REPORT
MOVB $ITEMB, 21S ;; SET ITEM NUMBER AS ERROR NUMBER
JSR PC, SATY4 ;; REPORT FATAL ERROR TO APT

21S: .BYTE 0
.BYTE 0

22S: BR 22S ;; APT ERROR LOOP
2S: TST @SWR ;; HALT ON ERROR
BPL 3S ;; SKIP IF CONTINUE
HALT ;; HALT ON ERROR!
3S: BIT #BIT09, @SWR ;; TEST FOR CHANGE IN SOFT-SWR
BEQ 4S ;; LOOP ON ERROR SWITCH SET?
BR IF NO
FUDGE $LPERR, (SP) ;; FUDGE RETURN FOR LOOPING
TST $ESCAPE ;; CHECK FOR AN ESCAPE ADDRESS
BEQ 5S ;; BR IF NONE
MOV $ESCAPE, (SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE

```

```

6186 032464
6187 032464 022737 024470 000042
6188 032472 001001
6189 032474 000000
6190 032476
6191 032476 000002
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209 032500 105737 001157
6210 032504 100002
6211 032506 000000
6212 032510 000430
6213 032512 010046
6214 032514 017600 000002
6215 032520 122737 000001 001230
6216 032526 001011
6217 032530 132737 000100 001231
6218 032536 001405
6219 032540 010037 032550
6220 032544 004737 033214
6221 032550 000000
6222 032552 132737 000040 001231
6223 032560 001003
6224 032562 112046
6225 032564 001005
6226 032566 005726
6227 032570 012600
6228 032572 062716 000002
6229 032576 000002
6230 032600 122716 000011
6231 032604 001430
6232 032606 122716 000200
6233 032612 001006
6234 032614 005726
6235 032616 104401
6236 032620 001205
6237 032622 105037 032756
6238 032626 000755
6239 032630 004737 032712
6240 032634 123726 001156
6241 032640 001350

```

```

5S:      CMP      #SENDAD,2#42      ;;ACT-11 AUTO-ACCEPT?
          BNE      6S              ;;BRANCH IF NO
          HALT                     ;;YES
6S:      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      $TFPLG      ;; IS THERE A TERMINAL?
          BPL      1S          ;; BR IF YES
          HALT      HERE IF NO TERMINAL
          BR       3S          ;; LEAVE
1S:      MOV      RD, -(SP)      ;; SAVE RD
          MOV      #2(SP), RD    ;; GET ADDRESS OF ASCIZ STRING
          CMPB     #APTENV, $ENV  ;; RUNNING IN APT MODE
          BNE     62S          ;; NO, GO CHECK FOR APT CONSOLE
          BITB     #APTSPool, $ENVM ;; SPOOL MESSAGE TO APT
          BEQ     62S          ;; NO, GO CHECK FOR CONSOLE
          MOV      RD, 61S      ;; SETUP MESSAGE ADDRESS FOR APT
          JSR     PC, $ATY3     ;; SPOOL MESSAGE TO APT
          .WORD   0            ;; MESSAGE ADDRESS
          BITB     #APTCsup, $ENVM ;; APT CONSOLE SUPPRESSED
          BNE     60S          ;; YES, SKIP TYPE OUT
          MOVB     (RD)+, -(SP)  ;; PUSH CHARACTER TO BE TYPED ONTO STACK
          BNE     4S          ;; BR IF IT ISN'T THE TERMINATOR
          TST     (SP)+          ;; IF TERMINATOR POP IT OFF THE STACK
          MOV      (SP)+, RD     ;; RESTORE RD
          ADD     #2, (SP)      ;; ADJUST RETURN PC
          RTI                          ;; RETURN
          BR     8S          ;; BRANCH IF <HT>
          CMPB     #HT, (SP)
          BEQ     8S
          CMPB     #CRLF, (SP)  ;; BRANCH IF NOT <CRLF>
          BNE     5S
          TST     (SP)+          ;; POP <CR><LF> EQUIV
          TYPE     $CRLF        ;; TYPE A CR AND LF
          CLRB     $CHARCNT     ;; CLEAR CHARACTER COUNT
          BR       2S          ;; GET NEXT CHARACTER
          JSR     PC, $TYPECD    ;; GO TYPE THIS CHARACTER
          CMPB     $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
          BNE     2S          ;; IF NO GO GET NEXT CHAR.

```

```

6242 032642 013746 001154          MOV      $NULL,-(SP)      ;;GET # OF FILLER CHARS. NEEDED
6243 032646 105366 000001          7S:     DECB      1(SP)    ;;AND THE NULL CHAR.
6244 032652 002770                    BLT      6S              ;;DOES A NULL NEED TO BE TYPED?
6245 032654 004737 032712          JSR      PC,$TYPEC      ;;BR IF NO--GO POP THE NULL OFF OF STACK
6246 032660 105337 032756          DECB      $CHARCNT      ;;GO TYPE A NULL
6247 032664 000770                    BR       7S              ;;DO NOT COUNT AS A COUNT
6248                                ;;LOOP

```

:HORIZONTAL TAB PROCESSOR

```

6251 032666 112716 000040          8S:     MOV      #'(SP)    ;;REPLACE TAB WITH SPACE
6252 032672 004737 032712          9S:     JSR      PC,$TYPEC  ;;TYPE A SPACE
6253 032676 132737 000007 032756          BIT      #7,$CHARCNT    ;;BRANCH IF NOT AT
6254 032704 001372                    BNE      9S              ;;TAB STOP
6255 032706 005726                    TST      ,SP,+          ;;POP SPACE OFF STACK
6256 032710 000724                    BR       2S              ;;GET NEXT CHARACTER
6257 032712 105777 146232          $TYPEC: TST      2$TPS     ;;WAIT UNTIL PRINTER IS READY
6258 032716 100375                    BPL      $TYPEC
6259 032720 116677 000002 146224          MOV      2(SP),2$TPB    ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6260 032726 122766 000015 000002          CMP      #CR,2(SP)     ;;IS CHARACTER A CARRIAGE RETURN?
6261 032734 001003                    BNE      1S              ;;BRANCH IF NO
6262 032736 105037 032756          CLRB     $CHARCNT      ;;YES--CLEAR CHARACTER COUNT
6263 032742 000406                    BR       $TYPEX         ;;EXIT
6264 032744 122766 000012 000002          1S:     CMP      #LF,2(SP) ;;IS CHARACTER A LINE FEED?
6265 032752 001402                    BEQ      $TYPEX         ;;BRANCH IF YES
6266 032754 105227                    INCB     (PC)+          ;;COUNT THE CHARACTER
6267 032756 000000                    $CHARCNT: .WORD      D  ;;CHARACTER COUNT STORAGE
6268 032760 000207                    $TYPEX:  RTS          PC

```

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

6271 032762 010046                    ;;*****
6272 032762 010146                    ;;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
6273 032764 010246                    ;;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
6274 032766 010346                    ;;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
6275 032768 010446                    ;;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
6276 032770 010546                    ;;*REPLACED WITH SPACES.
6277 032772 010646                    ;;*CALL:
6278 032774 010746                    ;;*
6279 032776 010846                    ;;*
6280 032778 010946                    ;;*
6281 032780 011046                    ;;*
6282 032782 011146                    ;;*
6283 032784 011246                    ;;*
6284 032786 011346                    ;;*
6285 032788 011446                    ;;*
6286 032790 011546                    ;;*
6287 032792 011646                    ;;*
6288 032794 011746                    ;;*
6289 032796 011846                    ;;*
6290 032798 011946                    ;;*
6291 032800 012046                    ;;*
6292 032802 012146                    ;;*
6293 032804 012246                    ;;*
6294 032806 012346                    ;;*
6295 032808 012446                    ;;*
6296 032810 012546                    ;;*
6297 032812 012646                    ;;*

```

```

*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
*      TYPDS                    ;;GO TO THE ROUTINE

```

```

$TYPDS:
MOV      R0,-(SP)      ;;PUSH R0 ON STACK
MOV      R1,-(SP)      ;;PUSH R1 ON STACK
MOV      R2,-(SP)      ;;PUSH R2 ON STACK
MOV      R3,-(SP)      ;;PUSH R3 ON STACK
MOV      R5,-(SP)      ;;PUSH R5 ON STACK
MOV      #20200,-(SP)   ;;SET BLANK SWITCH AND SIGN
MOV      20(SP),R5     ;;GET THE INPUT NUMBER
BPL      1S            ;;BR IF INPLT IS POS.
NEG      R5            ;;MAKE THE BINARY NUMBER POS.
MOV      #'-,1(SP)     ;;MAKE THE ASCII NUMBER NEG.
1S:     CLR      R0      ;;ZERO THE CONSTANTS INDEX
MOV      #DBLK,R3      ;;SETUP THE OUTPUT POINTER
MOV      #' ,(R3)+     ;;SET THE FIRST CHARACTER TO A BLANK
2S:     CLR      R2      ;;CLEAR THE BCD NUMBER

```


E10

APR 27 09:44:00 06-007-76 09:44

MACY11 27(1006) 06-007-76 09:54 PAGE 121
CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEG 0121

6339	033232	016001	033166	35:	MOV	\$DTBL(R0),R1	::GET THE CONSTANT
6340	033233	160105			SUB	R1,R5	::FORM THIS BCD DIGIT
6341	033234	002402			BLT	45	::BR IF DONE
6342	033235	002402			INC	R2	::INCREASE THE BCD DIGIT BY 1
6343	033236	000704			BR	25	
6344	033237	060105		45:	ADD	R1,R5	::ADD BACK THE CONSTANT
6345	033238	005702			TST	R2	::CHECK IF BCD DIGIT=0
6346	033239	001002			BNE	55	::FALL THROUGH IF 0
6347	033240	105706			TSTB	(SP)	::STILL DOING LEADING 0'S?
6348	033241	105706			BMI	75	::BR IF YES
6349	033242	105316		55:	ASLB	(SP)	::MSD?
6350	033243	105316			BCC	65	::BR IF NO
6351	033244	000001	177777	65:	MOVB	1(SP)-1,R3	::YES--SET THE SIGN
6352	033245	000001		75:	BTS	0,R3	::MAKE THE BCD DIGIT ASCII
6353	033246	000040			BIS	0,R3	::MAKE IT A SPACE IF NOT ALREADY A DIGIT
6354	033247	100000			MOVB	R2,(R3)+	::PUT THIS CHARACTER IN THE OUTPUT BUFFER
6355	033248	000000			TST	(R0)+	::JUST INCREMENTING
6356	033249	000010			CMPL	R0,#10	::CHECK THE TABLE INDEX
6357	033250	002148			BLT	25	::GO DO THE NEXT DIGIT
6358	033251	003002			BGT	95	::GO TO EXIT
6359	033252	010502			MOV	R5,R2	::GET THE LSD
6360	033253	000764		85:	BR	65	::GO CHANGE TO ASCII
6361	033254	105726			TSTB	(SP)+	::WAS THE LSD THE FIRST NON-ZERO?
6362	033255	100003			BPL	95	::BR IF NO
6363	033256	116663	177777	95:	MOVB	-1(SP),-2(R3)	::YES--SET THE SIGN FOR TYPING
6364	033257	105013			CLRB	(R3)	::SET THE TERMINATOR
6365	033258	012605			MOV	(SP)+,R5	::POP STACK INTO R5
6366	033259	012602			MOV	(SP)+,R3	::POP STACK INTO R3
6367	033260	012602			MOV	(SP)+,R2	::POP STACK INTO R2
6368	033261	012601			MOV	(SP)+,R1	::POP STACK INTO R1
6369	033262	012600			MOV	(SP)+,R0	::POP STACK INTO R0
6370	033263	104401	033176		TYPE	\$DBLK	::NOW TYPE THE NUMBER
6371	033264	016666	000002	000004	MOV	2(SP),4(SP)	::ADJUST THE STACK
6372	033265	012616			MOV	(SP)+,(SP)	
6373	033266	000002			RTI		::RETURN TO USER
6374	033267	023420			\$DTBL:	10000.	
6375	033268	001750				1000.	
6376	033269	000144				100.	
6377	033270	000012				10.	
6378	033271	000004			\$DBLK:	.BLKW 4	
6379					.SBTTL	APT COMMUNICATIONS ROUTINE	
6380					*****		
6381	033206	112737	000001	033452	\$ATY1:	MOVB #1,\$FFLG	::TO REPORT FATAL ERROR
6382	033214	112737	000001	033450	\$ATY3:	MOVB #1,\$MFLG	::TO TYPE A MESSAGE
6383	033222	000403				BR \$ATYC	
6384	033224	112737	000001	033452	\$ATY4:	MOVB #1,\$FFLG	::TO ONLY REPORT FATAL ERROR
6385	033232				\$ATYC:		
6386	033232	010046			MOV	R0,-(SP)	::PUSH R0 ON STACK
6387	033234	010146			MOV	R1,-(SP)	::PUSH R1 ON STACK
6388	033236	105737	033450		TSTB	\$MFLG	::SHOULD TYPE A MESSAGE?
6389	033242	001450			BEQ	55	::IF NOT: BR
6390	033244	122737	000001	001230	CMPL	#APTENV,\$ENV	::OPERATING UNDER APT?
6391	033252	001031			BNE	35	::IF NOT: BR
6392	033254	132737	000100	001231	BITB	#APTSPOOL,\$ENVM	::SHOULD SPOOL MESSAGES?
6393	033262	001425			BEQ	35	::IF NOT: BR

```

033264 017600 000004      MOV      24(SP),RO      ;;GET MESSAGE ADDR.
033270 062766 000002 000004    ADD      82,4,SP)      ;;BUMP RETURN ADDR.
033276 005737 001210    18:     TST      MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
033302 001375      BNE      18            ;;IF NOT: WAIT
033304 010037 001224    28:     MOV      RO,MSGADR      ;;PUT ADDR IN MAILBOX
033310 105720      TSTB     (RO)+         ;;FIND END OF MESSAGE
033312 001375      BNE      28
033314 163700 001224    SUB      MSGADR,RO      ;;SUB START OF MESSAGE
033320 006200      ASR      RO            ;;GET MESSAGE LGTH IN WORDS
033322 010037 001226    MOV      RO,MSGGLT      ;;PUT LENGTH IN MAILBOX
033326 012737 000004 001210    MOV      84,MSGTYPE     ;;TELL APT TO TAKE MSG.
033334 000413      BR       55
033336 017637 000004 033362 38:     MOV      24,SP,48      ;;PUT MSG ADDR IN JSR LINKAGE
033344 062766 000002 000004    ADD      82,4,SP)      ;;BLMP RETURN ADDRESS
033352 013746 177776      MOV      177776,-(SP)   ;;PUSH 177776 ON STACK
033356 004737 032500      JSR      PC,STYPE      ;;CALL TYPE MACRO
033364 000000      .WORD   0
033364 105737 033452    48:     TSTB     $FFLG         ;;SHOULD REPORT FATAL ERROR?
033370 001416      BEQ      125           ;;IF NOT: BR
033372 005737 001230    58:     TST      $ENV         ;;RUNNING UNDER APT?
033376 001413      BEQ      125           ;;IF NOT: BR
033400 005737 001210    68:     TST      MSGTYPE      ;;FINISHED LAST MESSAGE?
033404 001375      BNE      115           ;;IF NOT: WAIT
033406 017637 000004 001212    78:     MOV      24(SP),$FATAL  ;;GET ERROR #
033414 062766 000002 000004    ADD      82,4,SP)      ;;BUMP RETURN ADDR.
033422 005237 001210    88:     INC      MSGTYPE      ;;TELL APT TO TAKE ERROR
033426 105037 033452    98:     CLRB     $FFLG         ;;CLEAR FATAL FLAG
033432 105037 033451    CLR     $LFLG         ;;CLEAR LOG FLAG
033436 105037 033450    CLR     $MFLG         ;;CLEAR MESSAGE FLAG
033442 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
033444 012600      MOV      (SP)+,RO      ;;POP STACK INTO RO
033446 000207      RTS      PC           ;;RETURN
033450 000      $MFLG: .BYTE 0        ;;MESSG. FLAG
033451 000      $LFLG: .BYTE 0        ;;LOG FLAG
033452 000      $FFLG: .BYTE 0        ;;FATAL FLAG
033454      .EVEN
033454      APTSIZE=200
033454      APTENV=001
033454      APTSPOOL=100
033454      APTCSUP=040
033454      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS      ;;CALL FOR TYPEOUT
*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*
*$STYPCN---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST

```

G10

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
 02R6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 123
 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0:23

6410									
6411						;;*STYPOS OR STYPOC			
6412						;;*CALL:			
6413						;;* MOV NUM,-(SP)	;;:NUMBER TO BE TYPED		
6414						;;* TYPON	;;:CALL FOR TYPEOUT		
6415						;;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER			
6416						;;*CALL:			
6417						;;* MOV NUM,-(SP)	;;:NUMBER TO BE TYPED		
6418						;;* TYPOC	;;:CALL FOR TYPEOUT		
6419									
6420	033454	017646	000000			STYPOS: MOV 2(SP),-(SP)	;;:PICKUP THE MODE		
6421	033460	116637	000001	033677		MOV 1(SP),%SOFILL	;;:LOAD ZERO FILL SWITCH		
6422	033466	112637	033701			MOV 1(SP),%SOMODE+1	;;:NUMBER OF DIGITS TO TYPE		
6423	033472	062716	000002			ADD 2(SP)	;;:ADJUST RETURN ADDRESS		
6424	033476	000406				BR \$TYPON			
6425	033500	112737	000001	033677		STYPOC: MOV 1,%SOFILL	;;:SET THE ZERO FILL SWITCH		
6426	033506	112737	000006	033701		MOV 6,%SOMODE+1	;;:SET FOR SIX(6) DIGITS		
6427	033514	112737	000005	033676		STYPON: MOV 5,%SOCNT	;;:SET THE ITERATION COUNT		
6428	033522	010346				MOV R3,-(SP)	;;:SAVE R3		
6429	033524	010446				MOV R4,-(SP)	;;:SAVE R4		
6430	033526	010546				MOV R5,-(SP)	;;:SAVE R5		
6431	033530	113704	033701			MOV 1,%SOMODE+1,R4	;;:GET THE NUMBER OF DIGITS TO TYPE		
6432	033534	005404				NEG R4			
6433	033536	062704	000006			ADD 6,R4	;;:SUBTRACT IT FOR MAX. ALLOWED		
6434	033542	110437	033700			MOV R4,%SOMODE	;;:SAVE IT FOR USE		
6435	033546	113704	033677			MOV 1,%SOFILL,R4	;;:GET THE ZERO FILL SWITCH		
6436	033552	016605	000012			MOV 12(SP),R5	;;:PICKUP THE INPUT NUMBER		
6437	033556	005003				CLR R3	;;:CLEAR THE OUTPUT WORD		
6438	033560	006105				1\$: ROL R5	;;:ROTATE MSB INTO "C"		
6439	033562	000404				BR 3\$;;:GO DO MSB		
6440	033564	006105				2\$: ROL R5	;;:FORM THIS DIGIT		
6441	033566	006105				ROL R5			
6442	033570	006105				ROL R5			
6443	033572	010503				MOV R5,R3			
6444	033574	006103				3\$: ROL R3	;;:GET LSB OF THIS DIGIT		
6445	033576	105337	033700			DECB %SOMODE	;;:TYPE THIS DIGIT?		
6446	033602	100016				BPL 7\$;;:BR IF NO		
6447	033604	042703	177770			BIC 177770,R3	;;:GET RID OF JUNK		
6448	033610	001002				BNE 4\$;;:TEST FOR 0		
6449	033612	005704				TST R4	;;:SUPPRESS THIS 0?		
6450	033614	001403				BEQ 5\$;;:BR IF YES		
6451	033616	005204				4\$: INC R4	;;:DON'T SUPPRESS ANYMORE 0'S		
6452	033620	052703	000060			BIS 0,R3	;;:MAKE THIS DIGIT ASCII		
6453	033624	052703	000040			5\$: BIS 1,R3	;;:MAKE ASCII IF NOT ALREADY		
6454	033630	110337	033674			MOV R3,%S	;;:SAVE FOR TYPING		
6455	033634	104401	033674			TYPE 8\$;;:GO TYPE THIS DIGIT		
6456	033640	105337	033676			7\$: DECB %SOCNT	;;:COUNT BY 1		
6457	033644	003347				BGT 2\$;;:BR IF MORE TO DO		
6458	033646	002402				BLT 6\$;;:BR IF DONE		
6459	033650	005204				INC R4	;;:INSURE LAST DIGIT ISN'T A BLANK		
6460	033652	000744				BR 2\$;;:GO DO THE LAST DIGIT		
6461	033654	012605				6\$: MOV (SP)+,R5	;;:RESTORE R5		
6462	033656	012604				MOV (SP)+,R4	;;:RESTORE R4		
6463	033660	012603				MOV (SP)+,R3	;;:RESTORE R3		
6464	033662	016666	000002	000004		MOV 2(SP),4(SP)	;;:SET THE STACK FOR RETURNING		
6465	033670	012616				MOV (SP)+,(SP)			

H10

```

6466 033672 000002
6467 033674 000
6468 033675 000
6469 033676 000
6470 033677 000
6471 033700 000000
6472
6473
6474
6475
6476 033702 000000
6477 033704 000000
6478 033706 000000
6479 033710 000001
6480 033711
6481 033712
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491 033712 005037 033702
6492 033716 012737 033710 033704
6493 033724 013737 033704 033706
6494 033732 012737 033762 000060
6495 033740 012737 000200 000062
6496 033746 005777 145174
6497 033752 012777 000100 145164
6498 033760 000207
6499
6500
6501
6502
6503
6504
6505
6506
6507 033762 117746 145160
6508 033766 042716 177600
6509 033772 021627 000003
6510 033776 001007
6511 034000 104401 035110
6512 034004 004737 033712
6513 034010 005726
6514 034012 000137 031034
6515 034016 021627 000007
6516 034022 001004
6517 034024 022737 000176 001140
6518 034032 001500
6519
6520 034034
6521 034034 022737 000001 033702

```

```

RTI :: RETURN
BS: .BYTE 0 :: STORAGE FOR ASCII DIGIT
      .BYTE 000 :: TERMINATOR FOR TYPE ROUTINE
$OCNT: .BYTE 000 :: OCTAL DIGIT COUNTER
$OFILL: .BYTE 000 :: ZERO FILL SWITCH
$OMODE: .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" (1C) $TKINT IS CALLED AND
;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS ($TCP).
:
$TKSRV: MOVB $TKB,-(SP) :: PICKUP THE CHARACTER
        BIC #1C177,(SP) :: STRIP THE JUNK
        CMP (SP),#3 :: IS IT A CONTROL C?
        BNE 1$ :: BRANCH IF NO
        TYPE $CNTLC :: TYPE A CONTROL-C (1C)
        JSR PC,$TKINT :: INIT THE KEYBOARD
        TST (SP)+ :: CLEAN UP STACK
        JMP STOP :: CONTROL C RESTART
1$: CMP (SP),#7 :: IS IT A CONTROL G?
   BNE 2$ :: BRANCH IF NO
   CMP $SWREG,$SWR :: IS SOFT-SWR SELECTED?
   BEQ 6$ :: GO TO SWR CHANGE
2$: CMP #1,$TKCNT :: IS THE QUEUE FULL?

```

```

65522 034042 001004      BNE      3$          ;; BRANCH IF NO
65523 034044 104401 001200    TYPE     .3BELL     ;; RING THE TTY BELL
65524 034050 005726      TST      (SP)+      ;; CLEAN CHARACTER OFF OF STACK
65525 034052 000451      BR       5$          ;; EXIT
65526 034054 021627 000023    3$: CMP     SP),#23   ;; IS IT A CONTROL-S?
65527 034060 001021      BNE     32$         ;; BRANCH IF NO
65528 034062 005077 145056    CLR     2$TKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
65529 034066 005726      TST      (SP)+      ;; CLEAN CHAR OFF STACK
65530 034070 105777 145050    31$: TSTB   2$TKS    ;; WAIT FOR A CHAR
65531 034074 100375      BPL     31$        ;; LOOP UNTIL ITS THERE
65532 034076 117746 145044    MOVB   2$TKB,-(SP) ;; GET THE CHARACTER
65533 034102 042716 177600    BIC     #1C177,(SP) ;; MAKE IT 7-BIT ASCII
65534 034106 022627 000021    CMP     (SP)+,#21   ;; IS IT A CONTROL-Q?
65535 034112 001366      BNE     31$        ;; BRANCH IF NO
65536 034114 012777 000100 145022    MOV     #100,2$TKS ;; REENABLE TTY KEYBOARD INTERRUPTS
65537 034122 000002      RTI                    ;; RETURN
65538 034124 005237 033702    32$: INC     $TKCNT   ;; COUNT THIS CHARACTER
65539 034130 021627 000140    CMP     (SP),#140   ;; IS IT UPPER CASE?
65540 034134 002405      BLT     4$          ;; BRANCH IF YES
65541 034136 021627 000175    CMP     (SP),#175   ;; IS IT A SPECIAL CHAR?
65542 034142 003002      BGT     4$          ;; BRANCH IF YES
65543 034144 042716 000040    BIC     #40,(SP)    ;; MAKE IT UPPER CASE
65544 034150 112677 177530    4$: MOVB   (SP)+,2$TKQIN ;; AND PUT IT IN QUEUE
65545 034154 005237 033704    INC     $TKQIN      ;; UPDATE THE POINTER
65546 034160 023727 033704 033711    CMP     $TKQIN,#$TKQEND ;; GO OFF THE END?
65547 034166 001003      BNE     5$          ;; BRANCH IF NO
65548 034170 012737 033710 033704    MOV     #$TKQSRT,$TKQIN ;; RESET THE POINTER
65549 034176 000002      RTI                    ;; RETURN
65550
65551
65552
65553
65554
65555
65556 034200 022737 000176 001140    $CKSWR: CMP     #SWREG,SWR ;; IS THE SOFT-SWR SELECTED
65557 034206 001124      BNE     15$         ;; EXIT IF NOT
65558 034210 105777 144730    TSTB   2$TKS      ;; IS A CHAR WAITING?
65559 034214 100121      BPL     15$         ;; IF NOT, EXIT
65560 034216 117746 144724    MOVB   2$TKB,-(SP) ;; YES
65561 034222 042716 177600    BIC     #1C177,(SP) ;; MAKE IT 7-BIT ASCII
65562 034226 021627 000007    CMP     (SP),#7     ;; IS IT A CONTROL-G?
65563 034232 001300      BNE     2$          ;; IF NOT, PUT IT IN THE TTY QUEUE
65564
65565
65566
65567
65568
65569
65570 034234 123727 001134 000001    6$: CMPB   $AUTOB,#1 ;; ARE WE RUNNING IN AUTO-MODE?
65571 034242 001674      BEQ     2$          ;; BRANCH IF YES
65572 034244 005726      TST      (SP)+      ;; CLEAR CONTROL-G OFF STACK
65573 034246 004737 033712    JSR     PC,$TKINT   ;; FLUSH THE TTY INPUT QUEUE
65574 034252 005077 144666    CLR     2$TKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
65575 034256 112737 000001 001135    MOVB   #1,$INTAG   ;; SET INTERRUPT MODE INDICATOR
65576
65577 034264 104401 035122    TYPE     .3CNTLG    ;; ECHO THE CONTROL-G (1G)

```

```

*****
*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
*CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

```

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

```

6578	034270	104401	035127		SGTSWR: TYPE	,\$MSWR	:: TYPE CURRENT CONTENTS
6579	034274	013746	000176		MOV	\$WREG,-(SP)	:: SAVE SWREG FOR TYPEOUT
6580	034300	104402			TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
6581	034302	104401	035140		TYPE	,\$MNEW	:: PROMPT FOR NEW SWR
6582	034306	005046		19\$:	CLR	-(SP)	:: CLEAR COUNTER
6583	034310	005046			CLR	-(SP)	:: THE NEW SWR
6584	034312	105777	144626	7\$:	TSTB	\$STKS	:: CHAR THERE?
6585	034316	100375			BPL	7\$:: IF NOT TRY AGAIN
6586							
6587	034320	117746	144622		MOVB	\$STKB,-(SP)	:: PICK UP CHAR
6588	034324	042716	177600		BIC	#1C17,(SP)	:: MAKE IT 7-BIT ASCII
6589							
6590	034330	021627	000003		CMP	(SP),#3	:: IS IT A CONTROL-C?
6591	034334	001015			BNE	9\$:: BRANCH IF NOT
6592	034336	104401	035110		TYPE	,\$CNTLC	:: YES, ECHO CONTROL-C (↑C)
6593	034342	062706	000006		ADD	#6,SP	:: CLEAN UP STACK
6594	034346	123727	001135	000001	CMPB	\$INTAG,#1	:: REENABLE TTY KEYBOARD INTERRUPTS?
6595	034354	001003			BNE	8\$:: BRANCH IF NO
6596	034356	012777	000100	144560	MOV	#100,\$STKS	:: ALLOW TTY KEYBOARD INTERRUPTS
6597	034364	000137	031034	8\$:	JMP	STOP	:: CONTROL-C RESTART
6598							
6599							
6600	034370	021627	000025	9\$:	CMP	(SP),#25	:: IS IT A CONTROL-U?
6601	034374	001005			BNE	10\$:: BRANCH IF NOT
6602	034376	104401	035115		TYPE	,\$CNTLU	:: YES, ECHO CONTROL-U (↑U)
6603	034402	062706	000006	20\$:	ADD	#6,SP	:: IGNORE PREVIOUS INPUT
6604	034406	000737			BR	19\$:: LET'S TRY IT AGAIN
6605							
6606							
6607	034410	021627	000015	10\$:	CMP	(SP),#15	:: IS IT A <CR>?
6608	034414	001022			BNE	16\$:: BRANCH IF NO
6609	034416	005766	000004		TST	4(SP)	:: YES, IS IT THE FIRST CHAR?
6610	034422	001403			BEQ	11\$:: BRANCH IF YES
6611	034424	016677	000002	144506	MOV	2(SP),\$SWR	:: SAVE NEW SWR
6612	034432	062706	000006	11\$:	ADD	#6,SP	:: CLEAN UP STACK
6613	034436	104401	001205	14\$:	TYPE	,\$CR LF	:: ECHO <CR> AND <LF>
6614	034442	123727	001135	000001	CMPB	\$INTAG,#1	:: RE-ENABLE TTY KBD INTERRUPTS?
6615	034450	001003			BNE	15\$:: BRANCH IF NOT
6616	034452	012777	000100	144464	MOV	#100,\$STKS	:: RE-ENABLE TTY KBD INTERRUPTS
6617	034460	000002		15\$:	RTI		:: RETURN
6618	034462	004737	032712	16\$:	JSR	PC,\$TYPEC	:: ECHO CHAR
6619	034466	021627	000060		CMP	(SP),#60	:: CHAR < 0?
6620	034472	002420			BLT	18\$:: BRANCH IF YES
6621	034474	021627	000067		CMP	(SP),#67	:: CHAR > 7?
6622	034500	003015			BGT	18\$:: BRANCH IF YES
6623	034502	042726	000060		BIC	#60,(SP)+	:: STRIP-OFF ASCII
6624	034506	005766	000002		TST	2(SP)	:: IS THIS THE FIRST CHAR
6625	034512	001403			BEQ	17\$:: BRANCH IF YES
6626	034514	006316			ASL	(SP)	:: NO, SHIFT PRESENT
6627	034516	006316			ASL	(SP)	:: CHAR OVER TO MAKE
6628	034520	006316			ASL	(SP)	:: ROOM FOR NEW ONE.
6629	034522	005266	000002	17\$:	INC	2(SP)	:: KEEP COUNT OF CHAR
6630	034526	056616	177776		BIS	-2(SP),(SP)	:: SET IN NEW CHAR
6631	034532	000667			BR	7\$:: GET THE NEXT ONE
6632	034534	104401	001204	18\$:	TYPE	,\$QUES	:: TYPE ?<CR><LF>
6633	034540	000720			BR	20\$:: SIMULATE CONTROL-U

K10

JNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
 0296JC.P11 0E-007-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 127
 TTY INPUT ROUTINE

SEG 0127

.DSABL LSB

```

6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645 034542 011646
6646 C34544 016666 000004 000002
6647 034552 005066 000004
6648 034556 005046
6649 034560 012746 034566
6650 034564 000002
6651 034566
6652 034566 005737 033702
6653 034572 001775
6654 034574 005337 033702
6655 034600 117766 177102 000004
6656 034606 005237 033706
6657 034612 023727 033706 033711
6658 034620 001003
6659 034622 012737 033710 033706
6660 034630 000002
6661
6662
6663
6664
6665
6666
6667
6668 034632 010346
6669 034634 005046
6670 034636 012703 035066
6671 034642 022703 035110
6672 034646 101456
6673 034650 104410
6674 034652 112613
6675 034654 122713 000177
6676 034660 001022
6677 034662 005716
6678 034664 001007
6679 034666 112737 000134 035064
6680 034674 104401 035064
6681 034700 012716 177777
6682 034704 005303
6683 034706 020327 035066
6684 034712 103434
6685 034714 111337 035064
6686 034720 104401 035064
6687 034724 000746
6688 034726 005716
6689 034730 001406
  
```

```

*****
*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:
*      RDLIN         ;; INPUT A STRING FROM THE TTY
*      RETURN HERE  ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
*                   ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
:
$RDLIN: MOV      R3, -(SP)   ;; SAVE R3
        CLR      -(SP)     ;; CLEAR THE RUBOUT KEY
1$:     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$
        MOV    #-1, (SP)  ;; SET THE RUBOUT KEY
6$:     DEC     R3         ;; BACKUP BY ONE
        CMP    R3, #($TTYIN ;; STACK EMPTY?
        BLOS  4$         ;; BR IF YES
        MOVB  (R3), 9$    ;; SETUP TO TYPEOUT THE DELETED CHAR.
        TYPE  9$
        BR    2$         ;; GO READ ANOTHER CHAR.
5$:     TST    (SP)       ;; RUBOUT KEY SET?
        BEQ    7$         ;; BR IF NO
  
```



```

6690 034732 112737 000134 035064      MOVB      #'\,9$      ;;TYPE A BACK SLASH
6691 034740 104401 035064      TYPE      9$
6692 034744 005016      CLR      (SP)      ;;CLEAR THE RUBOUT KEY
6693 034746 122713 000025      7$: CMPB      #25,(R3)  ;;IS CHARACTER A CTRL U?
6694 034752 001093      BNE      8$      ;;BR IF NO
6695 034754 104401 035115      TYPE      $CNTLU    ;;TYPE A CONTROL "U"
6696 034760 000726      BR      1$      ;;GO START OVER
6697 034762 122713 000022      8$: CMPB      #22,(R3)  ;;IS CHARACTER A "r"?
6698 034766 001011      BNE      3$      ;;BRANCH IF NO
6699 034770 105013      CLRB      (R3)    ;;CLEAR THE CHARACTER
6700 034772 104401 001205      TYPE      $CRLF    ;;TYPE A "CR" & "LF"
6701 034776 104401 035066      TYPE      $TTYIN   ;;TYPE THE INPUT STRING
6702 035002 000717      BR      2$      ;;GO PICKUP ANOTHER CHACTER
6703 035004 104401 001204      4$: TYPE      $QUES   ;;TYPE A '?'
6704 035010 000712      BR      1$      ;;CLEAR THE BUFFER AND LOOP
6705 035012 111337 035064      3$: MOVB      (R3),9$  ;;ECHO THE CHARACTER
6706 035016 104401 035064      TYPE      9$
6707 035022 122723 000015      CMPB      #15,(R3)+  ;;CHECK FOR RETURN
6708 035026 001305      BNE      2$      ;;LOOP IF NOT RETURN
6709 035030 105063 177777      CLRB      -1(R3)   ;;CLEAR RETURN (THE 15)
6710 035034 104401 001206      TYPE      $LF     ;;TYPE A LINE FEED
6711 035040 005726      TST      (SP)+    ;;CLEAN RUBOUT KEY FROM THE STACK
6712 035042 012603      MOV      (SP)+,R3  ;;RESTORE R3
6713 035044 011646      MOV      (SP)-,(SP)  ;;ADJUST THE STACK AND PUT ADDRESS OF THE
6714 035046 016666 000004 000002      MOV      4(SP),2(SP)  ;;FIRST ASCII CHARACTER ON IT
6715 035054 012766 035066 000004      MOV      #TTYIN,4(SP)
6716 035062 000002      RTI
6717 035064 000      9$: .BYTE      0      ;;RETURN
6718 035065 000      .BYTE      0      ;;STORAGE FOR ASCII CHAR. TO TYPE
6719 035066 000022      $TTYIN: .BLKB     22  ;;TERMINATOR
6720 035110 041536 005015 000      $CNTLC: .ASCIZ  /C<15><12>  ;;RESERVE 22 BYTES FOR TTY INPUT
6721 035115 136 006525 000012      $CNTLU: .ASCIZ  /U<15><12>  ;;CONTROL "C"
6722 035122 043536 005015 000      $CNTLG: .ASCIZ  /G<15><12>  ;;CONTROL "U"
6723 035127 015 051412 051127      $MSWR: .ASCIZ  <15><12>/SWR = /  ;;CONTROL "G"
6724 035134 036440 000040      $MNEW: .ASCIZ  / NEW = /
6725 035140 020040 042516 020127
6726 035146 020075 000
6727 035152
6728 .EVEN
6729 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
6730
6731 ;;*****
6732 ;;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
6733 ;;*CHANGE IT TO BINARY.
6734 ;;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
6735 ;;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
6736 ;;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
6737 ;;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
6738 ;;*CALL:
6739 ;;* RDOCT      ;;READ AN OCTAL NUMBER
6740 ;;* RETURN HERE  ;;LOW ORDER BITS ARE ON TOP OF THE STACK
6741 ;;*           ;;HIGH ORDER BITS ARE IN $HI OCT
6742 035152 011646 000004 000002      $RDOCT: MOV      (SP)-,(SP)  ;;PROVIDE SPACE FOR THE
6743 035154 016666 000004 000002      MOV      4(SP),2(SP)  ;;INPUT NUMBER
6744 035162 010046      MOV      R0,-(SP)    ;;PUSH R0 ON STACK
6745 035164 010146      MOV      R1,-(SP)    ;;PUSH R1 ON STACK

```


M10

```

6746 035166 010246          MOV      R2,-(SP)          ;; PUSH R2 ON STACK
6747 035170 104411          1$:     RDLIN              ;; READ AN ASCIZ LINE
6748 035172 012600          MOV      (SP)+,R0         ;; GET ADDRESS OF 1ST CHARACTER
6749 035174 010037 035300          MOV      R0,$$           ;; AND SAVE IT
6750 035200 005001          CLR      R1              ;; CLEAR DATA WORD
6751 035202 005002          CLR      R2
6752 035204 112046          2$:     MOVVB      (R0)+,-(SP) ;; PICKUP THIS CHARACTER
6753 035206 001420          BEQ      3$              ;; IF ZERO GET OUT
6754 035210 122716 000060          CMPB     #'0,(SP)        ;; MAKE SURE THIS CHARACTER
6755 035214 003026          BGT      4$              ;; IS AN OCTAL DIGIT
6756 035216 122716 000067          CMPB     #'7,(SP)
6757 035222 002423          BLT      4$
6758 035224 006301          ASL      R1              ;; *2
6759 035226 006102          ROL      R2
6760 035230 006301          ASL      R1              ;; *4
6761 035232 006102          ROL      R2
6762 035234 006301          ASL      R1              ;; *8
6763 035236 006102          ROL      R2
6764 035240 042716 177770          BIC      #'07,(SP)       ;; STRIP THE ASCII JUNK
6765 035244 062601          ADD      (SP)+,R1        ;; ADD IN THIS DIGIT
6766 035246 000756          BR       2$              ;; LOOP
6767 035250 005726          3$:     TST      (SP)+         ;; CLEAN TERMINATOR FROM STACK
6768 035252 010166 000012          MOV      R1,12(SP)       ;; SAVE THE RESULT
6769 035256 010237 035310          MOV      R2,$HI OCT
6770 035262 012602          MOV      (SP)+,R2        ;; POP STACK INTO R2
6771 035264 012601          MOV      (SP)+,R1        ;; POP STACK INTO R1
6772 035266 012600          MOV      (SP)+,R0        ;; POP STACK INTO R0
6773 035270 000002          RTI
6774 035272 005726          4$:     TST      (SP)+         ;; CLEAN PARTIAL FROM STACK
6775 035274 105010          CLRB     (R0)            ;; SET A TERMINATOR
6776 035276 104401          TYPE
6777 035300 000000          5$:     .WORD     0           ;; TYPE UP THRU THE BAD CHAR.
6778 035302 104401 001204          TYPE     $QUES           ;; "?" "CR" & "LF"
6779 035306 000730          BR       1$              ;; TRY AGAIN
6780 035310 000000          $HI OCT: .WORD     0           ;; HIGH ORDER BITS GO HERE
6781          .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
6782
6783          ;; *****
6784          ;; *THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
6785          ;; *UNSIGNED OCTAL ASCIZ NUMBER.
6786          ;; *CALL
6787          ;; *   MOV      #PNTR,-(SP)          ;; POINTER TO LOW WORD OF BINARY NUMBER
6788          ;; *   JSR      PC,@#$DB20         ;; CALL THE ROUTINE
6789          ;; *   RETURN                       ;; THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
6790
6791
6792 035312 104413          $DB20: SAVREG              ;; SAVE ALL REGISTERS
6793 035314 016601 000002          MOV      2(SP),R1        ;; PICKUP THE POINTER TO LOW WORD
6794 035320 012705 035431          MOV      #$OCTVL+13.,R5 ;; POINTER TO DATA TABLE
6795 035324 012704 000014          MOV      #12.,R4         ;; DO ELEVEN CHARACTERS
6796 035330 012703 177770          MOV      #'07,R3        ;; MASK
6797 035334 012100          MOV      (R1)+,R0        ;; LOWER WORD
6798 035336 012101          MOV      (R1)+,R1        ;; HIGH WORD
6799 035340 005002          CLR      R2              ;; TERMINATOR
6800 035342 110245          1$:     MOVVB      R2,-(R5)     ;; PUT CHARACTER IN DATA TABLE
6801 035344 010002          MOV      R0,R2          ;; GET THIS DIGIT

```

N10

03:18:16 R16 DRIVE DIAGNOSTIC PART 3
03:18:16 P11 06-00-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 130
DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

SEG 0130

```

68003 035346 005304          DEC      R4          ;;COUNT THIS CHARACTER
68004 035348 003000          BGT     3$          ;;BR IF NOT THE LAST DIGIT
68005 035350 001405          BEQ     2$          ;;BR IF IT IS THE LAST DIGIT
68006 035352 005205          INC     R5          ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
68007 035354 010566 000002    MOV     R5,2(SP)   ;;ASCIZ CHAR. & PUT IT ON THE STACK
68008 035356 104414          RESREG          ;;RESTORE ALL REGISTERS
68009 035358 000207          RTS     PC          ;;RETURN TO USER
68010 035360 006203          2$:    ASR     R3          ;;POSITION THE MASK FOR THE LAST DIGIT
68011 035362 006001          3$:    ROR     R1          ;;POSITION THE BINARY NUMBER FOR
68012 035364 006000          ROR     R0          ;;THE NEXT OCTAL DIGIT
68013 035366 006001          ROR     R1
68014 035368 006000          ROR     R0
68015 035370 006001          ROR     R1
68016 035372 006000          ROR     R0
68017 035374 006001          ROR     R1
68018 035376 006000          ROR     R0
68019 035378 006001          ROR     R1
68020 035380 006000          ROR     R0
68021 035382 040302          BIC     R3,R2      ;;MASK OUT ALL JUNK
68022 035384 062702 000060    ADD     #'0,R2     ;;MAKE THIS CHAR. ASCII
68023 035386 000153          BR      1$          ;;GO PUT IT IN THE DATA TABLE
68024 035388 000016          $OCTVL: .BLKB 14.  ;;RESERVE DATA TABLE
68025 035390          .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, #SDB2D   ;; THE FIRST ADDRESS OF ASCIZ
*   RETRN          ;; IS ON THE STACK

SDB2D: SAVREG          ;;SAVE REGISTERS
MOV     2(SP),R2       ;;PICKUP THE DATA POINTER
MOV     #SDECVL,R0     ;;GET ADDRESS OF "SDECVL" 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     #STNPWR,R4     ;;ADDRESS OF TEN POWER
MOV     #STNPWR+2,R5
1$:    CLR     R3          ;;CLEAR PARTIAL
2$:    SUB     (R4),R1    ;;SUBTRACT TEN POWER
SBC     R2
SUB     (R5),R2
3$:    BLT     3$          ;;BR IF TEN POWER TO 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

```



```

000060 035722 010046
000061 035722 010146
000062 035724 010246
000063 035726 005046
000064 035730 016601 000012
000065 035732 100002
000066 035736 005216
000067 035740 005401
000068 035742 016602 000014
000069 035744 100002
000070 035750 005316
000071 035752 005402
000072 035754 012746 000021
000073 035756 005000
000074 035762 103001
000075 035764 060200
000076 035766 006000
000077 035770 006001
000078 035772 005316
000079 035774 001372
000080 035776 022616
000081 036000 001403
000082 036002 005400
000083 036004 005401
000084 036006 005600
000085 036010 005726
000086 036012 010066 000012
000087 036014 010166 000010
000088 036020 012602
000089 036024 012601
000090 036026 012600
000091 036030 000207
000092 036032

```

```

CMPB #0,(RO)+      ;; IS THIS AN ASCII "J" ?
BEQ 1$              ;; BR IF YES
2$: DEC RO          ;; BACKUP BY "1"
MOV RO,3$          ;; SAVE FOR TYPING
TYPE              ;; GO TYPE
3$: .WORD 0         ;; ASCIZ POINTER GOES HERE
MOV (SP)+,RO      ;; RESTORE RO
MOV (SP)+,(SP)    ;; RESTORE THE STACK
RTS PC            ;; RETURN
.SBTL INTEGER MULTIPLY ROUTINE

*****
*CALL
* MOV MULTIPLIER, -(SP)
* MOV MULTIPLICAND, -(SP)
* JSR PC, @MULT
* RETURN ;; PRODUCT IS ON THE STACK
*
* STACK PRODUCT
* ----
* TOP LSB'S
* +2 MSB'S

SMULT:
MOV RO, -(SP)      ;; PUSH RO ON STACK
MOV R1, -(SP)     ;; PUSH R1 ON STACK
MOV R2, -(SP)     ;; PUSH R2 ON STACK
CLR -(SP)         ;; CLEAR THE SIGN KEY
MOV 12(SP), R1    ;; GET THE MULTIPLICAND
BPL 1$            ;; BR IF PLUS
INC (SP)          ;; SET THE SIGN KEY
NEG R1            ;; MAKE THE MULTIPLICAND POSTIVE
1$: MOV 14(SP), R2 ;; GET THE MULTIPLIER
BPL 2$            ;; BR IF PLUS
DEC (SP)          ;; UPDATE THE SIGN KEY
NEG R2            ;; MAKE THE MULTIPLIER POSTIVE
2$: MOV #17, -(SP) ;; SET THE LOOP COUNT
CLR RO           ;; SETUP FOR THE MULTIPLY LOOP
3$: BCC 4$        ;; DON'T ADD IF MULTIPLICAND = 0
ADD R2, RO
4$: ROR RO        ;; POSITION THE PARITIAL PRODUCT AND
ROR R1           ;; THE MULTIPLICAND
DEC (SP)         ;; HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
BNE 3$          ;; BR IF NO
CMP (SP)+, (SP) ;; SHOULD PRODUCT BE NEGATIVE?
BEQ 5$          ;; GO TO EXIT IF NO
NEG RO          ;; YES--SO MAKE IT SO
ROR R1
SBC RO
5$: TST (SP)+    ;; CLEAR SIGN INFO. OFF OF STACK
MOV RO, 12(SP)  ;; PUT THE PRODUCT ON THE STACK (MSB'S)
MOV R1, 10(SP) ;; LSB'S
MOV (SP)+, R2  ;; POP STACK INTO R2
MOV (SP)+, R1  ;; POP STACK INTO R1
MOV (SP)+, RO  ;; POP STACK INTO RO
RTS PC

```

```

6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988 036034
6989 036034 010046
6990 036036 010046
6991 036040 010246
6992 036042 010346
6993 036044 010446
6994 036046 010546
6995 036050 016646 000022
6996 036054 016646 000022
6997 036060 016646 000022
6998 036064 016646 000022
6999 036070 000002
7000
7001
7002
7003 036072
7004 036072 012666 000022
7005 036076 012666 000022
7006 036102 012666 000022
7007 036106 012666 000022
7008 036112 012605
7009 036114 012604
7010 036116 012603
7011 036120 012602
7012 036122 012601
7013 036124 012600
7014 036126 000002
7015
7016
7017
7018
7019
7020
7021
7022
7023 036130 010046
7024 036132 016600 000002
7025 036136 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 R0,-(SP) ;; PUSH R0 ON STACK
MOV R1,-(SP) ;; PUSH R1 ON STACK
MOV R2,-(SP) ;; PUSH R2 ON STACK
MOV R3,-(SP) ;; PUSH R3 ON STACK
MOV R4,-(SP) ;; PUSH R4 ON STACK
MOV R5,-(SP) ;; PUSH R5 ON STACK
MOV 22(SP),-(SP) ;; SAVE PS OF MAIN FLOW
MOV 22(SP),-(SP) ;; SAVE PC OF MAIN FLOW
MOV 22(SP),-(SP) ;; SAVE PS OF CALL
MOV 22(SP),-(SP) ;; SAVE PC OF CALL
RTI

```

*RESTORE 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 R0,-(SP) ;; SAVE R0
MOV 2(SP),R0 ;; GET TRAP ADDRESS
TST -(R0) ;; BACKUP BY 2

```

E11

JN7BL5 RK6 DRIVE DIAGNOSTIC PART 3
 02R6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 134
 TRAP DECODER

SEG 0134

```

7026 036140 111000      MOVB   (R0),R0      ;;GET RIGHT BYTE OF TRAP
7027 036142 006300      ASL    RC           ;;POSITION FOR INDEXING
7028 036144 016300 036164  MOV    $TRAP2(R0),RC ;;INDEX TO TABLE
7029 036150 000200      RTS    R0           ;;GO TO ROUTINE
  
```

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

```

7030 036152 011646 000004 000002 $TRAP2: MOV   (SP),-(SP)  ;;MOVE THE PC DOWN
7031 036154 016666 000004 000002  MOV   4(SP),2(SP)      ;;MOVE THE PSW DOWN
7032 036162 000002      RTI                ;;RESTORE THE PSW
  
```

.SETTL TRAP TABLE

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

```

:          ROUTINE
:          -----
7045 036164 036152 $TRAP2: .WORD  $TRAP2
7046 036166 032500      $TYPE  ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
7047 036170 033500      $TYPOC ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
7048 036172 033454      $TYPOS ;;CALL=TYPOS     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
7049 036174 033514      $TYPON ;;CALL=TYPON     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
7050 036176 032762      $TYPDS ;;CALL=TYPDS     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
7051
7052 036200 034270      $GTSWR ;;CALL=GTSWR     TRAP+6(104406) GET SOFT-SWR SETTING
7053
7054 036202 034200      $CKSWR ;;CALL=CKSWR     TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
7055 036204 034542      $RDCHR ;;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
7056 036206 034632      $RDLIN ;;CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
7057 036210 035152      $RDOCT ;;CALL=RDOCT     TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
7058 036212 036034      $$AVREG ;;CALL=SAVREG    TRAP+13(104413) SAVE R0-R5 ROUTINE
7059 036214 036072      $RESREG ;;CALL=RESREG    TRAP+14(104414) RESTORE R0-R5 ROUTINE
7060 036216 030714      $SCOPI$ ;;CALL=SCOPI$   TRAP+15(104415) INTERNAL LOOP ON ERROR
7061
  
```

7062				
7063				
7064				
7065				
7066	036220	005015	047125	041111
7067	036226	051525	051040	030113
7068	036234	020066	051104	053111
7069	036242	020105	044504	043501
7070	036250	047516	052123	041511
7071	036256	005015	050011	051101
7072	036264	020124	063	
7073	036267	015	046412	044501
7074	036274	042116	041505	030455
7075	036302	026461	055104	033122
7076	036310	026512	026503	041120
7077	036316	005015		
7078	036320	005015	025011	025052
7079	036326	025052	041440	052501
7080	036334	044524	047117	025040
7081	036342	025052	025052	005015
7082	036350	005015	044124	051511
7083	036356	050040	047522	051107
7084	036364	046501	051440	047510
7085	036372	046125	020104	041040
7086	036400	020105	040510	052114
7087	036406	042105	047440	046116
7088	036414	020131	054502	052040
7089	036422	050131	047111	020107
7090	036430	047503	052116	047522
7091	036436	026514	103	
7092	036441	015	047412	044124
7093	036446	051105	044527	042523
7094	036454	041440	051101	051124
7095	036462	042111	042507	043040
7096	036470	051117	040515	052124
7097	036476	047111	020107	047101
7098	036504	026104	047440	020122
7099	036512	044124	020105	042504
7100	036520	044526	042503	
7101	036524	005015	040515	020131
7102	036532	042502	046040	043105
7103	036540	020124	047111	040440
7104	036546	020116	047125	042504
7105	036554	042524	046522	047111
7106	036562	042105	051440	040524
7107	036570	042524	005015	
7108	036574	005015	047111	052111
7109	036602	040511	046114	026131
7110	036610	042040	044522	042526
7111	036616	020123	047524	041040
7112	036624	020105	042524	052123
7113	036632	042105	051440	047510
7114	036640	046125	020104	040510
7115	036646	042526	006472	012
7116	036653	015	040412	020056
7117	036660	042510	042101	020123

.SBTTL SERVICE MESSAGES

MSG1: .ASCII <CR><LF>/UNIBUS RKO6 DRIVE DIAGNOSTIC/

.ASCII <CR><LF>/ PART 3/

.ASCII <CR><LF>/MAINDEC-11-DZR6J-C-PB/<CR><LF>

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

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

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

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

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

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

7118	036666	040515	052516	046101	
7119	036674	054514	046040	040517	
7120	036702	042504	104		
7121	036705	04015	041012	020056	.ASCII <CR><LF>/B. CORRECT PORT SELECTED/
7122	036712	047503	051122	041505	
7123	036720	020124	047520	052122	
7124	036726	051440	046105	041505	
7125	036734	042524	104		
7126	036737	015	041412	020056	.ASCII <CR><LF>/C. WRITE LOCK DISABLED/
7127	036744	051127	052111	020105	
7128	036752	047514	045503	042040	
7129	036760	051511	041101	042514	
7130	036766	104			
7131	036767	015	042012	020056	.ASCII <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
7132	036774	051104	053111	020105	
7133	037002	042522	042101	020131	
7134	037010	047111	044504	040503	
7135	037016	047524	020122	047117	
7136	037024	005015			
7137	037026	005015	051104	053111	.ASCII <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
7138	037034	051505	047040	052117	
7139	037042	052040	020117	042502	
7140	037050	052040	051505	042524	
7141	037056	020104	052515	052123	
7142	037064	044040	053101	105	
7143	037071	015	041012	052117	.ASCIZ <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
7144	037076	020110	047520	052122	
7145	037104	020123	042504	042523	
7146	037112	042514	052103	042105	
7147	037120	005015	000		
7148					
7149	037123	015	041012	020105	MSG2: .ASCIZ <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/
7150	037130	052523	042522	052040	
7151	037136	020117	052520	020124	
7152	037144	041523	040522	041524	
7153	037152	020110	040520	045503	
7154	037160	044440	020116	051104	
7155	037166	053111	020105	000060	
7156	037174	005015	051104	053111	MSG3: .ASCIZ <CR><LF>/DRIVE(S) TO BE TESTED: /
7157	037202	024105	024523	052040	
7158	037210	020117	042502	052040	
7159	037216	051505	042524	035104	
7160	037224	000040			
7161	037226	005015	054524	042520	MSG4: .ASCIZ <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440: /
7162	037234	041040	051525	020123	
7163	037242	042101	051104	051505	
7164	037250	020123	043111	047040	
7165	037256	052117	030440	033467	
7166	037264	032064	035060	020040	
7167	037272	000			
7168	037273	015	052012	050131	MSG5: .ASCIZ <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210: /
7169	037300	020105	047503	052116	
7170	037306	047522	046114	051105	
7171	037314	044440	052116	051105	
7172	037322	052522	052120	053040	
7173	037330	041505	047524	020122	

7174	037336	043111	047040	052117	
7175	037344	031040	030061	020072	
7176	037352	000040			
7177	037354	005015	047111	042524	MSG6: .ASCIZ <CR><LF>/INTERRUPT OCCURRED AT PC=/
7178	037362	051122	050125	020124	
7179	037370	041517	052503	051122	
7180	037376	042105	040440	020124	
7181	037404	041520	000075		
7182	037410	005015	051104	053111	MSG7: .ASCIZ <CR><LF>/DRIVE C WILL NOT BE TESTED/
7183	037416	020105	020060	044527	
7184	037424	046114	047040	052117	
7185	037432	041040	020105	042524	
7186	037440	052123	042105	000	
7187	037445	015	044412	052116	MSG8: .ASCIZ <CR><LF>/INTERLOCKS TEST/<CR><LF>
7188	037452	051105	047514	045503	
7189	037460	020123	042524	052123	
7190	037466	005015	000		
7191	037471	015	051012	046505	MSG9: .ASCIZ <CR><LF>/REMOVE UNIT SELECT PLUG/
7192	037476	053117	020105	047125	
7193	037504	052111	051440	046105	
7194	037512	041505	020124	046120	
7195	037520	043525	000		
7196	037523	015	005012	044527	MSG10: .ASCIZ <CR><LF><LF>/WILL TEST DRIVES:/
7197	037530	046114	052040	051505	
7198	037536	020124	051104	053111	
7199	037544	051505	000072		
7200	037550	005015	050012	053517	MSG11: .ASCIZ <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
7201	037556	051105	052440	020120	
7202	037564	042522	052123	051101	
7203	037572	020124	047524	052040	
7204	037600	051505	020124	006461	
7205	037606	000012			
7206	037610	047516	042516	005015	MSG12: .ASCIZ /NONE/<CR><LF>
7207	037616	000			
7208	037617	015	047012	020117	MSG13: .ASCII <CR><LF>/NO L OR P CLOCKS PRESENT/
7209	037624	020114	051117	050040	
7210	037632	041440	047514	045503	
7211	037640	020123	051120	051505	
7212	037646	047105	124		
7213	037651	015	040412	046114	.ASCIZ <CR><LF>/ALL TIMING TESTS BYPASSED/
7214	037656	052040	046511	047111	
7215	037664	020107	042524	052123	
7216	037672	020123	054502	040520	
7217	037700	051523	042105	000	
7218	037705	015	041012	050131	MSG14: .ASCIZ <CR><LF>/BYPASSING DRIVE /
7219	037712	051501	044523	043516	
7220	037720	042040	044522	042526	
7221	037726	000040			
7222	037730	005015	042012	044522	MSG15: .ASCIZ <CR><LF><LF>/DRIVE /
7223	037736	042526	000040		
7224	037742	005015	051104	053111	MSG16: .ASCIZ <CR><LF>/DRIVE SERIAL NO. /
7225	037750	020105	042523	044522	
7226	037756	046101	047040	027117	
7227	037764	000040			
7228	037766	005015	040503	052122	MSG17: .ASCIZ <CR><LF>/CARTRIDGE SERIAL NO. /
7229	037774	044522	043504	020105	

7230	040002	042523	044522	046101	
7231	040010	047040	027117	000040	
7232	040016	005015	047125	052111	MSG18: .ASCIZ <CR><LF>/UNIT SELECT PLUG TEST/<CR><LF>
7233	040024	051440	046105	041505	
7234	040032	020124	046120	043525	
7235	040040	052040	051505	006524	
7236	040046	000012			
7237	040050	005015	047520	052122	MSG19: .ASCIZ <CR><LF>/PORT SELECTION TESTS/<CR><LF>
7238	040056	051440	046105	041505	
7239	040064	044524	047117	052040	
7240	040072	051505	051524	005015	
7241	040100	000			
7242	040101	015	051012	047125	MSG20: .ASCIZ <CR><LF>/RUN-STOP SWITCH TEST/<CR><LF>
7243	040106	051455	047524	020120	
7244	040114	053523	052111	044103	
7245	040122	052040	051505	006524	
7246	040130	000012			
7247	040132	005015	041501	046040	MSG21: .ASCIZ <CR><LF>/AC LOW DETECTION TEST-PART 1/<CR><LF>
7248	040140	053517	042040	052105	
7249	040146	041505	044524	047117	
7250	040154	052040	051505	026524	
7251	040162	040520	052122	030440	
7252	040170	005015	000		
7253	040173	015	047012	047117	MSG22: .ASCIZ <CR><LF>/NON-EXECUTABLE FUNCTION (NXF) DETECTION TEST/<CR><LF>
7254	040200	042455	042530	052503	
7255	040206	040524	046102	020105	
7256	040214	052506	041516	044524	
7257	040222	047117	024040	054116	
7258	040230	024506	042040	052105	
7259	040236	041505	044524	047117	
7260	040244	052040	051505	006524	
7261	040252	000012			
7262	040254	005015	047516	041440	MSG23: .ASCIZ <CR><LF>/NO CLOCK INTERRUPTS PRESENT, ABORTING TIMING TESTS/
7263	040262	047514	045503	044440	
7264	040270	052116	051105	052522	
7265	040276	052120	020123	051120	
7266	040304	051505	047105	026124	
7267	040312	040440	047502	052122	
7268	040320	047111	020107	044524	
7269	040326	044515	043516	052040	
7270	040334	051505	051524	000	
7271	040341	015	053412	044522	MSG24: .ASCIZ <CR><LF>/WRITE LOCK TEST/<CR><LF>
7272	040346	042524	046040	041517	
7273	040354	020113	042524	052123	
7274	040362	005015	000		
7275	040365	040	051525	000	MSG25: .ASCIZ / US/
7276	040371	015	040412	020103	MSG26: .ASCIZ <CR><LF>/AC LOW DETECTION TEST-PART 2/<CR><LF>
7277	040376	047514	020127	042504	
7278	040404	042524	052103	047511	
7279	040412	020116	042524	052123	
7280	040420	050055	051101	020124	
7281	040426	006462	000012		
7282	040432	005015	040412	046114	MSG27: .ASCIZ <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
7283	040440	042040	044522	042526	
7284	040446	020123	042524	052123	
7285	040454	042105	005015	000012	

7296	040462	035015	052515	052114
7297	040470	050111	042514	042040
7298	040476	044522	042526	042040
7299	040504	052105	041505	044524
7290	040512	047117	052040	051505
7291	040520	006524	000012	
7292	040524	005015	046120	040505
7293	040532	042522	053440	044501
7294	040540	026124	044040	040505
7295	040546	051504	041040	044505
7296	040554	043516	046040	040517
7297	040562	042504	006504	000012
7298	040570	005015	042526	044522
7299	040576	054506	042040	047517
7300	040604	020122	040503	020116
7301	040612	047516	020127	042502
7302	040620	047440	042520	042516
7303	040626	020104	020046	042514
7304	040634	053101	020105	052111
7305	040642	047440	042520	006516
7306	040650	000012		
7307	040652	035015	006477	000012
7308	040660	005015	042522	047515
7309	040666	042526	042040	051511
7310	040674	020113	040520	045503
7311	040702	023040	041440	047514
7312	040710	042523	042040	047517
7313	040716	000122		
7314	040720	005015	042504	051120
7315	040726	051505	020123	051047
7316	040734	047125	051455	047524
7317	040742	023520	051440	044527
7318	040750	041524	020110	047524
7319	040756	023440	052522	023516
7320	040764	053440	044510	042514
7321	040772	042040	047517	020122
7322	041000	050117	047105	000
7323	041005	015	053012	051105
7324	041012	043111	020131	050123
7325	041020	047111	046104	020105
7326	041026	047504	051505	047040
7327	041034	052117	051440	040524
7328	041042	052122	023040	044040
7329	041050	040505	051504	042040
7330	041056	020117	047516	020124
7331	041064	047514	042101	005015
7332	041072	000		
7333	041073	015	042012	050105
7334	041100	042522	051523	023440
7335	041106	052522	026516	052123
7336	041114	050117	020047	053523
7337	041122	052111	044103	052040
7338	041130	020117	051047	047125
7339	041136	020047	044127	046111
7340	041144	020105	040503	052122
7341	041152	044522	043504	020105

MSG28: .ASCIZ <CR><LF>/MULTIPLE DRIVE DETECTION TEST/<CR><LF>

MSG29: .ASCIZ <CR><LF>/PLEASE WAIT, HEADS BEING LOADED/<CR><LF>

MSG30: .ASCIZ <CR><LF>/VERIFY DOOR CAN NOW BE OPENED & LEAVE IT OPEN/<CR><LF>

MSG31: .ASCIZ <CR><LF>/?/<CR><LF>

MSG32: .ASCIZ <CR><LF>/REMOVE DISK PACK & CLOSE DOOR/

MSG33: .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE DOOR OPEN/

MSG34: .ASCIZ <CR><LF>/VERIFY SPINDLE DOES NOT START & HEADS DO NOT LOAD/<CR><LF>

MSG35: .ASCIZ <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' WHILE CARTRIDGE REMOVED/

MSG36: .ASCIZ <CR><LF>/INSERT DISK PACK, DEPRESS 'RUN-STOP' TO 'RUN' & CLOSE DOOR/

MSG37: .ASCIZ <CR><LF>/DEPRESS SPACE BAR WHEN FINISHED<CR><LF>

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>

7398	041644	005015	042504	042523
7399	041652	042514	052103	050040
7400	041660	051117	020124	047111
7401	041666	052440	042523	023040
7402	041674	051440	046105	041505
7403	041702	020124	050117	047520
7404	041710	044523	042524	050040
7405	041716	051117	006524	000012
7406	041724	005015	042504	042523
7407	041732	042514	052103	053440
7408	041740	047522	043516	050040
7409	041746	051117	020124	020046
7410	041754	042523	042514	052103
7411	041762	041440	051117	042522
7412	041770	052103	050040	051117
7413	041776	006524	000012	
7414	042002	005015	042504	042523
7415	042010	042514	052103	041040
7416	042016	052117	020110	047520
7417	042024	052122	006523	000012
7418	042032	005015	042523	042514
7419	042040	052103	041440	051117
7420	042046	042522	052103	050040
7421	042054	051117	006524	000012
7422	042062	005015	047503	051122
7423	042070	041505	020124	047520
7424	042076	052122	047040	052117
7425	042104	051440	046105	041505
7426	042112	042524	026104	052040
7427	042120	054522	040440	040507
7428	042126	047111	005015	000
7429	042133	015	042012	020117
7430	042140	020101	040515	052516
7431	042146	046101	042040	044522
7432	042154	042526	046040	040517
7433	042162	104		
7434	042163	015	052012	042510
7435	042170	051440	044520	042116
7436	042176	042514	051440	047510
7437	042204	046125	020104	052123
7438	042212	051101	020124	047101
7439	042220	020104	044124	020105
7440	042226	051447	047524	023520
7441	042234	044440	042116	041511
7442	042242	052101	051117	051440
7443	042250	047510	046125	020104
7444	042256	047507	047440	043106
7445	042264	005015	051047	040505
7446	042272	054504	020047	044123
7447	042300	052517	042114	043440
7448	042306	020117	047117	044440
7449	042314	020116	050101	051120
7450	042322	054117	020056	020061
7451	042330	044515	006516	000012
7452	042336	005015	042526	044522
7453	042344	054506	042040	047517

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>

MSG43: .ASCIZ <CR><LF>/DESELECT BOTH PORTS/<CR><LF>

MSG44: .ASCIZ <CR><LF>/SELECT CORRECT PORT/<CR><LF>

MSG45: .ASCIZ <CR><LF>/CORRECT PORT NOT SELECTED, TRY AGAIN/<CR><LF>

MSG46: .ASCII <CR><LF>/DO A MANUAL DRIVE LOAD/

.ASCII <CR><LF>/THE SPINDLE SHOULD START AND THE 'STOP' INDICATOR SHOULD GO OFF

.ASCIZ <CR><LF>/'READY' SHOULD GO ON IN APPROX. 1 MIN/<CR><LF>

MSG47: .ASCIZ <CR><LF>/VERIFY DOOR CANNOT BE OPENED (DO NOT FORCE)/<CR><LF>

7454	042352	020122	040503	047116
7455	042360	052117	041040	020105
7456	042366	050117	047105	042105
7457	042374	024040	047504	047040
7458	042402	052117	043040	051117
7459	042410	042503	006451	000012
7460	042416	005015	042504	051120
7461	042424	051505	020123	044124
7462	042432	020105	051047	047125
7463	042440	051455	047524	023520
7464	042446	051440	044527	041524
7465	042454	020110	047524	023440
7466	042462	052123	050117	047
7467	042467	015	053012	051105
7468	042474	043111	020131	044124
7469	042502	020105	042510	042101
7470	042510	020123	047125	047514
7471	042516	042101	020054	044124
7472	042524	020105	051047	040505
7473	042532	054504	020047	044514
7474	042540	044107	020124	047507
7475	042546	051505	047440	043106
7476	042554	005015	020046	044124
7477	042562	020105	051447	047524
7478	042570	023520	046040	043511
7479	042576	052110	043440	042517
7480	042604	020123	047117	005015
7481	042612	000		
7482	042613	015	052012	051125
7483	042620	020116	043117	020106
7484	042626	041501	050040	053517
7485	042634	051105	043040	047522
7486	042642	020115	042502	044510
7487	042650	042116	052040	042510
7488	042656	051040	030113	006466
7489	042664	000012		
7490	042666	005015	053523	052111
7491	042674	044103	040440	020103
7492	042702	047520	042527	020122
7493	042710	040502	045503	047440
7494	042716	000116		
7495	042720	005015	047515	042515
7496	042726	052116	051101	046111
7497	042734	020131	042522	047515
7498	042742	042526	023040	044440
7499	042750	051516	051105	020124
7500	042756	044124	020105	040523
7501	042764	042515	052440	044516
7502	042772	020124	042523	042514
7503	043000	052103	050040	052514
7504	043006	107		
7505	043007	015	052012	020117
7506	043014	042522	042523	020124
7507	043022	047526	052514	042515
7508	043030	053040	046101	042111
7509	043036	005015	000	

MSG48: .ASCII <CR><LF> 'DEPRESS THE 'RUN-STOP' SWITCH TO 'STOP'/'

.ASCII <CR><LF>/VERIFY THE HEADS UNLOAD, THE 'READY' LIGHT GOES OFF/'

.ASCIZ <CR><LF>/& THE 'STOP' LIGHT GOES ON/<CR><LF>

MSG49: .ASCIZ <CR><LF> 'TURN OFF AC POWER FROM BEHIND THE RK06'<CR><LF>

MSG50: .ASCIZ <CR><LF>/SWITCH AC POWER BACK ON/'

MSG51: .ASCII <CR><LF>/MOMENTARILY REMOVE & INSERT THE SAME UNIT SELECT PLUG/'

.ASCIZ <CR><LF>/TO RESET VOLUME VALID/<CR><LF>

7510	043041	015	042012	050105
7511	043046	042522	051523	051440
7512	043054	040520	042503	052040
7513	043062	020117	047504	052040
7514	043070	051505	124	
7515	043073	015	047412	020122
7516	043100	047503	052116	047522
7517	043106	026514	020105	047524
7518	043114	041040	050131	051501
7519	043122	020123	047105	044524
7520	043130	042522	052040	051505
7521	043136	006524	000012	
7522	043142	005015	044504	040523
7523	043150	046102	020105	044124
7524	043156	020105	051127	052111
7525	043164	020105	047514	045503
7526	043172	051440	044527	041524
7527	043200	026110	053040	051105
7528	043206	043111	020131	044514
7529	043214	044107	020124	047507
7530	043222	051505	047440	043106
7531	043230	005015	000	
7532	043233	015	042412	040516
7533	043240	046102	020105	044124
7534	043246	020105	051127	052111
7535	043254	020105	047514	045503
7536	043262	051440	044527	041524
7537	043270	026110	053040	051105
7538	043276	043111	020131	044514
7539	043304	044107	020124	047507
7540	043312	051505	047440	006516
7541	043320	000012		
7542	043322	005015	054105	052111
7543	043330	052040	051505	020124
7544	043336	044527	044124	047440
7545	043344	044522	044507	040516
7546	043352	020114	047125	052111
7547	043360	051440	046105	041505
7548	043366	020124	046120	043525
7549	043374	047040	027117	000040
7550	043402	005015	046101	043511
7551	043410	046516	047105	020124
7552	043416	040503	052122	044522
7553	043424	043504	020105	051525
7554	043432	042105		
7555	043434	005015	051120	043517
7556	043442	040522	020115	044527
7557	043450	046114	041040	050131
7558	043456	051501	020123	044124
7559	043464	020105	051127	052111
7560	043472	020105	047514	045503
7561	043500	052040	051505	124
7562	043505	015	040412	042116
7563	043512	051040	040505	026504
7564	043520	051127	052111	020105
7565	043526	040504	040524	050040

MSG52: .ASCII <CR><LF>/DEPRESS SPACE TO DO TEST/

.ASCIZ <CR><LF>/OR CONTROL-E TO BYPASS ENTIRE TEST/<CR><LF>

MSG53: .ASCIZ <CR><LF>/DISABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES OFF/<CR><LF>

MSG54: .ASCIZ <CR><LF>/ENABLE THE WRITE LOCK SWITCH, VERIFY LIGHT GOES ON/<CR><LF>

MSG55: .ASCIZ <CR><LF>/EXIT TEST WITH ORIGINAL UNIT SELECT PLUG NO. /

MSG56: .ASCII <CR><LF>/ALIGNMENT CARTRIDGE USED/

.ASCII <CR><LF>/PROGRAM WILL BYPASS THE WRITE LOCK TEST/

.ASCIZ <CR><LF>/AND READ-WRITE DATA PORTION OF AC LOW DETECTION TEST-PART 2/<CR>

76111 050131
76112 050131
76113 050131
76114 050131
76115 050131
76116 050131
76117 050131
76118 050131
76119 050131
76120 050131
76121 050131
76122 050131
76123 050131
76124 050131
76125 050131
76126 050131
76127 050131
76128 050131
76129 050131
76130 050131
76131 050131
76132 050131
76133 050131
76134 050131
76135 050131
76136 050131
76137 050131
76138 050131
76139 050131
76140 050131
76141 050131
76142 050131
76143 050131
76144 050131
76145 050131
76146 050131
76147 050131
76148 050131
76149 050131
76150 050131
76151 050131
76152 050131
76153 050131
76154 050131
76155 050131
76156 050131
76157 050131
76158 050131
76159 050131
76160 050131
76161 050131
76162 050131
76163 050131
76164 050131
76165 050131
76166 050131
76167 050131
76168 050131
76169 050131
76170 050131
76171 050131
76172 050131
76173 050131
76174 050131
76175 050131
76176 050131
76177 050131
76178 050131
76179 050131
76180 050131
76181 050131
76182 050131
76183 050131
76184 050131
76185 050131
76186 050131
76187 050131
76188 050131
76189 050131
76190 050131
76191 050131
76192 050131
76193 050131
76194 050131
76195 050131
76196 050131
76197 050131
76198 050131
76199 050131
76200 050131

MSG57: .ASCIZ (CR)(LF)/VERIFY BATTERY RETRACT FUNCTIONAL/(CR)(LF)

MSG58: .ASCIZ (CR)(LF)/LOAD HEADS ON ALL DRIVES TO BE TESTED FOR MDS/(CR)(LF)

MSG59: .ASCIZ (CR)(LF)/INSERT SAME UNIT SELECT PLUG NUMBER IN ANY 2 DRIVES TO BE TESTE

MSG60: .ASCII (CR)(LF)/INSERT CORRECT UNIT SELECT PLUGS & LOAD HEADS/

.ASCIZ (CR)(LF)/ON PREVIOUS 2 DRIVES/(CR)(LF)

MSG61: .ASCIZ (CR)(LF)/MULTIPLE DRIVES FOUND ON DRIVE NO.

MSG62: .ASCII (CR)(LF)/BYPASSING MULT. DRIVE SELECT TEST/

7622	044224	051501	044523	043516
7623	044224	046440	046125	027124
7624	044224	042040	044522	042526
7625	044224	051440	046105	041505
7626	044224	020124	042524	052123
7627	044224	050115	047117	054514
7628	044224	030440	042040	044522
7629	044224	042526	050040	041505
7630	044224	042523	052116	050115
7631	044224	050115		
7632	044224	042012	042012	050105
7633	044224	042522	051523	051440
7634	044224	042520	042503	041040
7635	044224	051101	052040	020117
7636	044224	047504	040440	047516
7637	044350	044124	051105	031040
7638	044356	042040	044522	042526
7639	044364	123		
7640	044365	015	047412	020122
7641	044372	047503	052116	047522
7642	044400	026514	020105	047524
7643	044406	042440	044530	020124
7644	044414	042524	052123	
7645	044420	050115	044450	051516
7646	044426	051105	020124	047503
7647	044432	051122	041505	020124
7648	044442	047125	052111	051440
7649	044450	046105	041505	020124
7650	044456	046120	043525	020123
7651	044464	042502	047506	042522
7652	044472	042440	044530	044524
7653	044500	043516	006451	000012
7654	044506	050115	044124	020105
7655	044514	042523	047503	042116
7656	044522	050040	051117	044524
7657	044530	047117	047440	020106
7658	044536	044124	051511	052040
7659	044544	051505	020124	051511
7660	044552	052040	046505	047520
7661	044560	040522	044522	054514
7662	044566	041040	050131	051501
7663	044574	042523	104	
7664	044577	015	051412	042505
7665	044604	041440	046517	042515
7666	044612	052116	020123	047111
7667	044620	052040	042510	046040
7668	044626	051511	044524	043516
7669	044634	005015	000	
7670	044637	015	042012	050105
7671	044644	042522	051523	023440
7672	044652	052522	026516	052123
7673	044660	050117	020047	053523
7674	044666	052111	044103	052040
7675	044674	020117	051447	047524
7676	044702	023520	000	
7677	044705	015	042012	050105

.ASCII <CR><LF> ONLY : DRIVE PRESENT/<CR><LF>

MSG63: .ASCII <CR><LF>/DEPRESS SPACE BAR TO DO ANOTHER 2 DRIVES/

.ASCII <CR><LF>/OR CONTROL-E TO EXIT TEST/

.ASCII <CR><LF>/<INSERT CORRECT UNIT SELECT PLUGS BEFORE EXITING>/<CR><LF>

MSG64: .ASCII <CR><LF>/THE SECOND PORTION OF THIS TEST IS TEMPORARILY BYPASSED/

.ASCII <CR><LF>/SEE COMMENTS IN THE LISTING/<CR><LF>

MSG65: .ASCII <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'STOP' /

MSG66: .ASCII <CR><LF>/DEPRESS 'RUN-STOP' SWITCH TO 'RUN' /

7678	045367	015	050012	047522
7679	045367	015	050012	047522
7680	045367	015	050012	047522
7681	045367	015	050012	047522
7682	045367	015	050012	047522
7683	045367	015	050012	047522
7684	045367	015	050012	047522
7685	045367	015	050012	047522
7686	045367	015	050012	047522
7687	045367	015	050012	047522
7688	045367	015	050012	047522
7689	045367	015	050012	047522
7690	045367	015	050012	047522
7691	045367	015	050012	047522
7692	045367	015	050012	047522
7693	045367	015	050012	047522
7694	045367	015	050012	047522
7695	045367	015	050012	047522
7696	045367	015	050012	047522
7697	045367	015	050012	047522
7698	045367	015	050012	047522
7699	045367	015	050012	047522
7700	045367	015	050012	047522
7701	045367	015	050012	047522
7702	045367	015	050012	047522
7703	045367	015	050012	047522
7704	045367	015	050012	047522
7705	045367	015	050012	047522
7706	045367	015	050012	047522
7707	045367	015	050012	047522
7708	045367	015	050012	047522
7709	045367	015	050012	047522
7710	045367	015	050012	047522
7711	045367	015	050012	047522
7712	045367	015	050012	047522
7713	045367	015	050012	047522
7714	045367	015	050012	047522
7715	045367	015	050012	047522
7716	045367	015	050012	047522
7717	045367	015	050012	047522
7718	045367	015	050012	047522
7719	045367	015	050012	047522
7720	045367	015	050012	047522
7721	045367	015	050012	047522
7722	045367	015	050012	047522
7723	045367	015	050012	047522
7724	045367	015	050012	047522
7725	045367	015	050012	047522
7726	045367	015	050012	047522
7727	045367	015	050012	047522
7728	045367	015	050012	047522
7729	045367	015	050012	047522
7730	045367	015	050012	047522
7731	045367	015	050012	047522
7732	045367	015	050012	047522
7733	045367	015	050012	047522

.ASCIZ <CR><LF>/8 DEPRESS SPACE WHEN 'READY' LIGHT GOES ON<CR><LF>

MSG67: .ASCIZ <CR><LF>/DESELECT PORT SWITCH ON ALL OTHER DRIVES<CR><LF>

MSG68: .ASCIZ <CR><LF>/VERIFY BOTH DRIVES UNLOADED<CR><LF>

MSG69: .ASCIZ <CR><LF> DEPRESS SPACE WHEN 'READY' LIGHT GOES ON<CR><LF>

MSG70: .ASCIZ <CR><LF>/VERIFY HEADS LOAD<CR><LF>

MSG71: .ASCIZ <CR><LF>/SELECT CORRECT PORT SWITCH ON ALL OTHER DRIVES<CR><LF>

MSG72: .ASCIZ <CR><LF>/ABORTING BALANCE OF TESTS<CR><LF>

MSG74: .ASCIZ <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT<CR><LF>

7734	045407	051107	046501	040440
7735	045408	051502	052122	050040
7736	045410	047105	044504	043516
7737	045416	027056	050056	042514
7738	045424	051501	020105	040527
7739	045432	052111	000	
7740	045435	015	044012	046101
7741	045442	020124	042520	042116
7742	045450	047111	027107	027056
7743	045456	046120	040505	042523
7744	045464	053440	044501	000124
7745	045472	005015	051120	043517
7746	045500	040522	020115	041101
7747	045506	051117	042524	000104
7748	045514	005015	050103	020125
7749	045522	040510	052114	042105
7750	045530	000		
7751	045531	015	051412	051117
7752	045536	054522	020054	051127
7753	045544	052111	020105	047514
7754	045552	045503	051440	047510
7755	045560	046125	020104	047516
7756	045566	020124	042502	042040
7757	045574	050105	042522	051523
7758	045602	042105		
7759	045604	005015	044127	046111
7760	045612	020105	047117	052040
7761	045620	040522	045503	031040
7762	045628	051440	041505	047524
7763	045634	051522	030440	026071
7764	045642	031040	020060	051117
7765	045650	031040	061	
7766	045653	015	050012	042514
7767	045660	051501	020105	051124
7768	045666	020131	043501	044501
7769	045674	006516	000012	
7770				
7771				
7772				
7773	045700	005015	051105	047522
7774	045706	026122	047440	046116
7775	045714	020131	020060	044124
7776	045722	052522	033440	040440
7777	045730	046114	053517	042105
7778	045736	020054	051124	020131
7779	045744	043501	044501	006516
7780	045752	000012		
7781	045754	051104	053111	020105
7782	045762	020043	047111	051040
7783	045770	041513	031123	041440
7784	045776	047101	047516	020124
7785	046004	042502	051040	040505
7786	046012	020104	040502	045503
7787	046020	041440	051117	042522
7788	046026	052103	054514	044440
7789	046034	020116	045522	051115

MSG75: .ASCIZ <CR><LF>/HALT PENDING...PLEASE WAIT/

MSG76: .ASCIZ <CR><LF>/PROGRAM ABORTED/

MSG77: .ASCIZ <CR><LF>/CPU HALTED/

MSG100: .ASCII <CR><LF>/SORRY, WRITE LOCK SHOULD NOT BE DEPRESSED/

.ASCII <CR><LF>/WHILE ON TRACK 2 SECTORS 19, 20 OR 21/

.ASCIZ <CR><LF>/PLEASE TRY AGAIN/<CR><LF>

.SBTTL ERROR MESSAGES

EM1: .ASCIZ <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>

EM2: .ASCIZ /DRIVE # IN RKCS2 CANNOT BE REAC BACK CORRECTLY IN RKMR2/

7846	046526	042522	044507	052123		
7847	046534	051105	000			
7848	046537	104	040522	044440	EM11:	.ASCIZ /DRA IN RKDS & MED IN RKCS2 BOTH SET/
7849	046544	020116	045522	051504		
7850	046552	023040	047040	042105		
7851	046560	044440	020116	045522		
7852	046566	051503	020062	047502		
7853	046574	044124	051440	052105		
7854	046602	000				
7855	046603	103	047117	051124	EM12:	.ASCIZ /CONTROLLER NOT READY IN RKCS1/
7856	046610	046117	042514	020122		
7857	046616	047516	020124	042522		
7858	046624	042101	020131	047111		
7859	046632	051040	041513	030523		
7860	046640	000				
7861	046641	116	020117	052101	EM13:	.ASCIZ /NO ATTN IN RKASOF/
7862	046646	047124	044440	020116		
7863	046654	045522	051501	043117		
7864	046662	000				
7865	046663	127	047522	043516	EM14:	.ASCIZ /WRONG ATTN IN RKASOF/
7866	046670	040440	052124	020116		
7867	046676	047111	051040	040513		
7868	046704	047523	000106			
7869	046710	051104	054504	047040	EM15:	.ASCIZ /DRDY NOT CLEARED IN RKMR2/
7870	046716	052117	041440	042514		
7871	046724	051101	042105	044440		
7872	046732	020116	045522	051115		
7873	046740	000062				
7874	046742	051504	020103	047516	EM16:	.ASCIZ /DSC NOT SET IN RKMR2/
7875	046750	020124	042523	020124		
7876	046756	047111	051040	046513		
7877	046764	031122	000			
7878	046767	115	051505	040523	EM17:	.ASCIZ /MESSAGE A0 ERROR/
7879	046774	042507	040440	020060		
7880	047002	051105	047522	000122		
7881	047010	042515	051523	043501	EM18:	.ASCIZ /MESSAGE B0 ERROR/
7882	047016	020105	030102	042440		
7883	047024	051122	051117	000		
7884	047031	115	051505	040523	EM19:	.ASCIZ /MESSAGE A1 ERROR/
7885	047036	042507	040440	020061		
7886	047044	051105	047522	000122		
7887	047052	042515	051523	043501	EM20:	.ASCIZ /MESSAGE B1 ERROR/
7888	047060	020105	030502	042440		
7889	047066	051122	051117	000		
7890	047073	103	051105	020122	EM21:	.ASCIZ /CERR SET IN RKCS1/
7891	047100	042523	020124	047111		
7892	047106	051040	041513	030523		
7893	047114	000				
7894	047115	104	047517	020122	EM22:	.ASCIZ /DOOR STATUS IN RKMR2 NOT CLEARED/
7895	047122	052123	052101	051525		
7896	047130	044440	020116	045522		
7897	047136	051115	020062	047516		
7898	047144	020124	046103	040505		
7899	047152	042522	000104			
7900	047156	050123	047111	046104	EM23:	.ASCIZ /SPINDLE ON SET IN RKMR2/
7901	047164	020105	047117	051440		

7903	047172	052105	044440	020116	
7903	047200	045522	051115	000062	
7904	047206	047326	052514	042515	EM24: .ASCIZ /VOLUME VALID NOT SET IN RKMR2/
7905	047214	053040	046101	042111	
7906	047222	047040	052117	051440	
7907	047230	052105	044440	020116	
7908	047236	045522	051115	000062	
7909	047244	040503	052122	044522	EM25: .ASCIZ /CARTRIDGE STATUS IN RKMR2 NOT CLEARED/
7910	047252	043504	020105	052123	
7911	047260	052101	051525	044440	
7912	047266	020116	045522	051115	
7913	047274	020062	047516	020124	
7914	047302	046103	040505	042522	
7915	047310	000104			
7916	047312	047526	052514	042515	EM26: .ASCIZ /VOLUME VALID NOT CLEARED IN RKMR2/
7917	047320	053040	046101	042111	
7918	047326	047040	052117	041440	
7919	047334	042514	051101	042105	
7920	047342	044440	020116	045522	
7921	047350	051115	000062		
7922	047354	042516	020104	047516	EM27: .ASCIZ /NED NOT SET IN RKCS2/
7923	047362	020124	042523	020124	
7924	047370	047111	051040	041513	
7925	047376	031123	000		
7926	047401	126	046117	046525	EM28: .ASCIZ /VOLUME VALID SET IN RKMR2/
7927	047406	020105	040526	044514	
7928	047414	020104	042523	020124	
7929	047422	047111	051040	046513	
7930	047430	031122	000		
7931	047433	104	041523	047040	EM29: .ASCIZ /DSC NOT SET IN RKMR2/
7932	047440	052117	051440	052105	
7933	047446	044440	020116	045522	
7934	047454	051115	000062		
7935	047460	052101	047124	047040	EM30: .ASCIZ /ATTN NOT RESET IN RKASOF/
7936	047466	052117	051040	051505	
7937	047474	052105	044440	020116	
7938	047502	045522	051501	043117	
7939	047510	000			
7940	047511	116	042105	047040	EM31: .ASCIZ /NED NOT CLEARED IN RKCS2/
7941	047516	052117	041440	042514	
7942	047524	051101	042105	044440	
7943	047532	020116	045522	051503	
7944	047540	000062			
7945	047542	050123	047111	046104	EM32: .ASCIZ /SPINDLE ON NOT SET IN RKMR2/
7946	047550	020105	047117	047040	
7947	047556	052117	051440	052105	
7948	047564	044440	020116	045522	
7949	047572	051115	000062		
7950	047576	051104	052111	020105	EM33: .ASCIZ /DRIVE NOT READY IN RKMR2/
7951	047604	047516	020124	042522	
7952	047612	042101	020131	047111	
7953	047620	051040	046513	031122	
7954	047626	000			
7955	047627	104	047517	020122	EM34: .ASCIZ /DOOR STATUS BIT NOT SET IN RKMR2/
7956	047634	052123	052101	051525	
7957	047642	041040	052111	047040	

7958	047650	052117	051440	052105	
7959	047656	044440	020116	045522	
7960	047664	051115	000062		
7961	047670	042510	042101	020123	EM35: .ASCIZ /HEADS HOME NOT SET IN RKMR2/
7962	047676	047510	042515	047040	
7963	047704	052117	051440	052105	
7964	047712	044440	020116	045522	
7965	047720	051115	000062		
7966	047724	054503	020114	042101	EM36: .ASCIZ /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
7967	047732	051104	044440	020116	
7968	047740	045522	051115	020063	
7969	047746	047516	020124	040523	
7970	047754	042515	040440	020123	
7971	047762	045522	041504	000	
7972	047767	101	020103	047514	EM37: .ASCIZ /AC LOW NOT SET IN RKMR3/
7973	047774	020127	047516	020124	
7974	050002	042523	020124	047111	
7975	050010	051040	046513	031522	
7976	050016	000			
7977	050017	101	020103	047514	EM38: .ASCIZ /AC LOW DID NOT SET FAULT IN RKMR3/
7978	050024	020127	044504	020104	
7979	050032	047516	020124	042523	
7980	050040	020124	040506	046125	
7981	050046	020124	047111	051040	
7982	050054	046513	031522	000	
7983	050061	103	046131	042040	EM39: .ASCIZ /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
7984	050066	043111	020106	020046	
7985	050074	043117	051506	052105	
7986	050102	044440	020116	045522	
7987	050110	051115	020062	047516	
7988	050116	020124	046103	040505	
7989	050124	042522	000104		
7990	050130	054503	020114	042101	EM40: .ASCIZ /CYL ADDR IN RKMR3 NOT CLEARED/
7991	050136	051104	044440	020116	
7992	050144	045522	051115	020063	
7993	050152	047516	020124	046103	
7994	050160	040505	042522	000104	
7995	050166	054503	020114	042101	EM41: .ASCIZ /CYL ADDR IN RKMR3 DID NOT REMAIN CLEARED/
7996	050174	051104	044440	020116	
7997	050202	045522	051115	020063	
7998	050210	044504	020104	047516	
7999	050216	020124	042522	040515	
8000	050224	047111	041440	042514	
8001	050232	051101	042105	000	
8002	050237	116	042105	047040	EM42: .ASCIZ /NED NOT SET IN RKCS2/
8003	050244	052117	051440	052105	
8004	050252	044440	020116	045522	
8005	050260	051503	000062		
8006	050264	041501	047514	047040	EM43: .ASCIZ /ACLO NOT CLEARED IN RKMR3/
8007	050272	052117	041440	042514	
8008	050300	051101	042105	044440	
8009	050306	020116	045522	051115	
8010	050314	000063			
8011	050316	047526	052514	042515	EM44: .ASCIZ /VOLUME VALID NOT CLEARED IN RKMR2/
8012	050324	053040	046101	042111	
8013	050332	047040	052117	041440	

8014	050340	042514	051101	042105	
8015	050346	044440	020116	045522	
8016	050354	051115	000062		
8017	050360	047526	052514	042515	EM45: .ASCIZ /VOLUME VALID SET IN RKMR2 AFTER HEADS LOADED/
8018	050366	053040	046101	042111	
8019	050374	051440	052105	044440	
8020	050402	020116	045522	051115	
8021	050410	020062	043101	042524	
8022	050416	020122	042510	042101	
8023	050424	020123	047514	042101	
8024	050432	042105	000		
8025	050435	116	047117	042455	EM46: .ASCIZ /NON-EXECUTABLE FUNCTION (NXF) NOT SET IN RKMR3/
8026	050442	042530	052503	040524	
8027	050450	046102	020105	052505	
8028	050456	047516	044524	047117	
8029	050464	024040	054116	024506	
8030	050472	047040	052117	051440	
8031	050500	052105	044440	020116	
8032	050506	045522	051115	000063	
8033	050514	054503	044514	042116	EM47: .ASCIZ /CYLINDER ADDRESS CHANGED FROM 0/
8034	050522	051105	040440	042104	
8035	050530	042522	051523	041440	
8036	050536	040510	043516	042105	
8037	050544	043040	047522	020115	
8038	050552	000060			
8039	050554	051127	052111	020105	EM48: .ASCIZ /WRITE LOCK IN RKMR2 NOT CLEARED/
8040	050562	047514	045503	044440	
8041	050570	020116	045522	051115	
8042	050576	020062	047516	020124	
8043	050604	046103	040505	042522	
8044	050612	000104			
8045	050614	051127	052111	020105	EM49: .ASCIZ /WRITE LOCK IN RKMR2 NOT SET/
8046	050622	047514	045503	044440	
8047	050630	020116	045522	051115	
8048	050636	020062	047516	020124	
8049	050644	042523	000124		
8050	050650	051127	052111	020105	EM50: .ASCIZ /WRITE LOCK ERROR IN RKMR3 NOT SET/
8051	050656	047514	045503	042440	
8052	050664	051122	051117	044440	
8053	050672	020116	045522	051115	
8054	050700	020063	047516	020124	
8055	050706	042523	000124		
8056	050712	051127	052111	020105	EM51: .ASCIZ /WRITE LOCK DID NOT OCCUR AT SECTOR BOUNDARY/
8057	050720	047514	045503	042040	
8058	050726	042111	047040	052117	
8059	050734	047440	041503	051125	
8060	050742	040440	020124	042523	
8061	050750	052103	051117	041040	
8062	050756	052517	042116	054522	
8063	050764	000			
8064	050765	125	051516	047040	EM52: .ASCIZ /UNS NOT SET IN RKMR3/
8065	050772	052117	051440	052105	
8066	051000	044440	020116	045522	
8067	051006	051115	000063		
8068					
8069	051012	047125	047514	042101	EM53: .ASCIZ /UNLOAD NOT SET IN RKMR2/

8070	051020	047040	052117	051440	
8071	051026	052105	044440	020116	
8072	051034	045522	051115	000062	
8073	051042	0430503	047116	052117	EM54: .ASCIZ /CANNOT FIND MULT. DRIVE SELECT IN RKCS2/
8074	051050	043040	047111	020104	
8075	051056	052515	052114	020056	
8076	051064	051104	053111	020105	
8077	051072	042523	042514	052103	
8078	051100	044440	020116	045522	
8079	051106	051503	000062		
8080	051112	052101	047124	047040	EM55: .ASCIZ /ATTN NOT CLEARED IN RKASOF/
8081	051120	052117	041440	042514	
8082	051126	051101	042105	044440	
8083	051134	020116	045522	051501	
8084	051142	043117	000		
8085	051145	125	042516	050130	EM56: .ASCIZ /UNEXPECTED MEMORY PARITY ERROR TRAP/
8086	051152	041505	042524	020104	
8087	051160	042515	047515	054522	
8088	051166	050040	051101	052111	
8089	051174	020131	051105	047522	
8090	051202	020122	051124	050101	
8091	051210	000			
8092	051211	103	051105	020122	EM57: .ASCIZ /CERR IN RKCS1 NOT SET/
8093	051216	047111	051040	041513	
8094	051224	030523	047040	052117	
8095	051232	051440	052105	000	
8096	051237	120	051101	052111	EM58: .ASCIZ /PARITY NOT SET IN RKMR3/
8097	051244	020131	047516	020124	
8098	051252	042523	020124	047111	
8099	051260	051040	046513	031522	
8100	051266	000			
8101	051267	103	047524	051440	EM59: .ASCIZ /CTO SET IN RKCS1/
8102	051274	052105	044440	020116	
8103	051302	045522	051503	000061	
8104	051310	042510	042101	020123	EM60: .ASCIZ /HEADS HOME NOT FOUND IN RKMR2/
8105	051316	047510	042515	047040	
8106	051324	052117	043040	052517	
8107	051332	042116	044440	020116	
8108	051340	045522	051115	000062	
8109	051346	054116	020106	044504	EM61: .ASCIZ /NXF DID NOT SET FAULT/
8110	051354	020104	047516	020124	
8111	051362	042523	020124	040506	
8112	051370	046125	000124		
8113	051374	052104	020105	042523	EM62: .ASCIZ /DTE SET IN RKER/
8114	051402	020124	047111	051040	
8115	051410	042513	000122		
8116	051414	046104	020124	042523	EM63: .ASCIZ /DLT SET IN RKCS2/
8117	051422	020124	047111	051040	
8118	051430	041513	031123	000	
8119	051435	122	042113	020103	EM64: .ASCII /RKDC & RKDA INDICATE THAT WRITE CHECK ERROR/
8120	051442	020046	045522	040504	
8121	051450	044440	042116	041511	
8122	051456	052101	020105	044124	
8123	051464	052101	053440	044522	
8124	051472	042524	041440	042510	
8125	051500	045503	042440	051122	

0126	051506	051117			
0127	051510	005015	041517	052503	.ASCIZ <CR><LF>/OCCURRED AT CYL 411, TRACK 2, SECTOR 2:/
0128	051516	051122	042105	040440	
0129	051524	020124	054503	020114	
0130	051532	030464	026061	052040	
0131	051540	040522	045503	031040	
0132	051546	020054	042523	052103	
0133	051554	051117	031040	000061	
0134	051562	042522	042101	044040	EM65: .ASCIZ /READ HEADER ERROR/
0135	051570	040505	042504	020122	
0136	051576	051105	047522	000122	
0137	051604	054503	020114	042101	EM66: .ASCIZ /CYL ADDR IN RKM3 INCORRECT/
0138	051612	051104	044440	020116	
0139	051620	045522	051115	020063	
0140	051626	047111	047503	051122	
0141	051634	041505	000124		
0142	051640	042516	020104	042523	EM67: .ASCIZ /MED SET IN RKCS2/
0143	051646	020124	047111	051040	
0144	051654	041513	031123	000	
0145	051661	103	047101	047516	EM68: .ASCIZ /CANNOT READ BAD SECTOR INFORMATION/
0146	051666	020124	042522	042101	
0147	051674	041040	042101	051440	
0148	051702	041505	047524	020122	
0149	051710	047111	047506	046522	
0150	051716	052101	047511	000116	
0151	051724	047516	042040	044522	EM69: .ASCII /NO DRIVES FOUND ON BUSS/
0152	051732	042526	020123	047506	
0153	051740	047125	020104	047117	
0154	051746	041040	051525	123	
0155	051753	015	051412	052105	.ASCIZ <CR><LF>/SETUP CORRECTLY & PRESS 'CONTINUE'<CR><LF>
0156	051760	050125	041440	051117	
0157	051766	042522	052103	054514	
0158	051774	023040	050040	042522	
0159	052002	051523	023440	047503	
0160	052010	052116	047111	042525	
0161	052016	006447	000012		
0162	052022	044127	046111	020105	EM70: .ASCIZ /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
0163	052030	040527	052111	047111	
0164	052036	020107	047506	020122	
0165	052044	047503	052116	020122	
0166	052052	042522	042101	020131	
0167	052060	051117	040440	052106	
0168	052066	051105	041440	047117	
0169	052074	051124	051040	040505	
0170	052102	054504	051040	041505	
0171	052110	042047	000		
0172	052113	104	052105	041505	EM71: .ASCIZ /DETECTED 10 BAD SECTORS...ABORTING TEST/
0173	052120	042524	020104	030061	
0174	052126	041040	042101	051440	
0175	052134	041505	047524	051522	
0176	052142	027056	040456	047502	
0177	052150	052122	047111	020107	
0178	052156	042524	052123	000	
0179	052163	104	052105	041505	EM72: .ASCIZ /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
0180	052170	042524	020104	051502	
0181	052176	020105	052502	020124	

0182	052204	047516	020124	044514	
0183	052212	052123	042105	044440	
0184	052220	020116	040502	020104	
0185	052226	042523	052103	051117	
0186	052234	043040	046111	000105	
0187	052242	042504	042524	052103	EM73: .ASCII /DETECTED BSE IN READ COMMAND/
0188	052250	042105	041040	042523	
0189	052256	044440	020116	042522	
0190	052264	042101	041440	046517	
0191	052272	040515	042116		
0192	052276	005015	052502	020124	.ASCIZ <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
0193	052304	047516	020124	047111	
0194	052312	050040	042522	044526	
0195	052320	052517	020123	051127	
0196	052326	052111	020105	047503	
0197	052334	046515	047101	020104	
0198	052342	047524	051440	046501	
0199	052350	020105	042523	052103	
0200	052356	051117	000		
0201	052361	122	055124	047040	EM74: .ASCIZ /RTZ NOT SET IN RKMR2/
0202	052366	052117	051440	052105	
0203	052374	044440	020116	045522	
0204	052402	051115	000062		
0205	052406	005015	042504	042524	EM75: .ASCIZ <CR><LF>/DETECTED 10 BAD CYLINDERS...ABORTING TEST/
0206	052414	052103	042105	030440	
0207	052422	020060	040502	020104	
0208	052430	054503	044514	042116	
0209	052436	051105	027123	027056	
0210	052444	041101	051117	044524	
0211	052452	043516	052040	051505	
0212	052460	000124			
0213	052462	047516	042040	044522	EM76: .ASCII /NO DRIVES FOUND IN DEVICE MAP (\$DEVN)/
0214	052470	042526	020123	047506	
0215	052476	047125	020104	047111	
0216	052504	042040	053105	041511	
0217	052512	020105	040515	020120	
0218	052520	022050	042504	046526	
0219	052526	051			
0220	052527	015	051412	052105	.ASCIZ <CR><LF>/SETUP CORRECTLY & RESTART/<CR><LF>
0221	052534	050125	041440	051117	
0222	052542	042522	052103	054514	
0223	052550	023040	051040	051505	
0224	052556	040524	052122	005015	
0225	052564	000			
0226	052565	127	044522	042524	EM80: .ASCIZ /WRITE CHECK ERROR SET IN RKCS2/
0227	052572	041440	042510	045503	
0228	052600	042440	051122	051117	
0229	052606	051440	052105	044440	
0230	052614	020116	045522	051503	
0231	052622	000062			
0232	052624	051127	052111	020105	EM81: .ASCIZ /WRITE CHECK COMMAND NOT FUNCTIONING/
0233	052632	044103	041505	020113	
0234	052640	047503	046515	047101	
0235	052646	020104	047516	020124	
0236	052654	052506	041516	044524	
0237	052662	047117	047111	000107	

8238	052670	042522	042101	042040	EM82:	.ASCIZ	READ DATA DID NOT COMPARE WITH WRITE DATA/
8239	052676	052101	020101	044504			
8240	052704	020104	047516	020124			
8241	052712	047503	050115	051101			
8242	052720	020105	044527	044124			
8243	052726	053440	044522	042524			
8244	052734	042040	052101	000101			
8245	052742	040504	040524	041440	EM83:	.ASCIZ	/DATA CHECK ERROR SET IN RKE/
8246	052750	042510	045503	042440			
8247	052756	051122	051117	051440			
8248	052764	052105	044440	020116			
8249	052772	045522	051105	000			
8250	052777	117	043106	042523	EM84:	.ASCIZ	/OFFSET REG IN RKMR2 NOT 177/
8251	053004	020124	042522	020107			
8252	053012	047111	051040	046513			
8253	053020	031122	020040	047516			
8254	053026	020124	030440	033467			
8255	053034	000					
8256	053035	117	043106	042523	EM85:	.ASCIZ	/OFFSET STATUS BIT IN RKMR2 CLEARED/
8257	053042	020124	052123	052101			
8258	053050	051525	041040	052111			
8259	053056	044440	020116	045522			
8260	053064	051115	020062	046103			
8261	053072	040505	042522	000104			
8262	053100	043117	051506	052105	EM86:	.ASCIZ	/OFFSET REG IN RKMR2 NOT 77/
8263	053106	051040	043505	044440			
8264	053114	020116	045522	051115			
8265	053122	020062	047516	020124			
8266	053130	033467	000				
8267	053133	127	044522	042524	EM87:	.ASCIZ	/WRITE CHECK FAILURE AT OFFSET IN RKASOF/
8268	053140	041440	042510	045503			
8269	053146	043040	044501	052514			
8270	053154	042522	040440	020124			
8271	053162	043117	051506	052105			
8272	053170	044440	020116	045522			
8273	053176	051501	043117	000			
8274	053203	116	020117	051127	EM88:	.ASCIZ	/NO WRITE CHECK ERROR/
8275	053210	052111	020105	044103			
8276	053216	041505	020113	051105			
8277	053224	047522	000122				
8278	053230	042510	042101	020123	EM89:	.ASCIZ	/HEADS HOME NOT CLEARED IN RKMR2/
8279	053236	047510	042515	047040			
8280	053244	052117	041440	042514			
8281	053252	051101	042105	044440			
8282	053260	020116	045522	051115			
8283	053266	000062					
8284	053270	051124	041501	020113	EM90:	.ASCIZ	/TRACK FOLL OK NOT SET IN RKMR2/
8285	053276	047506	046114	047440			
8286	053304	020113	047516	020124			
8287	053312	042523	020124	047111			
8288	053320	051040	046513	031122			
8289	053326	000					
8290	053327	122	053105	047040	EM91:	.ASCIZ	/REV NOT SET IN RKMR2/
8291	053334	052117	051440	052105			
8292	053342	044440	020116	045522			
8293	053350	051115	000062				

117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

DH1: .ASCIZ TEST NO. PC/
DH2: .ASCIZ /RKMR2 RKMR3 RKER RKDS RKCS1 RKCS2 RKASSF/
DH3: .ASCIZ /AFTER AC SWITCHED OFF/
DH4: .ASCIZ /AFTER PACK RE-INSERTED & HEADS LOADED/
DH5: .ASCIZ /RKMR2 RKMR3 RKER RKDS RKDC RKCS1 RKCS2/
DH6: .ASCIZ /RKMR2 RKMR3 RKDC FROM CYL TO CYL CYL DIFF/
DH7: .ASCIZ /RKMR2 RKMR3 RKER RKDS RKDA RKCS1 RKCS2/
DH8: .ASCIZ /AFTER DRIVE UNLOADED & DOOR OPENED/
DH9: .ASCIZ /AFTER START SPINDLE COMMAND REC'D BY DRIVE/

051104	053111	DH10:	.ASCIZ	/AT END OF HEAD LOADING/
051104	053111	DH11:	.ASCIZ	/AFTER MANUALLY LOADING HEADS WITH DOOR OPEN/
052106	051105	DH12:	.ASCIZ	/AFTER DISK PACK REMOVED/
052106	051105	DH13:	.ASCIZ	/AFTER MANUALLY LOADING HEADS WITH DISK PACK REMOVED/
052106	051105	DH14:	.ASCIZ	/AFTER VOLUME VALID RESET/
044124	052517	DH15:	.ASCIZ	/WITHOUT PACK COMMAND/
052106	051105	DH16:	.ASCIZ	/AFTER UNIT SELECT PLUG REMOVED/
042524	020122	DH17:	.ASCIZ	/AFTER RECAL COMMAND/
042524	020122	DH18:	.ASCIZ	/AFTER UNLOAD COMMAND/
052106	051105	DH19:	.ASCIZ	/AFTER PACK COMMAND/
042524	020122	DH20:	.ASCIZ	/AFTER SELECT DRIVE COMMAND/

FILE 06-001-1E 09:44

DATA HEADERS

00000000	00000000	00000000	00000000	DH21:	.ASCIZ	AFTER SUBSYSTEM CLEAR
00000000	00000000	00000000	00000000	DH22:	.ASCIZ	AFTER DRIVE CLEAR COMMAND
00000000	00000000	00000000	00000000	DH23:	.ASCIZ	AFTER WRONG PORT SELECTED
00000000	00000000	00000000	00000000	DH24:	.ASCIZ	AFTER OFFSET COMMAND
00000000	00000000	00000000	00000000	DH25:	.ASCIZ	AFTER SEEK COMMAND
00000000	00000000	00000000	00000000	DH26:	.ASCIZ	AFTER READ DATA COMMAND
00000000	00000000	00000000	00000000	DH27:	.ASCIZ	AFTER WRITE DATA COMMAND
00000000	00000000	00000000	00000000	DH28:	.ASCIZ	AFTER BOTH PORTS DESELECTED
00000000	00000000	00000000	00000000	DH29:	.ASCIZ	AFTER CORRECT PORT SELECTED
00000000	00000000	00000000	00000000	DH30:	.ASCIZ	AFTER READ HEADER COMMAND
00000000	00000000	00000000	00000000	DH31:	.ASCIZ	AFTER DRIVE MANUALLY LOADED

F13

IBM 8K DRIVE DIAGNOSTIC PART 3
029610.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 161
DATA HEADERS

SEG 015:

055616	043101	042524	020122	DH32:	.ASCIZ	/AFTER WRITE CHECK COMMAND/
055624	051127	052111	020105			
055632	047103	041505	020113			
055640	047503	046515	047101			
055646	040504	044522	043516	DH33:	.ASCIZ	/DURING SEEK COMMAND/
055650	040504	042505	020113			
055656	047503	046515	047101			
055672	050104					
055674	043101	042524	020122	DH34:	.ASCIZ	/AFTER START SPINDLE COMMAND/
055702	052123	051101	020124			
055710	050123	047111	046104			
055716	020105	047503	046515			
055724	043101	000104				
055730	043101	042524	020122	DH35:	.ASCIZ	/AFTER MANUALLY UNLOADING/
055736	040515	052516	046101			
055744	054514	052440	046116			
055752	0517	044504	043516			
055760	000					
055761	122	046513	031122	DH36:	.ASCIZ	/RKMR2 RKMR3 RKCS1 RKCS2 FROM SECT TO SECT/
055766	020040	051040	046513			
055774	031522	020040	051040			
055800	041513	030523	020040			
055810	051040	041513	031123			
055816	020040	051106	046517			
055824	051440	041505	020124			
055832	052040	020117	042523			
055840	052103	000				
055843	052101	052106	051105	DH37:	.ASCIZ	/AFTER TIMEOUT TO POWER DOWN/
055850	052040	046511	047505			
055856	052125	052040	020117			
055864	047520	042527	020122			
055872	047504	047127	000			
055877	101	052106	051105	DH38:	.ASCIZ	/AFTER AC POWERED UP/
0558104	040440	020103	047520			
0558112	042527	042522	020104			
0558120	050125	000				
0558123	101	052106	051105	DH39:	.ASCIZ	/AFTER WRITE HEADER COMMAND/
0558130	053440	044522	042524			
0558136	044040	040505	042504			
0558144	020122	047503	046515			
0558152	047101	000104				
0558156	045522	051115	004462	DH40:	.ASCIZ	/RKMR2 RKMR3 RKDA WORD# HEADER WAS SHOULD BE/
0558164	045522	051115	004463			
0558172	045522	040504	053411			
0558200	051117	021504	044011			
0558206	040505	042504	020122			
0558214	040527	020123	051440			
0558222	047510	046125	020104			
0558230	042502	000				
0558233	104	051125	047111	DH41:	.ASCIZ	/DURING RECAL COMMAND/
0558240	020107	042522	040503			
0558246	020114	047503	046515			
0558254	047101	000104				
0558260	047117	051440	041505	DH42:	.ASCIZ	/ON SECTORS 0,2,4,6 OR 8 CYL 410 TRACK 2/
0558266	047524	051522	030040			

8574	056274	031054	032054	033054	
8575	056302	047440	020122	020070	
8576	056310	041440	046131	032040	
8577	056316	030061	052040	040522	
8578	056324	05503	031040	040000	
8579	056331	106	051117	040515	DH44: .ASCIZ /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
8580	056336	020124	020046	046101	
8581	056344	020114	042522	042101	
8582	056352	053455	044522	042524	
8583	056360	052040	051505	051524	
8584	056366	053440	046111	020114	
8585	056374	042502	041040	050131	
8586	056402	051501	042523	000104	
8587	056410	043101	042524	020122	DH45: .ASCIZ /AFTER SEEK WITH VOLUME VALID=0/
8588	056416	042523	045505	053440	
8589	056424	052111	020110	047526	
8590	056432	052514	042515	053040	
8591	056440	046101	042111	030075	
8592	056446	000			
8593	056447	101	052106	051105	DH46: .ASCIZ /AFTER WRITE DATA WITH VOLUME VALID=0/
8594	056454	053440	044522	042524	
8595	056462	042040	052101	020101	
8596	056470	044527	044124	053040	
8597	056476	046117	046525	020105	
8598	056504	040526	044514	036504	
8599	056512	000060			
8600	056514	043101	042524	020122	DH47: .ASCIZ /AFTER SEEK COMMAND WITH MOVEMENT/
8601	056522	042523	045505	041440	
8602	056530	046517	040515	042116	
8603	056536	053440	052111	020110	
8604	056544	047515	042526	042515	
8605	056552	052116	000		
8606	056555	101	052106	051105	DH48: .ASCIZ /AFTER WRITE LOCK SWITCH DISABLED/
8607	056562	053440	044522	042524	
8608	056570	046040	041517	020113	
8609	056576	053523	052111	044103	
8610	056604	042040	051511	041101	
8611	056612	042514	000104		
8612	056616	043101	042524	020122	DH49: .ASCIZ /AFTER WRITE LOCK SWITCH ENABLED/
8613	056624	051127	052111	020105	
8614	056632	047514	045503	051440	
8615	056640	044527	041524	020110	
8616	056646	047105	041101	042514	
8617	056654	000104			
8618	056656	043101	042524	020122	DH50: .ASCIZ /AFTER WRITING WITH WRITE LOCK ENABLED/
8619	056664	051127	052111	047111	
8620	056672	020107	044527	044124	
8621	056700	053440	044522	042524	
8622	056706	046040	041517	020113	
8623	056714	047105	041101	042514	
8624	056722	000104			
8625	056724	043101	042524	020122	DH51: .ASCIZ /AFTER SEEK TO SELF COMMAND/
8626	056732	042523	045505	052040	
8627	056740	020117	042523	043114	
8628	056746	041440	046517	040515	
8629	056754	042116	000		

H13

UNIBUS RKE DRIVE DIAGNOSTIC PART 3
029630.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 163
DATA HEADERS

SEG 0163

8630	056757	127	052111	020110	DH52:	.ASCIZ	/WITH INTENTIONAL MISCOMPARE/
8631	056764	047111	042524	052116			
8632	056772	047511	040516	020114			
8633	057000	044515	041523	046517			
8634	057006	040520	042522	000			
8635	057013	104	051125	047111	DH53:	.ASCIZ	/DURING OFFSET COMMAND/
8636	057020	020107	043117	051506			
8637	057026	052105	041440	046517			
8638	057034	040515	042116	000			
8639	057041	101	020124	040515	DH54:	.ASCIZ	/AT MAX POSITIVE OFFSET/
8640	057046	020130	047520	044523			
8641	057054	044524	042526	047440			
8642	057062	043106	042523	000124			
8643	057070	052101	046440	054101	DH55:	.ASCIZ	/AT MAX NEGATIVE OFFSET/
8644	057076	047040	043505	052101			
8645	057104	053111	020105	043117			
8646	057112	051506	052105	000			
8647	057117	122	046513	031122	DH56:	.ASCIZ	/RKMR2 RKMR3 RKDC CYL # HEADER WORD 3/
8648	057124	051011	046513	031522			
8649	057132	051011	042113	004503			
8650	057140	054503	020114	004443			
8651	057146	042510	042101	051105			
8652	057154	053440	051117	020104			
8653	057162	000060					
8654	057164	043101	042524	020122	DH57:	.ASCIZ	/AFTER WRITE COMMAND WITH OFFSET/
8655	057172	051127	052111	020105			
8656	057200	047503	046515	047101			
8657	057206	020104	044527	044124			
8658	057214	047440	043106	042523			
8659	057222	000124					
8660	057224	040504	040524	053440	DH58:	.ASCIZ	/DATA WAS SHOULD BE/
8661	057232	051501	004440	044123			
8662	057240	052517	042114	041040			
8663	057246	000105					
8664	057250	043101	042524	020122	DH59:	.ASCIZ	/AFTER DRIVE SELECT COMMAND WITH EVEN PARITY/
8665	057256	051104	053111	020105			
8666	057264	042523	042514	052103			
8667	057272	041440	046517	040515			
8668	057300	042116	053440	052111			
8669	057306	020110	052105	047105			
8670	057314	050040	051101	052111			
8671	057322	000131					
8672	057324	043101	042524	020122	DH60:	.ASCIZ	/AFTER WRITE LOCK ENABLED DURING CONTINUOUS WRITING/
8673	057332	051127	052111	020105			
8674	057340	047514	045503	042440			
8675	057346	040516	046102	042105			
8676	057354	042040	051125	047111			
8677	057362	020107	047503	052116			
8678	057370	047111	047525	051525			
8679	057376	053440	044522	044524			
8680	057404	043516	000				
8681	057407	122	046513	031122	DH61:	.ASCII	/RKMR2 RKMR3 RKCS1 RKCS2 RKDA WORD EXPECTED/
8682	057414	051011	046513	031522			
8683	057422	051011	041513	030523			
8684	057430	051011	041513	031123			
8685	057436	051011	042113	004501			

8686	057444	047527	042122	042411						
8687	057452	050130	041505	042524						
8688	057460	104								
8689	057461	015	004412	004411	.ASCIZ	<CR><LF>/	2 WRL	WAS	WORD/	
8690	057466	040011	053440	046122						
8691	057474	053411	051501	053411						
8692	057502	051117	000104							
8693	057506	045522	051115	004462	DH62:	.ASCII	/RKMR2 RKMR3 RKCS1 RKCS2 RKDA	WORD	EXPECTED/	
8694	057514	045522	051115	004463						
8695	057522	045522	051503	004461						
8696	057530	045522	051503	004462						
8697	057536	045522	040504	053411						
8698	057544	051117	004504	054105						
8699	057552	042103	052103	042105						
8700	057560	005015	004411	004411	.ASCII	<CR><LF>/	BEFORE	WAS	WORD/	
8701	057566	042502	047506	042522						
8702	057574	053411	051501	053411						
8703	057602	051117	104							
8704	057605	015	004412	004411	.ASCIZ	<CR><LF>/	WRL/			
8705	057612	053411	046122	000						
8706	057617	101	052106	051105	DH63:	.ASCIZ	/AFTER WRITE LOCK ENABLED FROM AC FAILURE/			
8707	057624	053440	044522	042524						
8708	057632	046040	041517	020113						
8709	057640	047105	041101	042514						
8710	057646	020104	051106	046517						
8711	057654	040440	020103	040506						
8712	057662	046111	051125	000103						
8713	057670	043101	042524	020122	DH64:	.ASCIZ	/AFTER MDS DETECTED IN RKCS2/			
8714	057676	042115	020123	042504						
8715	057704	042524	052103	042105						
8716	057712	044440	020116	045522						
8717	057720	051503	000062							
8718	057724	043101	042524	020122	DH65:	.ASCIZ	/AFTER SEARCHING ALL DRIVES PRESENT/			
8719	057732	042523	051101	044103						
8720	057740	047111	020107	046101						
8721	057746	020114	051104	053111						
8722	057754	051505	050040	042522						
8723	057762	042523	052116	000						
8724	057767	124	051505	020124	DH66:	.ASCIZ	/TEST NO. TRAP PC/			
8725	057774	047516	004456	051124						
8726	060002	050101	050040	000103						
8727	060010	043101	042524	020122	DH67:	.ASCIZ	/AFTER TIMEOUT TO ENABLE WRITE LOCK/			
8728	060016	044524	042515	052517						
8729	060024	020124	047524	042440						
8730	060032	040516	046102	020105						
8731	060040	051127	052111	020105						
8732	060046	047514	045503	000						
8733	060053	122	046513	031122	DH68:	.ASCIZ	/RKMR2 RKMR3 RKCS1 RKCS2 RKDC RKDA/			
8734	060060	051011	046513	031522						
8735	060066	051011	041513	030523						
8736	060074	051011	041513	031123						
8737	060102	051011	042113	004503						
8738	060110	045522	040504	000						
8739	060115	105	050130	041505	DH69:	.ASCIZ	/EXPECT EXPECT EXPECT EXPECT EXPECT EXPECT			
8740	060122	004524	054105	042520						
8741	060130	052103	042411	050130						

8742	060136	041505	004524	054105
8743	060144	042520	052103	042411
8744	060152	050130	041505	004524
8745	060160	054105	042520	052103
8746	060166	042411	050130	041505
8747	060174	000124		
8748	060176	041501	052524	046101
8749	060204	040411	052103	040525
8750	060212	004514	041501	052524
8751	060220	046101	040411	052103
8752	060226	040525	004514	041501
8753	060234	052524	046101	040411
8754	060242	052103	040525	004514
8755	060250	041501	052524	046101
8756	060256	000		
8757	060257	101	004460	030102
8758	060264	040411	004461	030502
8759	060272	040411	004462	031102
8760	060300	041011	000063	
8761	060304	045522	051503	004461
8762	060312	045522	051503	004462
8763	060320	045522	051501	043117
8764	060326	051011	042513	004522
8765	060334	045522	051504	051011
8766	060342	042113	000103	
8767	060346	051515	020107	020101
8768	060354	020046	020102	047111
8769	060362	051040	046513	031122
8770	060370	023040	051040	046513
8771	060376	031522	051040	051505
8772	060404	020120	051101	020105
8773	060412	047111	040526	044514
8774	060420	000104		
8775	060422	047117	051440	041505
8776	060430	047524	051522	030440
8777	060436	026060	030440	026062
8778	060444	030440	026064	030440
8779	060452	026066	030440	020070
8780	060460	051117	031040	020060
8781	060466	054503	020114	030464
8782	060474	020060	051124	041501
8783	060502	020113	000062	
8784				
8785				
8786				
8787				
8788	060506	001214	001116	005434
8789	060514	005436	005422	005420
8790	060522	005406	005410	005424
8791	060530	001214	001116	005434
8792	060536	005436	005406	005410
8793	060544	001166	001502	001504
8794	060552	001214	001116	005434
8795	060560	005436	005426	001364
8796	060566	001366	001374	
8797	060572	001214	001116	005434

DH70: .ASCIZ /ACTUAL ACTUAL ACTUAL ACTUAL ACTUAL ACTUAL ACTUAL/

DH71: .ASCIZ /A0 B0 A1 B1 A2 B2 B3/

DH72: .ASCIZ /RKCS1 RKCS2 RKASOF RKER RKDS RKDC/

DH73: .ASCIZ /MSG A & B IN RKMR2 & RKMR3 RESP ARE INVALID/

DH74: .ASCIZ /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/

.SBTTL ERROR OUTPUT DATA

.EVEN
DT1: \$TESTN,\$ERRPC,\$HMR2,\$HMR3,\$HER,\$HDS,\$HCS1,\$HCS2,\$HASOF

DT3: \$TESTN,\$ERRPC,\$HMR2,\$HMR3,\$HCS1,\$HCS2,\$TMP3,\$WD1,\$WD2

DT4: \$TESTN,\$ERRPC,\$HMR2,\$HMR3,\$HDC,\$FRCYL,\$TOCYL,\$CALDIF

DT5: \$TESTN,\$ERRPC,\$HMR2,\$HMR3,\$HER,\$HDS,\$HDA,\$HCS1,\$HCS2

08198	060600	005436	005422	005420
08199	060606	005416	005406	005410
08200	060614	001214	001354	
08201	060620	001214	001116	005434
08202	060626	005436	005416	001522
08203	060634	001540	005444	
08204	060640	001214	001116	005434
08205	060646	005436	005426	001366
08206	060654	001364	001374	
08207	060660	001214	001116	005434
08208	060666	005436	005426	001366
08209	060674	001760		
08210	060676	001214	001116	005434
08211	060704	005436	005406	005410
08212	060712	005426	005416	
08213	060716	001214	001116	005476
08214	060724	005500	005502	005504
08215	060732	005456	005460	005462
08216	060740	005464		
08217	060742	005406	005410	005424
08218	060750	005422	005420	005426
08219				
08220	060756	001214	001116	005476
08221	060764	005500	005502	005504
08222	060772	005506	005510	
08223	060776	005456	005460	005462
08224	061004	005464	005466	005470
08225	061012	005406	005410	005424
08226	061020	005422	005420	005426
08227				
08228	061026	001214	001116	005476
08229	061034	005500	005502	005504
08230	061042	005506	005510	005514
08231	061050	005456	005460	005462
08232	061056	005464	005466	005470
08233	061064	005474		
08234	061066	005406	005410	005424
08235	061074	005422	005420	005426
08236				
08237				
08238				
08239	061102	000002		
08240	061104	002	000	
08241	061106	054027		
08242	061110	007	000	
08243				
08244	061112	000003		
08245	061114	000	000	
08246	061116	054012		
08247	061120	002	000	
08248	061122	057407		
08249	061124	007	000	
08250				
08251	061126	000003		
08252	061130	000	000	
08253	061132	054012		

DT6: \$TESTN,TRAPPC
DT7: \$TESTN,\$ERRPC,HMR2,HMR3,HDA,WDCNT,HCWD,TEMP1
DT8: \$TESTN,\$ERRPC,HMR2,HMR3,HDC,TOCYL,FRCYL,CALDIF
DT9: \$TESTN,\$ERRPC,HMR2,HMR3,HDC,TOCYL,RHTAB
DT10: \$TESTN,\$ERRPC,HMR2,HMR3,HCS1,HCS2,HDC,HDA
DT13: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
HCS1,HCS2,HASOF,HER,HDS,HDC
DT14: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
HCS1,HCS2,HASOF,HER,HDS,HDC
DT15: \$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
HCS1,HCS2,HASOF,HER,HDS,HDC

.SBTTL ERROR DATA FORMATS

DF1: 2
.BYTE 2,0
DH2
.BYTE 7,0
DF3: 3
.BYTE 0,0
DH1
.BYTE 2,0
DH61
.BYTE 7,0
DF4: 3
.BYTE 0,0
DH1

8854	061134	002	000		.BYTE 2.0
8855	061136	057506			DH62
8856	061140	007	000		.BYTE 7.0
8857					
8858	061142	000001		DFS:	1
8859	061144	002	000		.BYTE 2.0
8860					
8861	061146	000003		DF6:	3
8862	061150	000	000		.BYTE 0.0
8863	061152	054012			DH1
8864	061154	002	000		.BYTE 2.0
8865	061156	054243			DH6
8866	061160	006	000		.BYTE 6.0
8867					
8868	061162	000002		DF7:	2
8869	061164	002	000		.BYTE 2.0
8870	061166	060053			DH68
8871	061170	006	000		.BYTE 6.0
8872	061172	000003		DF10:	3
8873	061174	000	000		.BYTE 0.0
8874	061176	054012			DH1
8875	061200	002	000		.BYTE 2.0
8876	061202	054027			DH2
8877	061204	007	000		.BYTE 7.0
8878					
8879	061206	000004		DF12:	4
8880	061210	000	000		.BYTE 0.0
8881	061212	060346			DH73
8882	061214	000	000		.BYTE 0.0
8883	061216	054012			DH1
8884	061220	002	000		.BYTE 2.0
8885	061222	054027			DH2
8886	061224	007	000		.BYTE 7.0
8887					
8888	061226	000002		DF14:	2
8889	061230	002	000		.BYTE 2.0
8890	061232	056156			DH40
8891	061234	006	000		.BYTE 6.0
8892					
8893					
8894	061236	000003		DF15:	3
8895	061240	000	000		.BYTE 0.0
8896	061242	054012			DH1
8897	061244	002	000		.BYTE 2.0
8898	061246	054325			DH7
8899	061250	007	000		.BYTE 7.0
8900					
8901					
8902	061252	000004		DF17:	4
8903	061254	000	000		.BYTE 0.0
8904	061256	056331			DH44
8905	061260	000	000		.BYTE 0.0
8906	061262	054012			DH1
8907	061264	002	000		.BYTE 2.0
8908	061266	054027			DH2
8909	061270	007	000		.BYTE 7.0

8910	061272	000003	
8911	061274	000	000
8912	061276	054012	
8913	061300	002	000
8914	061302	057117	
8915	061304	005	000
8916			
8917	061306	000007	
8918	061310	000	000
8919	061312	054012	
8920	061314	002	000
8921	061316	060115	
8922	061320	000	000
8923	061322	060257	
8924	061324	004	000
8925	061326	060176	
8926	061330	000	000
8927	061332	060257	
8928	061334	004	000
8929	061336	060304	
8930	061340	006	000
8931			
8932	061342	000007	
8933	061344	000	000
8934	061346	054012	
8935	061350	002	000
8936	061352	060115	
8937	061354	000	000
8938	061356	060257	
8939	061360	006	000
8940	061362	060176	
8941	061364	000	000
8942	061366	060257	
8943	061370	006	000
8944	061372	060304	
8945	061374	006	000
8946			
8947	061376	000007	
8948	061400	000	000
8949	061402	054012	
8950	061404	002	000
8951	061406	060115	
8952	061410	000	000
8953	061412	060257	
8954	061414	007	000
8955	061416	060176	
8956	061420	000	000
8957	061422	060257	
8958	061424	007	000
8959	061426	060304	
8960	061430	006	000

DF20:	3		
	.BYTE	0,0	
	DH1		
	.BYTE	2,0	
	DH56		
	.BYTE	5,0	
DF21:	7		
	.BYTE	0,0	
	DH1		
	.BYTE	2,0	
	DH69		
	.BYTE	0,0	
	DH71		
	.BYTE	4,0	
	DH70		
	.BYTE	0,0	
	DH71		
	.BYTE	4,0	
	DH72		
	.BYTE	6,0	
DF22:	7		
	.BYTE	0,0	
	DH1		
	.BYTE	2,0	
	DH69		
	.BYTE	0,0	
	DH71		
	.BYTE	6,0	
	DH70		
	.BYTE	0,0	
	DH71		
	.BYTE	6,0	
	DH72		
	.BYTE	6,0	
DF23:	7		
	.BYTE	0,0	
	DH1		
	.BYTE	2,0	
	DH69		
	.BYTE	0,0	
	DH71		
	.BYTE	7,0	
	DH70		
	.BYTE	0,0	
	DH71		
	.BYTE	7,0	
	DH72		
	.BYTE	6,0	

```

:*****
:SBTTL TYPE ERROR ROUTINE
:*ENTRY JSR PC,TYP ERR
:*RETURN RTS PC
:*

```

8961
8962
8963
8964
8965


```

0966
0967
0968
0969
0970
0971 061432 104413
0972 061434 032777 020000 117476
0973 061442 001107
0974 061444 113700 001114
0975 061450 042700 177400
0976 061454 005300
0977 061456 006300
0978 061460 006300
0979 061462 006300
0980 061464 006200 005560

```

```

: *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.
: *****
↑YPERR: SAVREG
BIT #SW13,DSWR :INHIBIT ERROR TYPEOUTS?
BNE 20$ :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
ADD #ERRTB,RO :FORM ADDRESS OF ERROR ENTRY

```

001204	061524	MOV	R0 +.25	:GET EH POINTER
001205	001205	BEG	35	:BRANCH IF THERE ISN'T ONE
		TYPE	.SCLF	:TYPE CARRIAGE RETURN LINE FEED
		TYPE		:TYPE ERROR MESSAGE (EM)
		WORD	0	:EM POINTER GOES HERE
061522	001205	MOV	(R0)+.45	:GET DH POINTER
		BEG	55	:BRANCH IF THERE ISN'T ONE
		TYPE	.SCLF	:TYPE CR-LF
		TYPE		:TYPE DATA HEADER
		WORD	0	:DH POINTER GOES HERE
		MOV	R0 +.R1	:GET DT POINTER
		BEG	205	:BRANCH IF THERE ARE NONE
		OR	R4	:SET INDENT SWITCH
		MOV	(R0)+.R0	:GET DF POINTER
		MOV	(R0)+.R2	:STORE NUMBER OF DH'S
		BEG	175	:DH NUM IS 0-BRANCH
		OR	R4	:NO INDENT
001205		TYPE	.SCLF	
		MOV	(R0)+.R3	:GET & STORE NUMBER OF DATA WORDS
		STB	(R0)+	:BUMP PAST FORMAT WORD
		STB	R2	:TEST IF ANY DATA FOR THIS HEADER
		BEG	145	:NO - SKIP DATA PRINT
		MOV	2(R1)+.-(SP)	:PUT FIRST DATA WORD ON STACK
		TYPE		:TYPE IT
		DEC	R3	:MORE DATA WORDS
		BEG	145	:NO-BRANCH
061716		TYPE	.SPACE2	:TYPE SEPARATORS
		BR	15	:LOOP
		DEC	R2	:MORE DH'S?
		BEG	205	:NO-BRANCH
001205		TYPE	.SCLF	
000002		TST	2(R0)	:ONLY A DH IN THIS REQUEST?
		BEG	155	:YES-BRANCH BYPASS INDENT
		OR	R4	:INDENT?
		BNE	155	:NO-BRANCH
061716		TYPE	.SPACE2	:YES-TYPE SPACES
061630		MOV	(R0)+.165	:GET NEXT DH POINTER
		TYPE		:TYPE DH
		WORD	0	:DH POINTER GOES HERE
		STB	(R0)	:TYPE A DT?
		BNE	215	:YES-BRANCH
		ADD	2 R0	:INCREMENT OF POINTER
		BR	145	:SEE IF END OF DF BLOCK
001205		TYPE	.SCLF	
		TST	R4	:INDENT?
		BNE	105	:NO-BRANCH
061716		TYPE	.SPACE2	:YES-TYPE SPACES
		BR	105	:LOOP
010000	117246	RESREG		
		BIT	25W12,25W4	:SEE IF EXIT AFTER 20 ERRORS
		BEG	255	:BR IF NO
		CMP	25RFLG,20	:ELSE SEE IF HAVE 20 ERRORS
		BNE	255	:BR IF NO
		MOV	25STACK,SP	:ELSE RESTORE STACK
		JMP	ENCRV	:ANC BYPASS DRIVE
001103	000024	RTS	PC	

061716 020040 000
061722
062002
000000
000001
000002
000003
000004
000005
000006
000007
177776

000014
000040
000020
000000
000006

177562
177560
177566
177564

062002 000413
062004 000417
062006 013737 177776 061762
062014 013737 000016 177776
062022 010737 061760
062026 000137 063160

062032 012706 061742
062036 010637 061756
062042 000414

SPACE2: .ASCIZ :2 SPACES
: 0DT-11 -- VOOSA
: DEC-11-UODPA-A-LA

: COPYRIGHT 1969, 1970, 1972
: DIGITAL EQUIPMENT CORPORATION
: MAYNARD, MASSACHUSETTS 01754

.ENABL ABS,AMA
.EVEN
=.+6C
: REGISTER NAMING CONVENTIONS

R0 = 20 : REGISTER
R1 = 21 : NAMING
R2 = 22 : CONVENTIONS
R3 = 23
R4 = 24
R5 = 25
R6 = 26
R7 = 27
ST = 177776 ; STATUS REGISTER
O.TVEC = 14 ; TRT VECTOR LOCATION
O.SYM = 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.USR ;FAKE THE SAVED STACK
BR O.RST1 ;CLEAR BREAKPOINT TABLES

```

9139 062244 004037 063366
9140 062250 013777 062000 177716
9141 062256 113704 061764
9142 062262 106004
9143 062264 106004
9144 062266 106004
9145 062270 110437 177776
9146 062274 000127
9147 062276 000403
9148 062100 012737 000002 063070
9149 062106 105037 064007
9150 062112 012737 000340 000016
9151 062120 012737 063150 000014
9152 062126 000447
9153 062130 004537 063610
9154 062134 012704 064033
9155 062140 120024
9156 062142 001413
9157 062144 022704 064041
9158 062150 101373
9159 062152 042700 177770
9160 062156 010004
9161 062160 006304
9162 062162 062704 061742
9163 062166 005202
9164 062170 000444
9165 062172 162704 064024
9166 062176 000770
9167 062200 004737 063734
9168 062204 010502
9169 062206 061202
9170 062210 006202
9171 062212 103421
9172 062214 006302
9173 062216 005722
9174 062220 010205
9175 062222 000137 062474
9176 062226 005702
9177 062230 001406
9178 062232 006204
9179 062234 103410
9180 062236 006304
9181 062240 010437 061774
9182 062244 000412
9183
9184

```

```

O.RST: JSR O.O.SVR :SAVE REGISTERS
MOV O.UIN,RO.ADR1 :REMOVE THE BREAKPOINT
MOV O.PA,R4 :GET ODT PRIORITY
RORB R4 :SHIFT
RORB R4 :INTO
RORB R4 :POSITION
MOV R4,ST :STORE IN STATUS
(PC)+

O.RST1: JMP
BR O.45
MOV RTI,O.RTI :SET TO RTI IF 11/20 OR /05
CLR B O.P :DISALLOW PROCEED
MOV #O.STM,O.TVEC+2 :STATUS WORD TO TRT VECTOR + 2
MOV #O.BRK,O.TVEC :PC TO TRT VECTOR
BR O.RALL :CLEAR BREAKPOINT TABLES

:
: SPECIAL NAME HANDLER
: DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URD

O.REGT: JSR S.O.GET :SPECIAL NAME GET ONE MORE CHARACTER
MOV #O.TL,R4 :TABLE START ADDRESS
O.RSP: CMB R2,R4+ :IS THIS THE CORRECT CHARACTER?
BEQ O.SP :JUMP IF YES
CMP #O.TL+O.LG,R4 :IS THE SEARCH DONE?
BHI O.RSP :BRANCH IF NOT
BIC #177770,R0 :MASK OFF OCTAL
MOV R0,R4

O.SP1: ASL R4
ADD #O.URD,R4 :GENERATE ADDRESS
INC R2 :SET FOUND FLAG
BR O.SCAN :GO FIND NEXT CHARACTER
O.SP: SUB #O.TL-7,R4 :CORRECT CONSTANT
BR O.SP1

:
: * HANDLER - OPEN INDEXED ON THE PC

O.CRPC: JSR PC,O.TCLS
MOV R5,R2 :CURRENT ADDRESS IN R2
ADD #R2,R2 :COMPUTE
ASR R2 :MOVE ONE BIT TO CARRY
BCS O.ERR :ERROR IF ODD NUMBER
ASL R2 :RESTORE WORD
TST (R2)+ :AND INCREMENT BY TWO
MOV R2,R5 :UPDATE CAD
JMP O.OP2 :GO FINISH UP

:
: B HANDLER - SET AND REMOVE BREAKPOINTS

O.BKPT: TST R2 :IF NO NUMBER TYPED
BEQ O.RALL :REMOVE BREAKPOINT
ASR R4 :CHECK IF ODD
BCS O.ERR :JUMP IF ODD
ASL R4 :RESTORE ONE BIT
MOV R4,O.ADR1 :SET A BREAKPOINT
BR O.DCD

```

9149
9150
9151
9152
9153
9154
9155
9156
9157
9158
9159
9160
9161
9162
9163
9164
9165
9166
9167
9168
9169
9170
9171
9172
9173
9174
9175
9176
9177
9178
9179
9180
9181
9182
9183
9184
9185
9186
9187
9188
9189
9190
9191
9192
9193
9194
9195
9196
9197
9198
9199
9200
9201
9202
9203
9204

062246 012737 064050 061774
062254 000406

062256 052705 00000!
062262 012700 000077
062266 004537 063666
062272 004537 063766
062276 005004
062300 005002
062302 004537 063610
062306 022700 000060
062312 101013
062314 022700 000067
062320 103410
062322 042700 177770
062326 006304
062330 006304
062332 006304
062334 060004
062336 005202
062340 000760
062342 005001
062344 120061 064017
062350 001405
062352 005201
062354 020127 000014
062360 103336
062362 000770
062364 006301
062366 000171 062372

062372 062422
062374 062454
062376 062130
062400 062764
062402 062466
062404 062200
062406 062520
062410 062530
062412 062606
062414 062602
062416 062226
062420 063072
000030

O.RALL: MOV #0.TRTC,O.AOR1 ;CLEAR BREAKPOINT
BR O.DCD

: COMMAND DECODER - OCT11
: REGISTERS R0-R4 MAY BE USED
: REGISTER R5 WILL BE CONSIDERED SAFE

O.ERR: BIS #1,R5 ;CLOSE EVERYTHING
MOV #1,R0 ;? TO BE TYPED
JSR O.FTYP ;OUTPUT
JSR O.CPLS ;TYPE '<CR><LF>'
O.DCD: CLR R4 ;R4 CONTAINS THE CONVERTED OCTAL
O.DCD1: CLR R2 ;R2 IS THE NUMBER FOUND FLAG
O.SCAN: JSR O.GET ;GET A CHAR, RETURN IN R0
CMP #0,R0 ;COMPARE WITH ASCII 0
BHI O.CLGL ;CHECK LEGALITY IF NON-NUMERIC
CMP #7,R0 ;COMPARE WITH ASCII 7
BLO O.CLGL ;CHECK LEGALITY IF NOT OCTAL
BIC #177770,R0 ;CONVERT TO BCD
ASL R4 ;MAKE ROOM
ASL R4 ;IN
ASL R4 ;R4
ADD R0,R4 ;PACK THREE BITS IN R4
INC R2 ;R2 HAS NUMERIC FLAG
BF O.SCAN ;AND TRY AGAIN
O.CLGL: CLR R1 ;CLEAR INDEX
O.LGL1: CMPB R0,O.LGCHR(R1) ;DO THE CODES MATCH?
BEQ O.LGL2 ;JUMP IF YES
INC R1 ;SET INDEX FOR NEXT SEARCH
CMP R1,#O.CLGT ;IS THE SEARCH DONE?
BHS O.ERR ;OOPS!
BR O.LGL1 ;RE-LOOP
O.LGL2: ASL R1 ;MULTIPLY BY TWO
JMP #O.LGDR(R1) ;GO TO PROPER ROUTINE

O.LGDR: O.WRD ; / OPEN WORD
O.CRET ; CARRIAGE RETURN CLOSE
O.REGT ; REGISTER OPS
O.GO ; GO TO ADDRESS K
O.OP1 ; (<LF> MODIFY, CLOSE, OPEN NEXT
O.ORPC ; OPEN RELATED, INDEX - PC
O.BACK ; OPEN PREVIOUS
O.OFST ; O OFFSET
O.WSCH ; W SEARCH WORD
O.EFF ; E SEARCH EFFECTIVE ADDRESS
O.BKPT ; B BREAKPOINTS
O.PROC ; P PROCEED

O.LGL = .-O.LGDR ;LGL MUST EQUAL 2X CHLGT ALWAYS
: PROCESS / - OPEN WORD
O.WRD: TST R2 ;GET VALUE IF R2 IS NON-ZERO

```

92305 062424 001410
92306 062426 010405
92307 062430 006205
92308 062432 103711
92309 062434 006305
92310 062436 011500
92311 062440 004537 063524
92312 062444 000714
92313 062446 042705 000001
92314 062452 000766
92315
92316
92317
92318
92319
92320
92321
92322
92323
92324
92325 062466 004737 063734
92326 062472 005725
92327 062474 004537 063760
92328 062500 010500
92329 062502 004537 063524
92330 062506 012700 000057
92331 062512 004537 063666
92332 062516 000744
92333
92334
92335
92336 062520 004737 063734
92337 062524 005745
92338 062526 000762
92339
92340
92341 062530 006205
92342 062532 103737
92343 062534 006305
92344 062536 012700 000040
92345 062542 004537 063666
92346 062546 160504
92347 062550 005304
92348 062552 005304
92349 062554 010400
92350 062556 010402
92351 062560 004537 063524
92352 062564 010200
92353 062566 006200
92354 062570 103402
92355 062572 004537 063524
92356 062576 000137 062276
92357
92358
92359
92360

```

```

      BEQ      0.WRDA      :SKIP OTHERWISE
      MOV      R4,R5      : PUT VALUE IN CAC
0.WRD1: ASR      R5        :MOVE ONE BIT TO CARRY
0.EPR2: BCS      0.ERR    :JUMP IF ODD ADDRESS
      ASL      R5        :RESTORE THE CARRY BIT
      MOV      @R5,R0     :GET CONTENTS OF WORD
      JSR      5,J.CADV   :GO GET AND TYPE OUT 3CAD
      BR      0.DCD1     :GO BACK TO DECODER
0.WRDA: BIC      @1,R5    :CLEAR CLOSED BIT
      BR      0.WRD1     :GO BACK TO MAIN-LINE
:
: PROCESS CARRIAGE RETURN
0.CRET: JSR      PC,0.TCLS :CLOSE LOCATION
      BIS      @1,R5     :CLOSE EVERYTHING
      BR      0.DCD      :RETURN TO DECODER
:
: PROCESS <LF>, OPEN NEXT WORD
0.OP1: JSR      PC,0.TCLS :CLOSE PRESENT CELL
      TST      (R5)+     :GENERATE NEW ADDRESS
0.OP2: JSR      5,0.CRLF  :<CR><LF>
      MOV      R5,R0     :NUMBER TO TYPE
      JSR      5,0.CADV   :TYPE OUT ADDRESS
      MOV      @1,R0     :TYPE A /
      JSR      5,0.FTYP   :GO PROCESS IT
      BR      0.WRD1
:
: PROCESS ↑, OPEN PREVIOUS WORD
0.BACK: JSR      PC,0.TCLS :GENERATE NEW ADDRESS
      TST      -(R5)     :GO DO THE REST
      BR      0.OP2
:
: PROCESS 0, COMPUTE OFFSET
0.OFST: ASR      R5        :GET LOW ORDER BIT
      BCS      0.ERR2    :ERROR IF CLOSED
      ASL      R5        :RESTORE WORD
      MOV      @1,R0     :TYPE ONE BLANK
      JSR      5,0.FTYP   :AS A SEPARATOR
      SUB      R5,R4     :COMPUTE
      DEC      R4        :
      DEC      R4        : 16 BIT OFFSET
      MOV      R4,R0     :TYPE A
      MOV      R4,R2     :SAVE R4
      JSR      5,0.CADV   :NUMBER IN R0 - WORD MODE
      MOV      R2,R0
      ASR      R0        :DIVIDE BY TWO
      BCS      0.OF1     :BRANCH IF ODD
      JSR      5,0.CADV   :NUMBER IN R0 - BYTE MODE
0.OF1: JMP      0.DCD1   :ALL DONE
:
: SEARCHES - $MSK HAS THE MASK
:             $MSK+2 HAS THE FWA
:             $MSK+4 HAS THE LWA

```

9261					
9262					
9263					
9264					
9265					
9266	062602	005201		0.EFF:	INC R1 ;SET EFFECTIVE SEARCH
9267	062604	000401			BR 0.WDS
9268	062606	005201		0.WSCH:	CLR R1 ;SET WORD SEARCH
9269	062610	005702		0.WDS:	TST R2 ;CHECK FOR OBJECT FOUND
9270	062612	001621		0.ERR1:	BEQ 0.ERR ;ERROR IF NO OBJECT
9271	062614	013702	061770		MOV 0.MSK+2,R2 ;SET ORIGIN
9272	062620	013705	061766		MOV 0.MSK,R5 ;SET MASK
9273	062624	005105			COM R5 ;AND COMPLEMENT IT
9274	062626	020237	061772	0.WDS2:	CMP R2,0.MSK+4 ; IS THE SEARCH ALL DONE?
9275	062632	101217			BHI 0.DCD ; YES
9276	062634	011200			MOV 2R2,R0 ; GET OBJECT
9277	062636	005701			TST R1 ;NO
9278	062640	001027			BNE 0.EFF1 ;BRANCH IF EFFECTIVE SEARCH
9279	062642	010046			MOV R0,-(SP)
9280	062644	010403			MOV R4,R3 ;EXCLUSIVE OR
9281	062646	040400			BIC R4,R0 ; IS DONE
9282	062650	042603			BIC (SP)+,R3 ; IN A VERY
9283	062652	050003			BIS R0,R3 ; FANCY MANNER HERE
9284	062654	040503			BIC R5,R3 ;AND RESULT WITH MASK
9285	062656	001016		0.WDS3:	BNE 0.WDS4 ;RE-LOOP IF NO MATCH
9286	062660	010446			MOV R4,-(SP) ;REGISTERS R2,R4, AND R5 ARE SAFE
9287	062662	004537	063760		JSR 5,0.CRLF ;TYPE <CR,LF>
9288	062666	010200			MOV R2,R0 ;GET READY TO TYPE
9289	062670	004537	063524		JSR 5,0.CADV ; TYPE ADDRESS
9290	062674	012700	000057		MOV #1,R0 ;SLASH TO R0
9291	062700	004537	063666		JSR 5,0.FTYP ;TYPE IT
9292	062704	011200			MOV 2R2,R0 ;GET CONTENTS
9293	062706	004537	063524		JSR 5,0.CADV ;TYPE CONTENTS
9294	062712	012604			MOV (SP)+,R4 ;RESTORE R4
9295	062714	005722		0.WDS4:	TST (R2)+ ;INCREMENT TO NEXT CELL AND
9296	062716	000743			BR 0.WDS2 ; RETURN
9297	062720	020004		0.EFF1:	CMP R0,R4 ; IS (X)=K?
9298	062722	001755			BEQ 0.WDS3 ;TYPE IF EQUAL
9299	062724	010003			MOV R0,R3 ;(X) TO R3
9300	062726	060203			ADD R2,R3 ;(X)+X
9301	062730	005203			INC R3 ;(X)+X+2
9302	062732	005203			INC R3 ; IS (X)+X+2=k?
9303	062734	020304			CMP R3,R4 ;BRANCH IF EQUAL
9304	062736	001747			BEQ 0.WDS3 ;WIPE OUT EXTRANEIOUS BITS
9305	062740	042700	177400		BIC #177400,R0 ;EXTEND SIGN
9306	062744	110000			MOV R0,R0
9307	062746	000257			CCC
9308	062750	006300			ASL R0 ;MULTIPLY BY TWO
9309	062752	005200			INC R0 ;ADD TWO
9310	062754	005200			INC R0
9311	062756	060200			ADD R2,R0 ;ADD PC
9312	062760	020004			CMP R0,R4 ;IS THE RESULT A PROPER REL. BRANCH?
9313	062762	000735			BR 0.WDS3
9314					
9315					
9316					

PROCESS G - GO

H14

UNITBUS RK6 DRIVE DIAGNOSTIC PART 3
 0276JC.P11 06-OCT-76 09:44

MAY11 27 10061 06-007-76 09:54 PAGE 176
 TYPE ERROR ROUTINE

SEG 0176

9317	062764	105037	064007	0.G0:	CLRB	0.P	:DISALLOW PROCEED
9318	062770	006204			ASR	R4	:CHECK LOW ORDER BIT
9319	062772	103617			BIS	0.ERR2	:ERROR IF ODD NUMBER
9320	062774	006304			ASL	R4	:RESTORE WORD
9321	062776	010437	06176C		MOV	R4,C,UPC	:SET UP NEW PC
9322	063002	112737	000340	177776	MOVB	#0,STM,ST	:SET HIGH PRIORITY
9323	063010	004537	063456		JSR	5,0,RSTT	:RESTORE TELETYPE
9324	063014	105037	064006		0.TBIT:	0.T	:CLEAR BOTH
9325	063020	042737	000020	061762	BIC	#0,TBT,0,UST	:T-BIT FLAGS
9326	063026	017737	176742	062000	MOV	20,ADR1,0,WIN	:SAVE INSTRUCTION
9327	063034	013777	064050	176732	MOV	0,TRTC,20,ADR1	:REPLACE WITH TRAP
9328	063042	012600			0.G02:	MOV	(SP)+,R0
9329	063044	012601				MOV	(SP)+,R1
9330	063046	012602				MOV	(SP)+,R2
9331	063050	012603				MOV	(SP)+,R3
9332	063052	012604				MOV	(SP)+,R4
9333	063054	012605				MOV	(SP)+,R5
9334	063056	012606				MOV	(SP)+,SP
9335	063060	013746	061762		MOV	0,UST,-(SP)	:AND SP
9336	063064	013746	06176C		MOV	0,UPC,-(SP)	:AND STATUS
9337	063070	000006			0.RTIT:	RTT	:AND PC
9338							:CHANGED TO RTI FOR 11'20 AND /05
9339							
9340							:PROCESS P - PROCEED
9341							:ONLY ALLOWED AFTER A BREAKPOINT
9342	063072	105737	064007	0.PROC:	TSTB	0.P	:CHECK LEGALITY OF PROCEED
9343	063076	001645			BEQ	0.ERR1	:NOT LEGAL
9344	063100	105037	064007		CLRB	0.P	:CLEAR PROCEED FLAG
9345	063104	005702			TST	R2	:WAS COUNT SPECIFIED?
9346	063106	001402			BEQ	0,PR1	:NO
9347	063110	010437	061776		MOV	R4,0,CT	:YES, PUT AWAY COUNT
9348	063114	112737	000340	177776	0.PRI:	MOVB	#0,STM,ST
9349	063122	004537	063456		JSR	5,0,RSTT	:FORCE HIGH PRIORITY
9350	063126	112737	000340	177776	0.C1:	MOVB	#0,STM,ST
9351	063134	105237	064006		MOVB	0,T	:RESTORE TTY
9352	063140	052737	000020	061762	INCB	0,T	:SET HIGH PRIORITY
9353	063146	000735			BIS	#0,TBT,0,UST	:SET T-BIT FLAG
9354					BR	0.G02	:SET T-BIT
9355							
9356							:BREAKPOINT HANDLER
9357							:A TRT BREAKPOINT CAUSES 0.BRK TO BE ENTERED, WHICH SAVES
9358							:VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
9359							:AND GIVES CONTROL TO THE COMMAND DECODER
9360	063150	012637	061760	0.BRK:	MOV	(SP)+,0,UPC	:PRIORITY IS 7 UPON ENTRY
9361	063154	012637	061762		MOV	(SP)+,0,UST	:SAVE STATUS AND PC
9362	063160	004037	063366	0.BK1:	JSR	0,0,SVR	:SAVE VARIOUS REGISTERS
9363	063164	105737	064006		TSTB	0,T	:CHECK FOR T-BIT SET
9364	063170	001311			BNE	0,TBIT	:JUMP IF SET
9365	063172	013777	062000	176574	MOV	0,WIN,20,ADR1	:REMOVE BREAKPOINTS
9366	063200	105737	061764		TSTB	0,PRI	:CHECK IF PRIORITY
9367	063204	100003			BPL	0,BK2	:IS AS SAME AS USER PGM
9368	063206	113705	061762		MOVB	0,UST,R5	:PICK UP USER UST IF SO
9369	063212	000407			BR	0,BK3	:AND DON'T COMPUTE THE PRIORITY
9370	063214	113705	061764	0.BK2:	MOVB	0,PRI,R5	:OTHERWISE PICK UP ACTUAL PRIORITY
9371	063220	000257			CCC		:CLEAR CARRY
9372	063222	106005			RORB	R5	:SHIFT LOW ORDER BITS


```

9373 063224 106005 RORB RS ; INTO
9374 063226 106005 RORB RS ; HIGH ORDER
9375 063230 106005 RORB RS ; POSITION
9376 063232 110537 177776 0.BK3: MOVB RS,ST ; PUT THE STATUS AWAY WHERE IT BELONGS
9377 063236 013705 061760 MOV 0,UPC,RS ; GET PC, IT POINTS TO THE TRT
9378 063242 005745 TST -(RS) ; SUBTRACT TWO
9379 063244 010537 061760 MOV RS,0,UPC ; FROM THE USER'S PC
9380 063250 020537 061774 CMP RS,0,ADR1 ; COMPARE WITH LIST
9381 063254 001417 BEQ 0,B2 ; JUMP IF FOUND
9382 063256 004537 063424 JSR 5,0,SVTT ; SAVE TELETYPE STATUS
9383 063262 004537 063760 JSR 5,0,CRLF ;
9384 063266 012704 064012 MOV #0,BD,R4 ; ERROR, NOTHING FOUND
9385 063272 012703 064013 MOV #0,BD+1,R3 ;
9386 063276 004537 063652 JSR 5,0,TYPE ; OUTPUT "BE" FOR BAD ENTRY
9387 063302 010500 MOV RS,RO ;
9388 063304 042737 000020 061762 BIC #0,TBT,0,UST ; CLEAR OUT ANY POSSIBLE FAKE T-BIT
9389 063312 000420 BR 0,B3 ; AND CONTINUE
9390 063314 005337 061776 0.B2: DEC 0,CT ;
9391 063320 003302 BGT 0,C1 ; JUMP IF REPEAT
9392 063322 012737 000001 061776 MOV #1,0,CT ; RESET COUNT TO 1
9393 063330 105237 064007 INCB 0,P ; ALLOW PROCEED
9394 063334 004537 063424 JSR 5,0,SVTT ; SAVE TELETYPE STATUS, R4 IS SAFE
9395 063340 012700 000102 MOV #1,B,RO ;
9396 063344 004537 063666 JSR 5,0,FTYP ; TYPE "B"
9397 063350 013700 061774 MOV 0,ADR1,RO ; GET ADDRESS OF BREAK
9398 063354 004537 063524 0.B3: JSR 5,0,CADV ; TYPE ADDRESS
9399 063360 005005 CLR RS ; CLEAR CAD
9400 063362 000137 062272 JMP 0,DCD ; GO TO DECODER
9401 ;
9402 ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
9403 ;
9404 063366 012637 064004 0.SVR: MOV (SP)+,0,XXX ; PICK REGISTER FROM STACK AND SAVE
9405 063372 010637 061756 MOV SP,0,USP ; SAVE USER STACK ADDRESS
9406 063376 012706 061756 MOV #0,USP,SP ; SET TO INTERNAL STACK
9407 063402 010546 MOV RS,-(SP) ; SAVE
9408 063404 010446 MOV R4,-(SP) ; REGISTERS
9409 063406 010346 MOV R3,-(SP) ;
9410 063410 010246 MOV R2,-(SP) ;
9411 063412 010146 MOV R1,-(SP) ;
9412 063414 013746 064004 MOV 0,XXX,-(SP) ; PUT SAVED REGISTER ON STACK
9413 063420 005746 TST -(SP) ;
9414 063422 000200 RTS RO ;
9415 ;
9416 ; SAVE TELETYPE STATUS
9417 ;
9418 063424 113737 177560 064010 0.SVTT: MOVB 0,RCSR,0,CSR1 ; SAVE R C/SR
9419 063432 113737 177564 064011 MOVB 0,TCSR,0,CSR2 ; SAVE T C/SR
9420 063440 105037 177560 CLRB 0,RCSR ; CLEAR ENABLE AND MAINTENANCE
9421 063444 105037 177564 CLRB 0,TCSR ; BITS IN BOTH C/SR
9422 063450 004537 063760 JSR 5,0,CRLF ; TYPE <CR,LF>
9423 063454 000205 RTS RS ;
9424 ;
9425 ; RESTORE TELETYPE STATUS
9426 ;
9427 063456 004537 063760 0.RSTT: JSR 5,0,CRLF ; <CR,LF> BEFORE RESTORING
9428 063462 105737 177564 TSTB 0,TCSR ; WAIT READY ON PRINTER

```

```

9173 063466 100375
9174 063470 032737 004000 177560
9175 063476 001403
9176 063500 105737 177560
9177 063504 100375
9178 063506 113737 064010 177560
9179 063514 113737 064011 177560
9180 063522 000205
          BPL      -4
          BIT      #4000,0.RCSR      ;CHECK BUSY FLAG ON READER
          BEQ      0.RSE1            ;SKIP READY LOOP IF NOT BUSY
          TSTB    0.RCSR            ;WAIT READY
          SPL      -4                ;ON READER
          O.RSE1: MOVB   0.CSR1,0.RCSR ;RESTORE
                   MOVB   0.CSR2,0.TCSR ; THE STATUS REGISTERS
                   RTS      R5

;
; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
; WORD IS IN RC
;
0.CADV: MOV      R2,-(SP)            ;SAVE R2
          MOV      #0,BUF+6,R4      ;BUFFER START ADDRESS
          MOV      #'0,-(SP)        ;CONSTANT ASCII 0
0.SPC:  MOV      R0,R2              ; GET
          BIC      #177770,R2       ; OCTAL CHARACTER
          ADD      2,SP,R2           ; CONVERT TO ASCII
          MOVB    R2,-(R4)          ; STORE IN BUFFER
          ASR     R0                 ; SHIFT THIS MESS
          ASR     R0                 ; RIGHT
          ASR     R0                 ; THREE WHOLE PLACES
          CMP     R4,#0,BUF+1        ; DONE?
          BHI     0.SPC              ; NO
          BIC     #177776,R0         ; GET LAST BIT
          ADD     (SP)+,R0           ; CONVERT TO ASCII
          MOVB   R0,-(R4)           ; AND PUT IT AWAY
          MOV     #0,BUF+6,R3       ; LWA
          JSR    5,0,TYPE           ; TYPE WHOLE STRING OF CHARACTERS
          MOV     (SP)+,R2          ; RESTORE R2
          RTS      R5

;
; GENERAL CHARACTER INPUT ROUTINE
; CHARACTER INPUT GOES TO R0
;
0.GET:  TSTB    0.RCSR              ; WAIT FOR
          BPL     -4                 ; INPUT FROM KEYBOARD
          MOVB   0.RDB,R0            ; GET A CHARACTER
          JSR    5,0,FTYP            ; ECHO CHARACTER
          BIC    #177600,R0         ; STRIP OFF PARITY FROM CHARACTER
          BEQ    0.GET              ; IGNORE NULLS
          CMPB  #40,R0              ; CHECK FOR SPACES
          BEQ    0.GET              ; IGNORE NULLS
          CMPB  #'',R0              ; CHECK FOR SEMI-COLON
          BEQ    0.GET              ; IGNORE THEM IF FOUND
          RTS      R5

;
; GENERAL CHARACTER OUTPUT ROUTINE
; ADDRESS OF FIRST BYTE IN R4,
; ADDRESS OF LAST BYTE IN R3, (R3):(R4)
;
0.TYPE: CMP     R3,R4               ; CHECK FOR COMPLETION
          BLO    0,TYPE1            ; EXIT WHEN DONE
          MOVB  (R4)+,R0             ; GET A CHARACTER
          JSR   5,0,FTYP            ; TYPE ONE CHARACTER
          BR    0,TYPE              ; LOOP UNTIL DONE

```

```

9485
9486
9487
9488 063666 105737 177564
9489 063672 100375
9490 063674 110037 177566
9491 063700 120037 000045
9492 063704 001012
9493 063706 113746 000044
9494 063712 105737 177564
9495 063716 100375
9496 063720 105037 177566
9497 063724 105316
9498 063726 003371
9499 063730 005726
9500 063732 000205
9501
9502
9503
9504
9505 063734 006205
9506 063736 103405
9507 063740 006305
9508 063742 005702
9509 063744 001401
9510 063746 010415
9511 063750 000207
9512 063752 005746
9513 063754 000137 062256
9514
9515
9516
9517
9518 063760 012703 064015
9519 063764 000402
9520 063766 012703 064016
9521 063772 012704 064014
9522 063776 004537 063652
9523 064002 000205
9524
9525 064004 000000
9526 064006 000
9527 064007 000
9528
9529 064010 000
9530 064011 000
9531
9532
9533 064012 042502
9534
9535 064014 015
9536 064015 012
9537 064016 052
9538
9539 064017 057
9540 064020 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 R0,0#45 ;IS CHAR TO BE FILLED?
        BNE 0.TYP1 ;NO
        MOVB 0#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
: JPCN 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>*
C.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

```

L14

064021 044
064022 107
064023 012
064024 137
064025 136
064026 117
064027 127
064030 105
064031 102
064032 123
000014
064033 123
064034 123
064035 115
064036 000
064037 000
064040 102
000006
064041
064047 040
064047 040
064050 000003
061742 061742
061742 000000
061744 000000
061746 000000
061750 000000
061752 000000
061754 000000
061756 000000
061760 000000
061762 000000
061764 000007
061766 000000
061770 000000
061772 000000
061774 000000
061776 000000
062000 000000
000001

```

: BYTE :S : :S
: BYTE :C : :C
: BYTE 012 : <LF>
: BYTE : : :
: BYTE : : :
: BYTE :O : :O
: BYTE :M : :M
: BYTE :D : :D
: BYTE :B : :B
O.CLGT = -.LGCH : : : : : : :
:TABLE LENGTH
O.TL: : BYTE :S : :DO : :
: BYTE :P : :NOT : :
: BYTE :M : :CHANGE : :
: BYTE 00 : :THE : :
: BYTE :B : :ORDER : :
: BYTE : : :HERE : :
O.LG = -.TL
O.BUF: = :46 ;6 CHAR. BUFFER WITH
: : : : : : :
:TRTC: TRT ;TRACE TRAP PROTOTYPE
:THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
O.URD: = O.ODT-40
: : : : : : :
: : : : : : :
: : : : : : :
O.USP: 00 : :USER SP
O.UPC: 00 : :USER PC
O.UST: 00 : :USER ST
O.PRI: 7 : :ODT PRIORITY
O.MSK: 0 : :MASK
: : : : : : :
: : : : : : :
: : : : : : :
BREAK POINT LISTS, ADRI = ADDRESS OF BREAKPOINT, CT = COUNT,
: : : : : : :
O.ADR1: 0
O.CT: 0
O.UIN: 0
.END

```

M14

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZR6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 182
CROSS REFERENCE TABLE -- USER SYMBOLS

SEG 019:

ABASE = 177440	1489	1530	1554*
ACDW1 = 000000	1489	1532	
ACDW2 = 000000	1489	1533	
ACLO = 000010	1185*		
ACPUOP = 000000	1489	1504	
ACT11 = 005522	1738*	2684*	
ADDW0 = 000000	1489	1534	
ADDW1 = 000000	1489	1535	
ADDW10 = 000000	1489	1544	
ADDW11 = 000000	1489	1545	
ADDW12 = 000000	1489	1546	
ADDW13 = 000000	1489	1547	
ADDW14 = 000000	1489	1548	
ADDW15 = 000000	1429	1549	
ADDW2 = 000000	1489	1536	
ADDW3 = 000000	1489	1537	
ADDW4 = 000000	1489	1538	
ADDW5 = 000000	1489	1539	
ADDW6 = 000000	1489	1540	
ADDW7 = 000000	1489	1541	
ADDW8 = 000000	1489	1542	
ADDW9 = 000000	1489	1543	
ADEVCT = 000000	1489	1495	
ADEVN = 000000	1489	1531	
AENV = 000000	1489	1500	
AENVN = 000000	1489	1501	
AFATAL = 000000	1489	1492	
AMADR1 = 000000	1489	1517	
AMADR2 = 000000	1489	1521	
AMADR3 = 000000	1489	1524	
AMADR4 = 000000	1489	1527	
AMAMS1 = 000000	1489	1511	
AMAMS2 = 000000	1489	1519	
AMAMS3 = 000000	1489	1522	
AMAMS4 = 000000	1489	1525	
AMSGAO = 000000	1489	1497	
AMSGLG = 000000	1489	1498	
AMSGTY = 000000	1489	1491	
AMTYP1 = 000000	1489	1512	
AMTYP2 = 000000	1489	1520	
AMTYP3 = 000000	1489	1523	
AMTYP4 = 000000	1489	1526	
APASS = 000000	1489	1494	
APRIOR = 000000	1489		
APTCSJ = 000040	6222	6394*	
APTENV = 000001	6169	6215	6350 6392*
APTSIZ = 000200	2612	6391*	
APTSPO = 000100	6217	6352	6393*
ASWREG = 000000	1489	1502	
ATESTN = 000000	1489	1493	
ATTN = 005376	1673*	5087	5107 5134
AUNIT = 000000	1489	1496	
A_SWR = 000000	1489	1503	
AVECT1 = 000000	1489	1528	
AVECT2 = 000000	1489	1529	
BADHDR = 005372	1663*	2692*	5877

Содержание
Содержание

5439*

5425*

Содержание
Содержание

Содержание
Содержание

8352*

Содержание
Содержание

5438*

5424*

Содержание
Содержание

Содержание
Содержание

2456

Содержание
Содержание

5427*

5423*

Содержание
Содержание

Содержание
Содержание

2021

Содержание
Содержание

5436*

5422*

Содержание
Содержание

Содержание
Содержание

1986

Содержание
Содержание

5435*

5425*

4398

4712

Содержание
Содержание

Содержание
Содержание

1923

Содержание
Содержание

5427*

5423

4331

4647

Содержание
Содержание

Содержание
Содержание

1817

Содержание
Содержание

5416*

4915

4555

4555

Содержание
Содержание

Содержание
Содержание

1811

Содержание
Содержание

5415*

4515

4514

4514

Содержание
Содержание

Содержание
Содержание

1805

Содержание
Содержание

5414*

5401

4391

4391

Содержание
Содержание

Содержание
Содержание

1799

Содержание
Содержание

5413*

5315

4326

4326

Содержание
Содержание

Содержание
Содержание

1794

Содержание
Содержание

5412*

5315

4326

4326

Содержание
Содержание

Содержание
Содержание

1791

Содержание
Содержание

3811

3811

3811

3811

Содержание
Содержание

Содержание
Содержание

1771

Содержание
Содержание

3811

3811

3811

3811

Содержание
Содержание

Содержание
Содержание

1751

Содержание
Содержание

3811

3811

3811

3811

Содержание
Содержание

Содержание
Содержание

1751

Содержание
Содержание

3811

3811

3811

3811

Содержание
Содержание

Содержание
Содержание

1751

Содержание
Содержание

3811

3811

3811

3811

Содержание
Содержание

Содержание
Содержание

1751

06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00

2061	2066	2071	84128					
84738	1954	1960	84538					
8885	8885	8938						
84766								
2236	2466	84848		2006	2016	2536	2541	84888
18622	1892	1903		1980	1996	2001	2036	84938
1872	1887	1943						
1938	2441	83608						
2406	85388							
1867	1908	1913	2021	2026	85198			
85278								
2306	85468							
2321	85518							
2491	2496	2501	2506	85558				
2091	2096	2101	83618					
2346	2351	2356	2361	2396	85878			
86068								
86128								
2201	2211	2216	2221	2246	2461	86188		
8914								
2161	2166	86648						

06-007-76 09:54
06-007-76 09:54

MACY11 27.10061 06-007-76 09:54 PAGE 129
CROSS REFERENCE TABLE -- USER SYMBOLS

SEC 0:00

ERRREC= 000004
ESEC 001420
E.A0 005476
E.A1 005502

000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200
000201
000202
000203
000204
000205
000206
000207
000208
000209
000210
000211
000212
000213
000214
000215
000216
000217
000218
000219
000220
000221
000222
000223
000224
000225
000226
000227
000228
000229
000230
000231
000232
000233
000234
000235
000236
000237
000238
000239
000240
000241
000242
000243
000244
000245
000246
000247
000248
000249
000250
000251
000252
000253
000254
000255
000256
000257
000258
000259
000260
000261
000262
000263
000264
000265
000266
000267
000268
000269
000270
000271
000272
000273
000274
000275
000276
000277
000278
000279
000280
000281
000282
000283
000284
000285
000286
000287
000288
000289
000290
000291
000292
000293
000294
000295
000296
000297
000298
000299
000300
000301
000302
000303
000304
000305
000306
000307
000308
000309
000310
000311
000312
000313
000314
000315
000316
000317
000318
000319
000320
000321
000322
000323
000324
000325
000326
000327
000328
000329
000330
000331
000332
000333
000334
000335
000336
000337
000338
000339
000340
000341
000342
000343
000344
000345
000346
000347
000348
000349
000350
000351
000352
000353
000354
000355
000356
000357
000358
000359
000360
000361
000362
000363
000364
000365
000366
000367
000368
000369
000370
000371
000372
000373
000374
000375
000376
000377
000378
000379
000380
000381
000382
000383
000384
000385
000386
000387
000388
000389
000390
000391
000392
000393
000394
000395
000396
000397
000398
000399
000400
000401
000402
000403
000404
000405
000406
000407
000408
000409
000410
000411
000412
000413
000414
000415
000416
000417
000418
000419
000420
000421
000422
000423
000424
000425
000426
000427
000428
000429
000430
000431
000432
000433
000434
000435
000436
000437
000438
000439
000440
000441
000442
000443
000444
000445
000446
000447
000448
000449
000450
000451
000452
000453
000454
000455
000456
000457
000458
000459
000460
000461
000462
000463
000464
000465
000466
000467
000468
000469
000470
000471
000472
000473
000474
000475
000476
000477
000478
000479
000480
000481
000482
000483
000484
000485
000486
000487
000488
000489
000490
000491
000492
000493
000494
000495
000496
000497
000498
000499
000500

80064*
80069*
80073*
80080*
80085*
80094*
80101*
80109*
80116*
80119*
80142*
80151*
80155*
80172*
80179*
80187*
80201*
80205*
80213*
80223*
80226*
80232*
80238*
80245*
80250*
80256*
80262*
80267*
80274*
80278*
7831*
8299*
4201
2597
6095
1594
1717
4412
1719
4368
2598*
6098*
3210*
4518*
3212*
4427
2609*
6101*
3235*
4594*
3237*
4689
2617*
3289*
4674*
3291*
4748
2618*
3669*
4733*
3802*
4754*
2631*
3800*
5175*
3979*
9035
2632*
4057*
5182
4059*
2697*
5184*
4102*
2701*
5213
4152*
2709*
9813
4198*
2746*
8920
4255*
2764*
8929
4414*
2764*
4353*
4520*

2460
8032*
8145*
2481
8162*
2481
8162*

M15

UNIBUS RK6 DRIVE DIAGNOSTIC PART 3
DZRB6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 195
CROSS REFERENCE TABLE -- USER SYMBOLS

SEG 019-

O. RALL	062246	9106	9141	9151#					
O. RCSR =	177560	9070#	9418	9420*	9430	9432	9434*	9464	
O. ROB =	177562	9069#	9466						
O. REGT	062130	9111#	9190						
O. RSE1	063506	9431	9434#						
O. RSP	062140	9113#	9116						
O. RST	062044	9084	9093#						
O. RSTT	063456	9323	9349	9427#					
O. RSTI	062074	9092	9100#						
O. RTIT	063070	9102*	9337#						
O. SCAN	062302	9122	9165#	9176					
O. SP	062172	9114	9123#						
O. SPC	063536	9444#	9452						
O. SPI	062160	9119#	9124						
O. STTH =	003340	9059#	9104	9322	9348	9350			
O. STRT	062032	9083	9090#						
O. SVR	063366	9093	9362	9404#					
O. SVTT	063424	9382	9394	9418#					
O. T	064006	9324*	9351*	9363	9526#				
O. TBT	063014	9324#	9364						
O. TBT =	000020	9060#	9325	9352	9388				
O. TC	063752	9506	9512#						
O. TCLS	063734	9128	9218	9224	9235	9505#			
O. TCSR =	177564	9072#	9419	9421*	9428	9435*	9488	9494	
O. TDB =	177566	9071#	9490*	9496*					
O. TL	064033	9112	9115	9123	9553#	9559			
O. TRTC	064050	9151	9327	9566#					
O. TVEC =	000014	9058#	9086	9104*	9105*				
O. TYPE	063652	9386	9457	9480#	9484	9522			
O. TYP1	063732	9481	9492	9500#					
O. TYP2	063712	9494#	9495	9498					
O. UIN	062000	9094	9326*	9365	9590#				
O. UPC	061760	9087*	9321*	9336	9360*	9377	9379*	9578#	
O. URD	061742	9090	9120	9571#					
O. USP	061756	9091*	9405*	9406	9577#				
O. UST	061762	9085*	9325*	9335	9352*	9361*	9368	9388*	9579#
O. WDS	062610	9267	9269#						
O. WDS2	062626	9274#	9296						
O. WDS3	062656	9285#	9298	9304	9313				
O. WDS4	062714	9285	9295#						
O. WRC	062422	9198	9204#						
O. WRCA	062446	9205	9213#						
O. WRD1	062430	9207#	9214	9231					
O. WSCH	062606	9196	9268#						
O. XXX	064004	9404*	9412	9525#					
O. 45	062106	9101	9103#						
PACK =	000003	1118#	3133	3303	3431	3714	3820	4623	
PARAM	001356	1570#	2547*	2550*	2660				
PARSRT	010050	1294	2547#						
PAT =	000020	1199#	3349						
PCA =	004000	1206#							
PCD =	010000	1207#							
PCLKF	005552	1755#	2703*	2710*	5735	5756			
PCVEC	001352	1563#	2704	2711					
PCYL	001372	1580#							
PFSRT	012434	3069#	6011						

מס' : 2

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

מס' : 1

H16

UNIBUS R16 DRIVE DIAGNOSTIC PART 3
CRASH FILE 06-007-76 09:44

MACY 11 27 10261 06-007-76 09:54 PAGE 203
CROSS REFERENCE TABLE -- USER SYMBOLS

SEG 0202

STIMES	001174	8820	9882											
STK0	001146	1479	2982											
STK0NT	033702	3622	4290											
STK0INT	033712	4146	4790											
STK0EN	033711	4488	5190											
STK0IN	033704	4488	5190											
STK0OU	033706	4488	5190											
STK0SA	033710	4488	5190											
STKS	001144	4488	5190											
STKSRV	033702	4488	5190											
STMP0	001160	4488	5190											
STMP1	001162	4488	5190											
STMP2	001164	4488	5190											
STMP3	001166	4488	5190											
STMP4	001170	4488	5190											
STMP5	001172	4488	5190											
STN	000023	4488	5190											
STAPR	035542	6840	7040											
STAPB	001152	6840	7040											
STAPG	001157	6840	7040											
STAPL	001150	6840	7040											
STAPR	036130	6840	7040											
STAPR2	036132	6840	7040											
STRP	000016	6840	7040											
STRPAD	036164	6840	7040											
STSTM	001004	6840	7040											
STSTMH	001102	6840	7040											
STYIN	035062	6840	7040											
STYBN	*****	6840	7040											
STYPOS	032762	6840	7040											
STYPE	032500	6840	7040											
STYPEC	032712	6840	7040											
STYPEX	032760	6840	7040											
STYPC	033500	6840	7040											
STYON	033514	6840	7040											
STYOS	033454	6840	7040											
SUNIT	001222	6840	7040											
SYSTR	001010	6840	7040											
SYSTR	001224	6840	7040											
SYSTR	001260	6840	7040											
SYSTR	001262	6840	7040											
SYSTR	032032	6840	7040											
SYSTR	000000	6840	7040											
SOFIL	033677	6840	7040											
S4OCAT	*****	6840	7040											
	= 062002	6840	7040											
		1284	1288	1293	1296	1303	1304	1306	1308	1309	1315	1316	1318	1320
		1442	1485	1653	1654	1655	1657	1658	1659	2578	2593	2594	4904	4908

UNIBUS RISK DRIVE DIAGNOSTIC PART 3
CARBON.F11 06-007-76 09:44

MACY11 (27.1006) 06-007-76 09:54 PAGE 204
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0203

5998	6054	6137	6138	6192	6271	6337	6390	6475	6479	6480	6481	6713
6781	6819	6881	9046	9047	9200	9429	9433	9465	9489	9551		

.SAS* = *****
.Sx = 001000

L16

UNIBUS Rk6 DRIVE DIAGNOSTIC PART 3
DZR6JC.P11 06-OCT-76 09:44

MACY11 27(1006) 06-OCT-76 09:54 PAGE 208
CROSS REFERENCE TABLE -- MACRO NAMES

SEG 0206

% ERRORS DETECTED: 0 HARD 2 SOFT
DEFAULT GLOBALS GENERATED: 0

DZR6JC.DZR6JC.SEO/SOL/NL:MD/EG:SDC/CRF NL:TOC/DOC=DZR6JC.P11
RUN-TIME: 76 69 9 SECONDS
RUN-TIME RATIO: 236 156=1.5
CORE USED: 31K (61 PAGES)

DOCUMENT PAGES: 206

