

# RK611/RK06

DRIVE DIAGNOSTIC PART 2  
MD-11-DZR6I-D

EP-DZR6I-D-DL-A

APR 1977

COPYRIGHT © 1977

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of 144 frames of diagnostic data, arranged in 12 rows and 12 columns. Each frame displays a different set of data, likely representing various drive parameters or test results. The data is presented in a structured, tabular format, with some frames containing text labels and others containing numerical or binary values. The overall layout is consistent across the grid, allowing for easy comparison of data across different frames.



B01

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZR6I-D-D
PRODUCT NAME:	UNIBUS RK06 DISK DRIVE DIAGNOSTIC: PART 2
DATE:	JANUARY 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GARY PAPAZIAN

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

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

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

COPYRIGHT (C) 1976, 1977 BY DIGITAL EQUIPMENT CORPORATION

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

TABLE OF CONTENTS

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

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 2 OF THE DRIVE DIAGNOSTICS TO INSURE THAT THE DISK IS CAPABLE OF PERFORMING READ AND WRITE DATA OPERATIONS IN BOTH 20 AND 22 SECTOR FORMATS. WORST CASE PATTERNS, SPIRAL WRITING AND READING, AND ALL OFFSET OPERATIONS ARE PERFORMED. ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING.



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

AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF PART 2, THE DRIVE IS READY FOR PART 3 OF THE DRIVE DIAGNOSTICS.

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

\*\*\*\*\*CAUTION\*\*\*\*\*

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

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

PDP-11  
CONSOLE TELETYPE  
16K MEMORY  
KW11-L OR KW11-P CLOCK  
RK06 UNIBUS CONTROLLER (RK611)  
1 TO 8 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. A 22 SECTOR FORMATTED PACK IS REQ'D, BUT WILL BE A RESULT OF RUNNING DRIVE DIAGNOSTIC PART 1 (SEE BELOW).

2.2 PRELIMINARY TESTING & PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFULLY FOLLOWED BY THE RK06 DRIVE DIAGNOSTIC- PART 1.

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

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

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

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

## 3.2 XXDP

THIS PROGRAM CAN BE CHAINED BY XXDP & WILL NOT OVERLAY THE LOADER.

## CHAIN MODE OPERATION (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

## DUMP MODE OPERATION (MANUAL)

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

## 3.3 ACT/APT

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

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

## AUTOMATIC MODE (MONITOR)

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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



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

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

## 3.3.1 APT ETABLE DEFINITIONS

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

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

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

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

TOTAL TIME: 2 MIN, 40 SEC

A BREAKDOWN OF THE MORE LENGTHY TESTS ARE SHOWN BELOW:

TEST 20	DRIVE OFF TRACK	45 SEC
TEST 21	UNLOAD HEADS	10 SEC
TEST 22	LOAD HEADS	20 SEC
TEST 23	ROTATIONAL TIMING	15 SEC
TEST 24	MAX SEEK TIMES	15 SEC
TEST 26	137 CYL SEEK TIMES	10 SEC
TEST 27	MAX VELOCITY TIMES	15 SEC

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

3.9 FAULT ISOLATION  
TO BE DETERMINED.

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

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

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

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

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

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

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

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

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

POSITION.

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

4.1.4 PRESS 'START'

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

THE PROGRAM BEGINS TESTING ONLY THOSE DRIVES SPECIFIED BY THE INPUT DIALOGUE. IF A SPECIFIED DRIVE CANNOT BE FOUND BY THE PROGRAM IT WILL BE FLAGGED AS AN ERROR THAT THE DRIVE WAS NOT AVAILABLE. THEN BEGINNING WITH THE LOWEST NUMERICAL DRIVE AND PROCEEDING IN SEQUENTIAL ORDER, ALL VALID DRIVES WILL BE TESTED. ONE PASS THROUGH THE TEST SEQUENCE WILL BE PERFORMED ON EACH DRIVE BEFORE MOVING TO THE NEXT DRIVE IN SEQUENCE. THE DRIVE TO BE TESTED WILL BE TYPED AT THE BEGINNING OF EACH PASS. "END OF PASS" WILL BE TYPED AFTER TESTING ALL DRIVES.

4.2 STARTING LOCATIONS

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

NOTE: THE DRIVE PRESENT CONDITION IS:

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

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

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

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

SWITCH	FUNCTION
-----	-----

15 HALT ON ERROR



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

14 LOOP ON TEST  
 13 INHIBIT ERROR TYPEOUT  
 12 BYPASS DRIVE AFTER 20 ERRORS  
 11 INHIBIT ITERATION  
 10 BELL ON ERROR  
 9 LOOP ON ERROR  
 8 LOOP ON TEST IN SW<07:00>

## 4.3.1 SW&lt;15&gt;

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

## 4.3.2 SW&lt;14&gt;

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&lt;13&gt;

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

## 4.3.4 SW&lt;12&gt;

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

## 4.3.5 SW&lt;11&gt;

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&lt;10&gt;

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

## 4.3.7 SW&lt;09&gt;

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.

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

## 4.3.8 SW&lt;08&gt;

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

## 4.4 'SOFTWARE' SWITCH REGISTER

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

SWR = NNNNNN    NEW =

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

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

## 4.5 INPUT DIALOGUE

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

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

## 4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

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

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 2

MAINDEC-11-DZR6I-D-PB

DRIVES TO BE TESTED: 1,3<CR>

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>

WILL TEST DRIVES:

1  
3

DRIVE 1

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

## 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 2  
MAINDEC-11-DZR6I-D-PB

WILL TEST DRIVES:

0  
1

DRIVE 0

DRIVE SERIAL NO. AAA  
CARTRIDGE SERIAL NO. BBB

DRIVE 1

DRIVE SERIAL NO. CCC  
CARTRIDGE SERIAL NO. DDD

END PASS #1

WILL TEST DRIVES:

0  
1

DRIVE 0

DRIVE 1

END PASS # 2

(ETC)

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

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

## 4.7 HALTING THE PROGRAM

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

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

IF HEADS ARE LOADED &amp; FORMATTING IS VALID,

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

THE PROGRAM WILL:

- 1. ECHO IC
- 2. TYPE "CPU HALTED"
- 3. HALT THE PROGRAM

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

- 1. ECHO IC
- 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

A. WRITE TESTS

THESE TESTS CHECK THE ABILITY OF THE DRIVE TO WRITE & READ  
WORSE CASE PATTERNS; PERFORM ALL OFFSETS & PERFORM ALL  
SPIRAL WRITING.

B. SERVO & SPINDLE TIMING TESTS

THESE TESTS CHECK & TYPE HEAD LOAD, UNLOAD & INDEX TIMING,  
ALSO MIN, MAX, AND AVERAGE SEEK TIMES, AND MAX VELOCITY  
OF THE HEADS ARE MEASURED & TYPED.

5.2 TEST DESCRIPTIONS

\*\*\*\*\*  
BASIC CONTROLLER TESTS, SIZING & SETUP  
\*\*\*\*\*



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

## TEST 1 REFERENCE ALL CONTROLLER REGISTERS

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

## TEST 2 SIZE THE BUSS

THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE MANUAL MODE.  
EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.  
CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE DRIVE WILL BE TESTED. IF SET, THE PROGRAM WILL BYPASS TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF MDS, LIFE OR NED BEING SET; OR BOTH NED & DRA RESET INDICATING THE OTHER PORT IS ACCESSED.

## TEST 3 VERIFY OPERATOR DRIVE SELECTIONS

THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT DEFAULTED. EVERY DRIVE FROM 0 TO 7 IS ADDRESSED & CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE PROGRAM WILL ASSUME THE DRIVE IS PRESENT. IT WILL THEN CHECK TO SEE THAT THE DRIVE WAS INPUTTED FOR TESTING. IF NOT, IT WILL BE AN ERROR. IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED ONLY IF THE ERROR WAS A RESULT OF MDS OR LIFE SET OR BOTH NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFOR TO VERIFY IT WAS NOT SPECIFIED.

## TEST 4 FIND NEXT DRIVE TO BE TESTED

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

## TEST 5 PRINT DRIVE SERIAL NUMBER

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

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

TEST 7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.  
THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE  
FACTORY DURING MANF. ALL BAD SECTOR INFO (BSE) WILL BE STORED  
AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.  
IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO  
IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,  
A MESSAGE WILL BE TYPED INDICATING THAT ALL  
FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.  
THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITI

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

\*\*\*\*\*  
WRITE TESTS  
\*\*\*\*\*

TEST 10 BASIC WRITE DATA TEST; 1 WORD

THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,  
ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &  
A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT  
PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP  
FOR A WRITE ERROR.

TEST 11 BASIC WRITE DATA TEST; FULL SECTOR

THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE  
A FULL SECTOR. ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND  
& CHECKED BY A RD DATA COMMAND. A FURTHER CHECK IS PERFORMED  
BY THE WRT CHK COMMAND.  
THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.

TEST 12 20 SECTOR FORMAT TEST

DATA IS WRITTEN ON A FULL TRACK IN 20 SECTOR FORMAT.  
MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.  
CYLINDER, TRACK, SECTOR 0 IS USED.

TEST 13 TEST OFFSET & RTC LOGIC

THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.  
THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE  
'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG

BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A  
DIFFERENT CYLINDER IS ISSUED  
IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT  
CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.

ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED

TEST 14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET POSITIONS UNTIL A FAILURE OCCURES. THE OFFSET POSITIONS ARE TYPED OUT.

OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STATUS 00 & 10. ALL HEADS ARE TESTED AT CYL 0.

IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT

- OR
- A. HEADS DID NOT MOVE AT ALL
  - B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP ARE EXCEPTIONALLY GOOD.

NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR, BUT AN INDICATION OF SURFACE, HEAD, & R/W ELECTRONICS QUALITY ONLY

TEST 15 WRITE WITH HEADS OFFSET

THIS TEST VERIFIES THAT WHEN ATTEMPTING TO WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR & THE DRIVE WILL WRITE SINCE THE WRITE COMMAND HAS AN IMPLIED RTC. THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY

TEST 16 TEST CURRENT CROSS-OVER CYLINDERS

THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:

SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.

A WRITE CHECK IS THEN PERFORMED TO VERIFY DATA WAS PROPERLY WRITTEN. THIS TEST IS PERFORMED FOR ALL 3 HEADS.

CYLINDER X: 63 127 191 255 319 383  
CYLINDER Y: 64 128 192 256 320 384

TEST 17 TEST HEAD SWITCHING TIME

TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.

1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS TO SECTOR 25(8) IS ISSUED.
2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).
3. SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN

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

THE START OF THE WRITE COMMAND (FROM SECTOR 25(8) HEAD 0; TO SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS

THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2

THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT

#### TEST 20 DRIVE OFF TRACK TEST

THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND THE ALLOTTED 3MS.

1. INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS (UNIQUE TO EACH CYLINDER)
2. A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE HEADS WILL IMMEDIATELY BE WRITTEN.
3. IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS, DRIVE OFF TRACK ERROR WILL SET.

IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDER 100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER BEFORE DOING THE NEXT CYLINDER

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

#### 6.0 ERROR REPORTING

#### 6.1 ERROR INTERPRETATION

WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN. MSG A(00) MSG B(01) RKR, 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

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

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLES:

EXAMPLE #1

MESSAGE AD ERROR  
AFTER START SPINDLE CMD & FWD SET

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

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR  
MESSAGE REGISTERS AO, BO, 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.

EXAMPLE #2:

NO ATTN IN RKASOF  
AFTER UNLOAD COMMAND

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

[ END OF DOCUMENT ]

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  
 1001  
 1002  
 1003  
 1004  
 1005  
 1006  
 1007  
 1008  
 1009  
 1010  
 1011  
 1012  
 1013  
 1014  
 1015  
 1016  
 1017  
 1018  
 1019  
 1020  
 1021  
 1022  
 1023  
 1024  
 1025  
 1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036

167400  
 000001

```

;*** PGM REV 028 ***
.NLIST CND,MC,MD
.LIST ME
.ENABL ABS,AMA

;DEFINE SYSMAC MACROS

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

```

.TITLE UNIBUS RK06 DRIVE DIAGNOSTIC PART 2
;COPYRIGHT (C) 1976
;DIGITAL EQUIPMENT CORP.
;MAYNARD, MASS. 01754
*
*PROGRAM BY GARY PAPAIZIAN
*
*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
*
  
```

```

.SBTTL OPERATIONAL SWITCH SETTINGS
*
* SWITCH USE
* -----
* 15 HALT ON ERROR
* 14 LOOP ON TEST
* 13 INHIBIT ERROR TYPEOUTS
* 12 ABORT DRIVE AFTER 20 ERRORS
* 11 INHIBIT ITERATIONS
* 10 BELL ON ERROR
* 9 LOOP ON ERROR
* 8 LOOP ON TEST IN SWR<7:0>
  
```

```

.SBTTL SUMMARY OF STARTING LOCATIONS
*
* 200 DEFAULT PARAMETERS
* 204 DEFAULT PARAMETERS & BYPASS WRITE TESTS
* 214 DEFAULT PARAMETERS & BYPASS TIMING TESTS
* 220 INPUT PARAMETERS
* 224 INPUT PARAMETERS & BYPASS WRITE TESTS
* 230 INPUT PARAMETERS & BYPASS TIMING TESTS
* 240 ODT11
*
  
```

```

1037 .SBTTL BASIC DEFINITIONS
1038
1039
1040 001100
1041
1042
1043
1044
1045 000011
1046 000012
1047 000015
1048 000200
1049 177776
1050
1051 177774
1052 177772
1053 177570
1054 177570
1055
1056
1057 000000
1058 000001
1059 000002
1060 000003
1061 000004
1062 000005
1063 000006
1064 000007
1065 000006
1066 000007
1067
1068
1069 000000
1070 000040
1071 000100
1072 000140
1073 000200
1074 000240
1075 000300
1076 000340
1077
1078
1079 100000
1080 040000
1081 020000
1082 010000
1083 004000
1084 002000
1085 001000
1086 000400
1087 000200
1088 000100
1089 000040
1090 000020
1091 000010
1092 000004

.*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL

.*MISCELLANEOUS DEFINITIONS
HT= 11 ;;CODE FOR HORIZONTAL TAB
LF= 12 ;;CODE FOR LINE FEED
CR= 15 ;;CODE FOR CARRIAGE RETURN
CRLF= 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

```

1093 000002  
1094 000001  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107 100000  
1108 040000  
1109 020000  
1110 010000  
1111 004000  
1112 002000  
1113 001000  
1114 000400  
1115 000200  
1116 000100  
1117 000040  
1118 000020  
1119 000010  
1120 000004  
1121 000002  
1122 000001  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135 000004  
1136 000010  
1137 000014  
1138 000014  
1139 000014  
1140 000020  
1141 000024  
1142 000030  
1143 000034  
1144 000060  
1145 000064  
1146 000240  
1147  
1148

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  
TRIVEC= 14 ; TRACE TRAP  
BPTVEC= 14 ; BREAKPOINT TRAP (BPT)  
IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) \*\*SCOPE\*\*  
PWRVEC= 24 ; POWER FAIL  
EMIVEC= 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

K02

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 22  
RK06 CONTROLLER REGISTER DEFINITION

SEQ 0022

```

1149
1150 ;          $BASE=177440
1151
1152 000000 RKCS1= 0 ; CONTROL AND STATUS REGISTER 1
1153 000002 RKWC= 2 ; WORD COUNT REGISTER
1154 000004 RKBA= 4 ; BUS ADDRESS REGISTER
1155 000006 RKDA= 6 ; DESIRED TRACK SECTOR REGISTER
1156 000010 RKCS2= 10 ; CONTROL AND STATUS REGISTER 2
1157 000012 RKDS= 12 ; DRIVE STATUS REGISTER
1158 000014 RKER= 14 ; ERROR REGISTER
1159 000016 RKASOF= 16 ; ATTENTION SUMMARY AND OFFSET REGISTER
1160 000020 RKDC= 20 ; DESIRED CYLINDER REGISTER
1161 000024 RKDB= 24 ; DATA BUFFER
1162 000026 RKMR1= 26 ; MAINTENANCE REGISTER 1
1163 000034 RKMR2= 34 ; MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1164 000036 RKMR3= 36 ; MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1165 000030 RKECPS= 30 ; ECC POSITION INFORMATION
1166 000032 RKECPT= 32 ; ECC PATTERN INFORMATION
1167
1168 .SBTTL CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1169
1170 ;          DRIVE COMMANDS
1171
1172 000001 SELDRV= 1 ; SELECT DRIVE (GET STATUS)
1173 000003 PACK= 3 ; PACK ACKNOWLEDGE
1174 000005 CLEAR= 5 ; DRIVE CLEAR
1175 000007 UNLOAD= 7 ; UNLOAD
1176 000011 SRTSPL= 11 ; START SPINDLE
1177 000013 RECAL= 13 ; RECALIBRATE
1178 000015 OFFSET= 15 ; OFFSET
1179 000017 SEEK= 17 ; SEEK
1180 000021 RDDATA= 21 ; READ DATA
1181 000023 WRDATA= 23 ; WRITE DATA
1182 000025 RDHEAD= 25 ; READ HEADER
1183 000027 WRHEAD= 27 ; WRITE HEADER AND DATA
1184 000031 WRTCHK= 31 ; WRITE CHECK
1185
1186 000001 GO= BIT0 ; GO BIT
1187 000100 IE= BIT6 ; INTERRUPT ENABLE
1188 000200 RDY= BIT7 ; CONTROLLER READY
1189 000400 BA16= BIT8 ; BUS ADDRESS BIT 16
1190 001000 BA17= BIT9 ; BUS ADDRESS BIT 17
1191 002000 CDT= BIT10 ; CONTROLLER DRIVE TYPE (0=RK06)
1192 004000 CTO= BIT11 ; CONTROLLER TIMEOUT
1193 010000 CFMT= BIT12 ; CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1194 020000 DCPAR= BIT13 ; SERCON PARITY ERROR DETECTED BY CONTROLLER
1195 040000 DI= BIT14 ; DRIVE INTERRUPT
1196 100000 CERR= BIT15 ; CONTROLLER ERROR
1197 100000 CCLR= BIT15 ; CONTROLLER CLEAR
1198
1199 .SBTTL CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)
1200
1201 000007 DRVMSK= 7 ; MASK FOR DRIVE SELECTION CODE
1202 000010 RLS= BIT3 ; DESELECT OR RELEASE DRIVE IN BITS 0-2
1203 000020 BAI= BIT4 ; BUS ADDRESS INCREMENT INHIBIT
1204 000040 SCLR= BIT5 ; SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES

```

1205	000100	IR=	BIT6	: INPUT READY
1206	000200	OR=	BIT7	: OUTPUT READY
1207	000400	UFE=	BIT8	: UNIT FIELD ERROR
1208	001000	MDS=	BIT9	: MULTIPLE DRIVE SELECT
1209	002000	PGE=	BIT10	: PROGRAMMING ERROR
1210	004000	NEM=	BIT11	: NON-EXISTENT MEMORY
1211	010000	NED=	BIT12	: NON-EXISTENT DRIVE
1212	020000	UPE=	BIT13	: UNIBUS PARITY ERROR
1213	040000	WCE=	BIT14	: WRITE CHECK ERROR
1214	100000	DLT=	BIT15	: DATA LATE ERROR
1215				
1216		.SBTTL	ERROR REGISTER BIT DEFINITION (RKER:14)	
1217				
1218	000001	ILF=	BIT0	: ILLEGAL FUNCTION CODE
1219	000002	SKI=	BIT1	: SEEK INCOMPLETE
1220	000004	NXF=	BIT2	: NON-EXECUTABLE FUNCTION
1221	000010	DRPAR=	BIT3	: DRIVE DETECTED SERCON PARITY ERROR
1222	000020	FMTE=	BIT4	: FORMAT ERROR
1223	000040	DTYPE=	BIT5	: DRIVE TYPE ERROR
1224	000100	ECH=	BIT6	: ECC HARD
1225	000200	BSE=	BIT7	: BAD SECTOR ERROR
1226	000400	HVRC=	BIT8	: HEADER VRC ERROR
1227	001000	COE=	BIT9	: CYLINDER ADDRESS OVERFLOW ERROR
1228	002000	IDAE=	BIT10	: INVALID DISK ADDRESS ERROR: HEAD/CYL
1229	004000	WLE=	BIT11	: WRITE LOCK ERROR
1230	010000	DTE=	BIT12	: DRIVE TIMING ERROR
1231	020000	OPI=	BIT13	: OPERATION (SEARCH) INCOMPLETE
1232	040000	UNS=	BIT14	: DRIVE UNSAFE
1233	100000	DCK=	BIT15	: DATA CHECK
1234				
1235		.SBTTL	STATUS REGISTER BIT DEFINITION (RKDS:12)	
1236				
1237	000001	DRA=	BIT0	: DRIVE AVAILABLE (CONTROLLER IS SET IF THIS BIT IS RESET)
1238				
1239	000004	OFST=	BIT2	: DRIVE OFFSET
1240	000010	ACLO=	BIT3	: AC LOW
1241	000020	DCLO=	BIT4	: DC LOW
1242	000040	DROT=	BIT5	: DRIVE OFF TRACK
1243	000100	VV=	BIT6	: VOLUME VALID
1244	000200	DRDY=	BIT7	: DRIVE READY
1245	000400	DDT=	BIT8	: DRIVE TYPE (0=RK06)
1246	004000	WRL=	BIT11	: WRITE LOCK
1247	020000	PIP=	BIT13	: POSITIONING IN PROGRESS
1248	040000	DSC=	BIT14	: DRIVE STATUS CHANGE
1249	100000	SVAL=	BIT15	: STATUS VALID
1250				
1251		.SBTTL	MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)	
1252				
1253	000017	MESMSK=	17	: MESSAGE MASK
1254	000020	PAT=	BIT4	: FORCE EVEN PARITY ON SERCON MESSAGE LINES
1255	000040	DMD=	BIT5	: DIAGNOSTIC MODE
1256	000100	MSP=	BIT6	: MAINTENANCE SECTOR PULSE
1257	000200	MIND=	BIT7	: MAINTENANCE INDEX
1258	000400	MCLK=	BIT9	: MAINTENANCE CLOCK
1259	001000	MERD=	BIT4	: MAINTENANCE ENCODED READ DATA
1260	002000	MEWD=	BIT5	: MAINTENANCE ENCODED WRITE DATA



## M02

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13MACY11 27(1006) 01-FEB-77 04:10 PAGE 24  
MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)

SEQ 0024

1261	004000	PCA= BIT11	:PRECOMPENSATION ADVANCE
1262	010000	PCD= BIT12	:PRECOMPENSATION DELAY
1263	020000	ECCW= BIT13	:ECC WORD IS BEING READ OR WRITTEN
1264	040000	WRTGAT= BIT14	:WRITE GATE
1265	100000	RDGATE= BIT15	:READ GATE
1266			
1267		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1268			
1269	000040	D.DRA= BITS	:DRIVE AVAILABLE
1270	000100	D.VV= BIT6	:VOLUME VALID
1271	000200	D.DRDY= BIT7	:DRIVE READY
1272	000400	D.DOT= BIT8	:DRIVE TYPE (0=RK06)
1273	001000	D.FORM= BIT9	:DRIVE FORMAT
1274	002000	D.OFF= BIT10	:OFFSET ON
1275	004000	D.WRL= BIT11	:WRITE LOCK
1276	010000	D.SPIN= BIT12	:SPINDLE ON
1277	020000	D.PIP= BIT13	:POSITIONING IN PROGRESS
1278	040000	D.DSC= BIT14	:DRIVE STATUS CHANGE
1279			
1280		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1281			
1282	000020	D.SSP= BIT4	:SERVO SIG PRES
1283	000040	D.HOHM= BITS	:HEADS HOME
1284	000100	D.BRHM= BIT6	:BRUSHES HOME
1285	000200	D.DOOR= BIT7	:DOOR INTERLOCKED
1286	000400	D.CART= BIT8	:CARTRIDGE INTERLOCK
1287	001000	D.SPOK= BIT9	:SPEED OK
1288	002000	D.FWD= BIT10	:FORWARD
1289	004000	D.REV= BIT11	:REVERSE
1290	010000	D.LOAD= BIT12	:HEADS LOADING
1291	020000	D.RTZ= BIT13	:RETURN TO ZERO
1292	040000	D.UNLD= BIT14	:HEADS UNLOADING
1293			
1294		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1295			
1296	000040	D.IDAE= BITS	:INVALID DISK ADDRESS ERROR:HEAD/CYL
1297	000100	D.ACLO= BIT6	:AC LOW
1298	000200	D.FLT= BIT7	:DRIVE FAULT
1299	000400	D.ILF= BIT8	:ILLEGAL FUNCTION CODE
1300	001000	D.PAR= BIT9	:DRIVE DETECTED SERCON PARITY ERROR
1301	002000	D.SKI= BIT10	:SEEK INCOMPLETE
1302	004000	D.WLE= BIT11	:WRITE LOCK ERROR
1303	010000	D.SFLS= BIT12	:SPEED LOSS
1304	020000	D.DROT= BIT13	:DRIVE OFF TRACK
1305	040000	D.UNS= BIT14	:R/W UNSAFE
1306			
1307		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1308			
1309	000020	D.SECT= BIT4	:SECTOR ERROR
1310	000040	D.WCUR= BITS	:WRITE CURRENT AND NO WRITE GATE
1311	000100	D.WGAT= BIT6	:WRITE GATE AND NO TRANSITIONS
1312	000200	D.HOFL= BIT7	:HEAD FAULT
1313	000400	D.MHD= BIT8	:MULTIPLE HEAD SELECT
1314	001000	D.XERR= BIT9	:INDEX ERROR
1315	002000	D.TIB= BIT10	:TRIBIT ERROR
1316	004000	D.PLO= BIT11	:PLO ERROR

# N02

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 25  
DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)

SEQ 0025

1317	010000	D.NMOV= BIT12	;SEEK AND NO MOTION
1318	020000	D.LIMD= BIT13	;LIMIT DETECT ON SEEK
1319	040000	D.SUNS= BIT14	;SERVO UNSAFE
1320			
1321		.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)	
1322			
1323	000007	M.DRV= 7	;DRIVE CODE, ALL BYTES
1324	017760	M.CDIF= 17760	;CYLINDER DIFF, BYTE 10
1325	017760	M.OFST= 17760	;OFFSET VALUE, BYTE 10
1326	077770	M.SER= 77770	;DRIVE SERIAL #, BYTE 11
1327			
1328		.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)	
1329			
1330	000003	M.ID= 3	;BYTE ID, ALL BYTES
1331	017760	M.CADD= 17760	;CYLINDER ADDRESS, BYTE 10
1332	040000	M.ALGN= BIT14	;ALIGN SIGN, BYTE 10
1333	000760	M.SECT= 760	;SECTOR COUNT, BYTE 11
1334	007000	M.HEAD= 7000	;HEAD DECODE, BYTE 11
1335	100000	M.PAR= BIT15	;PARITY, MESS A/B, ALL BYTES

```

1336
1337
1338
1339      000000
1340
1341
1342
1343      000174
1344      000174 000000
1345      000176 000000
1346
1347      000200 000137 012542
1348      000220 000220
1349      000220 000137 012532
1350
1351      000240 000240
1352      000240 000137 055422
1353
1354      .SBTTL ACT11 HOOKS
1355
1356      ;*****
1357      ;HOOKS REQUIRED BY ACT11
1358      $SVPC=      ;SAVE PC
1359      =46
1360      SENDAD      ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
1361      =52
1362      .WORD 100000      ;;2)SET LOC.52 TO 100000
1363      =$SVPC      ;; RESTORE PC
1364      =1000
1365      .SBTTL APT PARAMETER BLOCK
1366
1367      ;*****
1368      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1369      ;*****
1370      .SX=      ;SAVE CURRENT LOCATION
1371      =24      ;SET POWER FAIL TO POINT TO START OF PROGRAM
1372      200      ;FOR APT START UP
1373      =44      ;POINT TO APT INDIRECT ADDRESS PNTR.
1374      $APTHDR ;POINT TO APT HEADER BLOCK
1375      =.SX      ;RESET LOCATION COUNTER
1376      ;*****
1377      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1378      ;INTERFACE SPEC.
1379
1380      $APTHD:
1381      $SHIBTS: .WORD 0      ;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1382      $MBAOR: .WORD $MAIL ;ADDRESS OF APT MAILBOX (BITS 0-15)
1383      $STMT: .WORD 280.    ;RUN TIM OF LONGEST TEST
1384      $PASTM: .WORD 600.   ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1385      $SUNITH: .WORD 600.  ;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1386      .WORD $ETEND-$MAIL/2 ;LENGTH MAILBOX-ETABLE(WORDS)
1387
1388
1389      .LIST MD
1390
1391

```

1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447

;USE LOOP X TO OMIT SUBCLR

```

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

```

```

; THIS MACRO FILLS EXPECTED MSG A0, B0, A1, B1, A2, B2 & B3 WITH STANDARD BITS
; A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
; NOTE: A CAN BE ANY BIT COMBINATION DESIRED.

```

```

.MACRO F.EAB A
  MOV #<A!D.SPIN!D.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
.ENDM F.EAB

```

```

; THIS MACRO ASSUMES DRIVE MSG A0, B0, A1, B1 WILL ALWAYS BE TESTED
; USE A,C,D,E FOR MSG A0, B0, A1, B1 ERROR NUMBERS RESP.
; USE G=T.A2 TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
; H=T.B2 TO READ MSG B2 & PUT INFO INTO 'CYLADD'
; I=T.B3 TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEAD'

```

```

; USE F=<ERROR DESCRIPTION>
.MACRO CHECK A,C,D,E,F,G,H,I
  JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
  .WORD G!H!I ;& MSGS SPECIFIED HERE
  ERROR A ;MSG A0 ERROR F
  ERROR C ;MSG B0 ERROR
  ERROR D ;MSG A1 ERROR
  ERROR E ;MSG B1 ERROR
.ENDM CHECK

```

```

; A=CYL DIFF/OFFSET ERROR #
; B=CYL ADDR ERROR #
; C=<ERROR DESCRIPTION>
.MACRO CWD2 A,B,C,'D','E'

```

```

1448          TST      CYLDIF      ;SEE IF MSG A2=0
1449          BEQ      D            ;BR IF YES
1450          ERROR   A            ;MSG A2 NOT CLEARED C
1451 D:         TST      CYLADD      ;SEE IF MSG B2=0
1452          BEQ      E            ;BR IF YES
1453          ERROR   B            ;MSG B2 NOT CLEARED C
1454
1455 E:         .ENDM   CWD2
1456
1457 .MACRO   DRCLR   ?A
1458
1459          MOV      #CCLR,RKCS1(R5)
1460          MOV      $UNIT,RKCS2(R5) ;DRIVE#
1461          MOV      #CLEAR,RKCS1(R5) ;DRIVE CLEAR CMD
1462          MOV      T10,TEMP1      ;SETUP TIMEOUT
1463          JSR      PC,FRDY        ;FIND RDY
1464          ERROR   151            ;NO RDY AFTER DRIVE CLEAR CMD
1465          JSR      PC,TSTATN     ;TEST FOR ATTN
1466          BR      A
1467          ERROR   154            ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1468
1469 A:
1470 .ENDM   DRCLR
1471
1472 ;
1473 ;USE CALIB X TO OMIT CHECKING MSGS A0, B0, A1, B1, A2 & B2
1474 ;
1475 .MACRO   CALIB   A,?C
1476
1477          MOV      #CCLR,RKCS1(R5)
1478          MOV      $UNIT,RKCS2(R5)
1479          MOV      #RECAL,RKCS1(R5) ;RECAL CMD
1480
1481          ;RESET CYL DIFF/OFFSET & CYL ADDR REG
1482          ;IN RKMR2 & RKMR3 RESP.
1483          MOV      T10,TEMP1      ;SETUP TIMEOUT
1484          JSR      PC,FRDY        ;FIND RDY
1485          ERROR   124            ;RDY NOT SET AFTER RECAL CMD
1486
1487          MOV      #1,RKMR1(R5)   ;SELECT WORD 1
1488          JSR      PC,GSTAT
1489          BIT      #D.RTZ,HMR2
1490          BNE     C
1491          ERROR   244            ;RTZ NOT SET DURING RECAL CMD
1492 C:         MOV      T10,TEMP2      ;SETUP TIMEOUT
1493          JSR      PC,FATT1      ;FIND ATTN
1494          ERROR   55            ;NO ATTN AFTER RECAL CMD
1495 .IF B
1496          F.EAB   DSC
1497          CHECK   221,275,222,276,<AFTER RECAL CMD>,T.A2,T.B2,T.B3
1498          CWD2   47,50,<AFTER RECAL CMD>
1499 .ENDC
1500          DRCLR
1501
1502 .ENDM   CALIB
1503

```



```

1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559

```

```

; QUICK START SPINDLE
; MACRO QKSRT
;
; JSR PC, SUBCLR ; CERR AFTER SCLR
; ERROR 24
;
; MOV #SRTSPL, RKCS1(R5) ; START SPINDLE CMD
; MOV T10, TEMP1 ; SET TIMEOUT
; JSR PC, FRDY ; FIND RDY
; ERROR 121 ; RDY NOT SET AFTER ST SPIN CMD.
;
; MOV T100, TEMP2 ; SETUP TIMEOUT
; JSR PC, FATT1 ; FIND ATTN
; ERROR 74 ; NO ATTN AFTER ST SPIN CMD.
;
; CLR UNLD
; ENDM QKSRT
;
; A=WRHEAD/<CFMT!WRHEAD>
; USE WRHDR <A>,X TO OMIT CHECKING A0, B0, A1 & B1
; MACRO WRHDR A,C,K,'D
;
; MOV #<A>, RKCS1(R5) ; WRITE HEADER CMD
; MOV T5000, TEMP1 ; SETUP TIMEOUT
; JSR PC, FRDY ; FIND RDY
; ERROR 200 ; NO RDY AFTER WRITE HEADER CMD
;
; IF B K
; JSR PC, GSTAT ; GET FRESH STATUS
; ENDC
; IF NB K
; MOV #<CFMT!SELDRV>, RKCS1(R5)
; MOV T10, TEMP1
; JSR PC, FRDY
; ERROR 117 ; NO RDY AFTER SELDRV CMD
; ENDC
; BIT #CERR, HCS1
; BEQ D
; ERROR 201 ; CERR AFTER WRITE HEADER CMD
; TYPE MSG26 ; ABORTING BAL OF TESTS
; JMP $EOP
;
; D:
; IF B C
; F.EAB D
; CHECK 277,267,300,270,<AFTER WRITE HEADER CMD>,0,0,0
; ENDC
; ENDM WRHDR
;
; A=RDHEAD/<CFMT!RDHEAD>
; USE RDHDR <A>,X TO OMIT CHECKING A0, B0, A1, B1

```

```

1560 ;
1561 .MACRO RDHDR A,C,'D,'E
1562
1563     MOV     #RHTAB,RO
1564     MOV     #<A>,RKCS1(R5) ; READ HEADER CMD
1565     MOV     T5000,TEMP1 ; SETUP TIMEOUT
1566     JSR     PC,FRDY ; FIND ROY
1567     ERROR   171 ; NO ROY AFTER READ HEADER CMD
1568     BIT     #CERR,HCS1
1569     BEQ     D
1570     ERROR   174 ; CERR AFTER READ HEADER CMD
1571     TYPE    MSG26 ; ABORTING BAL OF TESTS
1572     JMP     $EOP
1573
1574 D:     MOV     RKDB(R5),(RO)+ ; 1'ST WORD FROM SILO TO RHTAB
1575     MOV     RKDB(R5),(RO)+ ; 2'ND WORD
1576     MOV     RKDB(R5),(RO)+ ; 3'RD WORD
1577
1578
1579     BIT     #DLT,RKCS2(R5)
1580     BEQ     E
1581     JSR     PC,GSTAT
1582     ERROR   173 ; DLT AFTER READ HEADER CMD
1583     TYPE    MSG26 ; ABORTING BAL OF TESTS
1584     JMP     $EOP
1585
1586 E:
1587 .IF     B C
1588     F.EAB  D
1589     CHECK  301,271,302,272,<AFTER READ HEADER CMD>,T.A2,T.B2,0
1590
1591 .ENDC
1592
1593 .ENDM  RDHDR
1594
1595 .MACRO HDCHK3 ?A
1596
1597     RDHDR   RDHEAD
1598     CMP     RHTAB,TOCYL ; CHECK WORD 0 ONLY, CYL#
1599     BEQ     A ; BR IF SAME
1600     ERROR   51 ; WRONG CYL# ON HEADER
1601
1602 A:
1603
1604 .ENDM  HDCHK3
1605
1606 ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1607
1608 .MACRO HOTBL A,B,C
1609
1610     MOV     A,CALADD ; SETUP
1611     MOV     #B,HEAD ; TO FILL
1612     MOV     #C,FORMAT ; HEADER
1613     JSR     PC,FHDTAB ; TABLE
1614
1615 .ENDM  HOTBL

```

```

1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671

```

```

; QUICK SEEK. ENTER WITH CYL# IN RKDC
; MACRO QKSEEK ?A
;
; MOV #SEEK,RKCS1(R5) ; SEEK CMD
; MOV T50,TEMP1 ; SETUP TIMEOUT
; JSR PC,FRDY ; FIND RDY
; ERROR 131 ; NO RDY AFTER SEEK CMD
;
; MOV T50000,TEMP1 ; SETUP TIMEOUT
; JSR PC,FATT2 ; FIND ATTN
; ERROR 132 ; NO ATTN AFTER SEEK CMD
;
; BIT #CERR,HCS1
; BEQ A
; ERROR 210 ; CERR AFTER SEEK CMD
;
A:
; ENDM QKSEEK
;
; A=WRDATA/<CFMT!WRDATA>
; C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
; D=ADDR TO JMP TO BYPASS TEST
; E: IF BLANK WILL CHECK A0, B0, A1 & B1 AT THE END OF WRITING
; E: IF NON BLANK WILL OMIT CHECKING A0 THRU B1
;
; MACRO WDATA A,C,D,E,J,K,?F,?G,?H,?I
;
; MOV #<A>,RKCS1(R5) ; WRITE DATA CMD
; MOV T50000,TEMP1 ; SETUP TIMEOUT
; JSR PC,FRDY ; FIND RDY
; ERROR 11 ; NO RDY AFTER WRITE DATA CMD
;
; IF B K
; JSR PC,GSTAT ; GET FRESH STATUS
; ENDC
; IF NB K
; MOV #<CFMT!SELDRV>,RKCS1(R5)
; MOV T10,TEMP1
; JSR PC,FRDY
; ERROR 11? ; NO RDY AFTER SELDRV CMD
; ENDC
; BIT #CERR,HCS1
; BEQ I ; BR IF NO ERRORS
;
; BIT #BSE,HER ; SEE IF BAD SECTOR FLAG
; BEQ G ; BR IF NO
; JSR PC,TRUERR ; ELSE SEE IF SECTOR LISTED IN BSE TABLE
; BR H ; RETURN HERE IF NO
; IF B J
;
; INC SECTOR ; RETURN HERE IF YES

```

```

1672          CMP      SECTOR,#10.      ;ARE 10 CONSEC. SECTORS BAD
1673          BNE     F                    ;BR IF NO
1674          ERROR   46                    ;ABORTING TEST DETECTED 10 BAD SECTORS
1675          JMP     D                    ;BYPASS TEST
1676          F:     MOV     #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
1677          JMP     C
1678          .ENDC
1679          .IF     NB      J              ;RET HERE IF YES
1680          JMP     35
1681          .ENDC
1682          G:     ERROR   12              ;CERR WITH WRITE DATA CMD
1683          F.EAB  0
1684          CHECK  52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1685          TYPE   MSG26                    ;ABORTING BAL OF TESTS
1686          JMP     $EOP
1687          H:     ERROR   63              ;BAD SECTOR NOT LISTED IN TABLE
1688          I:
1689          .IF     B      E
1690          F.EAB  0
1691          CHECK  52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
1692          .ENDC
1693          .ENDM  WDATA
1694
1695          ;
1696          ;A=RDDATA/<CFMT!RDDATA>
1697          ;USE RDATA <A>,X TO OMIT CHECKING A0, B0, A1 & B1
1698          ;
1699          ;
1700          .MACRO  RDATA  A,C,K,?D,?E,?F,?G,?H
1701
1702          MOV     #<A>,RKCS1(R5) ;READ DATA CMD
1703          MOV     T5000,TEMP1 ;SETUP TIMEOUT
1704          JSR    PC,FRDY ;FIND RDY
1705          ERROR  13 ;NO RDY AFTER READ DATA CMD
1706          .IF     B      K
1707          JSR    PC,GSTAT ;GET FRESH STATUS
1708          .ENDC
1709          .IF     NB      K
1710          MOV     #<CFMT!SELDRV>,RKCS1(R5)
1711          MOV     T10,TEMP1
1712          JSR    PC,FRDY
1713          ERROR  11? ;NO RDY AFTER SELDRV CMD
1714          .ENDC
1715          BIT     #CERR,HCS1
1716          BEQ    G
1717          BIT     #BSE,HER ;SEE IF BAD SECTOR
1718          BEQ    E
1719          ERROR  65 ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
1720          BR     H
1721          D:     TYPE   MSG26 ;ABORTING BAL OF TESTS
1722          JMP     $EOP
1723
1724          E:     BIT     #DCK,HER ;SEE IF DATA CHECK ERROR
1725          BEQ    F
1726          ERROR  21 ;DATA CHECK ERROR AFTER READ CMD (ECC)
1727

```

1728  
 1729  
 1730  
 1731  
 1732  
 1733  
 1734  
 1735  
 1736  
 1737  
 1738  
 1739  
 1740  
 1741  
 1742  
 1743  
 1744  
 1745  
 1746  
 1747  
 1748  
 1749  
 1750  
 1751  
 1752  
 1753  
 1754  
 1755  
 1756  
 1757  
 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

```

      BR      H
F:    ERROR  14          ;CERR AFTER READ DATA CMD.
H:    F.EAB  0
      CHECK  54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
      BR      0
G:
  .IF   B      C
      F.EAB  0
      CHECK  54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
  .ENDC
  .ENDM  RDATA

;
;A=WRTCHK/<CFMT!WRTCHK>
;C=EXPECTED DATA FOR TYPEOUT
;USE WRCHK <A>,DATA0,X TO OMIT CHECKING A0, B0, A1 & B1
;
.MACRO WRCHK  A,C,D,K,?E,?F
      MOV    #<A>,RKCS1(R5) ;WRITE CHECK CMD
      MOV    T5000,TEMP1    ;SETUP TIMEOUT
      JSR    PC,FRDY        ;FIND RDY
      ERROR  15            ;NO RDY AFTER WRITE CHECK CMD
  .IF   B      K
      JSR    PC,GSTAT      ;GET FRESH STATUS
  .ENDC
  .IF   NB     K
      MOV    #<CFMT!SELDV>,RKCS1(R5)
      MOV    T10,TEMP1
      JSR    PC,FRDY
      ERROR  11?          ;NO RDY AFTER SELDRV CMD
  .ENDC
      BIT    #CERR,HCS1
      BEQ    F
      BIT    #WCE,HCS2    ;SEE IF WRITE CHECK ERROR
      BEQ    E
      MOV    RKDB(R5),WD1 ;ACTUAL WORD FOR PRINTOUT
      MOV    C,WD2        ;EXPECTED WORD FOR TYPEOUT
      ERROR  16            ;WCE AFTER WRITE CMD
      BR      F
E:    ERROR  22          ;CERR AFTER WRITE CHECK CMD
      F.EAB  0
      CHECK  57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
      TYPE   MSG26        ;ABORTING BAL OF TESTS
      JMP    $EOP
F:
  .IF   B      0
      F.EAB  0
      CHECK  57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2.T.B2,0
  .ENDC
  
```

1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839

```
.ENDM WRCHK

.MACRO OFFSET ?A
    MOV    #A,$ESCAPE
    MOV    #OFFSET,RKCS1(R5) ;OFFSET CMD
    MOV    T100,TEMP1 ;SETUP TIMEOUT
    JSR    PC,FRDY
    ERROR  33 ;NO RDY AFTER OFFSET CMD

    MOV    #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
    CLR    E.B0
    MOV    #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
    MOV    #1,E.B1
    CHECK  35,61,36,62,<DURING OFFSET CMD>,0,0,0

A:    CLR    $ESCAPE
    MOV    T5000,TEMP1 ;SETUP TIMEOUT
    JSR    PC,FATT2 ;FIND ATTN
    ERROR  34 ;NO ATTN AFTER OFFSET CMD

    F.EAB <D.DSC!D.OFF>
    CHECK  260,261,37,40,<AFTER OFFSET CMD>,T.A2,T.B2,0

.ENDM OFFSET

.MACRO EOPGM
    SCOPE
    MOV    #1,$TIMES
    MOV    #STACK,SP
    INC    $DEVCT ;INCR COUNT FOR # DRIVES CHECKED
    CMP    DRIVS,$DEVCT ;ARE ALL DRIVES PRESENT TESTED?
    BEQ    $EOP1+2 ;BR IF YES
    JMP    NUDRV ;ELSE TEST NEXT DRIVE PRESENT
$EOP1: SCOPE
.ENDM EOPGM

;A= ERROR #
;B = ERROR CONDITION

.MACRO OFFDIR A,B ?C ?D
    MOV    R2,RKASOF(R5) ;REFRESH RKASOF

    BIT    #BIT7,R2
    BNE    C ;BR IF NEG OFFSET

    CMP    R2,CYLDIF ;CHECK POS OFFSET
    BEQ    D
    ERROR  A ;OFFSET IN A2 NOT = RKASOF
    BR    D ;B

C:    CMP    R1,CYLDIF ;CHECK NEG OFFSET
    BEQ    D
```

K03

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 35  
APT PARAMETER BLOCK

SEQ 0035

1840  
1841  
1842  
1843  
1844  
1845  
1846

ERROR A ;OFFSET IN A2 NOT = RKASOF  
;B  
D:  
.ENDM OFFDIR  
.NLIST MD

.SBTTL COMMON TAGS

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

.=1100

1847  
1848  
1849  
1850  
1851  
1852  
1853 001100  
1854 001100  
1855 001100 000000  
1856 001102 000  
1857 001103 000  
1858 001104 000000  
1859 001106 000000  
1860 001110 000000  
1861 001112 000000  
1862 001114 000  
1863 001115 001  
1864 001116 000000  
1865 001120 000000  
1866 001122 000000  
1867 001124 000000  
1868 001126 000000  
1869 001130 000000  
1870 001132 000000  
1871 001134 000  
1872 001135 000  
1873 001136 000000  
1874 001140 177570  
1875 001142 177570  
1876 001144 177560  
1877 001146 177562  
1878 001150 177564  
1879 001152 177566  
1880 001154 000  
1881 001155 002  
1882 001156 012  
1883 001157 000  
1884 001160 000000  
1885 001162 000000  
1886 001164 000000  
1887 001166 000000  
1888 001170 000000  
1889 001172 000000  
1890 001174 000000  
1891 001176 000000  
1892 001200 177607 000377  
1893 001204 077  
1894 001205 015  
1895 001206 000012  
1896  
1897  
1898  
1899  
1900  
1901 001210  
1902 001210 000000

SCMTAG: .WORD 0 ;;START OF COMMON TAGS  
\$TSTNM: .BYTE 0 ;;CONTAINS THE TEST NUMBER  
\$ERFLG: .BYTE 0 ;;CONTAINS ERROR FLAG  
\$ICNT: .WORD 0 ;;CONTAINS SUBTEST ITERATION COUNT  
\$LPADR: .WORD 0 ;;CONTAINS SCOPE LOOP ADDRESS  
\$LPERR: .WORD 0 ;;CONTAINS SCOPE RETURN FOR ERRORS  
\$ERTTL: .WORD 0 ;;CONTAINS TOTAL ERRORS DETECTED  
\$ITEMB: .BYTE 0 ;;CONTAINS ITEM CONTROL BYTE  
\$ERMAX: .BYTE 1 ;;CONTAINS MAX. ERRORS PER TEST  
\$ERRPC: .WORD 0 ;;CONTAINS PC OF LAST ERROR INSTRUCTION  
\$GDADR: .WORD 0 ;;CONTAINS ADDRESS OF 'GOOD' DATA  
\$BODADR: .WORD 0 ;;CONTAINS ADDRESS OF 'BAD' DATA  
\$GDADR: .WORD 0 ;;CONTAINS 'GOOD' DATA  
\$BODADR: .WORD 0 ;;CONTAINS 'BAD' DATA  
\$RESERVED: .WORD 0 ;;RESERVED--NOT TO BE USED  
\$AUTOB: .BYTE 0 ;;AUTOMATIC MODE INDICATOR  
\$INTAG: .BYTE 0 ;;INTERRUPT MODE INDICATOR  
\$SWR: .WORD DSWR ;;ADDRESS OF SWITCH REGISTER  
\$DISPLAY: .WORD DDISP ;;ADDRESS OF DISPLAY REGISTER  
\$TKS: 177560 ;;TTY KBD STATUS  
\$TKB: 177562 ;;TTY KBD BUFFER  
\$TPS: 177564 ;;TTY PRINTER STATUS REG. ADDRESS  
\$TPB: 177566 ;;TTY PRINTER BUFFER REG. ADDRESS  
\$NULL: .BYTE 0 ;;CONTAINS NULL CHARACTER FOR FILLS  
\$FILLS: .BYTE 2 ;;CONTAINS # OF FILLER CHARACTERS REQUIRED  
\$FILLC: .BYTE 12 ;;INSERT FILL CHARS. AFTER A "LINE FEED"  
\$TPFLG: .BYTE 0 ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)  
\$TMP0: .WORD 0 ;;USER DEFINED  
\$TMP1: .WORD 0 ;;USER DEFINED  
\$TMP2: .WORD 0 ;;USER DEFINED  
\$TMP3: .WORD 0 ;;USER DEFINED  
\$TMP4: .WORD 0 ;;USER DEFINED  
\$TMP5: .WORD 0 ;;USER DEFINED  
\$TIMES: 0 ;;MAX. NUMBER OF ITERATIONS  
\$ESCAPE: 0 ;;ESCAPE ON ERROR ADDRESS  
\$BELL: .ASCIZ <207><377><377> ;;CODE FOR BELL  
\$QUES: .ASCII /?/ ;;QUESTION MARK  
\$CRLF: .ASCII <15> ;;CARRIAGE RETURN  
\$LF: .ASCIZ <12> ;;LINE FEED

.SBTTL APT MAILBOX-ETABLE

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



M03

UNIBUS RKO6 DRIVE DIAGNOSTIC PART 2  
DZR610.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 37  
APT MAILBOX-ETABLE

SEQ 0037

1903	001212	000000	\$FATAL: .WORD	AFATAL	:: FATAL ERROR NUMBER
1904	001214	000000	\$TESTN: .WORD	ATESTN	:: TEST NUMBER
1905	001216	000000	\$PASS: .WORD	APASS	:: PASS COUNT
1906	001220	000000	\$DEVCT: .WORD	ADEVCT	:: DEVICE COUNT
1907	001222	000000	\$UNIT: .WORD	AUNIT	:: I/O UNIT NUMBER
1908	001224	000000	\$MSGAD: .WORD	AMSGAD	:: MESSAGE ADDRESS
1909	001226	000000	\$MSGLG: .WORD	AMSGLG	:: MESSAGE LENGTH
1910	001230		\$ETABLE:		:: APT ENVIRONMENT TABLE
1911	001230	000	\$ENV: .BYTE	AENV	:: ENVIRONMENT BYTE
1912	001231	000	\$ENVM: .BYTE	AENVM	:: ENVIRONMENT MODE BITS
1913	001232	000000	\$SWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
1914	001234	000000	\$USWR: .WORD	AUSWR	:: USER SWITCHES
1915	001236	000000	\$CPUOP: .WORD	ACPUOP	:: CPU TYPE, OPTIONS
1916			*		BITS 15-11=CPU TYPE
1917			*		11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1918			*		11/70=06, PDQ=07, Q=10
1919			*		BIT 10=REAL TIME CLOCK
1920			*		BIT 9=FLOATING POINT PROCESSOR
1921			*		BIT 8=MEMORY MANAGEMENT
1922	001240	000	\$MAMS1: .BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
1923	001241	000	\$MTYP1: .BYTE	AMTYP1	:: MEM. TYPE, BLK#1
1924			*		MEM. TYPE BYTE -- (HIGH BYTE)
1925			*		900 NSEC CORE=001
1926			*		300 NSEC BIPOLAR=002
1927			*		500 NSEC MOS=003
1928	001242	000000	\$MADR1: .WORD	AMADR1	:: HIGH ADDRESS, BLK#1
1929			*		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1930	001244	000	\$MAMS2: .BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
1931	001245	000	\$MTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
1932	001246	000000	\$MADR2: .WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
1933	001250	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
1934	001251	000	\$MTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
1935	001252	000000	\$MADR3: .WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
1936	001254	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
1937	001255	000	\$MTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
1938	001256	000000	\$MADR4: .WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
1939	001260	000000	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
1940	001262	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
1941	001264	177440	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
1942	001266	000000	\$DEVN: .WORD	ADEVN	:: DEVICE MAP
1943	001270	000000	\$CDW1: .WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
1944	001272	000000	\$CDW2: .WORD	ACDW2	:: CONTROLLER DESCRIPTION WORD#2
1945	001274	000000	\$DDW0: .WORD	ADDW0	:: DEVICE DESCRIPTOR WORD#0
1946	001276	000000	\$DDW1: .WORD	ADDW1	:: DEVICE DESCRIPTOR WORD#1
1947	001300	000000	\$DDW2: .WORD	ADDW2	:: DEVICE DESCRIPTOR WORD#2
1948	001302	000000	\$DDW3: .WORD	ADDW3	:: DEVICE DESCRIPTOR WORD#3
1949	001304	000000	\$DDW4: .WORD	ADDW4	:: DEVICE DESCRIPTOR WORD#4
1950	001306	000000	\$DDW5: .WORD	ADDW5	:: DEVICE DESCRIPTOR WORD#5
1951	001310	000000	\$DDW6: .WORD	ADDW6	:: DEVICE DESCRIPTOR WORD#6
1952	001312	000000	\$DDW7: .WORD	ADDW7	:: DEVICE DESCRIPTOR WORD#7
1953	001314		\$ETEND:		
1954			.MEXIT		
1955		177440	ABASE=	177440	:: DEFAULT BUSS ADDRESS
1956	001314	000210	RKVEC:	210	:: DEFAULT CONTROLLER INTERRUPT VECTOR
1957	001316	000240	RKPRI:	PRS	:: PRIORITY
1958	001320	172540	PKS:	172540	:: P-CLOCK STATUS REG

1959	001322	172542	PKSB:	172542	;P-CLOCK SET BUFFER
1960	001324	172544	PKRB:	172544	;P-CLOCK READ BUFFER
1961	001326	177546	LKS:	177546	;L-CLOCK STATUS REG.
1962					
1963	001330	000100	LCVEC:	100	;L-CLOCK INTERRUPT VECTOR
1964	001332	000104	PCVEC:	104	;P-CLOCK INTERRUPT VECTOR.
1965					
1966		000114	MEMVEC=	114	;MEMORY PARITY VECTOR
1967		172100	MEMBAS=	172100	;MEMORY PARITY OPTION CSR START ADDR
1968	001334	000000	TRAPPC:	0	;PC FOR MEM CHECK ENABLE TRAP
1969					
1970	001336	000000	PARAM:	0	;1 FOR 220 START, NO DEFAULT
1971	001340	000000	FTITLE:	0	;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
1972					
1973	001342	000000	DRVPTR:	0	;CONTAINS THE POINTER TO THE DRIVE FLAG
1974					; (DRIVO-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
1975	001344	000000	FRCYL:	0	;FROM CYLINDER
1976	001346	000000	TOCYL:	0	;TO CYLINDER
1977	001350	000000	CCYL:	0	;CURRENT CYL, USED IN N SQUARE TEST
1978	001352	000000	PCYL:	0	;PREV CYL, USED IN N SQUARE TEST
1979	001354	000000	CALDIF:	0	;CALC CYL DIFF USED IN N SQUARE TEST
1980	001356	000000	CYLDIF:	0	;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
1981	001360	000000	CYLOAD:	0	;CYL ADDR, RIGHT JUSTIFIED FROM RKMR3
1982	001362	000000	CALADD:	0	;CYL ADDR USED IN FHD7AB ROUTINE
1983					
1984	001364	000074	HZ:	60.	;60 FOR 60 CPS
1985					;50 FOR 50 CPS
1986	001366	000000	COUNT:	0	;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
1987					;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
1988	001370	000000	SEC:	0	;SECOND COUNTER
1989	001372	000000	TIMUP:	0	;FLAG TO INDICATE TIME IS UP
1990	001374	000000	SECNT:	0	;SECTOR COUNT
1991	001376	000000	PSEC:	0	;PREVIOUS SECTOR
1992	001400	000000	ESEC:	0	;EXPECTED SECTOR
1993	001402	000000	SECTOR:	0	;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
1994					
1995	001404	000001	T1:	1	;TIMEOUT CONSTANTS
1996	001406	000012	T10:	10.	
1997	001410	000062	T50:	50.	
1998	001412	000764	T500:	500.	
1999	001414	000144	T100:	100.	
2000	001416	011610	T5000:	5000.	
2001	001420	141520	T50000:	50000.	
2002					
2003	001422	000077	CYL:	63.	;CYLINDER NUMBERS USED IN
2004	001424	000177		127.	;CURRENT CROSSOVER TEST
2005	001426	000277		191.	
2006	001430	000377		255.	
2007	001432	000477		319.	
2008	001434	000577		383.	
2009					
2010	001436	000000	WD1:	0	;ACTUAL HEADER/DATA WORD
2011	001440	000000	WD2:	0	;EXPECTED DATA WORD
2012					
2013	001442	000000	OFFERR:	0	;SET WHEN WRITE CHECK ERROR ON OFFSET
2014					

2015											
2016	001444	000000			HEAD:	0					: HEAD NUMBER
2017	001446	000000			HEADP:	0					: HEAD # FROM H.B3, RT. JUSTIFIED
2018	001450	000000			HDI:	0					: SHIFTED HEAD# FOR FORMATTER ROUTINE
2019	001452	000000			FORMAT:	0					: FORMAT TYPE
2020	001454	000000			FMT1:	0					: SHIFTED FORMAT FOR FORMATTER ROUTINE
2021	001456	000000			WDCNT:	0					: WORD COUNT
2022											
2023	001460	000000			DATA0:	0					: ALL 0'S
2024	001462	052525			DATA01:	52525					: 0101 PATT
2025	001464	177777			DATA1:	177777					: ALL 1'S
2026	001466	133467			DPAT1:	133467					
2027	001470	070627			DPAT2:	70627					
2028											
2029	001472	000000			WORD:	0					: HEADER/DATA WORD
2030	001474	000000			HOWD:	0					: HEADER WORD FROM RKDB
2031											
2032	001476	000000			BSERR:	0					: CANNOT READ BSE INFO WHEN SET
2033	001500	000000			LIMERR:	0					: LIMIT DETECT ERROR FLAG
2034	001502	000000			BYPCERR:	0					: SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
2035	001504	000000			CHKFLG:	0					: WORDS TO BE TESTED
2036											
2037	001506	000102			HDTAB:	.BLKW 66.					: CALCULATED HEADER WORD TABLE
2038	001712	000102			RHTAB:	.BLKW 66.					: FILLED AFTER READ HEADER CMD
2039	002116	000102			SRTTAB:	.BLKW 66.					: ABOVE RHTAB SORTED STARTING FORM
2040											: SECTOR 0 BY SORT ROUTINE
2041	002322	000400			BSE20H:	.BLKW 256.					: 20 SECTOR HARDWARE BSE INFO
2042	003322	000400			BSE22H:	.BLKW 256.					: 22 SECTOR HARDWARE BSE INFO.
2043	004322	000400			BSE20S:	.BLKW 256.					: 20 SECTOR SOFTWARE BSE INFO.
2044	005322	000400			BSE22S:	.BLKW 256.					: 22 SECTOR SOFTWARE BSE INFO.
2045	006322	000400			ROTAB:	.BLKW 256.					: FILLED AFTER READ DATA CMD
2046											
2047	007322	000000			UNLD:	0					: SET TO 0 IF HEADS ARE LOADED
2048											: SET TO 1 IF HEADS UNLOADED
2049	007324	000000			BADHDR:	0					: SET TO 0 IF FORMATTING OK
2050											: SET TO 1 IF FORMATTING ALTERED
2051	007326	000000			HPEND:	0					: SET TO 0 IF HALT NOT PENDING
2052											: SET TO 1 IF HALT PENDING
2053											
2054											: THE ABOVE 3 FLAGS ARE USED
2055											: BY 'STOP' ROUTINE TO BRING
2056											: THE CPU TO A VALID HALT.
2057											
2058											
2059	007330	001	002	004	ATTN:	.BYTE 1,2,4,10,20,40,100,200					: ;ATN 0-7 RESP.
2060	007333	010	020	040							
2061	007336	100	200								
2062						.EVEN					
2063											
2064											
2065											: THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
2066											: THEY ARE LOADED AFTER RDY IS REC'D FROM WRDY ROUTINE.
2067											
2068											
2069	007340	000000			HCS1:	0					: HOLD RKCS1
2070	007342	000000			HCS2:	0					: HOLD RKCS2

2071 007344 000000  
 2072 007346 000000  
 2073 007350 000000  
 2074 007352 000000  
 2075 007354 000000  
 2076 007356 000000  
 2077 007360 000000  
 2078 007362 000000  
 2079 007364 000000  
 2080 007366 000000  
 2081 007370 000000  
 2082 007372 000000  
 2083 007374 000000

HWC: 0 ;HOLD RKWC  
 HBA: 0 ;ETC.  
 HDA: 0  
 HDS: 0  
 HER: 0  
 HASOF: 0  
 HDC: 0  
 HDB: 0  
 HMR1: 0  
 HMR2: 0  
 HMR3: 0  
 HPOS: 0  
 HPAT: 0

2086 007376 000000  
 2087 007400 000000  
 2088 007402 000000  
 2089 007404 000000  
 2090 007406 000000

TEMP1: 0 ;TEMPORARY STORAGE.  
 TEMP2: 0  
 TEMP3: 0  
 TEMP4: 0  
 TEMPS: 0

2094 007410 000000  
 2095 007412 000000  
 2096 007414 000000  
 2097 007416 000000  
 2098 007420 000000  
 2099 007422 000000  
 2100 007424 000000  
 2101 007426 000000

...  
 ;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).  
 ...  
 H.A0: 0  
 H.B0: 0  
 H.A1: 0  
 H.B1: 0  
 H.A2: 0  
 H.B2: 0  
 H.A3: 0  
 H.B3: 0

2105 007430 000000  
 2106 007432 000000  
 2107 007434 000000  
 2108 007436 000000  
 2109 007440 000000  
 2110 007442 000000  
 2111 007444 000000  
 2112 007446 000000

...  
 ;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.  
 ...  
 E.A0: 0  
 E.B0: 0  
 E.A1: 0  
 E.B1: 0  
 E.A2: 0  
 E.B2: 0  
 E.A3: 0  
 E.B3: 0

2116 000001  
 2117 000002  
 2118 000004

...  
 ;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.  
 ...  
 †.A2=BIT0 ;TEST MSG A2 IF SET  
 T.B2=BIT1  
 T.B3=BIT2

2124 007450 000000  
 2125 007452 000000  
 2126 007454 000000

...  
 ;ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.  
 ...  
 DDUMP: 0 ;FLAG - SET WHEN IN DDP DUMP MODE  
 DDPCH: 0 ;FLAG - SET WHEN IN DDP CHAIN MODE  
 ACT11: 0 ;FLAG - SET WHEN IN ACT11 MODE OF OPERATION

2127	007456	000000	PPTP: 0	; FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
2128	007460	000000	DRIVS: 0	; CONTAINS THE NUMBER OF DRIVES PRESENT
2129				
2130			; THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE	
2131			; IS PRESENT AND IS TO BE TESTED.	
2132				
2133	007462	000000	DRIV0: 0	; FLAG SET TO 1 WHEN DRIVE 0 PRESENT
2134	007464	000000	DRIV1: 0	; FOR DRIVE 1
2135	007466	000000	DRIV2: 0	; FOR DRIVE 2
2136	007470	000000	DRIV3: 0	; FOR DRIVE 3
2137	007472	000000	DRIV4: 0	; FOR DRIVE 4
2138	007474	000000	DRIV5: 0	; FOR DRIVE 5
2139	007476	000000	DRIV6: 0	; FOR DRIVE 6
2140	007500	000000	DRIV7: 0	; FOR DRIVE 7
2141				
2142	007502	000000	LCLKF: 0	; L-CLOCK FLAG PRESENT FLAG
2143	007504	000000	PCLKF: 0	; P-CLOCK FLAG PRESENT FLAG
2144	007506	000000	DOTIM: 0	; SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
2145	007510	000000	SIZFLG: 0	; SET IF DEFAULT DO SIZING IN TEST 1

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:

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

;ERROR 2 ;DETECTED MDS  
EM5  
DH1  
DT1  
DF1

;ERROR 3 ;DETECTED LIFE  
EM6  
DH1  
DT1  
DF1

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

;ERROR 5 ;DR PRESENT BUT NOT SPECIFIED BY OPERATOR  
EM8  
DH1  
DT1  
DF1

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

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

2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160 007512  
2161  
2162  
2163 007512 046136  
2164 007514 051572  
2165 007516 053654  
2166 007520 054456  
2167  
2168  
2169 007522 046355  
2170 007524 051572  
2171 007526 053654  
2172 007530 054456  
2173  
2174  
2175 007532 046376  
2176 007534 051572  
2177 007536 053654  
2178 007540 054456  
2179  
2180  
2181 007542 046417  
2182 007544 051572  
2183 007546 053654  
2184 007550 054456  
2185  
2186 007552 046506  
2187 007554 051572  
2188 007556 053654  
2189 007560 054456  
2190  
2191  
2192 007562 046562  
2193 007564 051572  
2194 007566 053654  
2195 007570 054456  
2196  
2197  
2198 007572 046636  
2199 007574 051572  
2200 007576 053654  
2201 007600 054456

2202				
2203			:ERROR 10	
2204	007602	046721	EM11	;DRA & NED BOTH SET
2205	007604	051572	DH1	
2206	007606	053654	DT1	
2207	007610	054456	DF1	
2208			:ERR 11	
2209	007612	046765	EM12	;NO RDY
2210	007614	052555	DH27	;AFTER WRITE DATA CMD
2211	007616	053654	DT1	
2212	007620	054606	DF10	
2213			:ERR 12	
2214	007622	047361	EM21	;CERR SET
2215	007624	052555	DH27	
2216	007626	053654	DT1	
2217	007630	054606	DF10	
2218			:ERR 13	
2219	007632	046765	EM12	;NO RDY
2220	007634	052525	DH26	;AFTER READ DATA CMD
2221	007636	053654	DT1	
2222	007640	054606	DF10	
2223			:ERR 14	
2224	007642	047361	EM21	;CERR SET
2225	007644	052525	DH26	
2226	007646	053654	DT1	
2227	007650	054606	DF10	
2228			:ERR 15	
2229	007652	046765	EM12	;NO RDY
2230	007654	052705	DH32	;AFTER WRITE CHECK CMD
2231	007656	053654	DT1	
2232	007660	054606	DF10	
2233			:ERR 16	
2234	007662	050547	EM80	;WRITE CHECK ERROR SET
2235	007664	052705	DH32	;AFTER WRITE CHECK CMD
2236	007666	053766	DT6	
2237	007670	054476	DF3	
2238			:ERR 17	
2239	007672	050606	EM81	;WRITE CHECK CMD NOT FUNCTIONING
2240	007674	053434	DH52	;WITH INTENTIONAL MISCOMPARE
2241	007676	053654	DT1	
2242	007700	054606	DF10	
2243			:ERR 20	
2244	007702	050652	EM82	;READ DATA NOT COMPARE WITH WRITE DATA
2245	007704	052525	DH26	;AFTER READ DATA CMD
2246	007706	053766	DT6	
2247	007710	054476	DF3	
2248			:ERR 21	
2249	007712	050724	EM83	;DATA CHECK ERROR
2250	007714	052525	DH26	
2251	007716	053654	DT1	
2252	007720	054606	DF10	
2253			:ERR 22	
2254	007722	047361	EM21	;CERR SET
2255	007724	052705	DH32	;AFTER WRITE CHECK CMD
2256	007726	053654	DT1	
2257	007730	054606	DF10	

2258			;ERR 23		
2259	007732	047276		EM18	;MSG B0 ERROR
2260	007734	052555		DH27	;AFTER WRITE DATA CMD
2261	007736	054212		DT13	
2262	007740	054736		DF21	
2263			;ERROR 24		
2264	007742	047361		EM21	;CERR SET
2265	007744	052375		DH21	;AFTER SCLR
2266	007746	053654		DT1	
2267	007750	054606		DF10	
2268			;ERR 25		
2269	007752	047340		EM20	;MSG B1 ERROR
2270	007754	052555		DH27	
2271	007756	054212		DT13	
2272	007760	054736		DF21	
2273			;ERR 26		
2274	007762	047276		EM18	
2275	007764	052525		DH26	;AFTER READ DATA CMD
2276	007766	054212		DT13	
2277	007770	054736		DF21	
2278			;ERROR 27		
2279				EM24	;VOL VALID NOT SET
2280	007772	047610		DH19	;AFTER PACK CMD
2281	007774	052317		DT1	
2282	007776	053654		DF10	
2283	010000	054606			
2284			;ERR 30		
2285	010002	047340		EM20	;MSG B1 ERROR
2286	010004	052525		DH26	;AFTER READ DATA CMD.
2287	010006	054212		DT13	
2288	010010	054736		DF21	
2289			;ERR 31		
2290	010012	047276		EM18	;MSG B0 ERROR
2291	010014	052705		DH32	;AFTER WRITE CHECK CMD
2292	010016	054212		DT13	
2293	010020	054736		DF21	
2294			;ERR 32		
2295	010022	047340		EM20	;MSG B1 ERROR
2296	010024	052705		DH32	
2297	010026	054212		DT13	
2298	010030	054736		DF21	
2299			;ERR 33		
2300	010032	046765		EM12	;CONTR NOT READY
2301	010034	052455		DH24	;AFTER OFFSET CMD
2302	010036	053654		DT1	
2303	010040	054606		DF10	
2304			;ERR 34		
2305	010042	047023		EM13	;NO ATTN
2306	010044	052455		DH24	
2307	010046	053654		DT1	
2308	010050	054606		DF10	
2309			;ERR 35		
2310	010052	047255		EM17	;MSG A0 ERROR
2311	010054	053470		DH53	;DURING OFFSET COMMAND
2312	010056	054212		DT13	
2313	010060	054736		DF21	



2314			;ERR 36		
2315	010062	047317		EM19	;MSG A1 ERROR
2316	010064	053470		DH53	
2317	010066	054212		DT13	
2318	010070	054736		DF21	
2319			;ERR 37		
2320	010072	047317		EM19	;MSG A1 ERROR
2321	010074	052455		DH24	;AFTER OFFSET CMD
2322	010076	054212		DT13	
2323	010100	054736		DF21	
2324			;ERR 40		
2325	010102	047340		EM20	;MSG B1 ERROR
2326	010104	052455		DH24	
2327	010106	054212		DT13	
2328	010110	054736		DF21	
2329			;ERR 41		
2330	010112	047045		EM14	;UNEXP MEM PCRTY TRAP
2331	010114	051763		DH8	;TEST #, TRAP PC
2332	010116	053714		DT3	
2333	010120	054472		DF2	
2334			;ERR 42		
2335	010122	050263		EM41	;CYL ADDR IN B2 DID NOT REMAIN CLEARED
2336	010124	052455		DH24	
2337	010126	054272		DT14	
2338	010130	054772		DF22	
2339			;ERR 43		
2340	010132	051052		EM85	
2341	010134	052423		DH22	
2342	010136	053654		DT1	
2343	010140	054606		DF10	
2344			;ERR 44		
2345	010142	047103		EM15	;WCE AT CYL 411, TRK 2, SEC 21
2346	010144	051572		DH1	
2347	010146	053654		DT1	
2348	010150	054522		DF4	
2349			;ERR 45		
2350	010152	051052		EM85	;OFFSET BIT IN RKMR2 CLEARED
2351	010154	053401		DH51	;AFTER SEEK TO SELF
2352	010156	053654		DT1	
2353	010160	054606		DF10	
2354			;ERR 46		
2355	010162	047643		EM25	;DETECTED 10 BAD SECTORS
2356	010164	052555		DH27	;AFTER WRITE DATA CMD.
2357	010166	053654		DT1	
2358	010170	054606		DF10	
2359			;ERROR 47		
2360	010172	050156		EM39	;CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2361	010174	052273		DH17	;AFTER RECAL CMD
2362	010176	054272		DT14	
2363	010200	054772		DF22	
2364			;ERROR 50		
2365	010202	050225		EM40	;CYL ADDR IN RKMR3 NOT CLEARED
2366	010204	052273		DH17	;AFTER RECAL CMD
2367	010206	054272		DT14	
2368	010210	054772		DF22	
2369			;ERR 51		

2370	010212	051214	EM93	;WRONG CYL# IN HEADER WORD (MISPOSITION)
2371	010214	052502	DH25	;AFTER SEEK CMD
2372	010216	054146	DT9	
2373	010220	054712	DF20	
2374			;ERR 52	
2375	010222	047255	EM17	;MSG A0 ERROR
2376	010224	052555	DH27	;AFTER WRITE DATA CMD
2377	010226	054212	DT13	
2378	010230	054736	DF21	
2379			;ERR 53	
2380	010232	047317	EM19	;MSG A1 ERROR
2381	010234	052555	DH27	
2382	010236	054212	DT13	
2383	010240	054736	DF21	
2384			;ERR 54	
2385	010242	047255	EM17	;MSG A0 ERROR
2386	010244	052525	DH26	;AFTER READ DATA CMD
2387	010246	054212	DT13	
2388	010250	054736	DF21	
2389			;ERROR 55	
2390	010252	047023	EM13	;NO ATTN
2391	010254	052273	DH17	;AFTER RECAL CMD
2392	010256	053654	DT1	
2393	010260	054606	DF10	
2394			;ERR 56	
2395	010262	047317	EM19	;MSG A1 ERROR
2396	010264	052525	DH26	
2397	010266	054212	DT13	
2398	010270	054736	DF21	
2399			;ERR 57	
2400	010272	047255	EM17	;MSG A0 ERROR
2401	010274	052705	DH32	;AFTER WRITE CHECK CMD
2402	010276	054212	DT13	
2403	010300	054736	DF21	
2404			;ERR 60	
2405	010302	047317	EM19	;MSG A1 ERROR
2406	010304	052705	DH32	
2407	010306	054212	DT13	
2408	010310	054736	DF21	
2409			;ERR 61	
2410	010312	047276	EM18	;MSG B0 ERROR
2411	010314	053470	DH53	;DURING OFFSET CMD
2412	010316	054212	DT13	
2413	010320	054736	DF21	
2414			;ERR 62	
2415	010322	047340	EM20	;MSG B1 ERROR
2416	010324	053470	DH53	
2417	010326	054212	DT13	
2418	010330	054736	DF21	
2419			;ERR 63	
2420	010332	047715	EM26	;BSE ERROR IN WRITE CMD NOT ON BSE TABLE
2421	010334	052555	DH27	;AFTER WRITE DATA CMD
2422	010336	053654	DT1	
2423	010340	054606	DF10	
2424			;ERR 64	
2425	010342	051153	EM88	;DID NOT FIND SECTOR 0 FROM INDEX

2426	010344	053516	DH54	;AFTER FORMAT CHANGE AND READY REC'D
2427	010346	053654	DT1	
2428	010350	054606	DF10	
2429			;ERR 65	
2430	010352	047774	EM27	;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
2431	010354	051572	DH1	
2432	010356	053654	DT1	
2433	010360	054456	DF1	
2434			;ERR 66	
2435	010362	000000	0	
2436	010364	000000	0	
2437	010366	000000	0	
2438	010370	000000	0	
2439			;ERR 67	
2440	010372	000000	0	
2441	010374	000000	0	
2442	010376	000000	0	
2443	010400	000000	0	
2444			;ERROR 70	
2445	010402	000000	0	
2446	010404	000000	0	
2447	010406	000000	0	
2448	010410	000000	0	
2449			;ERR 71	
2450	010412	000000	0	
2451	010414	000000	0	
2452	010416	000000	0	
2453	010420	000000	0	
2454			;ERROR 72	
2455	010422	000000	0	
2456	010424	000000	0	
2457	010426	000000	0	
2458	010430	000000	0	
2459			;ERR 73	
2460	010432	000000	0	
2461	010434	000000	0	
2462	010436	000000	0	
2463	010440	000000	0	
2464			;ERR 74	
2465	010442	047023	EM13	;NO ATTN
2466	010444	052057	DH10	;AT END OF HEAD LOADING
2467	010446	053654	DT1	
2468	010450	054606	DF10	
2469			;ERR 75	
2470	010452	047403	EM22	;NO DRIVS IN \$DEVN
2471	010454	051572	DH1	
2472	010456	053654	DT1	
2473	010460	054456	DF1	
2474			;ERR 76	
2475	010462	047510	EM23	;NO DRIVS ON BUSS
2476	010464	051572	DH1	
2477	010466	053654	DT1	
2478	010470	054456	DF1	
2479			;ERR 77	
2480	010472	000000	0	
2481	010474	000000	0	

2582	010476	000000	0	
2583	010500	000000	0	
2584			;ERR 100	
2585	010502	000000	0	
2586	010504	000000	0	
2587	010506	000000	0	
2588	010510	000000	0	
2589			;ERROR 101	
2590	010512	051275	EM94	;OFFSET NOT CLEARED
2591	010514	053254	DH47	;AFTER READ HEADER WITH MOVEMENT
2592	010516	054272	DT14	
2593	010520	054772	DF22	
2594			;ERROR 102	
2595	010522	051331	EM95	;FORMAT NOT SET
2596	010524	052555	DH27	;AFTER WRITE DATA CMD
2597	010526	053654	DT1	
2598	010530	054606	DF10	
2599			;ERR 103	
2500	010532	051331	EM95	
2501	010534	052705	DH32	;AFTER WRITE CHECK CMD
2502	010536	053654	DT1	
2503	010540	054606	DF10	
2504			;ERR 104	
2505	010542	050156	EM39	;OFFSET NOT RESET
2506	010544	053614	DH57	;AFTER WRITE CMD WITH OFFSET
2507	010546	054272	DT14	
2508	010550	054772	DF22	
2509			;ERR 105	
2510	010552	050225	EM40	;CYL ADDR NOT 0
2511	010554	053614	DH57	
2512	010556	054272	DT14	
2513	010560	054772	DF22	
2514			;ERR 106	
2515	010562	051365	EM96	;CANNOT FIND SECTOR 23(8)
2516	010564	051572	DH1	
2517	010566	053654	DT1	
2518	010570	054456	DF1	
2519			;ERR 107	
2520	010572	051416	EM97	;HEAD SWITCHING TOO LONG
2521	010574	052555	DH27	;AFTER WRITE DTA CMD
2522	010576	053654	DT1	
2523	010600	054646	DF15	
2524			;ERR 110	
2525	010602	000000	0	
2526	010604	000000	0	
2527	010606	000000	0	
2528	010610	000000	0	
2529			;ERR 111	
2530	010612	000000	0	
2531	010614	000000	0	
2532	010616	000000	0	
2533	010620	000000	0	
2534			;ERR 112	
2535	010622	051503	EM100	;DRIVE OFF TRACK SET
2536	010624	052555	DH27	;AFTER WRITE DATA CMD
2537	010626	053720	DT4	

2538	010630	054562	DF6	
2539			;ERR 113	
2540	010632	050113	EM36	;CYL ADDR IN RKMR3 INCORRECT
2541	010634	052555	DH27	
2542	010636	053720	DT4	
2543	010640	054562	DF6	
2544			;ERROR 114	
2545	010642	051115	EM86	;OFFSET IN A2 NOT = RKASOF
2546	010644	052455	DH24	;AFTER OFFSET CMD
2547	010646	054272	DT14	
2548	010650	054772	DF22	
2549			;ERR 115	
2550	010652	051115	EM86	
2551	010654	052423	DH22	;AFTER DRIVE CLEAR CMD
2552	010656	054272	DT14	
2553	010660	054772	DF22	
2554			;ERROR 116	
2555	010662	046765	EM12	;CONT NOT RDY
2556	010664	052317	DH19	;AFTER PACK CMD
2557	010666	053654	DT1	
2558	010670	054606	DF10	
2559			;ERROR 117	
2560	010672	046765	EM12	;CONT NOT RDY
2561	010674	052342	DH20	;AFTER SEL DR CMD
2562	010676	053654	DT1	
2563	010700	054606	DF10	
2564			;ERROR 120	
2565	010702	046765	EM12	
2566	010704	052375	DH21	;AFTER SUBSYS CLEAR
2567	010706	053654	DT1	
2568	010710	054606	DF10	
2569			;ERROR 121	
2570	010712	046765	EM12	
2571	010714	052004	DH9	;AFTER START SPINDLE CMD
2572	010716	053654	DT1	
2573	010720	054606	DF10	
2574			;ERROR 122	
2575	010722	051540	EM101	;DID NOT GO TO CYL 10
2576	010724	052626	DH30	;AFTER READ HEADER CMD
2577	010726	054272	DT14	
2578	010730	054772	DF22	
2579			;ERROR 123	
2580	010732	051115	EM86	;A2 OFFSET NOT = RKASOF
2581	010734	053401	DH51	;AFTER SEEK TO SELF
2582	010736	054272	DT14	
2583	010740	054772	DF22	
2584			;ERROR 124	
2585	010742	046765	EM12	
2586	010744	052273	DH17	;AFTER RECAL CMD
2587	010746	053654	DT1	
2588	010750	054606	DF10	
2589			;ERR 125	
2590	010752	050460	EM73	;CTO SET
2591	010754	050761	EM84	;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
2592	010756	053654	DT1	
2593	010760	054536	DF5	

2594			;ERR 126		
2595	010762	050526		EM79	;NED SET
2596	010764	050761		EM84	
2597	010766	053654		DT1	
2598	010770	054536		DF5	
2599			;ERR 127		
2600	010772	046355		EM5	;MDS SET
2601	010774	050761		EM84	
2602	010776	053654		DT1	
2603	011000	054536		DF5	
2604			;ERROR 130		
2605	011002	000000		0	
2606	011004	000000		0	
2607	011006	000000		0	
2608	011010	000000		0	
2609			;ERROR 131		
2610	011012	046765		EM12	;NO RDY
2611	011014	052502		DH25	;AFTER SEEK CMD
2612	011016	053654		DT1	
2613	011020	054606		DF10	
2614			;ERROR 132		
2615	011022	047023		EM13	;NO ATTN
2616	011024	052502		DH25	
2617	011026	053654		DT1	
2618	011030	054606		DF10	
2619			;ERROR 133		
2620	011032	000000		0	
2621	011034	000000		0	
2622	011036	000000		0	
2623	011040	000000		0	
2624			;ERROR 134		
2625	011042	000000		0	
2626	011044	000000		0	
2627	011046	000000		0	
2628	011050	000000		0	
2629			;ERROR 135		
2630	011052	000000		0	
2631	011054	000000		0	
2632	011056	000000		0	
2633	011060	000000		0	
2634			;ERROR 136		
2635	011062	000000		0	
2636	011064	000000		0	
2637	011066	000000		0	
2638	011070	000000		0	
2639			;ERROR 137		
2640	011072	050156		EM39	;CYL DIFF/OFFSET IN RKM2 NOT CLEARED
2641	011074	052502		DH25	
2642	011076	053654		DT1	
2643	011100	054606		DF10	
2644			;ERR 140		
2645	011102	047255		EM17	;MSG AD ERROR
2646	011104	053401		DH51	;AFTER SEEK TO SELF
2647	011106	054212		DT13	
2648	011110	054736		DF21	
2649			;ERR 141		

2650	011112	047276	EM18
2651	011114	053401	DH51
2652	011116	054212	DT13
2653	011120	054736	DF21
2654			;ERR 142
2655	011122	047317	EM19
2656	011124	053401	DH51
2657	011126	054212	DT13
2658	011130	054736	DF21
2659			;ERR 143
2660	011132	047340	EM20
2661	011134	053401	DH51
2662	011136	054212	DT13
2663	011140	054736	DF21
2664			;ERROR 144
2665	011142	000000	0
2666	011144	000000	0
2667	011146	000000	0
2668	011150	000000	0
2669			;ERROR 145
2670	011152	000000	0
2671	011154	000000	0
2672	011156	000000	0
2673	011160	000000	0
2674			;ERROR 146
2675	011162	000000	0
2676	011164	000000	0
2677	011166	000000	0
2678	011170	000000	0
2679			;ERROR 147
2680	011172	000000	0
2681	011174	000000	0
2682	011176	000000	0
2683	011200	000000	0
2684			;ERROR 150
2685	011202	000000	0
2686	011204	000000	0
2687	011206	000000	0
2688	011210	000000	0
2689			;ERROR 151
2690	011212	046765	EM12
2691	011214	052423	DH22
2692	011216	053654	DT1
2693	011220	054606	DF10
2694			;ERROR 152
2695	011222	000000	0
2696	011224	000000	0
2697	011226	000000	0
2698	011230	000000	0
2699			;ERROR 153
2700	011232	000000	0
2701	011234	000000	0
2702	011236	000000	0
2703	011240	000000	0
2704			;ERROR 154
2705	011242	050331	EM55

;NO RDY  
;AFTER CLEAR CMD

;ATTN NOT CLEARED

2706	011244	052423			DH22
2707	011246	053654			DT1
2708	011250	054606			DF10
2709			;ERROR	155	
2710	011252	000000			0
2711	011254	000000			0
2712	011256	000000			0
2713	011260	000000			0
2714			;ERROR	156	
2715	011262	000000			0
2716	011264	000000			0
2717	011266	000000			0
2718	011270	000000			0
2719			;ERROR	157	
2720	011272	000000			0
2721	011274	000000			0
2722	011276	000000			0
2723	011300	000000			0
2724			;ERROR	160	
2725	011302	000000			0
2726	011304	000000			0
2727	011306	000000			0
2728	011310	000000			0
2729			;ERROR	161	
2730	011312	000000			0
2731	011314	000000			0
2732	011316	000000			0
2733	011320	000000			0
2734			;ERROR	162	
2735	011322	000000			0
2736	011324	000000			0
2737	011326	000000			0
2738	011330	000000			0
2739			;ERROR	163	
2740	011332	000000			0
2741	011334	000000			0
2742	011336	000000			0
2743					0
2744					
2745					
2746					
2747					
2748					
2749					
2750					
2751					
2752					
2753					
2754					
2755					
2756					
2757					
2758					
2759					
2760					
2761					



2762					
2763					
2764	011340	000000		0	
2765			; ERROR	164	
2766	011342	000000		0	
2767	011344	000000		0	
2768	011346	000000		0	
2769	011350	000000		0	
2770			; ERROR	165	
2771	011352	000000		0	
2772	011354	000000		0	
2773	011356	000000		0	
2774	011360	000000		0	
2775			; ERROR	166	
2776	011362	000000		0	
2777	011364	000000		0	
2778	011366	000000		0	
2779	011370	000000		0	
2780			; ERROR	167	
2781	011372	000000		0	
2782	011374	000000		0	
2783	011376	000000		0	
2784	011400	000000		0	
2785			; ERROR	170	
2786	011402	000000		0	
2787	011404	000000		0	
2788	011406	000000		0	
2789	011410	000000		0	
2790			; ERROR	171	
2791	011412	046765		EM12	; NO RDY
2792	011414	052626		DH30	; AFTER READ HEADER CMD
2793	011416	053654		DT1	
2794	011420	054606		DF10	
2795			; ERROR	172	
2796	011422	000000		0	
2797	011424	000000		0	
2798	011426	000000		0	
2799	011430	000000		0	
2800			; ERROR	173	
2801	011432	050364		EM63	; DLT SET
2802	011434	052626		DH30	
2803	011436	053654		DT1	
2804	011440	054646		DF15	
2805			; ERROR	174	
2806	011442	047361		EM21	; CERR SET
2807	011444	052626		DH30	
2808	011446	053654		DT1	
2809	011450	054646		DF15	
2810			; ERROR	175	
2811	011452	050156		EM39	; CYL DIFF NOT CLEARED
2812	011454	052057		DH10	; AT END OF HEAD LOADING
2813	011456	053654		DT1	
2814	011460	054606		DF10	
2815			; ERROR	176	
2816	011462	050225		EM40	; CYL ADDR NOT CLEARED.
2817	011464	052057		DH10	

2818	011466	053654		DT1	
2819	011470	054606		DF10	
2820			;ERROR 177	0	
2821	011472	000000		0	
2822	011474	000000		0	
2823	011476	000000		0	
2824	011500	000000		0	
2825			;ERROR 200	EM12	;NO RDY
2826	011502	046765		DH39	;AFTER WRITE HEADER CMD
2827	011504	052737		DT1	
2828	011506	053654		DF15	
2829	011510	054646			
2830			;ERROR 201	EM21	;CERR SET
2831	011512	047361		DH39	
2832	011514	052737		DT1	
2833	011516	053654		DF15	
2834	011520	054646			
2835			;ERROR 202	EM65	;READ HEADER ERROR
2836	011522	050405		DH1	
2837	011524	051572		DT7	
2838	011526	054032		DF14	
2839	011530	054626			
2840			;ERROR 203	0	
2841	011532	000000		0	
2842	011534	000000		0	
2843	011536	000000		0	
2844	011540	000000		0	
2845			;ERROR 204	0	
2846	011542	000000		0	
2847	011544	000000		0	
2848	011546	000000		0	
2849	011550	000000		0	
2850			;ERROR 205	0	
2851	011552	000000		0	
2852	011554	000000		0	
2853	011556	000000		0	
2854	011560	000000		0	
2855			;ERROR 206	0	
2856	011562	000000		0	
2857	011564	000000		0	
2858	011566	000000		0	
2859	011570	000000		0	
2860			;ERROR 207	EM36	;CYL ADDR IN RKMR3 INCORRECT
2861	011572	050113		DH25	;AFTER SEEK CMD
2862	011574	052502		DT4	
2863	011576	053720		DF6	
2864	011600	054562			
2865			;ERROR 210	EM21	;CERR SET
2866	011602	047361		DH25	
2867	011604	052502		DT1	
2868	011606	053654		DF10	
2869	011610	054606			
2870			;ERROR 211	0	
2871	011612	000000		0	
2872	011614	000000		0	
2873	011616	000000		0	

2874	011620	000000	0	
2875			;ERROR 212	
2876	011622	000000	0	
2877	011624	000000	0	
2878	011626	000000	0	
2879	011630	000000	0	
2880			;ERROR 213	
2881	011632	000000	0	
2882	011634	000000	0	
2883	011636	000000	0	
2884	011640	000000	0	
2885			;ERROR 214	
2886	011642	000000	0	
2887	011644	000000	0	
2888	011646	000000	0	
2889	011650	000000	0	
2890			;ERROR 215	
2891	011652	000000	0	
2892	011654	000000	0	
2893	011656	000000	0	
2894	011660	000000	0	
2895			;ERROR 216	
2896	011662	000000	0	
2897	011664	000000	0	
2898	011666	000000	0	
2899	011670	000000	0	
2900			;ERROR 217	
2901	011672	000000	0	
2902	011674	000000	0	
2903	011676	000000	0	
2904	011700	000000	0	
2905			;ERROR 220	
2906	011702	000000	0	
2907	011704	000000	0	
2908	011706	000000	0	
2909	011710	000000	0	
2910			;ERROR 221	
2911	011712	047255	EM17	;MSG A0 ERROR
2912	011714	052273	DH17	
2913	011716	054212	DT13	
2914	011720	054736	DF21	
2915			;ERROR 222	
2916	011722	047317	EM19	;MSG A1 ERROR
2917	011724	052273	DH17	
2918	011726	054212	DT13	
2919	011730	054736	DF21	
2920			;ERROR 223	
2921	011732	000000	0	
2922	011734	000000	0	
2923	011736	000000	0	
2924	011740	000000	0	
2925			;ERROR 224	
2926	011742	000000	0	
2927	011744	000000	0	
2928	011746	000000	0	
2929	011750	000000	0	

2930			;ERROR 225	
2931	011752	000000	0	
2932	011754	000000	0	
2933	011756	000000	0	
2934	011760	000000	0	
2935			;ERROR 226	
2936	011762	046765	EM12	;NO RDY
2937	011764	052525	DH26	;AFTER READ DATA CMD
2938	011766	053654	DT1	
2939	011770	054606	DF10	
2940			;ERROR 227	
2941	011772	047361	EM21	;CERR SET
2942	011774	052525	DH26	
2943	011776	053654	DT1	
2944	012000	054646	DF15	
2945			;ERROR 230	
2946	012002	047212	EM16	;CANNOT READ BSE INFO
2947	012004	052123	DH13	;ON SEC 10, 12, 14, 16, 18, 20
2948	012006	053654	DT1	
2949	012010	054666	DF17	
2950			;ERROR 231	
2951	012012	047212	EM16	
2952	012014	052207	DH14	;ON SEC 11, 13, 15, 17, 19, 21
2953	012016	053654	DT1	
2954	012020	054666	DF17	
2955			;ERROR 232	
2956	012022	000000	0	
2957	012024	000000	0	
2958	012026	000000	0	
2959	012030	000000	0	
2960			;ERROR 233	
2961	012032	047212	EM16	;CANNOT READ BSE INFO
2962	012034	053053	DH42	;ON SECT 0,2,4,6,8
2963	012036	053654	DT1	
2964	012040	054666	DF17	
2965			;ERROR 234	
2966	012042	047212	EM16	
2967	012044	053124	DH43	;ON SECT 1,3,5,7,9
2968	012046	053654	DT1	
2969	012050	054666	DF17	
2970			;ERROR 235	
2971	012052	050427	EM69	;ALIGN CARTRIDGE USED
2972	012054	053175	DH44	;WILL BYPASS FORMAT & ALL R/W TESTS
2973	012056	053654	DT1	
2974	012060	054606	DF10	
2975			;ERROR 236	
2976	012062	000000	0	
2977	012064	000000	0	
2978	012066	000000	0	
2979	012070	000000	0	
2980			;ERROR 237	
2981	012072	000000	0	
2982	012074	000000	0	
2983	012076	000000	0	
2984	012100	000000	0	
2985			;ERROR 240	

2986	012102	000000	0	
2987	012104	000000	0	
2988	012106	000000	0	
2989	012110	000000	0	
2990			;	ERROR 241
2991	012112	000000	0	
2992	012114	000000	0	
2993	012116	000000	0	
2994	012120	000000	0	
2995			;	ERROR 242
2996	012122	000000	0	
2997	012124	000000	0	
2998	012126	000000	0	
2999	012130	000000	0	
3000				
3001			;	ERROR 243
3002	012132	050113		EM36
3003	012134	052502		DH25
3004	012136	054100		DT8
3005	012140	054562		DF6
3006			;	ERR 244
3007	012142	050501		EM74
3008	012144	053026		DH41
3009	012146	053654		DT1
3010	012150	054606		DF10
3011			;	ERR 245
3012	012152	000000	0	
3013	012154	000000	0	
3014	012156	000000	0	
3015	012160	000000	0	
3016			;	ERR 246
3017	012162	000000	0	
3018	012164	000000	0	
3019	012166	000000	0	
3020	012170	000000	0	
3021			;	ERR 247
3022	012172	000000	0	
3023	012174	000000	0	
3024	012176	000000	0	
3025	012200	000000	0	
3026			;	ERR 250
3027	012202	000000	0	
3028	012204	000000	0	
3029	012206	000000	0	
3030	012210	000000	0	
3031			;	ERR 251
3032	012212	000000	0	
3033	012214	000000	0	
3034	012216	000000	0	
3035	012220	000000	0	
3036			;	ERR 252
3037	012222	000000	0	
3038	012224	000000	0	
3039	012226	000000	0	
3040	012230	000000	0	
3041			;	ERR 253

;CYL ADDR IN RKMR3 INCORRECT  
;AFTER SEEK CMD

;RTZ NOT SET  
;DURING RECAL CMD

3042	012232	000000	0	
3043	012234	000000	0	
3044	012236	000000	0	
3045	012240	000000	0	
3046			0	
3047	012242	000000	0	
3048	012244	000000	0	
3049	012246	000000	0	
3050	012250	000000	0	
3051			0	
3052	012252	000000	0	
3053	012254	000000	0	
3054	012256	000000	0	
3055	012260	000000	0	
3056			0	
3057	012262	000000	0	
3058	012264	000000	0	
3059	012266	000000	0	
3060	012270	000000	0	
3061			0	
3062	012272	000000	0	
3063	012274	000000	0	
3064	012276	000000	0	
3065	012300	000000	0	
3066			0	
3067	012302	047255	EM17	;MSG A0 ERROR
3068	012304	052455	DH24	;AFTER OFFSET CMD
3069	012306	054212	DT13	
3070	012310	054736	DF21	
3071			0	
3072	012312	047276	EM18	;MSG B0 ERROR
3073	012314	052455	DH24	
3074	012316	054212	DT13	
3075	012320	054736	DF21	
3076			0	
3077	012322	000000	0	
3078	012324	000000	0	
3079	012326	000000	0	
3080	012330	000000	0	
3081			0	
3082	012332	000000	0	
3083	012334	000000	0	
3084	012336	000000	0	
3085	012340	000000	0	
3086			0	
3087	012342	000000	0	
3088	012344	000000	0	
3089	012346	000000	0	
3090	012350	000000	0	
3091			0	
3092	012352	047276	EM18	;MSG B0 ERROR
3093	012354	052423	DH22	;AFTER DRIVE CLEAR CMD
3094	012356	054212	DT13	
3095	012360	054736	DF21	
3096			0	
3097	012362	047340	EM20	;MSG B1 ERROR

3098	012364	052423	DH22	
3099	012366	054212	DT13	
3100	012370	054736	DF21	
3101				;ERR 267
3102	012372	047276	EM18	;MSG B0 ERROR
3103	012374	052737	DH39	;AFTER WRITE HEADER CMD
3104	012376	054212	DT13	
3105	012400	054736	DF21	
3106				;ERR 270
3107	012402	047340	EM20	;MSG B1 ERROR
3108	012404	052737	DH39	
3109	012406	054212	DT13	
3110	012410	054736	DF21	
3111				;ERR 271
3112	012412	047276	EM18	
3113	012414	052626	DH30	;AFTER RD. HDR. CMD.
3114	012416	054212	DT13	
3115	012420	054736	DF21	
3116				;ERR 272
3117	012422	047340	EM20	
3118	012424	052626	DH30	
3119	012426	054212	DT13	
3120	012430	054736	DF21	
3121				;ERR 273
3122	012432	047255	EM17	;MSG A0 ERROR
3123	012434	052423	DH22	;AFTER DRV CLR CMD
3124	012436	054212	DT13	
3125	012440	054736	DF21	
3126				;ERR 274
3127	012442	047317	EM19	;MSG A1 ERROR
3128	012444	052423	DH22	
3129	012446	054212	DT13	
3130	012450	054736	DF21	
3131				;ERR 275
3132	012452	047276	EM18	;MSG B0 ERROR
3133	012454	052273	DH17	;AFTER RECAL CMD
3134	012456	054212	DT13	
3135	012460	054736	DF21	
3136				;ERR 276
3137	012462	047340	EM20	;MSG B1 ERROR
3138	012464	052273	DH17	
3139	012466	054212	DT13	
3140	012470	054736	DF21	
3141				;ERR 277
3142	012472	047255	EM17	;MSG A0 ERROR
3143	012474	052737	DH39	;AFTER WRITE HEADER CMD
3144	012476	054212	DT13	
3145	012500	054736	DF21	
3146				;ERR 300
3147	012502	047317	EM19	;MSG A1 ERROR
3148	012504	052737	DH39	
3149	012506	054212	DT13	
3150	012510	054736	DF21	
3151				;ERR 301
3152	012512	047255	EM17	
3153	012514	052626	DH30	;AFT RD HDR. CMD

3154	012516	054212	DT13
3155	012520	054736	DF21
3156			:ERR 302
3157	012522	047317	EM19
3158	012524	052626	DH30
3159	012526	054212	DT13
3160	012530	054736	DF21
3161			



```

3162
3163
3164
3165 012532 012737 000001 001336 PARSRT: MOV #1,PARAM ;SET FLAG FOR 220 START
3166 012540 000402 BR PRGSRT ;START PROGRAM
3167
3168 012542 005037 001336 START: CLR PARAM ;CLEAR FOR 200 START
3169 012546 000005 PRGSRT: RESET ;CLEAR ALL INT ENABLE & INIT
3170 012550 012706 001100 MOV #STACK,SP ;SETUP STACK POINTER
3171 012554 012746 000000 MOV #PRO,-(SP) ;PSW LOADED TO BE
3172 012560 012746 012566 MOV #IS,-(SP) ;LSI-11 COMPATABLE
3173 012564 000002 RTI ;ENABLE ALL INTERRUPTS
3174
3175 012566 004737 041044 IS: JSR PC,STKINT ;SETUP KB VECTOR ADDR, PRIORITY 4
3176 ; & TURN ON KB INTERRUPT
3177
3178
3179 ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3180 ;*** ANY DEVICE WHICH SETS ITS ***
3181 ;*** INTERRUPT ENABLE BIT WILL ***
3182 ;*** SERVICED. ***
3183
3184 ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'STS')
3185 ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3186 ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3187
3188 ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3189
3190
3191 ;SYSMAC 'SETUP'
3192 .SBTTL INITIALIZE THE COMMON TAGS
3193 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3194 MOV #CMTAG,R6 ;:FIRST LOCATION TO BE CLEARED
3195 CLR (R6)+ ;:CLEAR MEMORY LOCATION
3196 CMP #SWR,R6 ;:DONE?
3197 BNE -6 ;:LOOP BACK IF NO
3198 MOV #STACK,SP ;:SETUP THE STACK POINTER
3199 ;;INITIALIZE A FEW VECTORS
3200 MOV #SCOPE,#IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
3201 MOV #340,#IOTVEC+2 ;:LEVEL 7
3202 MOV #ERROR,#EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
3203 MOV #340,#EMTVEC+2 ;:LEVEL 7
3204 MOV #TRAP,#TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
3205 MOV #340,#TRAPVEC+2 ;:LEVEL 7
3206 MOV #SPWRON,#PWAVEC ;:POWER FAILURE VECTOR
3207 MOV #340,#PWAVEC+2 ;:LEVEL 7
3208 MOV $ENDCT,$EOPCT ;:SETUP END-OF-PROGRAM COUNTER
3209 CLR $TIMES ;:INITIALIZE NUMBER OF ITERATIONS
3210 CLR $ESCAPE ;:CLEAR THE ESCAPE ON ERROR ADDRESS
3211 MOV #1,$SERMAX ;:ALLOW ONE ERROR PER TEST
3212 MOV #,$SLPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
3213 MOV #,$SLPERR ;:SETUP THE ERROR LOOP ADDRESS
3214 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3215 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
3216 MOV #ERRVEC,-(SP) ;:SAVE ERROR VECTOR
3217 MOV #64,$ERRVEC ;:SET UP ERROR VECTOR
3218 MOV #DSWR,$SWR ;:SETUP FOR A HARDWARE SWICH REGISTER

```

```

3218 012752 012737 177570 001142      MOV    #DDISP,DISPLAY    ;; AND A HARDWARE DISPLAY REGISTER
3219 012760 022777 177777 166152      CMP    #-1,#SWR         ;; TRY TO REFERENCE HARDWARE SWR
3220 012766 001012                    BNE    66$              ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
3221                    ; AND THE HARDWARE SWR IS NOT = -1
3222 012770 000403                    BR     65$              ;; BRANCH IF NO TIMEOUT
3223 012772 012716 013000 64$:      MOV    #65$, (SP)      ;; SET UP FOR TRAP RETURN
3224 012776 000002                    RTI
3225 013000 012737 000176 001140 65$:      MOV    #SWREG,SWR      ;; POINT TO SOFTWARE SWR
3226 013006 012737 000174 001142      MOV    #DISPREG,DISPLAY
3227 013014 012637 000004 66$:      MOV    (SP)+,#ERRVEC  ;; RESTORE ERROR VECTOR
3228
3229 013020 005037 001216                    CLR    $PASS           ;; CLEAR PASS COUNT
3230 013024 132737 000200 001231      BITB  #APTSIZE,$ENVM  ;; TEST USER SIZE UNDER APT
3231 013032 001403                    BEQ    67$             ;; YES, USE NON-APT SWITCH
3232 013034 012737 001232 001140      MOV    #SSWREG,SWR    ;; NO, USE APT SWITCH REGISTER
3233 013042
3234 013042 012737 013106 000004 67$:      MOV    #1$,ERRVEC     ;; SETUP TIMEOUT VECTOR
3235 013050 012737 000340 000006      MOV    #PR7,ERRVEC+2
3236
3237 013056 012701 172100                    MOV    #MEMBAS,R1     ;; ADDR OF MEM CSR
3238 013062 005011 3$:      CLR    (R1)           ;; SEE IF CAN REFERENCE
3239 013064 012711 000001                    MOV    #1,(R1)        ;; SET ENABLE BIT IF YES
3240 013070 012737 036610 000114      MOV    #MEMERR,MEMVEC ;; LOAD VECTOR IF NO TIMEOUT
3241 013076 012737 000340 000116      MOV    #PR7,MEMVEC+2
3242 013104 000401                    BR     2$
3243
3244 013106 022626 1$:      CMP    (SP)+,(SP)+    ;; ADJ STACK
3245 013110 062701 000002 2$:      ADD    #2,R1          ;; TRY NEXT CSR
3246 013114 020127 172140      CMP    R1,#MEMBAS+40 ;; SEE IF TRIED ALL
3247 013120 001360                    BNE    3$             ;; BR IF NO
3248 013122 012737 000006 000004      MOV    #ERRVEC+2,ERRVEC ;; RESTORE TRAP CATCHER
3249 013130 005037 000006      CLR    ERRVEC+2
3250
3251 013134 004737 031672                    JSR    PC,CLRFLG      ;; CLEAR DDUMP THRU SIZFLG
3252 013140 005037 001220      CLR    $DEVCT
3253 013144 005037 001222      CLR    $UNIT
3254
3255
3256 ; FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
3257 ;
3258
3259 013150 005737 000042      START1: TST    42
3260 013154 001014                    BNE    1$             ;; BR IF AUTO
3261 013156 004737 031712                    JSR    PC,TITLE      ;; MANUAL, TYPE PROG ID
3262 013162 123727 000041 000013      CMPB  41,#13         ;; 13=LOADED BY XXDP
3263 013170 001010                    BNE    2$
3264 013172 005237 007450      INC    DDUMP          ;; SET RK06 DUMP MODE FLAG
3265 013176 104401 044242      TYPE  MSG2           ;; REPLACE DR0 PACK W/SCRATCH & DO<CR>
3266 013202 000137 013216                    JMP    $T2
3267 013206 000137 013262 1$:      JMP    $T3
3268 013212 005237 007456 2$:      INC    PPTP          ;; SET ACT/APT/PTP DUMP MODE FLAG
3269
3270
3271 ; CHECK IF ALL PARAMETERS DEFAULTED. IF NOT, BEGIN INPUT DIALOGUE
3272 ; WITH OPERATOR. THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE
3273 ; DRIVE NOS. SEPERATED BY COMMAS & TERMINATED BY <CR>

```

M05

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 63  
INITIALIZE THE COMMON TAGS

SEQ 0063

```

3274 ; EX: DRIVES TO BE TESTED: 1,2,4<CR>
3275 ;
3276 ;
3277 013216 005737 001336 ST2: TST PARAM
3278 013222 001002 BNE 1$ ;BR IF 220 START
3279 013224 000137 013314 JMP ST4 ;200 START, DEFAULT & SIZE THE BUSS
3280 013230 104401 044313 1$: TYPE MSG3 ;DRIVES TO BE TESTED
3281 013234 004737 031772 JSR PC,DRVS ;GET DR NOS.
3282 013240 104401 044345 TYPE MSG4 ;BUSS ADDR
3283 013244 004737 032132 JSR PC,GBA ;GET BA
3284 013250 104401 044412 TYPE MSG5 ;CONT INT VECTOR
3285 013254 004737 032160 JSR PC,GINT ;GET INT VECTOR
3286 013260 000427 BR ST5
3287 ;
3288 ;
3289 ;
3290 ;
3291 ;
3292 ;
3293 ;
3294 ;
3295 013262 123727 000041 000013 ST3: CMPB 41,#13 ;13=LOADED BY XXDP
3296 013270 001007 BNE 1$
3297 013272 005237 007452 INC DDPCH ;SET RK06 CHAIN MODE FLAG
3298 013276 004737 031712 JSR PC,TITLE
3299 013302 104401 044527 TYPE MSG7 ;DRO NOT TSTD
3300 013306 000402 BR ST4
3301 013310 005237 007454 1$: INC ACT11 ;SET ACT AUTO FLAG.
3302 ;
3303 013314 012737 177440 001264 ST4: MOV #177440,$BASE ;DEFAULT VALUE
3304 013322 012737 000210 001314 MOV #210,RKVEC ;DEFAULT VALUE
3305 013330 004737 032212 JSR PC,SETINT
3306 013334 005237 007510 INC SIZEFLG ;DO "SIZE THE BUSS" TEST
3307 ;
3308 013340 005037 007322 ST5: CLR UNLD ;INITIALIZE FLAGS
3309 013344 005037 007324 CLR BADHDR ;USED IN 'STOP' ROUTINE
3310 013350 005037 007326 CLR HPEND ;FOR VALID PROGRAM HALTS
3311 013354 005037 001176 CLR $ESCAPE
3312 013360 012737 007462 001342 MOV #DRIVO,DRVPTR ;SETUP
3313 013366 005037 001220 CLR $DEVCT ;NO. OF DRVS DONE
3314 013372 005037 001222 CLR $UNIT ;CURRENT DRV UNDER TEST
3315 013376 012737 013444 000004 MOV #1$ ERRVEC ;SETUP TIMEOUT ERROR VECTOR
3316 013404 005777 165716 TST $LK$ ;SEE IF L-CLOCK THERE
3317 013410 005237 007502 INC LC!KF ;PRESENT, SET FLAG.
3318 013414 013700 001330 MOV LCVEC,RO ;VECTOR ADDR
3319 013420 012737 013506 000004 MOV #2$ ERRVEC
3320 013426 005777 165666 TST $PK$ ;SEE IF P-CLOCK THERE
3321 013432 005237 007504 INC PCLKF ;PRESENT, SET FLAG
3322 013436 013700 001332 MOV PCVEC,RO ;VECTOR ADDR
3323 013442 000412 BR
3324 ;
3325 013444 022626 1$: CMP (SP)+,(SP)+ ;L-CLOCK NOT THERE, CLEAR STACK
3326 013446 012737 013512 000004 MOV #4$ ERRVEC
3327 013454 005777 165640 TST $PK$ ;SEE IF P-CLOCK THERE
3328 013460 005237 007504 INC PCLKF ;PRESENT, SET FLAG
3329 013464 013700 001332 MOV PCVEC,RO ;VECTOR ADDR

```

3330 013470 005237 007506  
 3331 013474 012720 036010  
 3332 013500 012710 000300  
 3333 013504 000407  
 3334  
 3335 013506 022626  
 3336 013510 000767  
 3337  
 3338 013512 022626  
 3339 013514 005037 007506  
 3340 013520 104401 045070  
 3341  
 3342

3\$: INC DOTIM ;INDICATES TIMING TESTS CAN BE DONE  
 MOV #CLOCK,(R0)+ ;SERVICE ROUTINE FOR CLOCKS  
 MOV #PR6,(R0)  
 BR TST1 ;;GO TO NEXT TEST

2\$: CMP (SP)+,(SP)+ ;P-CLOCK NOT THERE, CLEAR STACK  
 BR 3\$

4\$: CMP (SP)+,(SP)+ ;NEITHER CLOCK THERE, CLEAR STACK  
 CLR DOTIM ;TIMING TESTS CANNOT BE DONE.  
 TYPE ,MSG13 ;HEAD SW. TEST BYPASSED

.SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

3343  
3344  
3345  
3346  
3347  
3348  
3349  
3350  
3351  
3352  
3353  
3354  
3355  
3356  
3357  
3358  
3359  
3360  
3361  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385  
3386  
3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398

\*\*\*\*\*  
\*TEST 1 REFERENCE ALL CONTROLLER REGISTERS  
\*  
\* THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS  
\* CAN BE ACCESSED. THE INABILITY TO BE ACCESSED WILL  
\* RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE. ANY  
\* ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER  
\* TESTS AND JUMPING TO 'END OF PASS'  
\*\*\*\*\*

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

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

SS:
MOV #IS,ERRVEC ;SETUP TIMEOUT ERROR VECTOR
MOV #BASE,RS ;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 $EOP1
```

\*\*\*\*\*  
\*TEST 2 SIZE THE BUSS  
\*  
\* THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED  
\* EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE  
\* MANUAL MODE.  
\* EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.  
\* CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE  
\* DRIVE WILL BE TESTED. IF SET, THE PROGRAM WILL BYPASS  
\*\*\*\*\*

3399  
3400  
3401  
3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
3411  
3412  
3413  
3414  
3415  
3416  
3417  
3418  
3419  
3420  
3421  
3422  
3423  
3424  
3425  
3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439  
3440  
3441  
3442  
3443  
3444  
3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454

013700 000004  
013702 012737 000001 001174  
013710 012706 001100  
013714 005237 001502  
013720 132737 000200 001231  
013726 001002  
013730 000137 014044  
013734 104401 044636  
013740 005037 007460  
013744 005000  
013746 012701 007462  
013752 013702 001266  
013756 032702 000001  
013762 001410  
013764 005237 007460  
013770 005211  
013772 104401 001205  
013776 010046  
014000 104403  
014002 001  
014003 000  
014004 005721  
014006 005200  
014010 022700 000010  
014014 001402  
014016 006002  
014020 000756  
014022 005737 007460  
014026 001402  
014030 000137 014764  
014034 104075  
014036 000000  
014040 000137 013340  
014044 012765 000040 000010  
014052 013737 001406 007376  
014060 004737 032230  
014064 104120  
014066 005737 007510  
014072 001562  
014074 104401 044636

```
;* TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
;* MDS LIFE OR NED BEING SET: OR BOTH NED & DRA RESET IN-
;* DICATING THE OTHER PORT IS ACCESSED.
*****
TST: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
INC BYPCERR ;DO NOT TEST CERR IN 'FRDY'
BITB #BIT7,SENVN ;SEE IF USE APT SELECTED DRIVES
BNE 14$ ;BR IF YES
JMP 12$ ;ELSE DO NORM SIZING OR VERIFY
14$: TYPE ,MSG10 ;WILL TEST DRIVES
CLR DRIVS ;# OF DRIVES PRESENT
CLR RO ;DRV ADDR
MOV #DRIVO,R1 ;DRV FLAG
MOV $DEVN,R2 ;APT DEVICE MAP
15$: BIT #BIT0,R2 ;SEE IF DRV IN DEVICE MAP
BEQ 16$ ;BR IF NO
INC DRIVS ;ELSE INCR DRIVE COUNT
INC (R1) ;& SET DRIVE PRESENT FLAG
TYPE ,$CRLF
MOV RO,-(SP) ;;SAVE RO FOR TYPEOUT
;;TYPE DRIVE #
;;GO TYPE--OCTAL ASCII
;;TYPE 1 DIGIT(S)
;;SUPPRESS LEADING ZEROS
16$: TST (R1)+ ;ADV POINTER TO NEXT FLAG
INC RO ;INC DRIVE #
CMP #8,,RO ;ALL 8 TESTED?
BEQ 17$ ;BR IF YES
ROR R2 ;ELSE GET NEXT BIT OFF DEVICE MAP
BR 15$ ;& TRY AGAIN
17$: TST DRIVS ;SEE IF MORE DRIVES PRESENT
BEQ 18$ ;BR IF NO
JMP NUDRV ;ELSE EXIT TEST
18$: ERROR 75 ;NO DRIVES FOUND IN $DEVN
HALT ;SETUP CORRECTLY & PRESS 'CONTINUE'
JMP STS ;TO TRY AGAIN
12$: MOV #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
MOV T10,TEMP1 ;SET TIMEOUT
JSR PC,FRDY ;FIND RDY
ERROR 120 ;RDY NOT SET BY END OF SCLR
TST SIZFLG
BEQ TST3 ;DO NOT SIZE, GOTO NEXT TEST
TYPE ,MSG10 ;WILL TEST DRIVES
```

```

3455 014100 005037 007460          CLR    DRIVS          ;# OF DRIVES PRESENT
3456 014104 005000                   CLR    RO             ;DRV ADDR
3457 014106 012701 007462          MOV    #DRIVO,R1     ;DRV FLAG
3458 014112                   15:
3459 014112 104415                   SCOPI
3460 014114 012706 001100          MOV    #STACK,SP     ;RESTORE STK PTR
3461
3462 014120 012765 000040 000010    MOV    #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
3463 014126 013737 001406 007376    MOV    T10,TEMP1     ;SET TIMEOUT
3464 014134 004737 032230          JSR    PC,FRDY       ;FIND RDY
3465 014140 104120                   ERROR 120            ;RDY NOT SET BY END OF SCLR
3466 014142 010065 000010          MOV    RO,RKCS2(R5)  ;SELECT THE DRIVE ADDR
3467 014146 012765 000001 000000    MOV    #SELDRV,RKCS1(R5) ;SELECT DRIVE CMD
3468 014154 013737 001406 007376    MOV    T10,TEMP1
3469 014162 004737 032230          JSR    PC,FRDY       ;FIND RDY
3470 014166 104117                   ERROR 117            ;NO RDY AFTER SELECT DRIVE CMD.
3471 014170 032737 100000 007340    BIT    #CERR,HCS1
3472 014176 001046                   BNE   2$
3473 014200 013737 007366 007376    MOV    HMR2,TEMP1
3474 014206 042737 177770 007376    BIC    #1C<DRVMSK>,TEMP1
3475 014214 020037 007376          CMP    RO,TEMP1      ;S/B SAME
3476 014220 001016                   BNE   3$
3477 014222 005700                   TST   RO
3478 014224 001003                   BNE   4$
3479 014226 005737 007452          TST   DDPCH          ;SEE IF XXDP CHAIN MODE
3480 014232 001014                   BNE   5$
3481 014234 005237 007460          4$: INC    DRIVS          ;INC DRIVE COUNT.
3482 014240 005211                   INC    (R1)           ;SET DRIVE PRESENT FLAG
3483 014242 104401 001205          TYPE  $SCLF
3484 014246 010046                   MOV    RO,-(SP)      ;;SAVE RO FOR TYPEOUT
3485                                     ;;TYPE DR #
3486 014250 104403                   TYPOS                ;;GO TYPE--OCTAL ASCII
3487 014252 001                                     .BYTE 1              ;;TYPE 1 DIGIT(S)
3488 014253 000                                     .BYTE 0              ;;SUPPRESS LEADING ZEROS
3489 014254 000403                   BR    5$
3490
3491 014256 004737 032736          3$: JSR    PC,BYP       ;TYPE BYPASS DR #
3492 014262 104001                   ERROR 1              ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #
3493
3494 014264 005721                   5$: TST   (R1)+         ;SHIFT PTR TO NEXT DR. FLAG
3495 014266 005200                   INC   RO             ;INC DR #
3496 014270 022700 000010          CMP   #8.,RO        ;
3497 014274 001306                   BNE   1$             ;MORE LEFT.
3498 014276 005737 007460          TST   DRIVS
3499 014302 001054                   BNE   10$
3500 014304 104076                   ERROR 76            ;NO DRIVES FOUND ON BUSS
3501 014306 000000                   HALT
3502 014310 000137 013340          JMP   STS           ;SETUP CORRECTLY
3503                                     ;AND PRESS 'CONTINUE'
3504 014314 032737 001000 007342 2$: BIT    #MDS,HCS2
3505 014322 001015                   BNE   6$
3506 014324 032737 000400 007342    BIT    #LFE,HCS2
3507 014332 001015                   BNE   7$
3508 014334 032737 000001 007352    BIT    #DRA,HDS
3509 014342 001015                   BNE   8$
3510 014344 032737 010000 007342    BIT    #NED,HCS2

```

```

3511 014352 001424          BEQ    9$
3512 014354 000743          BR     5$
3513
3514 014356 004737 032736 6$:   JSR    PC,BYP      ;TYPE BYP DR #
3515 014362 104002          ERROR  2          ;MDS DETECTED
3516 014364 000737          BR     5$
3517
3518 014366 004737 032736 7$:   JSR    PC,BYP
3519 014372 104003          ERROR  3          ;UFE DETECTED
3520 014374 000733          BR     5$
3521
3522 014376 032737 010000 007342 8$:  BIT    #NED,HCS2
3523 014404 001713          BEQ    4$
3524 014406 104401 045211          TYPE  MSG15      ;DRV#
3525 014412 010046          MOV    RO,-(SP)   ;:SAVE RO FOR TYPEOUT
3526
3527 014414 104403          TYPOS
3528 014416 001          .BYTE  1          ;:TYPE DR#
3529 014417 000          .BYTE  0          ;:GO TYPE--OCTAL ASCII
3530 014420 104010          ERROR  10         ;:TYPE 1 DIGIT(S)
3531 014422 000720          BR     5$         ;:SUPPRESS LEADING ZEROS
3532
3533 014424 004737 032736 9$:   JSR    PC,BYP
3534 014430 104004          ERROR  4          ;NO DRA & NO NED = OTHER PORT SELECTED
3535 014432 000714          BR     5$
3536 014434 000137 014764 10$:  JMP    NUDRV
3537

```

```

:*****
: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.
:*****

```

```

3553 014440 000004          ST3:  SCOPE
3554 014442 012737 000001 001174  MOV    #1,STIMES      ;:DO 1 ITERATION
3555 014450 012706 001100          MOV    #STACK,SP     ;:RESTORE STK PTR
3556 014454 005000          CLR    RO             ;:DRIVE ADDR
3557 014456 012701 007462          MOV    #DRIVO,R1     ;:DRIVE FLAG
3558 014462
3559 014462 104415          SCOPE
3560 014464 012706 001100          MOV    #STACK,SP     ;:RESTORE STK PTR
3561
3562 014470 012765 000040 000010  MOV    #SCLR,RKCS2(R5) ;:SUBSYSTEM CLEAR
3563 014476 013737 001406 007376  MOV    T10,TEMP1     ;:SET TIME OUT
3564 014504 004737 032230          JSR    PC,FRDY       ;:FIND RDY
3565 014510 104120          ERROR  120          ;:NO RDY AFTER SCLR
3566 014512 010065 000010          MOV    RO,RKCS2(R5) ;:DRV ADDR

```



F06

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 69  
T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 0069

3567	014516	012765	000001	000000		MOV	#SELDV,RKCS1(R5)	;SELECT DRIVE CMD
3568	014524	013737	001406	007376		MOV	T10,TEMP1	
3569	014532	004737	032230			JSR	PC,FRDY	;FIND RDY
3570	014536	104117				ERROR	117	;NO RDY AFTER SELECT DRIVE CMD.
3571	014540	032737	100000	007340		BIT	#CERR,HCS1	
3572	014546	001036				BNE	2\$	
3573	014550	013737	007366	007376		MOV	HMR2,TEMP1	
3574	014556	042737	177770	007376		BIC	#1C(DRVMSK),TEMP1	
3575	014564	020037	007376			CMP	RO,TEMP1	;S/B SAME
3576	014570	001010				BNE	3\$	
3577	014572	005711			11\$:	TST	(R1)	
3578	014574	001417				BEQ	5\$	
3579	014576	005721			4\$:	TST	(R1)+	;SHIFT PTR TO NEXT DR FLAG
3580	014600	005200				INC	RO	;INC DR#
3581	014602	022700	000010			CMP	#8.,RO	
3582	014606	001325				BNE	1\$	;MORE LEFT
3583	014610	000467				BR	TST4	;GO TO NEXT TEST
3584								
3585	014612	004737	032736		3\$:	JSR	PC,BYP	;TRY BYPASS DRIVE#
3586	014616	104001				ERROR	1	;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
3587	014620	005711				TST	(R1)	
3588	014622	001765				BEQ	4\$	;BRANCH IF NOT SPEC BY INPUT
3589	014624	005337	007460		12\$:	DEC	DRIVS	;DECREMENT TOTAL DRIVS
3590	014630	005011				CLR	(R1)	;CLEAR DRIVE FLAG
3591	014632	000761				BR	4\$	
3592								
3593	014634	004737	032736		5\$:	JSR	PC,BYP	
3594	014640	104005				ERROR	5	;DR PRESENT BUT NOT SPECIFIED BY OPERATOR
3595	014642	000755				BR	4\$	
3596								
3597	014644	032737	001000	007342	2\$:	BIT	#MDS,HCS2	
3598	014652	001027				BNE	6\$	
3599	014654	032737	000400	007342		BIT	#UFE,HCS2	
3600	014662	001027				BNE	7\$	
3601	014664	032737	000001	007352		BIT	#DRA,MDS	
3602	014672	001005				BNE	8\$	
3603	014674	032737	010000	007342		BIT	#NED,HCS2	
3604	014702	001423				BEQ	9\$	
3605	014704	000404				BR	10\$	
3606	014706	032737	010000	007342	8\$:	BIT	#NED,HCS2	
3607	014714	001726				BEQ	11\$	
3608	014716	005711			10\$:	TST	(R1)	
3609	014720	001726				BEQ	4\$	
3610								
3611	014722	004737	032736			JSR	PC,BYP	;TYPE BYPASS DRIVE#
3612	014726	104006				ERROR	6	
3613	014730	000735				BR	12\$	
3614								
3615	014732	004737	032736		6\$:	JSR	PC,BYP	;TYPE BYPASS DRIVE#
3616	014736	104002				ERROR	2	;MDS DETECTED
3617	014740	000762				BR	8\$	
3618								
3619	014742	004737	032736		7\$:	JSR	PC,BYP	
3620	014746	104003				ERROR	3	;UFE DETECTED
3621	014750	000756				BR	8\$	
3622								

3623 014752 004737 032736  
 3624 014756 104004  
 3625 014760 000752  
 3626  
 3627 014762 001237  
 3628  
 3629  
 3630  
 3631  
 3632  
 3633  
 3634  
 3635  
 3636  
 3637 014764 005037 001502  
 3638  
 3639  
 3640  
 3641  
 3642  
 3643  
 3644  
 3645  
 3646  
 3647  
 3648  
 3649 014770 000004  
 3650 014772 012737 000001 001174  
 3651 015000 012706 001100  
 3652 015004 012737 000004 001214  
 3653 015012 012737 000004 001102  
 3654  
 3655 015020 005737 007460  
 3656 015024 001004  
 3657 015026 104401 045355  
 3658 015032 000137 031550  
 3659  
 3660 015036 013701 001342  
 3661 015042 005737 001220  
 3662 015046 001402  
 3663 015050 005237 001222  
 3664 015054 005721  
 3665 015056 001774  
 3666 015060 005737 007452  
 3667 015064 001403  
 3668 015066 005737 001222  
 3669 015072 001766  
 3670 015074 010137 001342  
 3671 015100 104401 045211  
 3672 015104 013700 001222  
 3673 015110 010046  
 3674  
 3675 015112 104403  
 3676 015114 001  
 3677 015115 000  
 3678

```

9$: JSR PC,BYP ;DRA & NED RESET - OTHER PORT SELECTED
   ERROR 4
   BR 8$

   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
        ;ALLOW CHECKING 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,$TESTN
      MOV #STN-1,$STSTNM

      TST DRIVS ;:ANY DRIVES PRESENT?
      BNE 4$ ;:YES BRANCH
      TYPE MSG27 ;:ALL DRIVES TESTED
      JMP $EOP1 ;:NO, GO TO END

4$: MOV DRVPTR,R1 ;:ADDR OF NEXT DRIVE FLAG
   TST $DEVCT ;:IS FIRST DRIVE BEING CHECKED
   BEQ 2$ ;:YES, BRANCH
   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 $SUNIT ;:YES, IS IT DRIVE 0?
   BEQ 1$ ;:IF YES, DON'T TEST DR 0
3$: MOV R1,DRVPTR ;:STORE POINTER TO THE NEXT DR. FLAG
   TYPE MSG15 ;:"DRIVE"
   MOV $SUNIT,R0 ;:SAVE R0 FOR TYPEOUT
   MOV R0,-($P) ;:DRIVE #
                        ;:GO TYPE--OCTAL ASCII
                        ;:TYPE 1 DIGIT(S)
                        ;:SUPPRESS LEADING ZEROS

```

H06

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 71  
T4 FIND NEXT DRIVE TO BE TESTED

SEQ 0C71

3679 015116 104401 001205

TYPE ,SCLRF

3680  
3681 015122

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

\*\*\*\*\*

3688

↑ST5: SCOPE

3689 015122 000004  
3690 015124 012737 000001 001174

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

3691 015132 012706 001100

3692

3693 015136 005737 001216

TST \$PASS ;GO TO NEXT IF NOT FIRST PASS  
BNE TST6

3694 015142 001046

3695 015144 004737 034140

JSR PC,SUBCLR ;DO SUBSYS CLEAR  
ERROR 24 ;CERR AFTER SCLR

3696 015150 104024

3697

3698 015152 104401 045223

TYPE ,MSG16 ;DRIVE SERIAL NO.

3699 015156 012765 000003 000026

MOV #3,RKMR1(R5) ;SELECT BYTE 3

3700 015164 004737 033566

JSR PC,GSTAT ;GET STATUS

3701 015170 013701 007366

MOV #MR2,R1 ;GET SERIAL #

3702 015174 012704 042546

MOV #SOCTVL,R4 ;GET ADDR CHAR BUFF

3703 015200 010446

MOV R4,-(SP) ;STORE ON STACK FOR \$SUPRS

3704 015202 012703 000003

MOV #3,R3 ;SETUP CHAR COOUNT

3705 015206 006101

ROL R1 ;INITIALIZE BIT POSITIONS

3706 015210 006101

ROL R1 ;GET NEXT 4 BITS

3707 015212 006101

1\$: ROL R1

3708 015214 006101

ROL R1

3709 015216 006101

ROL R1

3710 015220 006101

ROL R1 ;GET WORKING COPY

3711 015222 010100

MOV R1,R0 ;CLEAR ALL BUT LOW 4 BITS

3712 015224 042700 177760

BIC #177760,R0 ;CONVERT TO ASCII DIGIT

3713 015230 052700 000060

BIS #60,R0 ;PUT ASCII DIGIT INTO CHAR BUFF

3714 015234 110024

MOVB R0,(R4)+

3715 015236 005303

DEC R3 ;BR IF ALL 3 CHARS NOT DONE

3716 015240 001364

BNE 1\$ ;ELSE INSERT NULL TERMINATOR

3717 015242 105014

CLRB (R4)

3718

3719 015244 004737 043014

JSR PC,\$SUPRS ;TYPE

3720 015250 104401 001205

TYPE ,SCLRF

3721 015254 104401 001205

TYPE ,SCLRF

3722

\*\*\*\*\*

3723

\*TEST 6 SET VV WITH PACK COMMAND

3724

3725

3726

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

3727

\*\*\*\*\*

3728

↑ST6: SCOPE

3729 015260 000004

MOV #1,STIMES ;DO 1 ITERATION

3730 015262 012737 000001 001174

MOV #STACK,SP ;RESTORE STK PTR

3731 015270 012706 001100

3732

3733 015274 004737 034140

JSR PC,SUBCLR ;CERR AFTER SCLR

3734 015300 104024

ERROR 24

```

3735
3736 015302 032737 000100 007366 BIT #D.VV,HMR2
3737 015310 001024 BNE TST7 ;;GO T NEXT TEST IF VV SET
3738
3739 015312 104415 SCOPI
3740 015314 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
3741
3742 015320 004737 034140 JSR PC,SUBCLR
3743 015324 104024 ERROR 24 ;CERR AFTER SCLR
3744
3745 015326 012765 000003 000000 MOV #PACK,RKCS1(R5) ;CMD TO SET VV
3746 015334 012737 000010 007376 MOV #10,TEMP1
3747 015342 004737 032230 JSR PC,FRDY ;FIND RDY
3748 015346 104116 ERROR 116 ;RDY NOT SET AFTER PACK CMD
3749
3750 015350 032737 000100 007366 BIT #D.VV,HMR2
3751 015356 001001 BNE TST7 ;;GO TO NEXT TEST IF VV NOW SET
3752 015360 104027 ERROR 27 ;PACK DID NOT SET V.V.
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779 015362 000004
3780 015364 012737 000001 001174
3781 015372 012706 001100
3782
3783 015376 004737 034140 JSR PC,SUBCLR
3784 015402 104024 ERROR 24 ;CERR AFTER SCLR
3785
3786
3787 015404 012765 100000 000000 MOV #CCLR,RKCS1(R5)
3788 015412 013765 001222 000010 MOV $UNIT,RKCS2(R5)
3789 015420 012765 000013 000000 MOV #RECAL,RKCS1(R5) ;RECAL CMD
3790 ;RESET CYL DIFF/OFFSET & CYL ADDR REG

```

```

*****
*TEST 7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL *
*
* THIS TEST VERIFIES THAT CYL 410, TRACK 2 CAN BE READ.
* THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
* FACTORY DURING MANF. ALL BAD SECTOR INFO (BSE) WILL BE STORED
* AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
*
* SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
* SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED
*
* SECTORS 1,3,5,7,9 CONTAIN IDENTICAL INFO FOR 20 SECTOR HARDWARE DETECTED BAD SEC
* SECTORS 11,13,15,17,19,21 CONTAIN IDENTICAL INFO FOR 20 SECTOR SOFTWARE DETECTED
*
* IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
* IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
* A MESSAGE WILL BE TYPED INDICATING THAT ALL
* FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
* THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING
*
* THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.

```

```

*****
TST7: SCOPE
MOV #1,STIMES ;;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
MOV #CCLR,RKCS1(R5)
MOV $UNIT,RKCS2(R5)
MOV #RECAL,RKCS1(R5) ;RECAL CMD
;RESET CYL DIFF/OFFSET & CYL ADDR REG

```

J06

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR610.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 73  
T7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

SEQ 0073

```

3791
3792 015426 013737 001406 007376 MOV T10,TEMP1 ;IN RKMR2 & RKMR3 RESP.
3793 015434 004737 032230 JSR PC,FRDY ;SETUP TIMEOUT
3794 015440 104124 ERROR 124 ;FIND RDY
;RDY NOT SET AFTER RECAL CMD
3795
3796 015442 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
3797 015450 004737 033566 JSR PC,GSTAT
3798 015454 032737 020000 007366 BIT #D,RTZ,HMR2
3799 015462 001001 BNE 64$
3800 015464 104244 ERROR 244 ;RTZ NOT SET DURING RECAL CMD
3801 015466 013737 001406 007400 64$: MOV T10,TEMP2 ;SETUP TIMEOUT
3802 015474 004737 032544 JSR PC,FATT1 ;FIND ATTN
3803 015500 104055 ERROR 55 ;NO ATTN AFTER RECAL CMD
3804
3805 015502 012765 100000 000000 MOV #CCLR,RKCS1(R5)
3806 015510 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
3807 015516 012765 000005 000000 MOV #CLEAR,RKCS1(R5) ;DRIVE CLEAR CMD
3808 015524 013737 001406 007376 MOV T10,TEMP1 ;SETUP TIMEOUT
3809 015532 004737 032230 JSR PC,FRDY ;FIND RDY
3810 015536 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
3811 015540 004737 032512 JSR PC,TSTATN ;TEST FOR ATTN
3812 015544 000401 BR 65$
3813 015546 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
3814 015550 65$:
3815
3816
3817
3818 015550 004737 034140 JSR PC,SUBCLR
3819 015554 104024 ERROR 24 ;CERR AFTER SCLR
3820
3821 015556 005037 007400 CLR TEMP2 ;SECTOR CTR
3822 015562 005037 007402 CLR TEMP3 ;0=22 SECTOR HARDWARE DETECTED TABLE
;1=22 SECTOR SOFTWARE DETECTED TABLE
;2=20 SECTOR HARDWARE DETECTED TABLE
;3=20 SECTOR SOFTWARE DETECTED TABLE
3823 ;STORE 22 SECTOR HARDWARE BSE ADDR.
3824
3825
3826 015566 012737 003322 007404 MOV #BSE22H,TEMP4
3827 015574 013765 007404 000004 MOV TEMP4,RKBA(R5)
3828 015602 012737 001000 007406 MOV #1000,TEMP5 ;TRACK 2, SECTOR 0
3829 015610 013765 007406 000006 MOV TEMPS,RKDA(R5)
3830
3831 015616 012765 000632 000020 15: MOV #410.,RKDC(R5) ;CYL 410
3832 015624 012765 177400 000002 MOV #-256.,RKWC(R5) ;LOAD WORD CT
3833 015632 012765 000021 000000 MOV #RDDATA,RKCS1(R5) ;READ DATA COMMAND
3834 015640 013737 001420 007376 MOV T5000,TEMP1 ;SETUP TIMEOUT
3835 015646 004737 032230 JSR PC,FRDY ;FIND RDY
3836 015652 104226 ERROR 226 ;NO RDY AFTER READ DATA CMD
3837 015654 004737 033566 JSR PC,GSTAT ;GET FRESH DATA
3838 015660 032737 100000 007340 BIT #CERR,HCS1
3839 015666 001504 BEQ 8$
3840 015670 104227 ERROR 227 ;CERR AFTER READ DATA CMD
3841
3842 015672 012737 010340 007430 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
3843 015700 005037 007432 CLR E.B0 ;EXPECTED MSG B0
3844 015704 012737 001720 007434 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
3845 015712 012737 000001 007436 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
3846 015720 005037 007440 CLR E.A2 ;EXPECTED MSG A2

```

```

3847 015724 012737 000002 007442 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
3848 015732 012737 000003 007446 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
3849
3850 015740 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
3851 015744 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
3852 015746 104054 ERROR 5 ;MSG A0 ERROR AFTER READ DATA CMD
3853 015750 104026 ERROR 2 ;MSG B0 ERROR
3854 015752 104056 ERROR 56 ;MSG A1 ERROR
3855 015754 104030 ERROR 30 ;MSG B1 ERROR
3856
3857 015756 004737 034140 JSR PC,SUBCLR
3858 015762 104024 ERROR 24 ;CERR AFTER SUBCLR
3859
3860 015764 005237 007400 INC TEMP2
3861 015770 023727 007400 000005 CMP TEMP2,#5 ;READ ALL 5 SECTORS?
3862 015776 001023 BNE 5$
3863 016000 005737 007402 TST TEMP3
3864 016004 001002 BNE 2$
3865 016006 104233 ERROR 233 ;CANT READ SECTORS 0,2,4,6,8
3866 016010 000430 BR 3$
3867
3868 016012 023727 007402 000001 2$: CMP TEMP3,#1
3869 016020 001002 BNE 4$
3870 016022 104230 ERROR 230 ;CANT READ SECTORS 10,12...
3871 016024 000422 BR 3$
3872
3873 016026 023727 007402 000002 4$: CMP TEMP3,#2
3874 016034 001002 BNE 6$
3875 016036 104234 ERROR 234 ;CANT READ SECTORS 1,3,5 ...
3876 016040 000414 BR 3$
3877
3878 016042 104231 6$: ERROR 231 ;CANT READ SECTORS 11,13,15 ...
3879 016044 000412 BR 3$
3880
3881 016046 013765 007404 000004 5$: MOV TEMP4,RKBA(R5) ;RESTORE TABLE ADDR
3882 016054 062737 000002 007406 ADD #2,TEMP5 ;READ 2 SECTORS FROM LAST
3883 016062 013765 007406 000006 MOV TEMP5,RKDA(R5)
3884 016070 000652 BR 1$
3885
3886 016072 005237 001476 3$: INC BSERR ;SET BSE FLAG
3887 016076 000553 BR TST10 ;GO TO NEXT TEST
3888 016100 005737 003330 8$: TST BSE22H+6 ;TEST CARTRIDGE TYPE
3889 016104 001404 BEQ 9$ ;BRANCH IF DATA CARTRIDGE
3890 016106 104235 ERROR 235 ;ALIGNMENT CARTRIDGE USED
3891 016110 005237 001476 INC BSERR ;SET BSE ERROR FLAG
3892 016114 000476 BR 10$
3893
3894 016116 005237 007402 9$: INC TEMP3
3895 016122 023727 007402 000001 CMP TEMP3,#1
3896 016130 001020 BNE 11$
3897 016132 005037 007400 CLR TEMP2 ;SECTOR CTR
3898 016136 012737 005322 007404 MOV #BSE22S,TEMP4 ;STORE 22 SECTOR SOFTWARE BSE ADDR
3899 016144 013765 007404 000004 MOV TEMP4,RKBA(R5)
3900 016152 012737 001012 007406 MOV #1012,TEMP5 ;TRACK 2, SECTOR 12(8)
3901 016160 013765 007406 000006 MOV TEMP5,RKDA(R5)
3902 016166 000137 015616 JMP 1$ ;REPEAT

```

```

3903
3904 016172 023727 007402 000002 11$:  CMP      TEMP3,#2
3905 016200 001020          BNE      12$
3906 016202 005037 007400          CLR      TEMP2          ;SECTOR CTR
3907 016206 012737 002322 007404          MOV      #BSE20H,TEMP4 ;STORE 20 SECTOR HARDWARE BSE ADDR.
3908 016214 013765 007404 000004          MOV      TEMP4,RKBA(R5)
3909 016222 012737 001001 007406          MOV      #1001,TEMP5   ;TRACK 2, SECTOR 1
3910 016230 013765 007406 000006          MOV      TEMP5,RKDA(R5)
3911 016236 000137 015616          JMP      1$            ;REPEAT
3912
3913 016242 023727 007402 000003 12$:  CMP      TEMP3,#3
3914 016250 001020          BNE      10$
3915 016252 005037 007400          CLR      TEMP2          ;SECTOR CTR
3916 016256 012737 004322 007404          MOV      #BSE20S,TEMP4 ;STORE 20 SECTOR SOFTWARE BSE ADDR
3917 016264 013765 007404 000004          MOV      TEMP4,RKBA(R5)
3918 016272 012737 001013 007406          MOV      #1013,TEMP5   ;TRACK 2, SECTOR 13(8)
3919 016300 013765 007406 000006          MOV      TEMP5,RKDA(R5)
3920 016306 000137 015616          JMP      1$            ;REPEAT
3921
3922 016312 005737 001216          10$:  TST      $PASS
3923 016316 001043          BNE      TST10          ;;GO TO NEXT TST IF NOT 1'ST PASS
3924 016320 104401 045247          TYPE    ,MSG17         ;;CART SERIAL #
3925 016324 012746 003322          MOV      #BSE22H,-(SP)
3926 016330 004737 042444          JSR      PC,$DB20      ;CONVERT DBL BINARY WORD TO OCTAL
3927 016334 004737 043014          JSR      PC,$SUPRS     ;TYPE SERIAL #
3928 016340 104401 001205          TYPE    ,SCLRF
3929 016344 104401 001205          TYPE    ,SCLRF
3930
3931 016350 004737 034140          JSR      PC,SUBCLR
3932 016354 104024          ERROR   24            ;CERR AFTER SCLR
3933                                     ;GO BACK TO CYL 0
3934
3935 016356 012765 000017 000000          MOV      #SEEK,RKCS1(R5);SEEK CMD
3936 016364 013737 0014'  007376          MOV      T50,TEMP1    ;SETUP TIMEOUT
3937 016372 004737 032230          JSR      PC,FRDY      ;FIND RDY
3938 016376 104131          ERROR   131          ;NO RDY AFTER SEEK CMD
3939
3940 016400 013737 001420 007376          MOV      T50000,TEMP1 ;SETUP TIMEOUT
3941 016406 004737 032640          JSR      PC,FATT2     ;FIND ATTN
3942 016412 104132          ERROR   132          ;NO ATTN AFTER SEEK CMD
3943
3944 016414 032737 100000 007340          BIT      #CERR,HCS1
3945 016422 001401          BEQ     66$
3946 016424 104210          ERROR   210          ;CERR AFTER SEEK CMD
3947
3948 016426          66$:
3949
3950
3951
3952 .SBTTL WRITE TESTS
3953
3954
3955 ;*****
3956 ;*TEST 10 BASIC WRITE DATA TEST; 1 WORD
3957 ;*
3958 ;* THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,

```

M06

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 76  
T10 BASIC WRITE DATA TEST; 1 WORD

SEQ 0076

```

3959
3960
3961
3962
3963
3964
3965 016426 000004
3966 016430 012737 000001 001174
3967 016436 012706 001100
3968
3969 016442 005737 001476
3970 016446 001406
3971 016450 104401 045605
3972 016454 104401 045277
3973 016460 000137 031514
3974
3975 016464 004737 034140
3976 016470 104024
3977
3978 016472 005237 007324
3979
3980 016476 012700 001506
3981
3982 016502 005020
3983 016504 012720 140000
3984 016510 012720 140000
3985
3986 016514 020027 001712
3987 016520 001370
3988
3989 016522 012765 001506 000004
3990 016530 012765 177676 000002
3991
3992 016536 012765 000027 000000
3993 016544 013737 001420 007376
3994 016552 004737 032230
3995 016556 104200
3996 016560 004737 033566
3997 016564 032737 100000 007340
3998 016572 001405
3999 016574 104201
4000 016576 104401 045277
4001 016602 000137 031514
4002 016606
4003
4004
4005 016606 104415
4006 016610 012706 001100
4007
4008 016614 004737 034140
4009 016620 104024
4010
4011 016622 005037 001402
4012 016626 013765 001402 000006
4013 016634 012765 001464 000004
4014 016642 012765 177777 000002

```

```

; * ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &
; * A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT
; * PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP
; * FOR A WRITE ERROR.
; *
; * *****
; * ST10: SCOPE
; * MOV #1,STIMES ;DO 1 ITERATION
; * MOV #STACK,SP ;RESTORE STK PTR
; *
; * TST BSERR ;SEE IF ALIGN CART
; * BEQ 2$ ;BR IF NO
; * TYPE ,MSG40 ;BSE OR ALIGN CART USED
; * TYPE ,MSG26 ;ABORTING BAL OF TESTS
; * JMP $EOP
; *
2$: JSR PC,SUBCLR
; * ERROR 24 ;CERR AFTER SCLR
; *
; * INC BADHDR ;USED FOR VALID HALT
; *
; * MOV #HDTAB,RO ;MAKE ALL CYL 0 HEADERS IDENTICAL
; *
1$: CLR (RO)+ ;HEADER WORD 0: CYL 0
; * MOV #140000,(RO)+ ;HEADER WORD 1: SECTOR 0
; * MOV #140000,(RO)+ ;HEADER WORD 2: XOR OF 1 & 2
; *
; * CMP RO,#HDTAB+132. ;ALL HEADERS DONE? (22X6=132)
; * BNE 1$ ;BR IF NO
; *
; * MOV #HDTAB,RKBA(R5) ;HEADER TABLE
; * MOV #-66.,RKWC(R5) ;WORD COUNT
; *
; * MOV #<WRHEAD>,RKCS1(R5) ;WRITE HEADER CMD
; * MOV T50000,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$
; * ERROR 201 ;CERR AFTER WRITE HEADER CMD
; * TYPE ,MSG26 ;ABORTING BAL OF TESTS
; * JMP $EOP
; *
64$:
; *
; * SCOP1
; * MOV #STACK,SP ;RESTORE STK PTR
; *
; * JSR PC,SUBCLR
; * ERROR 24 ;CERR AFTER SCLR
; *
; * CLR SECTOR
; * MOV SECTOR,RKDA(R5) ;TRACK/SECTOR #
; * MOV #DATA1,RKBA(R5) ;DATA TO BE ALL 1'S
; * MOV #-1,RKWC(R5) ;WORD COUNT=1

```



N06

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR610.P11 21-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 77  
T10 BASIC WRITE DATA TEST; 1 WORD

SEQ 0077

4015									
4016									
4017	016650	012765	000023	000000		MOV	#(WRDATA),RKCS1(R5)	;WRITE DATA CMD	
4018	016656	013737	001420	007376		MOV	T50000,TEMP1	;SETUP TIMEOUT	
4019	016664	004737	032230			JSR	PC,FRDY	;FIND RDY	
4020	016670	104011				ERROR	11	;NO RDY AFTER WRITE DATA CMD	
4021	016672	004737	033566			JSR	PC,GSTAT	;GET FRESH STATUS	
4022	016676	032737	100000	007340		BIT	#CERR,HCS1		
4023	016704	001465				BEQ	66\$	;BR IF NO ERRORS	
4024									
4025	016706	032737	000200	007354		BIT	#BSE,HER	;SEE IF BAD SECTOR FLAG	
4026	016714	001421				BEQ	66\$	;BR IF NO	
4027	016716	004737	035610			JSR	PC,TRUERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE	
4028	016722	000455				BR	67\$	;RETURN HERE IF NO	
4029									
4030	016724	005237	001402			INC	SECTOR	;RETURN HERE IF YES	
4031	016730	023727	001402	000012		CMP	SECTOR,#10.	;ARE 10 CONSEC. SECTORS BAD	
4032	016736	001003				BNE	65\$	;BR IF NO	
4033	016740	104046				ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS	
4034	016742	000137	017144			JMP	5\$	;BYPASS TEST	
4035	016746	012765	100000	000000	65\$:	MOV	#CCLR,RKCS1(R5)	;TRY ANOTHER SECTOR	
4036	016754	000137	016626			JMP	3\$		
4037	016760	104012			66\$:	ERROR	12	;CERR WITH WRITE DATA CMD	
4038									
4039	016762	012737	010340	007430		MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0	
4040	016770	005037	007432			CLR	E.B0	;EXPECTED MSG B0	
4041	016774	012737	001720	007434		MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1	
4042	017002	012737	000001	007436		MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1	
4043	017010	005037	007440			CLR	E.A2	;EXPECTED MSG A2	
4044	017014	012737	000002	007442		MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2	
4045	017022	012737	000003	007446		MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3	
4046									
4047	017030	004737	032752			JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1	
4048	017034	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE	
4049	017036	104052				ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD	
4050	017040	104023				ERROR	23	;MSG B0 ERROR	
4051	017042	104053				ERROR	53	;MSG A1 ERROR	
4052	017044	104025				ERROR	25	;MSG B1 ERROR	
4053	017046	104401	045277			TYPE	MSG26	;ABORTING BAL OF TESTS	
4054	017052	000137	031514			JMP	\$EOP		
4055	017056	104063			67\$:	ERROR	63	;BAD SECTOR NOT LISTED IN TABLE	
4056	017060				68\$:				
4057									
4058	017060	012737	010340	007430		MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0	
4059	017066	005037	007432			CLR	E.B0	;EXPECTED MSG B0	
4060	017072	012737	001720	007434		MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1	
4061	017100	012737	000001	007436		MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1	
4062	017106	005037	007440			CLR	E.A2	;EXPECTED MSG A2	
4063	017112	012737	000002	007442		MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2	
4064	017120	012737	000003	007446		MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3	
4065									
4066	017126	004737	032752			JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1	
4067	017132	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE	
4068	017134	104052				ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD	
4069	017136	104023				ERROR	23	;MSG B0 ERROR	
4070	017140	104053				ERROR	53	;MSG A1 ERROR	

```

4071 017142 104025
4072 017144
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083 017144 000004
4084 017146 012737 000001 001174
4085 017154 012706 001100
4086
4087 017160 004737 034140
4088 017164 104024
4089
4090 017166 012765 001506 000004
4091 017174 012765 177676 000002
4092 017202 005037 001346
4093
4094 017206 013737 001346 001362
4095 017214 012737 000000 001444
4096 017222 012737 000000 001452
4097 017230 004737 035122
4098
4099
4100 017234 012765 000027 000000
4101 017242 013737 001420 007376
4102 017250 004737 032230
4103 017254 104200
4104 017256 004737 033566
4105 017262 032737 100000 007340
4106 017270 001405
4107 017272 104201
4108 017274 104401 045277
4109 017300 000137 031514
4110 017304
4111
4112 017304 005037 007324
4113 017310 104415
4114 017312 012706 001100
4115
4116 017316 004737 034140
4117 017322 104024
4118
4119 017324 005037 001402
4120 017330 013765 001402 000006
4121 017336 012765 001460 000004
4122 017344 013700 001460
4123 017350 052765 000020 000010
4124 017356 012765 177400 000002
4125
4126 017364 012765 000023 000000

```

```

ERROR 25 ;MSG B1 ERROR
SS:
*****
*TEST 11 BASIC WRITE DATA TEST; FULL SECTOR
*
* THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
* A FULL SECTOR. ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
* & CHECKED BY A RD DATA COMMAND. A FURTHER CHECK IS PERFORMED
* BY THE WRT CHK COMMAND.
* THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.
*****
↑ST11: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
MOV #HDTAB,RKBA(R5) ;RESTORE TO 22 SECTOR
MOV #-66.,RKWC(R5) ;STANDARD FORMAT
CLR TOCYL
MOV TOCYL,CALADD ;SETUP
MOV #0,HEAD ;TO FILL
MOV #0,FORMAT ;HEADER
JSR PC,FHDTAB ;TABLE
MOV #<WRHEAD>,RKCS1(R5) ;WRITE HEADER CMD
MOV T5000,TEMP1 ;SETUP TIMEOUT
JSR PC,FROY ;FIND ROY
ERROR 200 ;NO ROY AFTER WRITE HEADER CMD
JSR PC,GSTAT ;GET FRESH STATUS
BIT #CERR,HCS1
BEQ 64$
ERROR 201 ;CERR AFTER WRITE HEADER CMD
TYPE MSG26 ;ABORTING BAL OF TESTS
JMP $EOP
64$:
CLR BADHDR ;USED FOR VALID HALT
SCOPI
MOV #STACK,SP ;RESTORE STK PTR
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR
CLR SECTOR
MOV SECTOR,RKDA(R5) ;SETUP SECTOR
MOV #DATA0,RKBA(R5) ;WRITE ALL 0'S
MOV DATA0,R0
BIS #BAI,RKCS2(R5)
MOV #-256.,RKWC(R5)
MOV #<WRDATA>,RKCS1(R5) ;WRITE DATA CMD

```

4127	017372	013737	001420	007376	MOV	T50000,TEMP1	;SETUP TIMEOUT
4128	017400	004737	032230		JSR	PC,FRDY	;FIND ROY
4129	017404	104011			ERROR	11	;NO ROY AFTER WRITE DATA CMD
4130	017406	004737	033566		JSR	PC,GSTAT	;GET FRESH STATUS
4131	017412	032737	100000	007340	BIT	#CERR,HCS1	
4132	017420	001465			BEQ	6P\$	;BR IF NO ERRORS
4133							
4134	017422	032737	000200	007354	BIT	#BSE,MER	;SEE IF BAD SECTOR FLAG
4135	017430	001421			BEQ	66\$	;BR IF NO
4136	017432	004737	035610		JSR	PC,TRUERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4137	017436	000455			BR	67\$	;RETURN HERE IF NO
4138							
4139	017440	005237	001402		INC	SECTOR	;RETURN HERE IF YES
4140	017444	023727	001402	000012	CMP	SECTOR,#10.	;ARE 10 CONSEC. SECTORS BAD
4141	017452	001003			BNE	65\$	;BR IF NO
4142	017454	104046			ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS
4143	017456	000137	020752		JMP	7\$	;BYPASS TEST
4144	017462	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	;TRY ANOTHER SECTOR
4145	017470	000137	017330		JMP	8\$	
4146	017474	104012			ERROR	12	;CERR WITH WRITE DATA CMD
4147							
4148	017476	012737	010340	007430	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4149	017504	005037	007432		CLR	E.B0	;EXPECTED MSG B0
4150	017510	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4151	017516	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4152	017524	005037	007440		CLR	E.A2	;EXPECTED MSG A2
4153	017530	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4154	017536	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4155							
4156	017544	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4157	017550	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4158	017552	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
4159	017554	104023			ERROR	23	;MSG B0 ERROR
4160	017556	104053			ERROR	53	;MSG A1 ERROR
4161	017560	104025			ERROR	25	;MSG B1 ERROR
4162	017562	104401	045277		TYPE	MSG26	;ABORTING BAL OF TESTS
4163	017566	000137	031514		JMP	\$EOP	
4164	017572	104063			ERROR	63	;BAD SECTOR NOT LISTED IN TABLE
4165	017574						
4166							
4167	017574	012737	010340	007430	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4168	017602	005037	007432		CLR	E.B0	;EXPECTED MSG B0
4169	017606	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4170	017614	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4171	017622	005037	007440		CLR	E.A2	;EXPECTED MSG A2
4172	017626	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4173	017634	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4174							
4175	017642	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4176	017646	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4177	017650	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
4178	017652	104023			ERROR	23	;MSG B0 ERROR
4179	017654	104053			ERROR	53	;MSG A1 ERROR
4180	017656	104025			ERROR	25	;MSG B1 ERROR
4181	017660	104415			SCOP1		
4182	017662	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR

```

4183
4184 017666 004737 034140      JSR    PC,SUBCLR
4185 017672 104024      ERROR  24      ;CERR AFTER SCLR
4186
4187 017674 013765 001402 000006      MOV    SECTOR,RKDA(R5) ;SETUP SECTOR
4188 017702 012765 006322 000004      MOV    #ROTAB,RKBA(R5)
4189 017710 012765 177400 000002      MOV    #-256.,RKWC(R5)
4190
4191
4192 017716 012765 000021 000000      MOV    #<RDATA>,RKCS1(R5) ;READ DATA CMD
4193 017724 013737 001420 007376      MOV    T5000,TEMP1      ;SETUP TIMEOUT
4194 017732 004737 032230      JSR    PC,FRDY          ;FIND RDY
4195 017736 104013      ERROR  13          ;NO RDY AFTER READ DATA CMD
4196 017740 004737 033566      JSR    PC,GSTAT         ;GET FRESH STATUS
4197 017744 032737 100000 007340      BIT    #CERR,HCS1
4198 017752 001454      BEQ    72$
4199 017754 032737 000200 007354      BIT    #BSE,HER        ;SEE IF BAD SECTOR
4200 017762 001406      BEQ    70$
4201 017764 104065      ERROR  65          ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
4202 017766 000413      BR     73$
4203 017770 104401 045277      TYPE   MSG26          ;ABORTING BAL OF TESTS
4204 017774 000137 031514      JMP    $EOP
4205
4206 020000 032737 100000 007354 70$:  BIT    #DCK,HER        ;SEE IF DATA CHECK ERROR
4207 020006 001402      BEQ    71$
4208 020010 104021      ERROR  21          ;DATA CHECK ERROR AFTER READ CMD (ECC)
4209 020012 000401      BR     73$
4210
4211 020014 104014      71$:  ERROR  14          ;CERR AFTER READ DATA CMD.
4212
4213 020016      73$:
4214
4215 020016 012737 010340 007430      MOV    #<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4216 020024 000137 007432      CLR    E.B0          ;EXPECTED MSG B0
4217 020030 012737 001720 007434      MOV    #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4218 020036 012737 000001 007436      MOV    #1,E.B1       ;MSG ID FOR EXPECTED MSG B1
4219 020044 005037 007440      CLR    E.A2          ;EXPECTED MSG A2
4220 020050 012737 000002 007442      MOV    #2,E.B2       ;MSG ID FOR EXPECTED MSG B2
4221 020056 012737 000003 007446      MOV    #3,E.B3       ;MSG ID FOR EXPECTED MSG B3
4222
4223 020064 004737 032752      JSR    PC,CHKMSG       ;CHECK MSGS A0, B0, A1, B1
4224 020070 000003      .WORD T.A2!T.B2!0     ;& MSGS SPECIFIED HERE
4225 020072 104054      ERROR  54          ;MSG A0 ERROR AFTER READ DATA CMD
4226 020074 104026      ERROR  26          ;MSG B0 ERROR
4227 020076 104056      ERROR  56          ;MSG A1 ERROR
4228 020100 104030      ERROR  30          ;MSG B1 ERROR
4229 020102 000732      BR     69$
4230 020104      72$:
4231
4232 020104 012737 010340 007430      MOV    #<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4233 020112 005037 007432      CLR    E.B0          ;EXPECTED MSG B0
4234 020116 012737 001720 007434      MOV    #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4235 020124 012737 000001 007436      MOV    #1,E.B1       ;MSG ID FOR EXPECTED MSG B1
4236 020132 005037 007440      CLR    E.A2          ;EXPECTED MSG A2
4237 020136 012737 000002 007442      MOV    #2,E.B2       ;MSG ID FOR EXPECTED MSG B2
4238 020144 012737 000003 007446      MOV    #3,E.B3       ;MSG ID FOR EXPECTED MSG B3

```



# F07

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR610.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 82  
T11 BASIC WRITE DATA TEST; FULL SECTOR

SEQ 0082

```

4295 020410 012737 010340 007430  MOV  #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4296 020416 005037 007432  CLR  E.B0 ;EXPECTED MSG B0
4297 020422 012737 001720 007434  MOV  #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED,A1
4298 020430 012737 000001 007436  MOV  #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4299 020436 005037 007440  CLR  E.A2 ;EXPECTED MSG A2
4300 020442 012737 000002 007442  MOV  #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4301 020450 012737 000003 007446  MOV  #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4302
4303 020456 004737 032752  JSR  PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4304 020462 000003  .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
4305 020464 104057  ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
4306 020466 104031  ERROR 31 ;MSG B0 ERROR
4307 020470 104060  ERROR 60 ;MSG A1 ERROR
4308 020472 104032  ERROR 32 ;MSG B1 ERROR
4309 020474 104401 045277  TYPE MSG26 ;ABORTING BAL OF TESTS
4310 020500 000137 031514  JMP  $EOP
4311
4312 020504 75$:
4313
4314 020504 012737 010340 007430  MOV  #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4315 020512 005037 007432  CLR  E.B0 ;EXPECTED MSG B0
4316 020516 012737 001720 007434  MOV  #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4317 020524 012737 000001 007436  MOV  #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4318 020532 005037 007440  CLR  E.A2 ;EXPECTED MSG A2
4319 020536 012737 000002 007442  MOV  #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4320 020544 012737 000003 007446  MOV  #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4321
4322 020552 004737 032752  JSR  PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4323 020556 000003  .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
4324 020550 104057  ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
4325 020562 104031  ERROR 31 ;MSG B0 ERROR
4326 020564 104060  ERROR 60 ;MSG A1 ERROR
4327 020566 104032  ERROR 32 ;MSG B1 ERROR
4328
4329 020570 104415  SCOP1
4330 020572 012706 001100  MOV  #STACK,SP ;RESTORE STK PTR
4331
4332 020576 004737 034140  JSR  PC,SUBCLR
4333 020602 104024  ERROR 24 ;CERR AFTER SCLR
4334
4335 020604 012765 001460 000004  MOV  #DATA0,RKBA(R5) ;SETUP TO CHECK AGAINST WRONG DATA
4336 020612 012765 177400 000002  MOV  #-256,RKWC(R5)
4337 020620 013765 001402 000006  MOV  SECTOR,RKDA(R5)
4338 020626 012765 000031 000000  MOV  #WATCH,RKCS1(R5)
4339 020634 012737 141520 007376  MOV  #50000,TEMP1
4340 020642 004737 032230  JSR  PC,FRDY
4341 020646 104015  ERROR 15 ;NO RDY AFTER WRITE CHECK CMD
4342 020650 004737 033566  JSR  PC,GSTAT ;GET FRESH STATUS
4343 020654 032737 040000 007342  BIT  #WCE,HCS2 ;EXPECT MISCOMPARE
4344 020662 001001  BNE 65
4345 020664 104017  ERROR 17 ;WRITE CHECK CMD NOT FUNCTIONING
4346 ;WITH INTENTIONAL MISCOMPARE
4347 020666 65:
4348
4349 020666 012737 010340 007430  MOV  #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4350 020674 005037 007432  CLR  E.B0 ;EXPECTED MSG B0

```

G07

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 83  
T11 BASIC WRITE DATA TEST; FULL SECTOR

SEQ 0083

4351	020700	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP),E.A1	;EXPECTED A1
4352	020706	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4353	020714	005037	007440		CLR	E.A2	;EXPECTED MSG A2
4354	020720	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4355	020726	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4356							
4357	020734	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4358	020740	000000			.WORD	0!0!0	; & MSGS SPECIFIED HERE
4359	020742	104057			ERROR	57	;MSG A0 ERROR AFT WRT CHK CMD
4360	020744	104031			ERROR	31	;MSG B0 ERROR
4361	020746	104060			ERROR	60	;MSG A1 ERROR
4362	020750	104032			ERROR	32	;MSG B1 ERROR

75:

```

*****
:TEST 12      20 SECTOR FORMAT TEST
*
*      ALL 1'S ARE WRITTEN ON A FULL SECTOR IN 20 SECTOR FORMAT.
*      MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
*      CYL 0, TRACK 0, & SECTOR 0 IS USED.
*
*****

```

4372							
4373	020752	000004			TEST12:	SCOPE	
4374	020754	012737	000001	001174	MOV	#1,STIMES	;DO 1 ITERATION
4375	020762	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
4376							
4377	020766	004737	034140		JSR	PC,SUBCLR	
4378	020772	104024			ERROR	24	;CERR AFTER SCLR
4379							
4380	020774	012765	001506	000004	MOV	#HDTAB,RKBA(R5)	;HEADER WORD TABLE
4381	021002	012765	177704	000002	MOV	#-60.,RKWC(R5)	;WORD COUNT FOR 20 SECTOR FMT
4382	021010	005037	001346		CLR	TOCYL	
4383	021014	005237	007324		INC	BADHDR	;USED FOR VALID HALT
4384							
4385							
4386	021020	013737	001346	001362	MOV	TOCYL,CALADD	;SETUP
4387	021026	012737	000000	001444	MOV	#0,HEAD	;TO FILL
4388	021034	012737	000001	001452	MOV	#1,FORMAT	;HEADER
4389	021042	004737	035122		JSR	PC,FHDTAB	;TABLE
4390							
4391							
4392	021046	012765	010027	000000	MOV	#(CFMT!WRHEAD),RKCS1(R5)	;WRITE HEADER CMD
4393	021054	013737	001420	007376	MOV	T5000,TEMP1	;SETUP TIMEOUT
4394	021062	004737	032230		JSR	PC,FRDY	;FIND RDY
4395	021066	104200			ERROR	200	;NO RDY AFTER WRITE HEADER CMD
4396	021070	012765	010001	000000	MOV	#(CFMT!SELDRV),RKCS1(R5)	
4397	021076	013737	001406	007376	MOV	T10,TEMP1	
4398	021104	004737	032230		JSR	PC,FRDY	
4399	021110	104117			ERROR	117	;NO RDY AFTER SELDRV CMD
4400	021112	032737	100000	007340	BIT	#CERR,HCS1	
4401	021120	001405			BEQ	645	
4402	021122	104201			ERROR	201	;CERR AFTER WRITE HEADFR CMD
4403	021124	104401	045277		TYPE	,MSG26	;ABORTING BAL OF TESTS
4404	021130	000137	031514		JMP	\$EOP	

645:

4406

# H07

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR610.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 84  
T12 20 SECTOR FORMAT TEST

SEQ 0084

4407	021134	104415				SCOPI		
4408	021136	012706	001100			MOV	#STACK, SP	;RESTORE STK PTR
4409								
4410	021142	004737	034140			JSR	PC, SUBCLR	
4411	021146	104024				ERROR	24	;CERR AFTER SCLR
4412								
4413	021150	005037	001402			CLR	SECTOR	
4414	021154	013765	001402	000006	45:	MOV	SECTOR, RKDA(R5)	
4415	021162	012765	001464	000004		MOV	#DATA1, RKBA(R5)	;WRITE ALL 1'S
4416	021170	052765	000020	000010		BIS	#BAI, RKCS2(R5)	;BUSS ADDR INCR INHIBIT
4417	021176	012765	177400	000002		MOV	#-256., RKWC(R5)	;DO FULL SECTOR
4418								
4419								
4420	021204	012765	010023	000000		MOV	#<CFMT!WRDATA>, RKCS1(R5)	;WRITE DATA CMD
4421	021212	013737	001420	007376		MOV	T50000, TEMP1	;SETUP TIMEOUT
4422	021220	004737	032230			JSR	PC, FRDY	;FIND RDY
4423	021224	104011				ERROR	11	;NO RDY AFTER WRITE DATA CMD
4424	021226	012765	010001	000000		MOV	#<CFMT!SELDRV>, RKCS1(R5)	
4425	021234	013737	001406	007375		MOV	T10, TEMP1	
4426	021242	004737	032230			JSR	PC, FRDY	
4427	021246	104117				ERROR	117	;NO RDY AFTER SELDRV CMD
4428	021250	032737	100000	007340		BIT	#CERR, HCS1	
4429	021256	001465				BEQ	685	;BR IF NO ERRORS
4430								
4431	021260	032737	000200	007354		BIT	#BSE, HER	;SEE IF BAD SECTOR FLAG
4432	021266	001421				BEQ	665	;BR IF NO
4433	021270	004737	035610			JSR	PC, TRUERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4434	021274	000455				BR	675	;RETURN HERE IF NO
4435								
4436	021276	005237	001402			INC	SECTOR	;RETURN HERE IF YES
4437	021302	023727	001402	000012		CMP	SECTOR, #10.	;ARE 10 CONSEC. SECTORS BAD
4438	021310	001003				BNE	655	;BR IF NO
4439	021312	104046				ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS
4440	021314	000137	022256			JMP	35	;BYPASS TEST
4441	021320	012765	100000	000000	655:	MOV	#CLR, RKCS1(R5)	;TRY ANOTHER SECTOR
4442	021326	000137	021154			JMP	45	
4443	021332	104012			665:	ERROR	12	;CERR WITH WRITE DATA CMD
4444								
4445	021334	012737	010340	007430		MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>, E.A0	;EXPECTED MSG A0
4446	021342	005037	007432			CLR	E.B0	;EXPECTED MSG B0
4447	021346	012737	001720	007434		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>, E.A1	;EXPECTED A1
4448	021354	012737	000001	007436		MOV	#1, E.B1	;MSG ID FOR EXPECTED MSG B1
4449	021362	005037	007440			CLR	E.A2	;EXPECTED MSG A2
4450	021366	012737	000002	007442		MOV	#2, E.B2	;MSG ID FOR EXPECTED MSG B2
4451	021374	012737	000003	007446		MOV	#3, E.B3	;MSG ID FOR EXPECTED MSG B3
4452								
4453	021402	004737	032752			JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
4454	021406	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4455	021410	104052				ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
4456	021412	104023				ERROR	23	;MSG B0 ERROR
4457	021414	104053				ERROR	53	;MSG A1 ERROR
4458	021416	104025				ERROR	25	;MSG B1 ERROR
4459	021420	104401	045277			TYPE	MSG26	;ABORTING BAL OF TESTS
4460	021424	000137	031514			JMP	\$EOP	
4461	021430	104063			675:	ERROR	63	;BAD SECTOR NOT LISTED IN TABLE
4462	021432				685:			



4463	021432	012765	010001	000000	MOV	#(CFMT!SELDRV),RKCS1(R5)	
4464	021440	013737	001406	007376	MOV	T10,TEMP1	
4465	021446	004737	032230		JSR	PC,FRDY	;FIND RDY
4466	021452	104117			ERROR	117	;RDY NOT FOUND AFTER SELDRV CMD
4467	021454	032737	001000	007366	BIT	#D.FORM,HMR2	
4468	021462	001001			BNE	15	
4469	021464	104102			ERROR	102	;FORMAT NOT SET
4470							
4471	021466						
4472							
4473	021466	012737	010340	007430	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
4474	021474	005037	007432		CLR	E.B0	;EXPECTED MSG B0
4475	021500	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
4476	021506	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4477	021514	005037	007440		CLR	E.A2	;EXPECTED MSG A2
4478	021520	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4479	021526	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4480							
4481	021534	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4482	021540	000000			.WORD	0!0!0	;# MSGS SPECIFIED HERE
4483	021542	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
4484	021544	104023			ERROR	23	;MSH B0 ERROR
4485	021546	104053			ERROR	53	;MSG A1 ERROR
4486	021550	104025			ERROR	25	;MSG B1 ERROR
4487	021552	012765	177400	000002	MOV	#-256,RKWC(R5)	
4488	021560	012765	001464	000004	MOV	#DATA1,RKBA(R5)	
4489	021566	052765	000020	000010	BIS	#BAI,RKCS2(R5)	
4490	021574	013765	001402	000006	MOV	SECTOR,RKDA(R5)	
4491							
4492	021602	012765	010031	000000	MOV	#(CFMT!WRTCHK),RKCS1(R5)	;WRITE CHECK CMD
4493	021610	013737	001420	007376	MOV	T5000,TEMP1	;SETUP TIMEOUT
4494	021616	004737	032230		JSR	PC,FRDY	;FIND RDY
4495	021622	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
4496	021624	012765	010001	000000	MOV	#(CFMT!SELDRV),RKCS1(R5)	
4497	021632	013737	001406	007376	MOV	T10,TEMP1	
4498	021640	004737	032230		JSR	PC,FRDY	
4499	021644	104117			ERROR	117	;NO RDY AFTER SELDRV CMD
4500	021646	032737	100000	007340	BIT	#CERR,HCS1	
4501	021654	001453			BEQ	705	
4502	021656	032737	040000	007342	BIT	#WCE,HCS2	;SEE IF WRITE CHECK ERROR
4503	021664	001410			BEQ	695	
4504	021666	016537	000024	001436	MOV	RKDB(R5),WD1	;ACTUAL WORD FOR PRINTOUT
4505	021674	013737	001464	001440	MOV	DATA1,WD2	;EXPECTED WORD FOR TYPEOUT
4506	021702	104016			ERROR	16	;WCE AFTER WRITE CMD
4507	021704	000437			BR	705	
4508							
4509	021706	104022					
4510							
4511	021710	012737	010340	007430	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
4512	021716	005037	007432		CLR	E.B0	;EXPECTED MSG B0
4513	021722	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
4514	021730	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4515	021736	005037	007440		CLR	E.A2	;EXPECTED MSG A2
4516	021742	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4517	021750	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4518							

15:

695:

```

4519 021756 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4520 021762 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4521 021764 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
4522 021766 104031 ERROR 31 ;MSH B0 ERROR
4523 021770 104060 ERROR 60 ;MSG A1 ERROR
4524 021772 104032 ERROR 32 ;MSG B1 ERROR
4525 021774 104401 045277 TYPE MSG26 ;ABORTING BAL OF TESTS
4526 022000 000137 031514 JMP $EOP

4528 022004 70$: MOV #<CFMT!SELDRV>,RKCS1(R5)
4529 022004 012765 010001 000000 MOV T10,TEMP1
4530 022012 013737 001406 007376 JSR PC,FRDY ;FIND RDY
4531 022020 004737 032230 ERROR 11? ;NO RDY AFTER SELDRV CMD
4532 022024 104117 BIT #D.FORM,HMR2
4533 022026 032737 001000 007366 BNE 2$
4534 022034 001001 ERROR 103 ;FORMAT NOT SET
4535 022036 104103

4537 022040 2$: MOV #<D!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4538 4539 022040 012737 010340 007430 CLR E.B0 ;EXPECTED MSG B0
4540 022046 005037 007432 MOV #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4541 022052 012737 001720 007434 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4542 022060 012737 000001 007436 CLR E.A2 ;EXPECTED MSG A2
4543 022066 005037 007440 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4544 022072 012737 000002 007442 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4545 022100 012737 000003 007446

4547 022106 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4548 022112 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4549 022114 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
4550 022116 104031 ERROR 31 ;MSH B0 ERROR
4551 022120 104060 ERROR 60 ;MSG A1 ERROR
4552 022122 104032 ERROR 32 ;MSG B1 ERROR
4553 022124 104415 SCOP1
4554 022126 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
4555

4556 022132 004737 034140 JSR PC,SUBCLR
4557 022136 104024 ERROR 24 ;CERR AFTER SCLR
4558

4559 022140 012765 001506 000004 MOV #HDTAB,RKBA(R5) ;RESTORE CYL 0 TO 22 SECTOR FMT
4560 022146 012765 177676 000002 MOV #-66.,RKWC(R5)
4561 022154 005037 001346 CLR TOCYL

4564 022160 013737 001346 001362 MOV TOCYL,CALADD ;SETUP
4565 022166 012737 000000 001444 MOV #0,HEAD ;TO FILL
4566 022174 012737 000000 001452 MOV #0,FORMAT ;HEADER
4567 022202 004737 035122 JSR PC,FHDTAB ;TABLE
4568
4569

4570 022206 012765 000027 000000 MOV #<WRHEAD>,RKCS1(R5) ;WRITE HEADER CMD
4571 022214 013737 001420 007376 MOV T5000,TEMP1 ;SETUP TIMEOUT
4572 022222 004737 032230 JSR PC,FRDY ;FIND RDY
4573 022226 104200 ERROR 200 ;NO RDY AFTER WRITE HEADER CMD
4574 022230 004737 033566 JSR PC,GSTAT ;GET FRESH STATUS

```

```

4575 022234 032737 100000 007340
4576 022242 001405
4577 022244 104201
4578 022246 104401 045277
4579 022252 000137 031514
4580 022256
4581
4582 022256 005037 007324
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597 022262 000004
4598 022264 012737 000001 001174
4599 022272 012706 001100
4600
4601 022276 012702 000001
4602
4603 022302 004737 034140
4604 022306 104024
4605
4606 022310 010265 000016
4607
4608 022314 012737 022410 001176
4609 022322 012765 000015 000000
4610 022330 013737 001414 007376
4611 022336 004737 032230
4612 022342 104033
4613
4614 022344 012737 032140 007430
4615 022352 005037 007432
4616 022356 012737 001720 007434
4617 022364 012737 000001 007436
4618
4619 022372 004737 032752
4620 022376 000000
4621 022400 104035
4622 022402 104061
4623 022404 104036
4624 022406 104062
4625
4626 022410 005037 001176
4627 022414 013737 001416 007376
4628 022422 004737 032640
4629 022426 104034
4630

```

```

BIT #CERR,HCS1
BEQ 71$
ERROR 201 ;CERR AFTER WRITE HEADER CMD
TYPE MSG26 ;ABORTING BAL OF TESTS
JMP $EOP

71$:

3$: CLR BADHOR ;USED FOR VALID HALT

*****
*TEST 13 TEST OFFSET & RTC LOGIC
*
* THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.
* THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE
* 'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG
* BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A
* DIFFERENT CYLINDER IS ISSUED
* IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT
* CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.
* ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED
*****

T13: SCOPE
MOV #1,$TIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR

MOV #1,R2 ;MIN POS OFFSET

1$: JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR

MOV R2,RKASOF(R5) ;SET OFFSET

MOV #64,$$ESCAPE
MOV #OFFSET,RKCS1(R5) ;OFFSET CMD
MOV T100,TEMP1 ;SETUP TIMEOUT
JSR PC,FROY
ERROR 33 ;NO RDY AFTER OFFSET CMD

MOV #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
CLR E.B0
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
MOV #1,E.B1

JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
WORD 0!0!0 ;& MSGS SPECIFIED HERE
ERROR 35 ;MSG A0 ERROR DURING OFFSET CMD
ERROR 61 ;MSG B0 ERROR
ERROR 36 ;MSG A1 ERROR
ERROR 62 ;MSG B1 ERROR

64$: CLR $$ESCAPE
MOV T5000,TEMP1 ;SETUP TIMEOUT
JSR PC,FATT2 ;FIND ATTN
ERROR 34 ;NO ATTN AFTER OFFSET CMD

```

```

4631
4632 022430 012737 052340 007430 MOV #<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4633 022436 005037 007432 CLR E.B0 ;EXPECTED MSG B0
4634 022442 012737 001720 007434 MOV #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4635 022450 012737 000001 007436 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4636 022456 005037 007440 CLR E.A2 ;EXPECTED MSG A2
4637 022462 012737 000002 007442 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4638 022470 012737 000003 007446 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4639
4640 022476 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4641 022502 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4642 022504 104260 ERROR 260 ;MSG A0 ERROR AFTER OFFSET CMD
4643 022506 104261 ERROR 261 ;MSG B0 ERROR
4644 022510 104037 ERROR 37 ;MSG A1 ERROR
4645 022512 104040 ERROR 40 ;MSG B1 ERROR
4646
4647 022514 005737 001360 TST CYLADD
4648 022520 001401 BEQ 17$
4649 022522 104042 ERROR 42 ;CYL ADDR IN B2 WAS NOT 0
4650 ;AFTER OFFSET CMD FROM CYL 0
4651 022524 17$:
4652 022524 010265 000016 MOV R2,RKASOF(R5) ;REFRESH RKASOF
4653
4654 022530 032702 000200 BIT #BIT7,R2
4655 022534 001005 BNE 65$ ;BR IF NEG OFFSET
4656
4657 022536 020237 001356 CMP R2,CYLDIF ;CHECK POS OFFSET
4658 022542 001406 BEQ 66$
4659 022544 104114 ERROR 114 ;OFFSET IN A2 NOT = RKASOF
4660 022546 000404 BR 66$ ;AFTER OFFSET CMD
4661
4662 022550 020137 001356 65$: CMP R1,CYLDIF ;CHECK NEG OFFSET
4663 022554 001401 BEQ 66$
4664 022556 104114 ERROR 114 ;OFFSET IN A2 NOT = RKASOF
4665 ;AFTER OFFSET CMD
4666 022560 66$:
4667
4668 022560 012765 100000 000000 MOV #CLR,RKCS1(R5)
4669 022566 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
4670 022574 012765 000005 000000 MOV #CLR,RKCS1(R5) ;DRIVE CLEAR CMD
4671 022602 013737 001406 007376 MOV T10,TEMP1 ;SETUP TIMEOUT
4672 022610 004737 032230 JSR PC,FRDY ;FIND RDY
4673 022614 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
4674 022616 004737 032512 JSR PC,TSTATN ;TEST FOR ATTN
4675 022622 000401 BR 67$
4676 022624 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4677 022626 67$:
4678
4679 022626 012765 100000 000000 MOV #CLR,RKCS1(R5)
4680 022634 004737 033566 JSR PC,GSTAT
4681 022640 032737 002000 007366 BIT #D.OFF,HMR2
4682 022646 001001 BNE 4$
4683 022650 104043 ERROR 43 ;OFFSET BIT IN RKMR2 CLEARED
4684 ;AFTER DRIVE CLEAR CMD & SELECT DRV CMD
4685 022652 012737 012340 007430 4$: MOV #<D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED A0
4686 022660 005037 007432 CLR E.B0

```

M07

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 89  
T13 TEST OFFSET & RTC LOGIC

SEQ 0089

4687	022664	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	
4688	022672	012737	000001	007436	MOV	#1,E.B1	
4689							
4690	022700	004737	032752		JSR	PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1	
4691	022704	000003			WORD	T,A2!T.B2!0 ;& MSGS SPECIFIED HERE	
4692	022706	104273			ERROR	273 ;MSG A0 ERROR AFTER DRIVE CLEAR CMD	
4693	022710	104265			ERROR	265 ;MSH B0 ERROR	
4694	022712	104274			ERROR	274 ;MSG A1 ERROR	
4695	022714	104266			ERROR	266 ;MSG B1 ERROR	
4696	022716	010265	000016		MOV	R2,RKASOF(R5) ;REFRESH RKASOF	
4697							
4698	022722	032702	000200		BIT	#BIT7,R2	
4699	022726	001005			BNE	68\$ ;BR IF NEG OFFSET	
4700							
4701	022730	020237	001356		CMP	R2,CYLDIF ;CHECK POS OFFSET	
4702	022734	001406			BEQ	69\$	
4703	022736	104115			ERROR	115 ;OFFSET IN A2 NOT = RKASOF	
4704	022740	000404			BR	69\$ ;AFTER DRIVE CLEAR CMD	
4705							
4706	022742	020137	001356		68\$: CMP	R1,CYLDIF ;CHECK NEG OFFSET	
4707	022746	001401			BEQ	69\$	
4708	022750	104115			ERROR	115 ;OFFSET IN A2 NOT = RKASOF	
4709							
4710	022752				69\$:		
4711							
4712	022752	012765	000017	000000	MOV	#SEEK,RKCS1(R5) ;SEEK CMD	
4713	022760	013737	001410	007376	MOV	T50,TEMP1 ;SETUP TIMEOUT	
4714	022766	004737	032230		JSR	PC,FRDY ;FIND RDY	
4715	022772	104131			ERROR	131 ;NO RDY AFTER SEEK CMD	
4716							
4717	022774	013737	001420	007376	MOV	T50000,TEMP1 ;SETUP TIMEOUT	
4718	023002	004737	032640		JSR	PC,FATT2 ;FIND ATTN	
4719	023006	104132			ERROR	132 ;NO ATTN AFTER SEEK CMD	
4720							
4721	023010	032737	100000	007340	BIT	#CERR,HCS1	
4722	023016	001401			BEQ	70\$	
4723	023020	104210			ERROR	210 ;CERR AFTER SEEK CMD	
4724							
4725	023022				70\$:		
4726							
4727							
4728	023022	032737	002000	007366	BIT	#D.OFF,HMR2	
4729	023030	001001			BNE	7\$	
4730	023032	104045			ERROR	45 ;OFFSET BIT CLEARED IN RKMR2 AFTER SEEK TO SELF.	
4731							
4732	023034				7\$:		
4733							
4734	023034	012737	052340	007430	MOV	#(D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0 ;EXPECTED MSG A0	
4735	023042	005037	007432		CLR	E.B0 ;EXPECTED MSG B0	
4736	023046	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1 ;EXPECTED A1	
4737	023054	012737	000001	007436	MOV	#1,E.B1 ;MSG ID FOR EXPECTED MSG B1	
4738	023062	005037	007440		CLR	E.A2 ;EXPECTED MSG A2	
4739	023066	012737	000002	007442	MOV	#2,E.B2 ;MSG ID FOR EXPECTED MSG B2	
4740	023074	012737	000003	007446	MOV	#3,E.B3 ;MSG ID FOR EXPECTED MSG B3	
4741							
4742	023102	004737	032752		JSR	PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1	

```

4743 023106 000003 .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
4744 023110 104140 ERROR 140 ; MSG A0 ERROR AFTER SEEK TO SELF
4745 023112 104141 ERROR 141 ; MSG B0 ERROR
4746 023114 104142 ERROR 142 ; MSG A1 ERROR
4747 023116 104143 ERROR 143 ; MSG B1 ERROR
4748 023120 010265 000016 MOV R2,RKASOF(R5) ; REFRESH RKASOF
4749
4750 023124 032702 000200 BIT #BIT7,R2
4751 023130 001005 BNE 71$ ; BR IF NEG OFFSET
4752
4753 023132 020237 001356 CMP R2,CYLDIF ; CHECK POS OFFSET
4754 023136 001406 BEQ 72$
4755 023140 104123 ERROR 123 ; OFFSET IN A2 NOT = RKASOF
4756 023142 000404 BR 72$ ; AFTER SEEK TO SELF
4757
4758 023144 020137 001356 71$: CMP R1,CYLDIF ; CHECK NEG OFFSET
4759 023150 001401 BEQ 72$
4760 023152 104123 ERROR 123 ; OFFSET IN A2 NOT = RKASOF
4761 ; AFTER SEEK TO SELF
4762 023154 72$:
4763
4764 023154 004737 034140 JSR PC,SUBCLR
4765 023160 104024 ERROR 24 ; CERR AFTER SCLR
4766
4767 023162 012737 000012 001346 MOV #10.,TOCYL
4768 023170 012765 000012 000020 MOV #10.,RKDC(R5) ; SETUP CYL 10
4769 ; DO ACTUAL IMPLIED SEEK TO CYL 10 TO VERIFY
4770 ; OFFSET BIT IN RKMR2 CLEARED
4771
4772
4773 023176 012700 001712 MOV #RHTAB,RO
4774 023202 012765 000025 000000 MOV #<RDHEAD>,RKCS1(R5) ; READ HEADER CMD
4775 023210 013737 001420 007376 MOV T5000,TEMP1 ; SETUP TIMEOUT
4776 023216 004737 032230 JSR PC,FRDY ; FIND RDY
4777 023222 104171 ERROR 171 ; NO RDY AFTER READ HEADER CMD
4778 023224 032737 100000 007340 BIT #CERR,HCS1
4779 023232 001405 BEQ 74$
4780 023234 104174 ERROR 174 ; CERR AFTER READ HEADER CMD
4781 023236 104401 045277 TYPE MSG26 ; ABORTING BAL OF TESTS
4782 023242 000137 031514 JMP $EOP
4783
4784 023246 016520 000024 74$: MOV RKDB(R5),(RO)+ ; 1'ST WORD FROM SILO TO RHTAB
4785 023252 016520 000024 MOV RKDB(R5),(RO)+ ; 2'ND WORD
4786 023256 016520 000024 MOV RKDB(R5),(RO)+ ; 3'RD WORD
4787
4788
4789 023262 032765 100000 000010 BIT #DLT,RKCS2(R5)
4790 023270 001407 BEQ 75$
4791 023272 004737 033566 JSR PC,GSTAT
4792 023276 104173 ERROR 173 ; DLT AFTER READ HEADER CMD
4793 023300 104401 045277 TYPE MSG26 ; ABORTING BAL OF TESTS
4794 023304 000137 031514 JMP $EOP
4795 023310 75$:
4796
4797 023310 012737 010340 007430 MOV #<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ; EXPECTED MSG A0
4798 023316 005037 007432 CLR E.B0 ; EXPECTED MSG B0

```

4799	023322	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRAH!D.SSP),E.A1	;EXPECTED A1
4800	023330	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4801	023336	005037	007440		CLR	E.A2	;EXPECTED MSG A2
4802	023342	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4803	023350	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4804							
4805	023356	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4806	023362	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4807	023364	104301			ERROR	301	;MSG A0 ERROR AFTER READ HEADER CMD
4808	023366	104271			ERROR	271	;MSG B0 ERROR
4809	023370	104302			ERROR	302	;MSG A1 ERROR
4810	023372	104272			ERROR	272	;MSG B1 ERROR
4811							
4812	023374	023737	001712	001346	CMP	RHTAB,TOCYL	;CHECK WORD 0 ONLY, CYL#
4813	023402	001401			BEQ	73\$	;BR IF SAME
4814	023404	104051			ERROR	51	;WRONG CYL# ON HEADER
4815	023406						73\$:
4816							
4817							
4818	023406	032737	002000	007366	BIT	#D.OFF,HMR2	
4819	023414	001401			BEQ	9\$	
4820	023416	104101			ERROR	101	;OFFSET NOT CLEARED AFTER READ HEADER WITH MOVEMENT
4821							
4822	023420	023727	001360	000012	CMP	CYLOAD,#10.	9\$:
4823	023426	001401			BEQ	10\$	
4824	023430	104122			ERROR	122	;DID NOT GO TO CYL 10
4825							
4826	023432	005737	001356		TST	CYLDIF	10\$:
4827	023436	001401			BEQ	16\$	
4828	023440	104101			ERROR	101	;OFFSET NOT CLEARED IN RKMR2
4829							
4830	023442	004737	034140		JSR	PC,SUBCLR	16\$:
4831	023446	104024			ERROR	24	;CERR AFTER SCLR
4832							
4833	023450	012765	000017	000000	MOV	#SEEK,RKCS1(R5)	;SEEK CMD
4834	023456	013737	001410	007376	MOV	T50,TEMP1	;SETUP TIMEOUT
4835	023464	004737	032230		JSR	PC,FRDY	;FIND RDY
4836	023470	104131			ERROR	131	;NO RDY AFTER SEEK CMD
4837							
4838	023472	013737	001420	007376	MOV	T50000,TEMP1	;SETUP TIMEOUT
4839	023500	004737	032640		JSR	PC,FATT2	;FIND ATTN
4840	023504	104132			ERROR	132	;NO ATTN AFTER SEEK CMD
4841							
4842	023506	032737	100000	007340	BIT	#CERR,HCS1	
4843	023514	001401			BEQ	76\$	
4844	023516	104210			ERROR	210	;CERR AFTER SEEK CMD
4845							
4846	023520						76\$:
4847							
4848							
4849	023520	032702	000200		BIT	#BIT7,R2	;SEE IF DOING NEG OFFSETS
4850	023524	001014			BNE	18\$	;BR IF YES
4851							
4852	023526	005202			INC	R2	
4853	023530	020227	000061		CMP	R2,#61	;SEE IF JUST DID MAX POS OFFSET
4854	023534	001402			BEQ	20\$	;BR IF YES

4855 023536 000137 022302  
 4856  
 4857 023542 012702 000201  
 4858 023546 012701 000101  
 4859 023552 000137 022302  
 4860  
 4861 023556 005201  
 4862 023560 005202  
 4863 023562 020227 000261  
 4864 023566 001402  
 4865 023570 000137 022302  
 4866  
 4867  
 4868  
 4869  
 4870  
 4871  
 4872  
 4873  
 4874  
 4875  
 4876  
 4877  
 4878  
 4879  
 4880  
 4881  
 4882  
 4883  
 4884  
 4885  
 4886  
 4887

JMP 1\$ ;ELSE DO NEXT POS OFFSET  
 20\$: MOV #201,R2 ;SETUP NEG OFFSET FOR RKASOF  
 MOV #101,R1 ;SETUP NEG OFFSET OFOR MSG A  
 JMP 1\$ ;DO NEG OFFSET  
 18\$: INC R1  
 INC R2  
 CMP R2,#261 ;SEE IF ALL NEG OFFSETS DONE  
 BEQ TST14 ;GO TO NEXT TST  
 JMP 1\$ ;DO ANOTHER

\*\*\*\*\*  
 TEST 14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY  
 WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN  
 PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET  
 POSITIONS UNTIL A FAILURE OCCURES. THE FAILING OFFSET POSITIONS  
 ARE TYPED OUT.  
 OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STATUS 00 & 10.  
 ALL HEADS ARE TESTED AT CYLINDER 0  
 IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT  
 OR A. HEADS DID NOT MOVE AT ALL  
 B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP  
 ARE EXCEPTIONALLY GOOD.  
 NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR  
 BUT AN INDICATION OF SURFACE, HEAD & R/W ELECTRONICS QUALITY ONLY.

4888 023574 000004  
 4889 023576 012737 000001 001174  
 4890 023604 012706 001100  
 4891  
 4892 023610 004737 034140  
 4893 023614 104024  
 4894  
 4895 023616 005037 001402  
 4896 023622 013765 001402 000006  
 4897 023630 012765 001464 000004  
 4898 023636 052765 000020 000010  
 4899 023644 012765 177400 000002  
 4900  
 4901  
 4902  
 4903 023652 012765 000023 000000  
 4904 023660 013737 001420 007376  
 4905 023666 004737 032230  
 4906 023672 104011  
 4907 023674 004737 033566  
 4908 023700 032737 100000 007340  
 4909 023706 001465  
 4910

\*\*\*\*\*  
 TST14: SCOPE  
 MOV #1,STIMES ;DO 1 ITERATION  
 MOV #STACK,SP ;RESTORE STK PTR  
 JSR PC,SUBCLR  
 ERROR 24 ;CERR AFTER SCLR  
 11\$: CLR SECTOR  
 MOV SECTOR,RKDA(R5)  
 MOV #DATA1,RKBA(R5) ;WRITE ALL 1'S  
 BIS #BAI,RKCS2(R5) ;BUS ADDR INCR INHIB  
 MOV #-256.,RKWC(R5) ;SECTOR 0 ONLY  
 ;WILL DO IMPLIED SEEK TO CYL 0  
 ;WAS ON CYL 1 FROM LAST TEST  
 MOV #<WRDATA>,RKCS1(R5) ;WRITE DATA CMD  
 MOV T5000,TEMP1 ;SETUP TIMEOUT  
 JSR PC,FRDY ;FIND RDY  
 ERROR 11 ;NO RDY AFTER WRITE DATA CMD  
 JSR PC,GSTAT ;GET FRESH STATUS  
 BIT #CERR,HCS1  
 BEQ 67\$ ;BR IF NO ERRORS



4911	023710	032737	000200	007354	BIT	#BSE, MER	; SEE IF BAD SECTOR FLAG
4912	023716	001421			BEQ	65\$	; BR IF NO
4913	023720	004737	035610		JSR	PC, TRUEERR	; ELSE SEE IF SECTOR LISTED IN BSE TABLE
4914	023724	000455			BR	66\$	; RETURN HERE IF NO
4915							
4916	023726	005237	001402		INC	SECTOR	; RETURN HERE IF YES
4917	023732	023727	001402	000012	CMP	SECTOR, #10.	; ARE 10 CONSEC. SECTORS BAD
4918	023740	001003			BNE	64\$	; BR IF NO
4919	023742	104046			ERROR	46	; ABORTING TEST DETECTED 10 BAD SECTORS
4920	023744	000137	025406		JMP	10\$	; BYPASS TEST
4921	023750	012765	100000	000000	MOV	#CCLR, RKCS1(R5)	; TRY ANOTHER SECTOR
4922	023756	000137	023622		JMP	11\$	
4923	023762	104012			ERROR	12	; CERR WITH WRITE DATA CMD
4924							
4925	023764	012737	010340	007430	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA), E.A0	; EXPECTED MSG A0
4926	023772	005037	007432		CLR	E.B0	; EXPECTED MSG B0
4927	023776	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP), E.A1	; EXPECTED A1
4928	024004	012737	000001	007436	MOV	#1, E.B1	; MSG ID FOR EXPECTED MSG B1
4929	024012	005037	007440		CLR	E.A2	; EXPECTED MSG A2
4930	024016	012737	000002	007442	MOV	#2, E.B2	; MSG ID FOR EXPECTED MSG B2
4931	024024	012737	000003	007446	MOV	#3, E.B3	; MSG ID FOR EXPECTED MSG B3
4932							
4933	024032	004737	032752		JSR	PC, CHKMSG	; CHECK MSGS A0, B0, A1, B1
4934	024036	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4935	024040	104052			ERROR	52	; MSG A0 ERROR AFTER WRITE DATA CMD
4936	024042	104023			ERROR	23	; MSH B0 ERROR
4937	024044	104053			ERROR	53	; MSG A1 ERROR
4938	024046	104025			ERROR	25	; MSG B1 ERROR
4939	024050	104401	045277		TYPE	MSG26	; ABORTING BAL OF TESTS
4940	024054	000137	031514		JMP	\$EOP	
4941	024060	104063			ERROR	63	; BAD SECTOR NOT LISTED IN TABLE
4942	024062						
4943							
4944	024062	012737	010340	007430	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA), E.A0	; EXPECTED MSG A0
4945	024070	005037	007432		CLR	E.B0	; EXPECTED MSG B0
4946	024074	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP), E.A1	; EXPECTED A1
4947	024102	012737	000001	007436	MOV	#1, E.B1	; MSG ID FOR EXPECTED MSG B1
4948	024110	005037	007440		CLR	E.A2	; EXPECTED MSG A2
4949	024114	012737	000002	007442	MOV	#2, E.B2	; MSG ID FOR EXPECTED MSG B2
4950	024122	012737	000003	007446	MOV	#3, E.B3	; MSG ID FOR EXPECTED MSG B3
4951							
4952	024130	004737	032752		JSR	PC, CHKMSG	; CHECK MSGS A0, B0, A1, B1
4953	024134	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4954	024136	104052			ERROR	52	; MSG A0 ERROR AFTER WRITE DATA CMD
4955	024140	104023			ERROR	23	; MSH B0 ERROR
4956	024142	104053			ERROR	53	; MSG A1 ERROR
4957	024144	104025			ERROR	25	; MSG B1 ERROR
4958	024146	012765	001464	000004	MOV	#DATA1, RKBA(R5)	
4959	024154	052765	000020	000010	BIS	#BA1, RKCS2(R5)	
4960	024162	012765	177400	000002	MOV	#-256, RKWC(R5)	
4961	024170	013765	001402	000006	MOV	SECTOR, RKDA(R5)	
4962							
4963	024176	012765	000031	000000	MOV	#(WATCH), RKCS1(R5)	; WRITE CHECK CMD
4964	024204	013737	001420	007376	MOV	T5000, TEMP1	; SETUP TIMEOUT
4965	024212	004737	032230		JSR	PC, FRDY	; FIND RDY
4966	024216	104015			ERROR	15	; NO RDY AFTER WRITE CHECK CMD

# E08

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 94  
T14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

SEQ 0094

4967	024220	004737	033566		JSR	PC, GSTAT	;GET FRESH STATUS
4968	024224	032737	100000	007340	BIT	#CERR, HCS1	
4969	024232	001453			BEQ	69\$	
4970	024234	032737	040000	007342	BIT	#WCE, HCS2	;SEE IF WRITE CHECK ERROR
4971	024242	001410			BEQ	68\$	
4972	024244	016537	000024	001436	MOV	RKDB(R5), WD1	;ACTUAL WORD FOR PRINTOUT
4973	024252	013737	001464	001440	MOV	DATA1, WD2	;EXPECTED WORD FOR TYPEOUT
4974	024260	104016			ERROR	16	;WCE AFTER WRITE CMD
4975	024262	000437			BR	69\$	
4976							
4977	024264	104022			68\$:	ERROR	22 ;CERR AFTER WRITE CHECK CMD
4978							
4979	024266	012737	010340	007430	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>, E.A0	;EXPECTED MSG A0
4980	024274	005037	007432		CLR	E.B0	;EXPECTED MSG B0
4981	024300	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>, E.A1	;EXPECTED A1
4982	024306	012737	000001	007436	MOV	#1, E.B1	;MSG ID FOR EXPECTED MSG B1
4983	024314	005037	007440		CLR	E.A2	;EXPECTED MSG A2
4984	024320	012737	000002	007442	MOV	#2, E.B2	;MSG ID FOR EXPECTED MSG B2
4985	024326	012737	000003	007446	MOV	#3, E.B3	;MSG ID FOR EXPECTED MSG B3
4986							
4987	024334	004737	032752		JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
4988	024340	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4989	024342	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
4990	024344	104031			ERROR	31	;MSG B0 ERROR
4991	024346	104060			ERROR	60	;MSG A1 ERROR
4992	024350	104032			ERROR	32	;MSG B1 ERROR
4993	024352	104401	045277		TYPE	MSG26	;ABORTING BAL OF TESTS
4994	024356	000137	031514		JMP	\$EOP	
4995							
4996	024362				69\$:		
4997							
4998	024362	012737	010340	007430	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>, E.A0	;EXPECTED MSG A0
4999	024370	005037	007432		CLR	E.B0	;EXPECTED MSG B0
5000	024374	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>, E.A1	;EXPECTED A1
5001	024402	012737	000001	007436	MOV	#1, E.B1	;MSG ID FOR EXPECTED MSG B1
5002	024410	005037	007440		CLR	E.A2	;EXPECTED MSG A2
5003	024414	012737	000002	007442	MOV	#2, E.B2	;MSG ID FOR EXPECTED MSG B2
5004	024422	012737	000003	007446	MOV	#3, E.B3	;MSG ID FOR EXPECTED MSG B3
5005							
5006	024430	004737	032752		JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
5007	024434	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5008	024436	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
5009	024440	104031			ERROR	31	;MSG B0 ERROR
5010	024442	104060			ERROR	60	;MSG A1 ERROR
5011	024444	104032			ERROR	32	;MSG B1 ERROR
5012							
5013	024446	104401	044564		TYPE	MSG8	;READ WITH OFFSET TEST
5014	024452	005001			CLR	R1	;HEAD #
5015							
5016	024454	012700	000001		9\$:	MOV	#1, R0 ;INIT OFFSET COUNTER
5017	024460	104401	044623		TYPE	MSG9	;HEAD #
5018	024464	010146			MOV	R1, -(SP)	;SAVE R1 FOR TYPEOUT
5019							;TYPE HEAD #
5020	024466	104403			TYPOS		;GO TYPE--OCTAL ASCII
5021	024470	001			.BYTE	1	;TYPE 1 DIGIT(S)
5022	024471	000			.BYTE	0	;SUPPRESS LEADING ZEROS

# F08

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 95  
T14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

SEQ 0095

5023	024472	104401	001205		TYPE	,SCLF	
5024							
5025	024476	005037	001442	15:	CLR	OFFERR	;WRITE CHECK ERROR FLAG
5026							
5027	024502	004737	034140		JSR	PC,SUBCLR	
5028	024506	104024			ERROR	24	;CERR AFTER SCLR
5029							
5030	024510	010065	000016		MOV	R0,RKASOF(R5)	;OFFSET VALUE
5031	024514	000301			SWAB	R1	
5032	024516	010165	000006		MOV	R1,RKDA(R5)	;HEAD NO.
5033	024522	000301			SWAB	R1	
5034							
5035	024524	012737	024620	001176	MOV	#70\$, \$ESCAPE	
5036	024532	012765	000015	000000	MOV	#OFFSET,RKCS1(R5)	;OFFSET CMD
5037	024540	013737	001414	007376	MOV	T100,TEMP1	;SETUP TIMEOUT
5038	024546	004737	032230		JSR	PC,FRDY	
5039	024552	104033			ERROR	33	;NO RDY AFTER OFFSET CMD
5040							
5041	024554	012737	032140	007430	MOV	#<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5042	024562	005037	007432		CLR	E.B0	
5043	024566	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	
5044	024574	012737	000001	007436	MOV	#1,E.B1	
5045							
5046	024602	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5047	024606	000000			.WORD	0!0!0	; & MSGS SPECIFIED HERE
5048	024610	104035			ERROR	35	;MSG A0 ERROR DURING OFFSET CMD
5049	024612	104061			ERROR	61	;MSH B0 ERROR
5050	024614	104036			ERROR	36	;MSG A1 ERROR
5051	024616	104062			ERROR	62	;MSG B1 ERROR
5052							
5053	024620	005037	001176	007376	70\$: CLR	\$ESCAPE	
5054	024624	013737	001416		MOV	T500,TEMP1	;SETUP TIMEOUT
5055	024632	004737	032640		JSR	PC,FATT2	;FIND ATTN
5056	024636	104034			ERROR	34	;NO ATTN AFTER OFFSET CMD
5057							
5058							
5059	024640	012737	052340	007430	MOV	#<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5060	024646	005037	007432		CLR	E.B0	;EXPECTED MSG B0
5061	024652	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5062	024660	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5063	024666	005037	007440		CLR	E.A2	;EXPECTED MSG A2
5064	024672	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5065	024700	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5066							
5067	024706	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5068	024712	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5069	024714	104260			ERROR	260	;MSG A0 ERROR AFTER OFFSET CMD
5070	024716	104261			ERROR	261	;MSH B0 ERROR
5071	024720	104037			ERROR	37	;MSG A1 ERROR
5072	024722	104040			ERROR	40	;MSG B1 ERROR
5073							
5074							
5075	024724	012765	100000	000000	MOV	#CLR,RKCS1(R5)	
5076	024732	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)	;DRIVE#
5077	024740	012765	000005	000000	MOV	#CLEAR,RKCS1(R5)	;DRIVE CLEAR CMD
5078	024746	013737	001406	007376	MOV	T10,TEMP1	;SETUP TIMEOUT

5079	024754	004737	032230		JSR	PC,FRDY		;FIND RDY
5080	024760	104151			ERROR	151		;NO RDY AFTER DRIVE CLEAR CMD
5081	024762	004737	032512		JSR	PC,TSTATN		;TEST FOR ATTN
5082	024766	000401			BR	71\$		
5083	024770	104154			ERROR	154		;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5084	024772			71\$:				
5085								
5086	024772	012765	001464	000004	MOV	#DATA1,RKBA(R5)		
5087	025000	052765	000020	000010	B* S	#BAI,RKCS2(R5)		
5088	025006	012765	177400	000002	MOV	#-256.,RKWC(R5)		
5089	025014	013765	001402	000006	MOV	SECTOR,RKDA(R5)		
5090	025022	012765	000031	000000	MOV	#WATCH,RKCS1(R5)		;WRITE CHECK CMD
5091	025030	012737	141520	007376	MOV	#50000.,TEMP1		;SETUP TIMEOUT
5092	025036	004737	032230		JSR	PC,FRDY		;FIND RDY
5093	025042	104015			ERROR	15		;NO RDY AFTER WRITE CHECK CMD
5094	025044	004737	033566		JSR	PC,GSTAT		;GET FRESH STATUS
5095	025050	032737	040000	007342	BIT	#WCE,HCS2		
5096	025056	001421			BEQ	2\$		
5097								
5098	025060	016537	000024	001436	MOV	RKDB(R5),WD1		;GET MISCOMPARED WORD
5099	025066	005237	001442		INC	OFFERR		;BAD WRITE CHK ERROR=SET ERR FLG.
5100								
5101	025072	005737	001442		TST	OFFERR		
5102	025076	001411			BEQ	2\$		
5103	025100	104401	045543		TYPE	MSG39		;WRITE CHECK FAILURE AT OFFSET
5104	025104	010046			MOV	RO,-(SP)		;SAVE RO FOR TYPEOUT
5105								;TYPE OFFSET VALUE
5106	025106	104403			TYPOS			;GO TYPE--OCTAL ASCII
5107	025110	006			.BYTE	6		;TYPE 6 DIGITS
5108	025111	000			.BYTE	0		;SUPPRESS LEADING ZEROS
5109	025112	104401	001205		TYPE	,SCRLF		
5110	025116	104401	001205		TYPE	,SCRLF		
5111								
5112	025122	032700	000200	2\$:	BIT	#BIT7,RO		;SEE IF OFFSET IS + OR -
5113	025126	001023			BNE	5\$		;BR IF - OFFSET
5114								
5115	025130	020027	000060		CMP	RO,#60		
5116	025134	001412			BEQ	4\$		
5117	025136	005737	001442		TST	OFFERR		
5118	025142	001404			BEQ	3\$		
5119	025144	012700	000200	8\$:	MOV	#200,RO		;SETUP FOR NEG OFFSET
5120	025150	000137	024476		JMP	1\$		
5121								
5122	025154	005200		3\$:	INC	RO		
5123	025156	000137	024476		JMP	1\$		
5124								
5125	025162	005737	001442	4\$:	TST	OFFERR		
5126	025166	001366			BNE	8\$		;DO NEG OFFSETS
5127	025170	104401	045405		TY E	,MSG37		;NO WRITE CHECK ERROR AT MAX POS OFFSET
5128								;NOTE! EITHER HEADS DID NOT MOVE
5129								;OR READ/WRITE AMP IS EXCEPTIONALLY GOOD.
5130	025174	000763			BR	8\$		;DO NEG OFFSETS
5131								
5132	025176	020027	000260	5\$:	CMP	RO,#260		
5133	025202	001404			BEQ	6\$		
5134	025204	005737	001442		TST	OFFERR		

# H08

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR610.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 97  
T14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

SEQ 0097

```
5135 025210 001100      BNE    TST15      ;;GO TO NEXT TST
5136 025212 000760      BR     3$
5137
5138 025214 005737 001442    6$:    TST    OFFERR
5139 025220 001002      BNE    7$
5140 025222 104401 045463    TYPE  ,MSG38      ;NO WRITE CHECK ERROR AT MAX NEG OFFSET
5141                                     ;NOTE! EITHER HEADS DID NOT MOVE
5142                                     ;OR READ/WRITE AMP IS EXCEPTIONALLY GOOD.
5143 025226      7$:
5144
5145 025226 012765 100000 000000    MOV    #CCLR,RKCS1(R5)
5146 025234 013765 001222 000010    MOV    $UNIT,RKCS2(R5)
5147 025242 012765 000013 000000    MOV    #RECAL,RKCS1(R5)      ;RECAL CMD
5148                                     ;RESET CYL DIFF/OFFSET & CYL ADDR REG
5149                                     ;IN RKMR2 & RKMR3 RESP.
5150 025250 013737 001406 007376    MOV    T10,TEMP1      ;SETUP TIMEOUT
5151 025256 004737 032230    JSR    PC,FRDY        ;FIND RDY
5152 025262 104124    ERROR  124           ;RDY NOT SET AFTER RECAL CMD
5153
5154 025264 012765 000001 000026    MOV    #1,RKMR1(R5)    ;SELECT WORD 1
5155 025272 004737 033566    JSR    PC,GSTAT
5156 025276 032737 020000 007366    BIT    #D.RTZ,HMR2
5157 025304 001001      BNE    72$
5158 025306 104244    ERROR  244           ;RTZ NOT SET DURING RECAL CMD
5159 025310 013737 001406 007400    72$:  MOV    T10,TEMP2      ;SETUP TIMEOUT
5160 025316 004737 032544    JSR    PC,FATT1       ;FIND ATTN
5161 025322 104055    ERROR  55           ;NO ATTN AFTER RECAL CMD
5162
5163 025324 012765 100000 000000    MOV    #CCLR,RKCS1(R5)
5164 025332 013765 001222 000010    MOV    $UNIT,RKCS2(R5)    ;DRIVE#
5165 025340 012765 000005 000000    MOV    #CLEAR,RKCS1(R5)  ;DRIVE CLEAR CMD
5166 025346 013737 001406 007376    MOV    T10,TEMP1      ;SETUP TIMEOUT
5167 025354 004737 032230    JSR    PC,FRDY        ;FIND RDY
5168 025360 104151    ERROR  151          ;NO RDY AFTER DRIVE CLEAR CMD
5169 025362 004737 032512    JSR    PC,TSTATN      ;TEST FOR ATTN
5170 025366 000401      BR     73$
5171 025370 104154    ERROR  154          ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5172 025372      73$:
5173
5174
5175 025372 005201      INC    R1            ;HEAD CTR
5176 025374 020127 000003    CMP    R1,#3         ;SEE IF ALL HEADS DONE
5177 025400 001402      BEQ   10$           ;BR IF YES
5178 025402 000137 024454    JMP    9$            ;ELSE REPEAT ALL FOR NEXT HEAD
5179 025406 104401 044723    10$:  TYPE  ,MSG12       ;OFFSET FAILURES ARE NOT ERRORS
5180                                     ;*****
5181                                     ;TEST 15      WRITE WITH HEADS OFFSET
5182                                     ;*
5183                                     ;*      THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
5184                                     ;*      WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
5185                                     ;*      & THE DRIVE WILL WRITE
5186                                     ;*      SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
5187                                     ;*      THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY
5188                                     ;*
5189                                     ;*****
5190 025412 000004    TST15: SCOPE
```

5191	025414	012737	000001	001174		MOV	#1, \$TIMES	::DO 1 ITERATION
5192	025422	012706	001100			MOV	#STACK, SP	;RESTORE STK PTR
5193								
5194	025426	012700	000260			MOV	#260, RO	;MAX NEG OFFSET
5195								
5196	025432	004737	034140		15:	JSR	PC, SUBCLR	
5197	025436	104024				ERROR	24	;CERR AFTER SCLR
5198								
5199	025440	010065	000016			MOV	RO, RKASUF(R5)	;SET OFFSET
5200								
5201	025444	012737	025540	001176		MOV	#645, \$ESCAPE	
5202	025452	012765	000015	000000		MOV	#OFFSET, RKCS1(R5)	;OFFSET CMD
5203	025460	013737	001414	007376		MOV	T100, TEMP1	;SETUP TIMEOUT
5204	025466	004737	032230			JSR	PC, FRDY	
5205	025472	104033				ERROR	33	;NO RDY AFTER OFFSET CMD
5206								
5207	025474	012737	032140	007430		MOV	#<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>, E.A0	;EXPECTED MSG A0
5208	025502	005037	007432			CLR	E.B0	
5209	025506	012737	001720	007434		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>, E.A1	
5210	025514	012737	000001	007436		MOV	#1, E.B1	
5211								
5212	025522	004737	032752			JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
5213	025526	000000				.WORD	0!0!0	; & MSGS SPECIFIED HERE
5214	025530	104035				ERROR	35	;MSG A0 ERROR DURING OFFSET CMD
5215	025532	104061				ERROR	61	;MSH B0 ERROR
5216	025534	104036				ERROR	36	;MSG A1 ERROR
5217	025536	104062				ERROR	62	;MSG B1 ERROR
5218								
5219	025540	005037	001176		645:	CLR	\$ESCAPE	
5220	025544	013737	001416	007376		MOV	T5000, TEMP1	;SETUP TIMEOUT
5221	025552	004737	032640			JSR	PC, FATT2	;FIND ATTN
5222	025556	104034				ERROR	34	;NO ATTN AFTER OFFSET CMD
5223								
5224								
5225	025560	012737	052340	007430		MOV	#<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>, E.A0	;EXPECTED MSG A0
5226	025566	005037	007432			CLR	E.B0	;EXPECTED MSG B0
5227	025572	012737	001720	007434		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>, E.A1	;EXPECTED A1
5228	025600	012737	000001	007436		MOV	#1, E.B1	;MSG ID FOR EXPECTED MSG B1
5229	025606	005037	007440			CLR	E.A2	;EXPECTED MSG A2
5230	025612	012737	000002	007442		MOV	#2, E.B2	;MSG ID FOR EXPECTED MSG B2
5231	025620	012737	000003	007446		MOV	#3, E.B3	;MSG ID FOR EXPECTED MSG B3
5232								
5233	025626	004737	032752			JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
5234	025632	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5235	025634	104260				ERROR	260	;MSG A0 ERROR AFTER OFFSET CMD
5236	025636	104261				ERROR	261	;MSH B0 ERROR
5237	025640	104037				ERROR	37	;MSG A1 ERROR
5238	025642	104040				ERROR	40	;MSG B1 ERROR
5239								
5240								
5241	025644	012765	100000	000000		MOV	#CLR, RKCS1(R5)	
5242	025652	013765	001222	000010		MOV	\$UNIT, RKCS2(R5)	;DRIVE#
5243	025660	012765	000005	000000		MOV	#CLEAR, RKCS1(R5)	;DRIVE CLEAR CMD
5244	025666	013737	001406	007376		MOV	T10, TEMP1	;SETUP TIMEOUT
5245	025674	004737	032230			JSR	PC, FRDY	;FIND RDY
5246	025700	104151				ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD

5247	025702	004737	032512		JSR	PC,TSTATN	;TEST FOR ATTN
5248	025706	000401			BR	65\$	
5249	025710	104154			ERROR	154	;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5250	025712			65\$:			
5251							
5252							
5253	025712	005037	001402		CLR	SECTOR	
5254	025716	013765	001402	000006	4\$:	MOV	SECTOR,RKDA(R5)
5255	025724	012765	001460	000004	MOV	#DATA0,RKBA(R5)	;WRITE ALL 0'S
5256	025732	052765	000020	000010	BIS	#BA1,RKCS2(R5)	;BUS ADDR INCR INH
5257	025740	012765	177400	000002	MOV	#-256.,RKWC(R5)	;FULL SECTOR
5258							
5259	025746	012765	000023	000000	MOV	#(WRDATA),RKCS1(R5)	;WRITE DATA CMD
5260	025754	013737	001420	007376	MOV	T5000,TEMP1	;SETUP TIMEOUT
5261	025762	004737	032230		JSR	PC,FRDY	;FIND RDY
5262	025766	104011			ERROR	11	;NO RDY AFTER WRITE DATA CMD
5263	025770	004737	033566		JSR	PC,GSTAT	;GET FRESH STATUS
5264	025774	032737	100000	007340	BIT	#CERR,HCS1	
5265	026002	001465			BEQ	69\$	;BR IF NO ERRORS
5266							
5267	026004	032737	000200	007354	BIT	#BSE,HER	;SEE IF BAD SECTOR FLAG
5268	026012	001421			BEQ	67\$	;BR IF NO
5269	026014	004737	035610		JSR	PC,TRUERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE
5270	026020	000455			BR	68\$	;RETURN HERE IF NO
5271							
5272	026022	005237	001402		INC	SECTOR	;RETURN HERE IF YES
5273	026026	023727	001402	000012	CMP	SECTOR,#10.	;ARE 10 CONSEC. SECTORS BAD
5274	026034	001003			BNE	66\$	;BR IF NO
5275	026036	104046			ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS
5276	026040	000137	026760		JMP	3\$	;BYPASS TEST
5277	026044	012765	100000	000000	66\$:	MOV	#CCLR,RKCS1(R5)
5278	026052	000137	025716		JMP	4\$	;TRY ANOTHER SECTOR
5279	026056	104012			67\$:	ERROR	12
5280							;CERR WITH WRITE DATA CMD
5281	026060	012737	010340	007430	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
5282	026066	005037	007432		CLR	E.B0	;EXPECTED MSG B0
5283	026072	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
5284	026100	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5285	026106	005037	007440		CLR	E.A2	;EXPECTED MSG A2
5286	026112	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5287	026120	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5288							
5289	026126	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5290	026132	000003			.WORD	T.A2!T.B2!0	;# MSGS SPECIFIED HERE
5291	026134	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5292	026136	104023			ERROR	23	;MSG B0 ERROR
5293	026140	104053			ERROR	53	;MSG A1 ERROR
5294	026142	104025			ERROR	25	;MSG B1 ERROR
5295	026144	104401	045277		TYPE	MSG26	;ABORTING BAL OF TESTS
5296	026150	000137	031514		JMP	\$EOP	
5297	026154	104063			68\$:	ERROR	63
5298	026156				69\$:		;BAD SECTOR NOT LISTED IN TABLE
5299							
5300	026156	012737	010340	007430	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
5301	026164	005037	007432		CLR	E.B0	;EXPECTED MSG B0
5302	026170	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1

K08

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 100  
T15 WRITE WITH HEADS OFFSET

SEQ 0100

5303	026176	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5304	026204	005037	007440		CLR	E.A2	;EXPECTED MSG A2
5305	026210	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5306	026216	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5307							
5308	026224	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5309	026230	000003			WORD	T.A2!T.B2!0	;8 MSGS SPECIFIED HERE
5310	026232	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5311	026234	104023			ERROR	23	;MSG B0 ERROR
5312	026236	104053			ERROR	53	;MSG A1 ERROR
5313	026240	104025			ERROR	25	;MSG B1 ERROR
5314							
5315	026242	005737	001356		TST	CYLDIF	;SEE IF MSG A2=0
5316	026246	001401			BEQ	70\$	;BR IF YES
5317	026250	104104			ERROR	104	;MSG A2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5318	026252	005737	001360	70\$:	TST	CYLADD	;SEE IF MSG B2=0
5319	026256	001401			BEQ	71\$	;BR IF YES
5320	026260	104105			ERROR	105	;MSG B2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5321	026262			71\$:			
5322							
5323	026262	104415			SCOPI		
5324	026264	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
5325							
5326	026270	004737	034140		JSR	PC,SUBCLR	
5327	026274	104024			ERROR	24	;CERR AFTER SCLR
5328							
5329							
5330	026276	012765	001460	000004	MOV	#DATA0,RKBA(R5)	
5331	026304	052765	000020	000010	BIS	#BAI,RKCS2(R5)	
5332	026312	012765	177400	000002	MOV	#-256,RKWC(R5)	
5333	026320	013765	001402	000006	MOV	SECTOR,RKDA(R5)	
5334							
5335	026326	012765	000031	000000	MOV	#(WRTCHK),RKCS1(R5)	;WRITE CHECK CMD
5336	026334	013737	001420	007376	MOV	T5000,TEMP1	;SETUP TIMEOUT
5337	026342	004737	032230		JSR	PC,FRDY	;FIND RDY
5338	026346	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
5339	026350	004737	033566		JSR	PC,GSTAT	;GET FRESH STATUS
5340	026354	032737	100000	007340	BIT	#CERR,HCS1	
5341	026362	001453			BEQ	73\$	
5342	026364	032737	040000	007342	BIT	#WCE,HCS2	;SEE IF WRITE CHECK ERROR
5343	026372	001410			BEQ	72\$	
5344	026374	016537	000024	001436	MOV	RKDB(R5),WD1	;ACTUAL WORD FOR PRINTOUT
5345	026402	013737	001460	001440	MOV	DATA0,WD2	;EXPECTED WORD FOR TYPEOUT
5346	026410	104016			ERROR	16	;WCE AFTER WRITE CMD
5347	026412	000437			BR	73\$	
5348							
5349	026414	104022		72\$:	ERROR	22	;CERR AFTER WRITE CHECK CMD
5350							
5351	026416	012737	010340	007430	MOV	#(O!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
5352	026424	005037	007432		CLR	E.B0	;EXPECTED MSG B0
5353	026430	012737	001720	007434	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
5354	026436	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5355	026444	005037	007440		CLR	E.A2	;EXPECTED MSG A2
5356	026450	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5357	026456	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5358							



```

5359 026464 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5360 026470 000003 .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
5361 026472 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
5362 026474 104031 ERROR 31 ;MSH B0 ERROR
5363 026476 104060 ERROR 60 ;MSG A1 ERROR
5364 026500 104032 ERROR 32 ;MSG B1 ERROR
5365 026502 104401 045277 TYPE MSG26 ;ABORTING BAL OF TESTS
5366 026506 000137 031514 JMP $EOP

5367
5368 026512 73$:
5369
5370 026512 012737 010340 007430 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5371 026520 005037 007432 CLR E.B0 ;EXPECTED MSG B0
5372 026524 012737 001720 007434 MOV #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
5373 026532 012737 000001 007436 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5374 026540 005037 007440 CLR E.A2 ;EXPECTED MSG A2
5375 026544 012737 000002 007442 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5376 026552 012737 000003 007446 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5377
5378 026550 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5379 026564 000003 .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
5380 026566 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
5381 026570 104031 ERROR 31 ;MSH B0 ERROR
5382 026572 104060 ERROR 60 ;MSG A1 ERROR
5383 026574 104032 ERROR 32 ;MSG B1 ERROR
5384
5385 026576 020027 000260 CMP R0,#260
5386 026602 001004 BNE 2$ ;BR IF JUST DID POS OFFSET
5387 026604 012700 000060 MOV #60,R0 ;ELSE SETUP FOR POS OFFSET
5388 026610 000137 025432 JMP 1$

5389
5390 026614 2$:
5391
5392 026614 012765 100000 000000 MOV #CCLR,RKCS1(R5)
5393 026622 013765 001222 000010 MOV $UNIT,RKCS2(R5)
5394 026630 012765 000013 000000 MOV #RECAL,RKCS1(R5) ;RECAL CMD
5395 ;RESET CYL DIFF/OFFSET & CYL ADDR REG
5396 ;IN RKMR2 & RKMR3 RESP.
5397 026636 013737 001406 007376 MOV T10,TEMP1 ;SETUP TIMEOUT
5398 026644 004737 032230 JSR PC,FRDY ;FIND RDY
5399 026650 104124 ERROR 124 ;RDY NOT SET AFTER RECAL CMD
5400
5401 026652 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
5402 026660 004737 033566 JSR PC,GSTAT
5403 026664 032737 020000 007366 BIT #0,RTZ,HMR2
5404 026672 001001 BNE 74$
5405 026674 104244 ERROR 244 ;RTZ NOT SET DURING RECAL CMD
5406 026676 013737 001406 007400 MOV T10,TEMP2 ;SETUP TIMEOUT
5407 026704 004737 032544 JSR PC,FATT1 ;FIND ATTN
5408 026710 104055 ERROR 55 ;NO ATTN AFTER RECAL CMD
5409
5410 026712 012765 100000 000000 MOV #CCLR,RKCS1(R5)
5411 026720 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
5412 026726 012765 000005 000000 MOV #CLEAR,RKCS1(R5) ;DRIVE CLEAR CMD
5413 026734 013737 001406 007376 MOV T10,TEMP1 ;SETUP TIMEOUT
5414 026742 004737 032230 JSR PC,FRDY ;FIND RDY

```

5415 026746 104151  
5416 026750 004737 032512  
5417 026754 000401  
5418 026756 104154  
5419 026760  
5420  
5421  
5422 026760  
5423  
5424  
5425  
5426  
5427  
5428  
5429  
5430  
5431  
5432  
5433  
5434  
5435  
5436  
5437  
5438  
5439  
5440 026760 000004  
5441 026762 012737 000001 001174  
5442 026770 012706 001100  
5443  
5444 026774 012700 001422  
5445  
5446 027000 004737 034140  
5447 027004 104024  
5448  
5449 027006 011065 000020  
5450 027012 012765 001466 000004  
5451 027020 052765 000020 000010  
5452 027026 012765 076000 000002  
5453  
5454 027034 012765 000023 000000  
5455 027042 013737 001420 007376  
5456 027050 004737 032230  
5457 027054 104011  
5458 027056 004737 033566  
5459 027062 032737 100000 007340  
5460 027070 001451  
5461  
5462 027072 032737 000200 007354  
5463 027100 001405  
5464 027102 004737 035610  
5465 027106 000441  
5466 027110 000137 027576  
5467 027114 104012  
5468  
5469 027116 012737 010340 007430  
5470 027124 005037 007432

ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD  
JSR PC,TSTATN ;TEST FOR ATTN  
BR 75\$  
ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD  
75\$:  
3\$:  
\*\*\*\*\*  
\*TEST 16 TEST CURRENT CROSS-OVER CYLINDERS  
\*  
\* THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF  
\* CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:  
\*  
\* SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y  
\* WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.  
\*  
\* A WRITE CHECK IS THEN PERFORMED TO VERIFY DATA WAS PROPERLY WRITTEN.  
\* THIS TEST IS PERFORMED FOR ALL 3 HEADS.  
\*  
\* CYLINDER X: 63 127 191 255 319 383  
\* CYLINDER Y: 64 128 192 256 320 384  
\*  
\*\*\*\*\*  
†ST16: SCOPE  
MOV #1,STIMES ;;DO 1 ITERATION  
MOV #STACK,SP  
MOV #CYL,RO ;CYL ADDR TABLE  
1\$: JSR PC,SUBCLR  
ERROR 24 ;CERR AFTER SCLR  
MOV (RO),RKDC(R5) ;CYL #  
MOV #DPAT1,RKBA(R5) ;DATA PATTERN  
BIS #BAI,RKCS2(R5) ;BUSS ADDR INCREMENT INHIBIT  
MOV #-6\*22.\*256.,RKWC(R5) ;WORD COUNT TO SPIRAL & FILL 2 CYLINDERS  
MOV #<WRDATA>,RKCS1(R5) ;WRITE DATA CMD  
MOV T5000,TEMP1 ;SETUP TIMEOUT  
JSR PC,FROY ;FIND RDY  
ERROR 11 ;NO RDY AFTER WRITE DATA CMD  
JSR PC,GSTAT ;GET FRESH STATUS  
BIT #CERR,HCS1  
BEQ 67\$ ;BR IF NO ERRORS  
BIT #BSE,HER ;SEE IF BAD SECTOR FLAG  
BEQ 65\$ ;BR IF NO  
JSR PC,TRUERR ;ELSE SEE IF SECTOR LISTED IN BSE TABLE  
BR 66\$ ;RETURN HERE IF NO  
JMP 3\$ ;RET HERE IF YES  
65\$: ERROR 12 ;CERR WITH WRITE DATA CMD  
MOV #<O!D.SPIN!D.DROY!D.VV!D.LRA>,E.AO ;EXPECTED MSG AO  
CLR E.BO ;EXPECTED MSG BO

# N08

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 103  
T16 TEST CURRENT CROSS-OVER CYLINDERS

SEQ 0103

5471	027130	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5472	027136	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5473	027144	005037	007440		CLR	E.A2	;EXPECTED MSG A2
5474	027150	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5475	027156	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5476							
5477	027164	004737	032752		JSP	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5478	027170	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5479	027172	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5480	027174	104023			ERROR	23	;MSH B0 ERROR
5481	027176	104053			ERROR	53	;MSG A1 ERROR
5482	027200	104025			ERROR	25	;MSG B1 ERROR
5483	027202	104401	045277		TYPE	MSG26	;ABORTING BAL OF TESTS
5484	027206	000137	031514		JMP	\$EOP	
5485	027212	104063			ERROR	63	;BAD SECTOR NOT LISTED IN TABLE
5486	027214						
5487							
5488	027214	012737	010340	007430	MOV	#<D!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5489	027222	005037	007432		CLR	E.B0	;EXPECTED MSG B0
5490	027226	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5491	027234	012737	000001	007436	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5492	027242	005037	007440		CLR	E.A2	;EXPECTED MSG A2
5493	027246	012737	000002	007442	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5494	027254	012737	000003	007446	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5495							
5496	027262	004737	032752		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5497	027266	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5498	027270	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5499	027272	104023			ERROR	23	;MSH B0 ERROR
5500	027274	104053			ERROR	53	;MSG A1 ERROR
5501	027276	104025			ERROR	25	;MSG B1 ERROR
5502							
5503	027300	011065	000020		MOV	(R0),RKDC(R5)	
5504	027304	012765	00146E	000004	MOV	#DPAT1,RKBA(R5)	
5505	027312	052765	000020	000010	BIS	#BAI,RKCS2(R5)	
5506	027320	012765	076000	000002	MOV	#-6*22.*256.,RKWC(R5)	
5507							
5508	027326	012765	000031	000000	MOV	#<WRTCHK>,RKCS1(R5)	;WRITE CHECK CMD
5509	027334	013737	001420	007376	MOV	T5000,TEMP1	;SETUP TIMEOUT
5510	027342	004737	032230		JSR	PC,FRDY	;FIND RDY
5511	027346	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
5512	027350	004737	033566		JSR	PC,GSTAT	;GET FRESH STATUS
5513	027354	032737	100000	007340	BIT	#CERR,HCS1	
5514	027362	001453			BEQ	69\$	
5515	027364	032737	040000	007342	BIT	#WCE,HCS2	;SEE IF WRITE CHECK ERROR
5516	027372	001410			BEQ	68\$	
5517	027374	016537	000024	001436	MOV	RKDB(R5),WD1	;ACTUAL WORD FOR PRINTOUT
5518	027402	013737	001466	001440	MOV	DPAT1,WD2	;EXPECTED WORD FOR TYPEOUT
5519	027410	104016			ERROR	16	;WCE AFTER WRITE CMD
5520	027412	000437			BR	69\$	
5521							
5522	027414	104022			ERROR	22	;CERR AFTER WRITE CHECK CMD
5523							
5524	027416	012737	010340	007430	MOV	#<D!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5525	027424	005037	007432		CLR	E.B0	;EXPECTED MSG B0
5526	027430	012737	001720	007434	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1

```

5527 027436 012737 000001 007436 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5528 027444 005037 007440 CLR E.A2 ;EXPECTED MSG A2
5529 027450 012737 000002 007442 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5530 027456 012737 000003 007446 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5531
5532 027464 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5533 027470 000003 .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
5534 027472 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
5535 027474 104031 ERROR 31 ;MSH B0 ERROR
5536 027476 104060 ERROR 60 ;MSG A1 ERROR
5537 027500 104032 ERROR 32 ;MSG B1 ERROR
5538 027502 104401 045277 TYPE MSG26 ;ABORTING BAL OF TESTS
5539 027506 000137 031514 JMP $EOP
5540
5541 027512 69$:
5542
5543 027512 012737 010340 007430 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5544 027520 005037 007432 CLR E.B0 ;EXPECTED MSG B0
5545 027524 012737 001720 007434 MOV #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
5546 027532 012737 000001 007436 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5547 027540 005037 007440 CLR E.A2 ;EXPECTED MSG A2
5548 027544 012737 000002 007442 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5549 027552 012737 000003 007446 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5550
5551 027560 004737 032752 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5552 027564 000003 .WORD T.A2!T.B2!0 ; & MSGS SPECIFIED HERE
5553 027566 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
5554 027570 104031 ERROR 31 ;MSH B0 ERROR
5555 027572 104060 ERROR 60 ;MSG A1 ERROR
5556 027574 104032 ERROR 32 ;MSG B1 ERROR
5557 027576 022027 000577 3$: CMP (R0)+,#383. ;ALL CYLINDERS DONE?
5558 027602 001402 BEQ 4$ ;BR IF YES
5559 027604 000137 027000 JMP 1$ ;ELSE REPEAT
5560
5561 027610 4$:
5562
5563 027610 012765 100000 000000 MOV #CLR,RKCS1(R5)
5564 027616 013765 001222 000010 MOV $UNIT,RKCS2(R5)
5565 027624 012765 000013 000000 MOV #RECAL,RKCS1(R5) ;RECAL CMD
5566 ;RESET CYL DIFF/OFFSET & CYL ADDR REG
5567 ;IN RKMR2 & RKMR3 RESP.
5568 027632 013737 001406 007376 MOV T10,TEMP1 ;SETUP TIMEOUT
5569 027640 004737 032230 JSR PC,FRDY ;FIND RDY
5570 027644 104124 ERROR 124 ;RDY NOT SET AFTER RECAL CMD
5571
5572 027646 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
5573 027654 004737 033566 JSR PC,GSTAT
5574 027660 032737 020000 007366 BIT #D.RTZ,HMR2
5575 027666 001001 BNE 70$
5576 027670 104244 ERROR 244 ;RTZ NOT SET DURING RECAL CMD
5577 027672 013737 001406 007400 70$: MOV T10,TEMP2 ;SETUP TIMEOUT
5578 027700 004737 032544 JSR PC,FATT1 ;FIND ATTN
5579 027704 104055 ERROR 55 ;NO ATTN AFTER RECAL CMD
5580
5581 027706 012765 100000 000000 MOV #CLR,RKCS1(R5)
5582 027714 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#

```

```

5583 027722 012765 000005 000000      MOV      #CLEAR,RKCS1(R5)      ;DRIVE CLEAR CMD
5584 027730 013737 001406 007376      MOV      T10,TEMP1            ;SETUP TIMEOUT
5585 027736 004737 032230      JSR      PC,FRDY              ;FIND RDY
5586 027742 104151                ERROR    151                  ;NO RDY AFTER DRIVE CLEAR CMD
5587 027744 004737 032512      JSR      PC,TSTATN           ;TEST FOR ATTN
5588 027750 000401                BR       71$
5589 027752 104154                ERROR    154                  ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5590 027754

```

71\$:

```

5591
5592
5593 027754
5594
5595
5596
5597
5598
5599

```

2\$:

```

*****
*TEST 17      TEST HEAD SWITCHING TIME
*
*      TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.
*
*      1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS
*         TO SECTOR 25(8) IS ISSUED.
*      2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL
*         A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).
*      3. SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN
*         THE START OF THE WRITE COMMAND (FROM SECTOR 25(8), HEAD 0; TO
*         SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS
*
*      THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2
*
*      THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT

```

```

5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612 *****

```

\*\*\*\*\*

```

5613 027754 000004      TST17: SCOPE
5614 027756 012737 000001 001174      MOV      #1,STIMES           ;;DO 1 ITERATION
5615 027764 012706 001100                MOV      #STACK,SP
5616
5617 027770 005737 007506                TST      DOTIM               ;BYPASS THIS TEST IF
5618 027774 001001                BNE     1$                   ;NEITHER L OR P CLOCK PRESENT
5619 027776 000520                BR       TST20               ;;GO TO NEXT TEST
5620
5621 030000 012737 000025 007406 1$:      MOV      #25,TEMPS           ;HEAD 0, SECTOR 21 TO BE PUT IN RKDA
5622 030006 004737 034140 2$:      JSR      PC,SUBCLR          ;CERR AFTER SCLR
5623 030012 104024                ERROR    24
5624
5625 030014 004737 034710                JSR      PC,FSEC23          ;FIND SECTOR 23
5626 030020 104106                ERROR    106                ;CANNOT FIND SECTOR 23
5627 030022 000506                BR       TST20               ;;GO TO NEXT TEST
5628
5629 030024 012765 001462 000004      MOV      #DATA01,RKBA(R5)    ;DATA 0101
5630 030032 052765 000020 000010      BIS      #BAI,RKCS2(R5)     ;BUSS ADDR INCREMENT INHIBIT
5631 030040 012765 177000 000002      MOV      #-512,RKWC(R5)     ;WORD COUNT
5632 030046 013765 007406 000006      MOV      TEMPS,RKDA(R5)     ;HEAD & SECTOR
5633 030054 012765 000023 000000      MOV      #WRDATA,RKCS1(R5) ;WRITE DATA CMD
5634
5635 030062 012727 002000 007376      MOV      #2000,#TEMP1
5636 030070 004737 032720                JSR      PC,DL$              ;DO DELAY
5637
5638 030074 032765 000200 000000 7$:      BIT      #RDY,RKCS1(R5)     ;LOOK FOR CONTROLLER READY

```

```

5639 030102 001006          BNE      B$
5640 030104 004737 032230 JSR      PC,FRDY      ;FIND RDY AND GET FRESH STATUS
5641 030110 104011          ERROR    11          ;NO RDY AFTER SEL DRV CMD
5642 030112 004737 033566 JSR      PC,GSTAT
5643 030116 104107          ERROR    107         ;HEAD SWITCHING LONGER THAN DELAY
5644
5645 030120 032737 100000 007340 B$:      BIT      #CERR,HCS1
5646 030126 001444          BEQ      3$
5647 030130 104012          ERROR    12          ;CERR AFTER WRITE DATA
5648
5649 030132 012737 010340 007430 MOV      #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5650 030140 005037 007432 CLR      E.B0          ;EXPECTED MSG B0
5651 030144 012737 001720 007434 MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5652 030152 012737 000001 007436 MOV      #1,E.B1       ;MSG ID FOR EXPECTED MSG B1
5653 030160 005037 007440 CLR      E.A2          ;EXPECTED MSG A2
5654 030164 012737 000002 007442 MOV      #2,E.B2       ;MSG ID FOR EXPECTED MSG B2
5655 030172 012737 000003 007446 MOV      #3,E.B3       ;MSG ID FOR EXPECTED MSG B3
5656
5657 030200 004737 032752 JSR      PC,CHKMSG    ;CHECK MSGS A0, B0, A1, B1
5658 030204 000000          .WORD   0!0!0       ;8 MSGS SPECIFIED HERE
5659 030206 104052          ERROR    52          ;MSG A0 ERROR AFTER WRITE DATA CMD
5660 030210 104023          ERROR    23          ;MSG B0 ERROR
5661 030212 104053          ERROR    53          ;MSG A1 ERROR
5662 030214 104025          ERROR    25          ;MSG B1 ERROR
5663
5664 030216 023727 007406 000425 CMP      TEMPS,#425   ;HEAD 1,SECTOR 21 DONE?
5665 030224 001405          BEQ      TST20       ;GO TO NEXT TEST
5666 030226 012737 000425 007406 MOV      #425,TEMPS
5667 030234 000137 030006 JMP      2$          ;ELSE REPEAT FOR HEAD 1, SECTOR 21
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688

```

```

*****
*TEST 20      DRIVE OFF TRACK TEST
*
*      THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND
*      THE ALLOTTED 3MS.
*
*      1.  INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS
*           (UNIQUE TO EACH CYLINDER)
*      2.  A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO 1.
*           AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE
*           HEADS WILL IMMEDIATELY BE WRITTEN.
*      3.  IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS,
*           DRIVE OFF TRACK ERROR WILL SET.
*
*      IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDERS.
*      100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER
*      BEFORE DOING THE NEXT CYLINDER
*
*****

```

```

5689 030240 000004          TST20:  SCOPE
5690 030242 012737 000001 001174 MOV      #1,STIMES    ;DO 1 ITERATION
5691 030250 012706 001100 MOV      #STACK,SP    ;RESTORE STK PTR
5692 030254 005237 007324 INC      BADHOR        ;USED FOR VALID HALT
5693 030260 005037 001346 CLR      TOCYL
5694 030264 012737 100000 007406 MOV      #BIT15,TEMPS

```

## E09

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13MACY11 27(1006) 01-FEB-77 04:10 PAGE 107  
T20 DRIVE OFF TRACK TEST

SEQ 0107

5695									
5696	030272	004737	034140			15:	JSR	PC, SUBCLR	
5697	030276	104024					ERROR	24	;CERR AFTER SCLR
5698									
5699	030300	012700	001506				MOV	#HDTAB, RO	;FORMAT HEADERS ON ALL MAJOR CYL.
5700									
5701	030304	013720	001346			25:	MOV	TOCYL, (RO)+	;HEADER WORD 0: CYL #
5702	030310	012720	140000				MOV	#140000, (RO)+	;HEADER WORD 1: ALL SECTOR 0
5703	030314	012710	140000				MOV	#140000, (RO)	;HEADER WORD 2: XOR OF 0 & 1
5704	030320	053720	001346				BIS	TOCYL, (RO)+	;ADD CYL # TO WORD 2
5705									
5706	030324	020027	001712				CMP	RO, #HDTAB+132.	;ALL 22 SECTORS DONE? (22X6=132)
5707	030330	001365					BNE	25	;BR IF NO
5708									
5709	030332	012765	001506	000004			MOV	#HDTAB, RKBA(R5)	
5710	030340	012765	177676	000002			MOV	#-66., RKWC(R5)	
5711	030346	013765	001346	000020			MOV	TOCYL, RKDC(R5)	
5712									
5713	030354	012765	000027	000000			MOV	#<WRHEAD>, RKCS1(R5)	;WRITE HEADER CMD
5714	030362	013737	001420	007376			MOV	T50000, TEMP1	;SETUP TIMEOUT
5715	030370	004737	032230				JSR	PC, FRDY	;FIND RDY
5716	030374	104200					ERROR	200	;NO RDY AFTER WRITE HEADER CMD
5717	030376	004737	033566				JSR	PC, GSTAT	;GET FRESH STATUS
5718	030402	032737	100000	007340			BIT	#CERR, HCS1	
5719	030410	001405					BEQ	645	
5720	030412	104201					ERROR	201	;CERR AFTER WRITE HEADER CMD
5721	030414	104401	045277				TYPE	MSG26	;ABORTING BAL OF TESTS
5722	030420	000137	031514				JMP	\$EOP	
5723	030424					645:			
5724									
5725									
5726	030424	006137	007406				ROL	TEMPS	;SET CARRY ONLY ONCE
5727	030430	006137	001346				ROL	TOCYL	;SELECT NEXT MAJOR CYL
5728	030434	023727	001346	001000			CMP	TOCYL, #1000	;ALL MAJOR CYL FORMATTED?
5729	030442	001313					BNE	15	;BR IF NO
5730	030444	005065	000020				CLR	RKDC(R5)	;SETUP TO RETURN TO CYL 0
5731									
5732	030450	012765	000017	000000			MOV	#SEEK, RKCS1(R5)	;SEEK CMD
5733	030456	013737	001410	007376			MOV	T50, TEMP1	;SETUP TIMEOUT
5734	030464	004737	032230				JSR	PC, FRDY	;FIND RDY
5735	030470	104131					ERROR	131	;NO RDY AFTER SEEK CMD
5736									
5737	030472	013737	001420	007376			MOV	T50000, TEMP1	;SETUP TIMEOUT
5738	030500	004737	032640				JSR	PC, FAT12	;FIND ATTN
5739	030504	104132					ERROR	132	;NO ATTN AFTER SEEK CMD
5740									
5741	030506	032737	100000	007340			BIT	#CERR, HCS1	
5742	030514	001401					BEQ	655	
5743	030516	104210					ERROR	210	;CERR AFTER SEEK CMD
5744									
5745	030520					655:			
5746									
5747	030520	012737	031226	001176			MOV	#FORM, \$ESCAPE	
5748	030526	005000					CLR	RO	;ITERATION COUNTER
5749	030530	012737	000001	001346			MOV	#1, TOCYL	;SETUP TO CYL #
5750	030536	005037	001344				CLR	FRCYL	

```

5751
5752 030542 104415          SCOPI
5753 030544 012706 001100  MOV      #STACK,SP          ;RESTORE STK PTR
5754
5755 030550 013737 001346 001354  MOV      TOCYL,CALC.F      ;SETUP FOR ERROR PRINTOUT
5756
5757 030556 004737 034140      3$: JSR      PC,SUBCLR
5758 030562 104024          ERROR    24                ;CERR AFTER SCLR
5759
5760 030564 013765 001346 000020  MOV      TOCYL,RKDC(R5)    ;GO TO CYL #
5761 030572 012765 001464 000004  MOV      #DATA1,RKBA(R5)   ;ALL 1'S
5762 030600 052765 000020 000010  BIS      #BA1,RKCS2(R5)
5763 030606 012765 177400 000002  MOV      #-256.,RKWC(R5)  ; SECTOR TO BE ALL 1'S
5764
5765 030614 012765 000023 000000  MOV      #WRDATA,RKCS1(R5) ;WRITE DATA CMD
5766 030622 013737 001420 007376  MOV      T50000,TEMP1
5767 030630 004737 032230      JSR      PC,FRDY          ;FIND RDY
5768 030634 104011          ERROR    11                ;NO RDY AFTER WRITE DATA CMD.
5769
5770 030636 004737 033566          JSR      PC,GSTAT         ;GET FRESH STATUS
5771 030642 032737 020000 007370  BIT      #D.DROT,HMR3     ;SEE IF DRIVE OFF TRACK
5772 030650 001401          BEQ      5$
5773 030652 104112          ERROR    112             ;DRIVE OFF TRACK AFTER WRITE DATA CMD
5774
5775 030654 032737 100000 007340  5$: BIT      #CERR,HCS1
5776 030662 001401          BEQ      6$
5777 030664 104012          ERROR    12                ;CERR SET AFTER WRITE DATA CMD
5778
5779 030666          6$:
5780
5781 030666 012737 010340 007430  MOV      #<D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5782 030674 005037 007432          CLR      E.B0             ;EXPECTED MSG B0
5783 030700 012737 001720 007434  MOV      #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
5784 030706 012737 000001 007436  MOV      #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
5785 030714 005037 007440          CLR      E.A2            ;EXPECTED MSG A2
5786 030720 012737 000002 007442  MOV      #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
5787 030726 012737 000003 007446  MOV      #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
5788
5789 030734 004737 032752      JSR      PC,CHKMSG        ;CHECK MSGS A0, B0, A1, B1
5790 030740 000003          .WORD    T.A2!T.B2!0     ;& MSGS SPECIFIED HERE
5791 030742 104052          ERROR    52                ;MSG A0 ERROR AFTER WRITE DATA CMD
5792 030744 104023          ERROR    23                ;MSG B0 ERROR
5793 030746 104053          ERROR    53                ;MSG A1 ERROR
5794 030750 104025          ERROR    25                ;MSG B1 ERROR
5795 030752 023737 001360 001346  CMP      CYLADD,TOCYL
5796 030760 001401          BEQ      7$
5797 030762 104113          ERROR    113             ;CYL ADDR IN RKMR3 NOT = RKDC
5798
5799 030764          7$:
5800 030764 104415          SCOPI
5801 030766 012706 001100  MOV      #STACK,SP          ;RESTORE STK PTR
5802
5803 030772 004737 034140      JSR      PC,SUBCLR
5804 030776 104024          ERROR    24                ;CERR AFTER SCLR
5805
5806

```



5807	031000	012765	001464	000004		MOV	#DATA1,RKBA(R5)	
5808	031006	052765	000020	000010		BIS	#BA1,RKCS2(R5)	
5809	031014	012765	177400	000002		MOV	#-256.,RKWC(R5)	
5810								
5811	031022	012765	000023	000000		MOV	#WRDATA,RKCS1(R5)	
5812	031030	013737	001420	007376		MOV	T50000,TEMP1	
5813	031036	004737	032230			JSR	PC,FRDY	;FIND RDY
5814	031042	104011				ERROR	11	;NO RDY AFTER WRITE DATA CMD
5815								
5816	031044	004737	033566			JSR	PC,GSTAT	;GET FRESH STATUS
5817	031050	032737	020000	007370		BIT	#D.DROT,HMR3	
5818	031056	001401				BEQ	8\$	
5819	031060	104112				ERROR	112	;DRIVE OFF TRACK AFTER WRITE DATA CMD
5820								
5821	031062	032737	100000	007340	8\$:	BIT	#CERR,HCS1	
5822	031070	001401				BEQ	9\$	
5823	031072	104012				ERROR	12	;CERR AFTER WRITE DATA CMD
5824								
5825	031074				9\$:			
5826								
5827	031074	012737	010340	007430		MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5828	031102	005037	007432			CLR	E.B0	;EXPECTED MSG B0
5829	031106	012737	001720	007434		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5830	031114	012737	000001	007436		MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5831	031122	005037	007440			CLR	E.A2	;EXPECTED MSG A2
5832	031126	012737	000002	007442		MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5833	031134	012737	000003	007446		MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5834								
5835	031142	004737	032752			JSR	PC,CHKMSG	;CHECK MSGS A0 B0 A1 B1
5836	031146	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5837	031150	104052				ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5838	031152	104023				ERROR	23	;MSG B0 ERROR
5839	031154	104053				ERROR	53	;MSG A1 ERROR
5840	031156	104025				ERROR	25	;MSG B1 ERROR
5841	031160	005737	001360			TST	CYLADD	
5842	031164	001401				BEQ	10\$	
5843	031166	104042				ERROR	42	;NOT BACK TO CYL 0
5844								
5845	031170	005200			10\$:	INC	RO	
5846	031172	020027	000144			CMP	RO,#100.	;ALL ITERATIONS DONE?
5847	031176	001402				BEQ	13\$	;BR IF YES
5848	031200	000137	030556			JMP	3\$	;ELSE DO AGAIN
5849								
5850	031204	005000			13\$:	CLR	RO	;RESET ITERATION CTR
5851	031206	006337	001346			ASL	TOCYL	
5852	031212	023727	001346	001000		CMP	TOCYL,#1000	;ALL MAJOR CYL DONE?
5853	031220	001402				BEQ	FORM	;BR IF YES
5854	031222	000137	030556			JMP	3\$	;ELSE DO NEXT CYL
5855								
5856	031226	005037	001346		FORM:	CLR	TOCYL	;RESTORE TO ORIG 22 SECTOR FORMAT.
5857	031232	012737	100000	007406		MOV	#BIT15,TEMP5	
5858								
5859	031240	004737	034140		12\$:	JSR	PC,SUBCLR	
5860	031244	104024				ERROR	24	;CERR AFTER SCLR
5861								
5862	031246	012765	001506	000004		MOV	#HDTAB,RKBA(R5)	;REFORMAT ALL MAJOR CYLINDERS

5863	031254	012765	177676	000002	MOV	#-66.,RKWC(R5)	
5864	031262	013765	001346	000020	MOV	TOCYL,RKDC(R5)	
5865							
5866	031270	013737	001346	001362	MOV	TOCYL,CALADD	; SETUP
5867	031276	012737	000000	001444	MOV	#0,HEAD	; TO FILL
5868	031304	012737	000000	001452	MOV	#0,FORMAT	; HEADER
5869	031312	004737	035122		JSR	PC,FHDTAB	; TABLE
5870							
5871							
5872	031316	012765	000027	000000	MOV	#(WRHEAD),RKCS1(R5)	; WRITE HEADER CMD
5873	031324	013737	001420	007376	MOV	T50000,TEMP1	; SETUP TIMEOUT
5874	031332	004737	032230		JSR	PC,FRDY	; FIND RDY
5875	031336	104200			ERROR	200	; NO RDY AFTER WRITE HEADER CMD
5876	031340	004737	033566		JSR	PC,GSTAT	; GET FRESH STATUS
5877	031344	032737	100000	007340	BIT	#CERR,HCS1	
5878	031352	001405			BEQ	64\$	
5879	031354	104201			ERROR	201	; CERR AFTER WRITE HEADER CMD
5880	031356	104401	045277		TYPE	MSG26	; ABORTING BAL OF TESTS
5881	031362	000137	031514		JMP	\$EOP	
5882	031366						
5883							
5884							
5885	031366	006137	007406		ROL	TEMPS	
5886	031372	006137	001346		ROL	TOCYL	
5887	031376	023727	001346	001000	CMP	TOCYL,#1000	; ALL MAJOR CYL REFORMATTED?
5888	031404	001315			BNE	12\$	; BR IF NO
5889							
5890	031406	005065	000020		CLR	RKDC(R5)	; SETUP TO RETURN TO CYL 0
5891	031412	005037	001176		CLR	\$ESCAPE	
5892							
5893	031416	012765	000017	000000	MOV	#SEEK,RKCS1(R5)	; SEEK CMD
5894	031424	013737	001410	007376	MOV	T50,TEMP1	; SETUP TIMEOUT
5895	031432	004737	032230		JSR	PC,FRDY	; FIND RDY
5896	031436	104131			ERROR	131	; NO RDY AFTER SEEK CMD
5897							
5898	031440	013737	001420	007376	MOV	T50000,TEMP1	; SETUP TIMEOUT
5899	031446	004737	032640		JSR	PC,FATT2	; FIND ATTN
5900	031452	104132			ERROR	132	; NO ATTN AFTER SEEK CMD
5901							
5902	031454	032737	100000	007340	BIT	#CERR,HCS1	
5903	031462	001401			BEQ	65\$	
5904	031464	104210			ERROR	210	; CERR AFTER SEEK CMD
5905							
5906	031466						
5907							
5908							
5909	031466	005737	007326		TST	HPEND	; SEE IF HALT PENDING
5910	031472	001406			BEQ	4\$	; BR IF NO
5911	031474	005037	007326		CLR	HPEND	; CLEAR FOR FUTURE FORMATTING
5912	031500	005037	007324		CLR	BADHDR	; HEADERS NOW OK
5913	031504	000137	036172		JMP	STOP	; GO BACK & HALT CPU
5914							
5915	031510	005037	007324		CLR	BADHDR	; HEADERS NOW OK
5916							

64\$:

65\$:

4\$:

```

5917 .SBTTL END OF PASS ROUTINE
5918
5919 ;:*****
5920 ;:INCREMENT THE PASS NUMBER ($PASS)
5921 ;:TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
5922 ;:IF THERES A MONITOR GO TO IT
5923 ;:IF THERE ISN'T JUMP TO STS
5924
5925 031514 $EOP:
5926
5927 031514 000004 SCOPE
5928 031516 012737 000001 001174 MOV #1,$TIMES
5929 031524 012706 001100 MOV #STACK,SP
5930 031530 005237 001220 INC $DEVCT ;INCR COUNT FOR # DRIVES CHECKED
5931 031534 023737 007460 001220 CMP DRIVS,$DEVCT ;ARE ALL DRIVES PRESENT TESTED?
5932 031542 001403 BEQ $EOP1+2 ;BR IF YES
5933 031544 000137 014764 JMP NUDRV ;ELSE TEST NEXT DRIVE PRESENT
5934 031550 000004 $EOP1: SCOPE
5935 031552 005037 001102 CLR $STNM ;;ZERO THE TEST NUMBER
5936 031556 005037 001174 CLR $TIMES ;;ZERO THE NUMBER OF ITERATIONS
5937 031562 005237 001216 INC $PASS ;;INCREMENT THE PASS NUMBER
5938 031566 042737 100000 001216 BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
5939 031574 005327 DEC (PC)+ ;;LOOP?
5940 031576 000001 $EOPCT: .WORD 1
5941 031600 003022 BGT $DOAGN ;;YES
5942 031602 012737 MOV (PC)+,2(PC)+ ;;RESTORE COUNTER
5943 031604 000001 $ENDCT: .WORD 1
5944 031606 031576 $EOPCT
5945 031610 104401 031655 TYPE $SENDMG ;;TYPE "END PASS #"
5946 031614 013746 001216 MOV $PASS,-(SP) ;;SAVE $PASS FOR TYPEOUT
5947 031620 104405 TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
5948 031622 104401 031652 TYPE $ENULL ;;TYPE A NULL CHARACTER
5949 031626 013700 000042 $GET42: MOV #42,R0 ;;GET MONITOR ADDRESS
5950 031632 001405 BEQ $DOAGN ;;BRANCH IF NO MONITOR
5951 031634 000005 RESET ;;CLEAR THE WORLD
5952 031636 004710 $ENDAD: JSR PC,(R0) ;;GO TO MONITOR
5953 031640 000240 NOP ;;SAVE ROOM
5954 031642 000240 NOP ;;FOR
5955 031644 000240 NOP ;;ACT11
5956 031646 $DOAGN:
5957 031646 000137 JMP 2(PC)+ ;;RETURN
5958 031650 013340 $RTNAD: .WORD STS
5959 031652 377 377 000 $ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
5960 031655 015 042412 042116 $SENDMG: .ASCIZ <15><12>/END PASS #/
5961 031662 050040 051501 020123
5962 031670 000043

```

```

5963 .SBTTL SUBROUTINES
5964 ;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM
5965 ;
5966
5967 CLRFLG: MOV #DDUMP, R0
5968 031672 012700 007450
5969 031676 012701 177757
5970 031702 005020
5971 031704 005201
5972 031706 001375
5973 031710 000207
5974
5975
5976 ;TYPE PROGRAM ID IF FTITLE=0
5977 ;
5978
5979 031712 005737 001340
5980 031716 001024
5981 031720 005237 001340
5982 031724 104401 043352
5983
5984 031730 005737 000042
5985 031734 001012
5986 031736 123727 001230 000001
5987 031744 001406
5988 031746 023727 001140 000176
5989 031754 001005
5990 031756 104406
5991 031760 000403
5992 031762 112737 000001 001134
5993 031770
5994 031770 000207
5995
5996
5997 ;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
5998 ;DRIVS, DRIVO-DRIV7 REGISTERS APPROPRIATELY
5999 ;
6000
6001 031772 104411
6002 031774 012600
6003 031776 012701 177770
6004 032002 112002
6005 032004 042702 177400
6006 032010 012703 007462
6007 032014 012704 000060
6008
6009 032020 020402
6010 032022 001415
6011 032024 005723
6012 032026 005204
6013 032030 020427 000070
6014 032034 001371
6015 032036 005702
6016 032040 001022
6017 032042 020127 177770
6018 032046 001426

;PROGRAM ID
;ARE WE RUNNING UNDER XXDP/ACT?
;BRANCH IF YES
;ARE WE RUNNING UNDER APT?
;BRANCH IF YES
;SOFTWARE SWITCH REG SELECTED?
;BRANCH IF NO
;GET SOFT-SWR SETTINGS
;;SET AUTO-MODE INDICATOR

GDRVS: RDLIN
MOV (SP)+, R0 ;GET STARTING ADDR OF ASCII STRING
MOV #-8, R1 ;SET UP COUNT
1S: MOV (R0)+, R2 ;GET ASCII CHAR
BIC #177400, R2 ;MASK HI BYTE
MOV #DRIVO, R3 ;DRIVE FLAG ADDR
MOV #60, R4

2S: CMP R4, R2 ;WAS TYPED CHAR 0 THRU 7?
BEQ 3S ;BRANCH IF YES
TST (R3)+ ;NO, INCREMENT DR FLAG ADDR
INC R4
CMP R4, #70
BNE 2S ;S/B 0-7 OR TERMINATOR
TST R2
BNE 4S
CMP R1, #-8.
BEQ 6S ;DEFAULT ALL DRIVES

```

# K09

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
 DZR610.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 113  
 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0113

6019	032050	005037	007510	75:	CLR RTS	SIZFLG PC	;BYPASS TEST 1 (SIZING) ;FOUND TERMINATOR, EXIT
6020	032054	000207					
6021							
6022	032056	005213		35:	INC INC	DR3 DRIVS	;SET UP FLAG FOR THE DRIVE ;INCREMENT TOTAL # DRIVES TO BE TESTED
6023	032060	005237	007460		MOV	(R0)+,R2	;GET NEXT ASCII CHAR.
6024	032064	112002			BIC	#177400,R2	;MASK
6025	032066	042702	177400		CMP	#54,R2	;IS IT A COMMA?
6026	032072	022702	000054		BEQ	55	;YES, GO TO NEXT WORD.
6027	032076	001407			TST	R2	;NO, IS IT A TERMINATOR?
6028	032100	005702			BNE	45	;IF NOT, SOMETHING WRONG.
6029	032102	001001			BR	75	;FOUND TERMINATOR, EXIT
6030	032104	000761					
6031							
6032	032106	104401	046062	45:	TYPE	EMI	;ONLY 0-7 ALLOWED.
6033	032112	000137	012546		JMP	PRGSRT	;START ALL OVER
6034							
6035	032116	005201		55:	INC	R1	;S/B NO MORE THAN 8 DIFF
6036	032120	001330			BNE	15	;DRIVES TYPED IN.
6037	032122	000771			BR	45	;IF MORE, HAVE ERROR.
6038							
6039	032124	005237	007510	65:	INC	SIZFLG	;DO TEST 1 (SIZING)
6040	032130	000207			RTS	PC	;EXIT.
6041							
6042							
6043							
6044							
6045							
6046	032132	104412		GBA:	RDOCT		
6047	032134	012600			MOV	(SP)+,RO	;GET LOW ORDER FROM STACK
6048	032136	005700			TST	RO	
6049	032140	001403			BEQ	15	;BRANCH IF DEFAULT.
6050	032142	010037	001264		MOV	RO,\$BASE	
6051	032146	000207			RTS	PC	
6052	032150	012737	177440 001264	15:	MOV	#177440,\$BASE	;DEFAULT VALUE
6053	032156	000207			RTS	PC	
6054							
6055							
6056							
6057							
6058							
6059	032160	104412		GINT:	RDOCT		
6060	032162	012600			MOV	(SP)+,RO	;GET LOW ORDER FROM STACK
6061	032164	005700			TST	RO	
6062	032166	001405			BEQ	15	;BRANCH IF DEFAULT
6063	032170	010037	001314		MOV	RO,RKVEC	
6064	032174	004737	032212	25:	JSR	PC,SETINT	
6065	032200	000207			RTS	PC	
6066	032202	012737	000210 001314	15:	MOV	#210,RKVEC	;DEFAULT VALUE
6067	032210	000771			BR	25	
6068							
6069							
6070							
6071							
6072							
6073	032212	013700	001314	SETINT:	MOV	RKVEC,RO	
6074	032216	012720	036652		MOV	#INTER,(RO)+	;INTER ADDR TO RKVEC

```

6075 032222 013710 001316      MOV      RKPRI,(R0)      ;PR5 TO RKVEC+2
6076 032226 000207              RTS      PC
6077
6078
6079
6080      ; ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
6081      ; ENTER WITH A COUNT IN TEMP1
6082      ; RETURN IF RDY NOT PRESENT (ERROR CONDITION)
6083      ; RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
6084      ; STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6085
6086 032230 032765 000200 000000  FRDY:   BIT      #RDY,RKCS1(R5)
6087 032236 001010              BNE     IS
6088 032240 005337 007376      DEC     TEMP1
6089 032244 001371              BNE     FRDY
6090 032246 004737 032364      JSR    PC,HOLD          ;STORE ALL RK611 REGS IN HOLDING REGS.
6091 032252 004737 033504      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR
6092 032256 000207              RTS     PC              ;NO RDY, EXIT
6093 032260 062716 000002  IS:     ADD     #2,(SP)    ;SKIP OVER ERROR
6094 032264 004737 032364      JSR    PC,HOLD
6095 032270 004737 033504      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR
6096 032274 000207              RTS     PC
6097
6098      ; ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
6099
6100 032276 032765 000200 000000  FRDY1:  BIT      #RDY,RKCS1(R5)
6101 032304 001014              BNE     IS
6102 032306 005337 007376      DEC     TEMP1
6103 032312 001371              BNE     FRDY1
6104 032314 016537 000034 007366      MOV    RKMR2(R5),HMR2
6105 032322 016537 000036 007370      MOV    RKMR3(R5),HMR3
6106 032330 004737 033504      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
6107 032334 000207              RTS     PC              ;NO RDY, EXIT
6108 032336 062716 000002  IS:     ADD     #2,(SP)    ;SKIP OVER ERROR
6109 032342 016537 000034 007366      MOV    RKMR2(R5),HMR2
6110 032350 016537 000036 007370      MOV    RKMR3(R5),HMR3
6111 032356 004737 033504      JSR    PC,CKCERR       ;CHECK FOR SPECIAL CERR CONDITIONS
6112 032362 000207              RTS     PC
6113
6114
6115      ; STORE ALL RK611 REGISTERS IN HOLDING REGS
6116
6117
6118 032364 016537 000000 007340  HOLD:   MOV     RKCS1(R5),HCS1
6119 032372 016537 000010 007342      MOV    RKCS2(R5),HCS2
6120 032400 016537 000002 007344      MOV    RKWC(R5),HWC
6121 032406 016537 000004 007346      MOV    RKBA(R5),HBA
6122 032414 016537 000006 007350      MOV    RKDA(R5),HDA
6123 032422 016537 000012 007352      MOV    RKDS(R5),HDS
6124 032430 016537 000014 007354      MOV    RKER(R5),HER
6125 032436 016537 000016 007356      MOV    RKASOF(R5),HASOF
6126 032444 016537 000020 007360      MOV    RKDC(R5),HDC
6127 032452 016537 000026 007364      MOV    RKMR1(R5),HMR1
6128 032460 016537 000034 007366      MOV    RKMR2(R5),HMR2
6129 032466 016537 000036 007370      MOV    RKMR3(R5),HMR3
6130 032474 016537 000030 007372      MOV    RKECPS(R5),HPOS
    
```

```

6131 032502 016537 000032 007374      MOV      RKECPT(R5),HPAT
6132 032510 000207                      RTS      PC
6133
6134
6135      ;ROUTINE TO CHECK FOR CORRECT ATTN
6136      ;RETURN IF ATTN NOT PRESENT (ERROR CONDITION)
6137      ;RETURN +2 IF ATTN PRESENT (SKIP OVER ERROR)
6138
6139 032512 010446      †STATN: MOV      R4,-(SP)          ;SAV R4
6140 032514 013704 001222      MOV      $UNIT,R4
6141 032520 136437 007330 007357      BITB    ATTN(R4),HASOF+1
6142 032526 001404                      BEQ      1$ ;BRANCH IF ATTN NOT PRESENT
6143 032530 012604                      MOV      (SP)+,R4 ;RESTOR R4
6144 032532 062716 000002      ADD      #2,(SP) ;INCR RET ADDR TO JUMP OVER ERROR.
6145 032536 000207                      RTS      PC
6146 032540 012604      1$: MOV      (SP)+,R4 ;RESTOR R4
6147 032542 000207                      RTS      PC
6148
6149
6150      ;ROUTINE TO FIND ATTN WITHIN TIMES GREATER THAN 1 SEC
6151      ;ENTER WITH TIME IN SECONDS IN TEMP2
6152      ;RETURN IF NO ATTN (ERROR CONDITION)
6153      ;RETURN +2 IF ATTN FOUND
6154      ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6155
6156
6157 032544 010446      FATT1: MOV      R4,-(SP)          ;SAV R4
6158 032546 012737 177777 007376      3$: MOV      #-1,TEMP1
6159 032554 013704 001222      MOV      $UNIT,R4
6160 032560 136465 007330 000017      1$: BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN
6161 032566 001014                      BNE     2$
6162 032570 005337 007376      DEC     TEMP1
6163 032574 001371                      BNE     1$
6164 032576 005337 007400      DEC     TEMP2
6165 032602 001361                      BNE     3$
6166 032604 005065 000026      CLR     RKMR1(R5) ;SELECT WORD 0
6167 032610 004737 033566      JSR     PC,GSTAT ;GET LATEST STATUS
6168 032614 012604                      MOV     (SP)+,R4 ;RESTOR R4
6169 032616 000207                      RTS     PC
6170 032620 005065 000026      2$: CLR     RKMR1(R5)
6171 032624 004737 033566      JSR     PC,GSTAT ;GET STATUS AFTER ATTN SEEN
6172 032630 012604                      MOV     (SP)+,R4 ;RESTOR R4
6173 032632 062716 000002      ADD     #2,(SP) ;SKIP OVER ERROR
6174 032636 000207                      RTS     PC
6175
6176
6177      ;ROUTINE TO FIND ATTN WITHIN 1 SEC
6178      ;ENTER WITH COUNT IN TEMP1
6179      ;RETURN IF NO ATTN (ERROR)
6180      ;RETURN +2 IF ATTN FOUND
6181      ;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6182
6183
6184 032640 010446      FATT2: MOV      R4,-(SP)          ;SAV R4
6185 032642 013704 001222      2$: MOV      $UNIT,R4
6186 032646 136465 007330 000017      BITB    ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN

```

6187	032654	001011			BNE	1\$		
6188	032656	005337	007376		DEC	TEMP1		
6189	032662	001367			BNE	2\$		
6190	032664	005065	000026		CLR	RKMR1(R5)		; SELECT WORD 0
6191	032670	004737	033566		JSR	PC, GSTAT		; GET LATEST STATUS.
6192	032674	012604			MOV	(SP)+, R4		; RESTOR R4
6193	032676	000207			RTS	PC		
6194	032700	005065	000026	1\$:	CLR	RKMR1(R5)		
6195	032704	004737	033566		JSR	PC, GSTAT		
6196	032710	012604			MOV	(SP)+, R4		; RESTOR R4
6197	032712	062716	000002		ADD	#2, (SP)		; SKIP OVER ERROR
6198	032716	000207			RTS	PC		
6199								
6200								
6201								
6202								
6203								
6204	032720	005737	007376		DLY:	TST	TEMP1	; 5.6 US
6205	032724	001403				BEQ	1\$	; 1.9 US
6206	032726	005337	007376			DEC	TEMP1	; 6.8 US
6207	032732	000772				BR	DLY	; 2.5 US
6208	032734	000207		1\$:		RTS	PC	; 3.8 US
6209								
6210								
6211								
6212								
6213	032736	104401	045166		BYP:	TYPE	MSG14	; BYPASS DRIVE
6214	032742	010046				MOV	RO, -(SP)	; SAVE RO FOR TYPEOUT
6215								; TYPE DR#
6216	032744	104403				TYPOS		; GO TYPE--OCTAL ASCII
6217	032746	001				.BYTE	1	; TYPE 1 DIGIT(S)
6218	032747	000				.BYTE	0	; SUPPRESS LEADING ZEROS
6219	032750	000207				RTS	PC	
6220								
6221								
6222								
6223	032752	017637	000000	001504	CHKMSG:	MOV	2(SP), CHKFLG	; PASS MSGS TO BE TESTED
6224	032760	062716	000002			ADD	#2, (SP)	; BUMP RETURN ADDR TO 1ST ERROR
6225	032764	004737	033630			JSR	PC, GSTAT1	; GET ALL ACTUAL DRIVE & CONTR STATUS
6226								
6227	032770	053737	001222	007430		BIS	\$UNIT, E.A0	; SET UNIT #
6228	032776	053737	001222	007434		BIS	\$UNIT, E.A1	
6229	033004	053737	001222	007440		BIS	\$UNIT, E.A2	
6230	033012	053737	001222	007444		BIS	\$UNIT, E.A3	
6231								
6232	033020	013746	007376			MOV	TEMP1, -(SP)	; SAVE TEMP1
6233								
6234	033024	013737	007430	007376		MOV	E.A0, TEMP1	
6235	033032	004737	036066			JSR	PC, SBPAR	; GET PARITY FOR MSG A0
6236	033036	013737	007376	007430		MOV	TEMP1, E.A0	
6237								
6238	033044	013737	007434	007376		MOV	E.A1, TEMP1	
6239	033052	004737	036066			JSR	PC, SBPAR	; GET PARITY FOR MSG A1
6240	033056	013737	007376	007434		MOV	TEMP1, E.A1	
6241								
6242	033064	013737	007440	007376		MOV	E.A2, TEMP1	



## B10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 117  
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0117

6243	033072	004737	036066			JSR	PC,SBPAR		;GET PARITY FOR MSG A2
6244	033076	013737	007376	007440		MOV	TEMP1,E.A2		
6245									
6246	033104	013737	007432	007376		MOV	E.B0,TEMP1		
6247	033112	004737	036066			JSR	PC,SBPAR		;GET PARITY FOR MSG B0
6248	033116	013737	007376	007432		MOV	TEMP1,E.B0		
6249									
6250	033124	013737	007436	007376		MOV	E.B1,TEMP1		
6251	033132	004737	036066			JSR	PC,SBPAR		;GET PARITY FOR MSG B1
6252	033136	013737	007376	007436		MOV	TEMP1,E.B1		
6253									
6254	033144	013737	007442	007376		MOV	E.B2,TEMP1		
6255	033152	004737	036066			JSR	PC,SBPAR		;GET PARITY FOR MSG B2
6256	033156	013737	007376	007442		MOV	TEMP1,E.B2		
6257									
6258	033164	013737	007446	007376		MOV	E.B3,TEMP1		
6259	033172	004737	036066			JSR	PC,SBPAR		;GET PARITY FOR MSG B3
6260	033176	013737	007376	007446		MOV	TEMP1,E.B3		
6261									
6262	033204	012637	007376			MOV	(SP)+,TEMP1		;RESTORE TEMP1
6263	033210	013737	001176	001172		MOV	\$ESCAPE,\$TMP5		;SAVE ESCAPE
6264									
6265	033216	023737	007410	007430		CMP	H.A0,E.A0		;TEST MSG A0
6266	033224	001411				BEQ	2\$		;BR IF OK
6267	033226	012737	033240	001176		MOV	#1\$, \$ESCAPE		;ELSE SETUP ESCAPE
6268	033234	011646				MOV	(SP),-(SP)		;COPY RET ADDR.
6269	033236	000207				RTS	PC		; & RETURN TO MAINLINE ERROR
6270									
6271	033240	032777	001000	145672	1\$:	BIT	#SW9,\$SWR		;RET HERE FROM MAINLINE ERROR
6272	033246	001107				BNE	20\$		; & BR IF LOOP ON ERROR
6273	033250	062716	000002		2\$:	ADD	#2,(SP)		;BUMP RET ADDR TO NEXT ERROR
6274									
6275	033254	023737	007412	007432		CMP	H.B0,E.B0		;TEST MSG B0
6276	033262	001411				BEQ	5\$		;BR IF OK
6277	033264	012737	033276	001176		MOV	#4\$, \$ESCAPE		;ELSE SETUP ESCAPE
6278	033272	011646				MOV	(SP),-(SP)		;COPY RET ADDR
6279	033274	000207				RTS	PC		; & RETURN TO MAINLINE ERROR
6280									
6281	033276	032777	001000	145634	4\$:	BIT	#SW9,\$SWR		;RETURN HERE FROM MAINLINE ERROR
6282	033304	001070				BNE	20\$		; & BR IF LOOP ON ERROR
6283	033306	062716	000002		5\$:	ADD	#2,(SP)		;BUMP RET ADDR TO NEXT ERROR
6284									
6285	033312	023737	007414	007434		CMP	H.A1,E.A1		;TEST MSG A1
6286	033320	001411				BEQ	8\$		;BR IF OK
6287	033322	012737	033334	001176		MOV	#7\$, \$ESCAPE		
6288	033330	011646				MOV	(SP),-(SP)		
6289	033332	000207				RTS	PC		
6290									
6291	033334	032777	001000	145576	7\$:	BIT	#SW9,\$SWR		
6292	033342	001051				BNE	20\$		
6293	033344	062716	000002		8\$:	ADD	#2,(SP)		
6294									
6295	033350	023737	007416	007436		CMP	H.B1,E.B1		;TEST MSG B1
6296	033356	001411				BEQ	11\$		;BR IF OK
6297	033360	012737	033372	001176		MOV	#10\$, \$ESCAPE		
6298	033366	011646				MOV	(SP),-(SP)		

```

6299 033370 000207          RTS      PC
6300
6301 033372 032777 001000 145540 10$:  BIT      #SW9,2SWR
6302 033400 001032          BNE     20$
6303 033402 062716 000002          ADD     #2,(SP)
6304
6305 033406 032737 000001 001504 12$:  BIT      #T.A2,CHKFLG ;TEST MSG A2?
6306 033414 001402          BEQ     13$           ;BR IF NO
6307 033416 004737 034514          JSR     PC,RCYLD     ;PUT INFO CYLDIF, DO NOT CHECK
6308 033422 032737 000002 001504 13$:  BIT      #T.B2,CHKFLG ;TEST MSG B2?
6309 033430 001402          BEQ     14$           ;BR IF NO
6310 033432 004737 034566          JSR     PC,RCYLA    ;PUT INFO IN CYLADD, DO NOT CHECK
6311
6312 033436 032737 000004 001504 14$:  BIT      #T.B3,CHKFLG ;TEST MSG B3?
6313 033444 001404          BEQ     15$
6314 033446 004737 034624          JSR     PC,RSEC     ;PUT INFO IN SECTOR, DO NOT CHECK
6315 033452 004737 034662          JSR     PC,RHEAD    ;PUT INFO IN HEADA, DO NOT CHECK
6316
6317 033456 013737 001172 001176 15$:  MOV     $TMP5,$ESCAPE ;RESTORE ESCAPE
6318 033464 000207          RTS      PC
6319
6320 033466 012706 001100 20$:  MOV     #STACK,SP   ;RESET STACK PTR
6321 033472 013737 001172 001176  MOV     $TMP5,$ESCAPE ;RESTORE ESCAPE
6322 033500 000177 145404          JMP     $SLPERR
6323
6324 ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
6325 ; I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
6326
6327 033504 005737 001502          CKCERR: TST     BYPCERR
6328 033510 001025          BNE     4$
6329 033512 032737 100000 007340  BIT     #CERR,HCS1
6330 033520 001001          BNE     1$           ;BR IF CERR
6331 033522 000207          RTS      PC
6332
6333 033524 032737 004000 007340 1$:  BIT     #CTO,HCS1
6334 033532 001402          BEQ     2$           ;BR IF NOT CTO
6335 033534 104125          ERROR  125          ;CTO ERROR, MSG A & B INVALID
6336 033536 000207          RTS      PC
6337
6338 033540 032737 010000 007342 2$:  BIT     #NED,HCS2
6339 033546 001401          BEQ     3$           ;BR IF NOT NED
6340 033550 104126          ERROR  126          ;NED ERROR, MSG A & B INVALID
6341
6342 033552 032737 001000 007342 3$:  BIT     #MDS,HCS2
6343 033560 001401          BEQ     4$
6344 033562 104127          ERROR  127          ;MDS ERROR, MSG A & B INVALID
6345
6346 033564 000207          4$:  RTS      PC
6347
6348 ; THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
6349 ; IT THEN WAITS FOR CONTROLLER READY
6350 ; IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
6351 ;
6352
6353
6354 033566 013746 007376          GSTAT: MOV     TEMP1,-(SP) ;SAVE TEMP1

```

# D10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR61D.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 119  
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0119

```

6355 033572 013765 001222 000010
6356 033600 012765 000001 000000
6357 033606 013737 001406 007376
6358 033614 004737 032230
6359 033620 104117
6360 033622 012637 007376
6361 033626 000207
6362
6363
6364
6365
6366 033630 013746 007376
6367 033634 004737 032364
6368 033640 012765 100000 000000
6369 033646 013765 001222 000010
6370 033654 012765 000003 000026
6371 033662 012765 000001 000000
6372 033670 013737 001406 007376
6373 033676 004737 032276
6374 033702 104117
6375 033704 013737 007366 007424
6376 033712 013737 007370 007426
6377
6378 033720 012765 100000 000000
6379 033726 013765 001222 000010
6380 033734 012765 000002 000026
6381 033742 012765 000001 000000
6382 033750 013737 001406 007376
6383 033756 004737 032276
6384 033762 104117
6385 033764 013737 007366 007420
6386 033772 013737 007370 007422
6387
6388 034000 012765 100000 000000
6389 034006 013765 001222 000010
6390 034014 012765 000001 000026
6391 034022 012765 000001 000000
6392 034030 013737 001406 007376
6393 034036 004737 032276
6394 034042 104117
6395 034044 013737 007366 007414
6396 034052 013737 007370 007416
6397
6398 034060 012765 100000 000000
6399 034066 013765 001222 000010
6400 034074 012765 000001 000000
6401 034102 013737 001406 007376
6402 034110 004737 032276
6403 034114 104117
6404 034116 013737 007366 007410
6405 034124 013737 007370 007412
6406
6407 034132 012637 007376
6408 034136 000207
6409
6410
    
```

```

MOV SUNIT,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 SUNIT,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 SUNIT,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 SUNIT,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 SUNIT,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

MOV (SP)+,TEMP1 ;RESTORE TEMP1
RTS PC
    
```

# E10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR61D.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 120  
GE1 VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0120

```

6411 ; THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
6412 ; IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
6413 ; THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
6414 ; RETURN IF CERR SET
6415 ; RETURN +2 IF CERR CLEAR
6416
6417 034140 012765 000040 000010 SUBCLR: MOV #SCLR,RKCS2(R5) ;SUBSYS CLEAR
6418 034146 013737 001406 007376 MOV T10,TEMP1
6419 034154 004737 032230 JSR PC,FRDY ;FIND RDY
6420 034160 104120 ERROR 120 ;RDY NOT SET BY END OF SCLR
6421 034162 013765 001222 000010 MOV #UNIT,RKCS2(R5) ;CURRENT DRIVE #
6422 034170 005065 000026 CLR RKMR1(R5) ;SELECT WORD 0
6423 034174 004737 033566 JSR PC,GSTAT ;GET STATUS
6424 034200 032737 100000 007340 BIT #CERR,HCS1 ;CHECK FOR CONT ERROR
6425 034206 001401 BEQ 1$
6426 034210 000207 RTS PC
6427 034212 062716 000002 1$: ADD #2,(SP) ;SKIP OVER ERROR
6428 034216 000207 RTS PC
6429
6430 ; READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
6431
6432
6433 034220 012765 000003 000026 RDSEC: MOV #3,RKMR1(R5) ;WORD 3
6434 034226 004737 033566 JSR PC,GSTAT
6435 034232 013737 007370 001402 MOV HMR3,SECTOR
6436 034240 042737 177017 001402 BIC #C(M.SECT),SECTOR
6437 034246 006237 001402 ASR SECTOR ;RIGHT JUSTIFY
6438 034252 006237 001402 ASR SECTOR ;SECTOR
6439 034256 006237 001402 ASR SECTOR ;INFO
6440 034262 006237 001402 ASR SECTOR
6441 034266 000207 RTS PC
6442
6443 ; READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
6444
6445
6446 034270 012765 000002 000026 RDCYLD: MOV #2,RKMR1(R5) ;WORD 2
6447 034276 004737 033566 JSR PC,GSTAT
6448 034302 013737 007366 001356 MOV HMR2,CYLDIF
6449 034310 042737 160017 001356 BIC #C(M.CDIF),CYLDIF
6450 034316 006237 001356 ASR CYLDIF ;RIGHT JUSTIFY
6451 034322 006237 001356 ASR CYLDIF ;CYL DIFF/OFFSET
6452 034326 006237 001356 ASR CYLDIF ;INFO
6453 034332 006237 001356 ASR CYLDIF
6454 034336 023727 001356 000777 CMP CYLDIF,#777 ;CHK TO SEE IF RET IN COMPL. FORM
6455 034344 001002 BNE 1$ ;BR IF NOT
6456 034346 005037 001356 CLR CYLDIF ;CLR IF YES
6457 034352 000207 1$: RTS PC
6458
6459 ; QUICK SELECT DRIVE COMMAND TO OBTAIN CYL DIFF
6460
6461
6462 034354 013746 007376 QKCYLD: MOV TEMP1,-(SP) ;SAVE TEMP1
6463 034360 012765 000002 000026 MOV #2,RKMR1(R5) ;SELECT WORD 2
6464 034366 012765 000001 000000 MOV #SELDV,RKCS1(R5) ;SELECT DRIVE CMD
6465 034374 013737 001406 007376 MOV T10,TEMP1
6466 034402 032765 000200 000000 1$: BIT #RDY,RKCS1(R5) ;TEST FOR CONT RDY
6466 034410 001004 BNE 2$ ;BR IF THERE

```

# F10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 121  
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0121

```

6467 034412 005337 007376          DEC    TEMP1
6468 034416 001371                   BNE    1$
6469 034420 104117                   ERROR  117          ;NO RDY AFTER SEL DRV CMD
6470
6471 034422 016537 000034 001356 2$:  MOV    RKMR2(R5),CYLDIF
6472 034430 042737 160017 001356    BIC    #1C<M.CDIF>,CYLDIF ;GET CYL DIFF ONLY (NO SHIFTING)
6473 034436 012637 007376          MOV    (SP)+,TEMP1 ;RESTORE TEMP1
6474 034442 000207          RTS    PC
6475
6476          ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
6477
6478 034444 012765 000002 000026 RDCYLA: MOV    #2,RKMR1(R5) ;WORD 2
6479 034452 004737 033566          JSR    PC,GSTAT
6480 034456 013737 007370 001360    MOV    HMR3,CYLADD
6481 034464 042737 160017 001360    BIC    #1C<M.CADD>,CYLADD
6482 034472 006237 001360          ASR    CYLADD ;RIGHT JUSTIFY
6483 034476 006237 001360          ASR    CYLADD ;CYL ADDR
6484 034502 006237 001360          ASR    CYLADD ;INFO
6485 034506 006237 001360          ASR    CYLADD
6486 034512 000207          RTS    PC
6487
6488          ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
6489
6490 034514 013737 007420 001356 RCYLD: MOV    H.A2,CYLDIF
6491 034522 042737 160017 001356    BIC    #1C<M.CDIF>,CYLDIF ;CLEAR UNWANTED INFO
6492 034530 006237 001356          ASR    CYLDIF ;RIGHT JUSTIFY
6493 034534 006237 001356          ASR    CYLDIF
6494 034540 006237 001356          ASR    CYLDIF
6495 034544 006237 001356          ASR    CYLDIF
6496 034550 023727 001356 000777    CMP    CYLDIF,#777 ;CHK TO SEE IF RET IN COMPL. FORM
6497 034556 001002          BNE    1$ ;BR IF NO
6498 034560 005037 001356          CLR    CYLDIF ;ELSE CLEAR
6499 034564 000207          1$:  RTS    PC
6500
6501          ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
6502
6503 034566 013737 007422 001360 RCYLA: MOV    H.B2,CYLADD
6504 034574 042737 160017 001360    BIC    #1C<M.CADD>,CYLADD ;CLEAR UNWANTED INFO
6505 034602 006237 001360          ASR    CYLADD ;RIGHT JUSTIFY
6506 034606 006237 001360          ASR    CYLADD
6507 034612 006237 001360          ASR    CYLADD
6508 034616 006237 001360          ASR    CYLADD
6509 034622 000207          RTS    PC
6510
6511          ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
6512
6513 034624 013737 007426 001402 RSEC:  MOV    H.B3,SECTOR
6514 034632 042737 177017 001402    BIC    #1C<M.SECT>,SECTOR ;CLEAR UNWANTED INFO
6515 034640 006237 001402          ASR    SECTOR ;RIGHT JUSTIFY
6516 034644 006237 001402          ASR    SECTOR
6517 034650 006237 001402          ASR    SECTOR
6518 034654 006237 001402          ASR    SECTOR
6519 034660 000207          RTS    PC
6520
6521          ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
6522

```

```

6523 034662 013737 007426 001446 RHEAD: MOV H.B3,HEADR
6524 034670 042737 170777 001446 BIC #1C(M.HEAD),HEADR ;CLEAR UNWANTED INFO
6525 034676 006237 001446 ASR HEADR ;RIGHT JUSTIFY IT
6526 034702 000337 001446 SWAB HEADR
6527 034706 000207 RTS PC
6528
6529 ;FIND SECTOR 23
6530 ;RETURN IF NOT FOUND
6531 ;RETURN +4 IF FOUND
6532
6533 034710 013737 001416 007376 FSEC23: MOV T5000,TEMP1 ;SETUP TIMEOUT
6534 034716 004737 034220 1$: JSR PC,RDSEC ;READ SECTOR
6535 034722 023727 001402 000023 CMP SECTOR,#23 ;TEST FOR SECTOR 23(B)
6536 034730 001014 BNE 2$ ;BR IF NOT 23(B)
6537
6538 034732 004737 034220 JSR PC,RDSEC
6539 034736 023727 001402 000023 CMP SECTOR,#23
6540 034744 001412 BEQ 3$ ;BR IF READ SAME TWICE
6541 034746 004737 034220 JSR PC,RDSEC ;ELSE TRY 1 MORE TIME
6542 034752 023727 001402 000023 CMP SECTOR,#23
6543 034760 001404 BEQ 3$ ;BR IF 23(B)
6544
6545 034762 005337 007376 2$: DEC TEMP1
6546 034766 001353 BNE 1$ ;TRY AGAIN
6547 034770 000207 RTS PC
6548
6549 034772 062716 000004 3$: ADD #4,(SP) ;SKIP OVER ERROR
6550 034776 000207 RTS PC
6551
6552 ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
6553 ;ENTER WITH TIME IN SECONDS IN TEMP2
6554 ;RETURN IF NOT FOUND
6555 ;RETURN+2 IF FOUND - SKIP OVER ERROR
6556
6557 035000 012737 177777 007376 FHDHM: MOV #-1,TEMP1 ;ALL 1'S
6558 035006 012765 000001 000026 MOV #1,RKMR1(R5) ;WORD 1
6559 035014 004737 033566 1$: JSR PC,GSTAT
6560 035020 032737 000040 007366 BIT #0,HDHM,HMR2
6561 035026 001007 BNE 2$
6562 035030 005337 007376 DEC TEMP1
6563 035034 001367 BNE 1$
6564 035036 005337 007400 DEC TEMP2
6565 035042 001356 BNE FHDHM
6566 035044 000207 RTS PC
6567 035046 062716 000002 2$: ADD #2,(SP) ;SKIP OVER ERROR
6568 035052 000207 RTS PC
6569
6570 ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE THE TIMEOUT
6571 ;RETURN IF NOT FOUND
6572 ;RETURN+2 IF FOUND: SKIP OVER ERROR
6573
6574 035054 012737 000372 007376 FLOAD: MOV #250,TEMP1
6575 035062 012765 000001 000026 MOV #1,RKMR1(R5) ;WORD 1
6576 035070 004737 033566 1$: JSR PC,GSTAT
6577 035074 032737 010000 007366 BIT #0,LOAD,HMR2
6578 035102 001004 BNE 2$

```

# H10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 123  
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0123

6579	035104	005337	007376
6580	035110	001367	
6581	035112	000207	
6582	035114	062716	000002
6583	035120	000207	
6584			
6585			
6586			
6587			
6588			
6589			
6590	035122	010046	
6591	035124	010146	
6592	035126	012700	001506
6593	035132	005001	
6594	035134	013737	001444 001450
6595	035142	006337	001450
6596	035146	006337	001450
6597	035152	006337	001450
6598	035156	006337	001450
6599	035162	006337	001450
6600	035166	013737	001452 001454
6601	035174	000337	001454
6602	035200	006337	001454
6603			
6604	035204	013720	001362
6605	035210	010110	
6606	035212	053710	001450
6607	035216	053710	001454
6608	035222	004737	035302
6609			
6610	035226	013737	001362 007376
6611	035234	011037	007400
6612	035240	043737	001362 007400
6613	035246	042037	007376
6614	035252	053737	007376 007400
6615	035260	013720	007400
6616			
6617	035264	005201	
6618	035266	020127	000026
6619	035272	001344	
6620			
6621	035274	012601	
6622	035276	012600	
6623	035300	000207	
6624			
6625			
6626			
6627			
6628	035302	010246	
6629	035304	005737	001452
6630	035310	001016	
6631	035312	012702	003332
6632	035316	004737	035402
6633	035322	052710	100000
6634			

```

DEC      TEMP1
BNE      1$
RTS      PC
2$:      ADD      #2,(SP)      ;SKIP OVER ERROR
RTS      PC

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

FHDTAB:  MOV      RO,-(SP)      ;SAV RO
          MOV      R1,-(SP)      ;SAV R1
          MOV      #FHDTAB,R0    ;HEADER WORD TABLE ADDR
          CLR      R1            ;SECTOR COUNTER
          MOV      HEAD,HD1
          ASL      HD1
          ASL      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

1$:      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    ;
          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 SFCTORS DONE? (66 WORDS)
          BNE     1$           ;BR IF NO

          MOV      (SP)+,R1    ;RESTOR R1
          MOV      (SP)+,R0    ;RESTOR R0
          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     1$           ;BR IF 20 SECTOR FORMAT
          MOV      #BSE22H+8.,R2
          JSR      PC,FLGTST   ;GET HARDWARE DETECTED FLAG
          BIS      #BIT15,(RO) ;RETURN HERE IF GOOD SECTOR
    
```

```

6635 035326 012702 005332      MOV      #BSE225+8.,R2      ;ELSE RETURN HERE
6636 035332 004737 035402      JSR      PC,FLGTST         ;GET SOFTWARE DETECTED FLAG
6637 035336 052710 040000      BIS      #BIT14,(R0)       ;RETURN HERE IF GOOD SECTOR
6638
6639 035342 012602                MOV      (SP)+,R2         ;ELSE RETURN HERE
6640 035344 000207                RTS      PC
6641
6642 035346 012702 002332      1$:     MOV      #BSE20H+8.,R2
6643 035352 004737 035402      JSR      PC,FLGTST         ;GET HARDWARE DETECTED FLAG
6644 035356 052710 100000      BIS      #BIT15,(R0)       ;RETURN HERE IF GOOD SECTOR
6645
6646 035362 012702 004332      MOV      #BSE20S+8.,R2
6647 035366 004737 035402      JSR      PC,FLGTST         ;GET SOFTWARE DETECTED FLAG
6648 035372 052710 040000      BIS      #BIT14,(R0)       ;RETURN HERE IF GOOD SECTOR
6649
6650 035376 012602                MOV      (SP)+,R2         ;RESTORE R2
6651 035400 000207                RTS      PC
6652
6653
6654
6655
6656
6657
6658
6659 035402 010346      FLGTST: MOV      R3,-(SP)      ;SAVE R3
6660
6661 035404 021227 177777      1$:     CMP      (R2),#-1         ;SEE IF ALL 1'S
6662 035410 001002                BNE      2$                ;BR IF NO
6663 035412 012603                MOV      (SP)+,R3         ;RESTORE R3
6664 035414 000207                RTS      PC
6665
6666 035416 022237 001362      2$:     CMP      (R2)+,CALADD     ;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
6667 035422 001403                BEQ      3$                ;GO TO NEXT CYL WORD IN TABLE
6668 035424 062702 000002      ADD      #2,R2
6669 035430 000765                BR       1$
6670
6671 035432 013703 001444      3$:     MOV      HEAD,R3         ;GET HEAD # FROM FHDTAB ROUTINE
6672 035436 000303                SWAB    R3
6673 035440 050103                BIS      R1,R3             ;ADD SECTOR # FROM FHDTAB ROUTINE
6674 035442 022203                CMP      (R2)+,R3         ;SEE IF SECTOR/HEAD COMPARE
6675
6676 035444 001401                BEQ      4$                ;& INCR PTR TO NEXT CYL WORD
6677 035446 000756                BR       1$                ;BR IF COMPARE
6678
6679 035450 012603                4$:     MOV      (SP)+,R3         ;RESTORE R3
6680 035452 062716 000004      ADD      #4,(SP)         ;INCREMENT RET ADDR
6681 035456 000207                RTS      PC
6682
6683
6684
6685
6686
6687 035460 010046      SORT:   MOV      R0,-(SP)      ;SAVE R0
6688 035462 010146                MOV      R1,-(SP)         ;SAVE R1
6689 035464 004737 034220      JSR      PC,R0SEC
6690 035470 062737 000001 001402      ADD      #1,SECTOR

```

```

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

```

```

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

```



```

6691 035476 004737 035566          JSR    PC,MULT6          ;MULT SECTOR BY 6
6692
6693 035502 012700 000204          MOV    #132,R0
6694 035506 163700 001402          SUB    SECTOR,R0        ;RO-SECTOR TO RO = INDEX
6695 035512 014037 001402          MOV    RO,SECTOR
6696 035516 062737 001712 001402      ADD    #RHTAB,SECTOR    ;SAVE INDEX
6697
6698 035524 062700 001712          ADD    #RHTAB,R0        ;INDEX TO BOT HALF OF RHTAB
6699 035530 012701 002116          MOV    #SRTTAB,R1      ;INDEX TO TOP HALF OF SRTTAB
6700
6701 035534 012021          1$:  MOV    (R0)+,(R1)+    ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
6702 035536 020027 002116          CMP    RO,#RHTAB+132.
6703 035542 001374          BNE    1$
6704
6705 035544 012700 001712          MOV    #RHTAB,R0        ;PUT TOP OF RHTAB TO BOT OF SRTTAB
6706 035550 012021          2$:  MOV    (R0)+,(R1)+
6707 035552 020037 001402          CMP    RO,SECTOR
6708 035556 001374          BNE    2$
6709
6710 035560 012601          MOV    (SP)+,R1        ;RESTOR R1
6711 035562 012600          MOV    (SP)+,R0        ;RESTOR R0
6712 035564 000207          RTS    PC
6713
6714
6715          ;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
6716
6717 035566 006337 001402          MULT6: ASL    SECTOR      ;2 X SECTOR
6718 035572 013746 001402          MOV    SECTOR,-(SP)
6719 035576 006337 001402          ASL    SECTOR          ;4 X SECTOR
6720 035602 062637 001402          ADD    (SP)+,SECTOR    ;(4 X 5)+(2 X 5) = 6 X SECTOR
6721 035606 000207          RTS    PC
6722
6723
6724          ;THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
6725          ;CMD. IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
6726          ;CYLINDER AT CYL 410, TRACK 2.
6727
6728          ;RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
6729          ;RETURN+2 IF LISTED, SKIP OVER ERROR
6730
6731 035610 010446          TRUERR: MOV   R4,-(SP)          ;SAVE R4
6732
6733 035612 032737 010000 007340      BIT    #CFMT,HCS1      ;CHECK FORMAT
6734 035620 001014          BNE    2$              ;BR FOR 20 SECTOR FORMAT
6735
6736 035622 012704 003332          MOV    #BSE22H+B.,R4
6737 035626 004737 035710          JSR    PC,TERR1        ;SEE IF ON HARDWARE DETECTED TABLE
6738 035632 000422          BR     3$              ;RETURN HERE IF YES
6739
6740 035634 012704 005332          MOV    #BSE22S+B.,R4
6741 035640 004737 035710          JSR    PC,TERR1        ;SEE IF ON SOFTWARE DETECTED TABLE
6742 035644 000415          BR     3$              ;RETURN HERE IF YES
6743
6744 035646 012604          1$:  MOV    (SP)+,R4        ;RESTORE R4
6745 035650 000207          RTS    PC              ;RETURN WITHOUT JUMPING OVER ERROR
6746

```

# K10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 126  
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0126

```

6747 035652 012704 002332
6748 035656 004737 035710
6749 035662 000406
6750
6751 035664 012704 004332
6752 035670 004737 035710
6753 035674 000401
6754 035676 000763
6755
6756 035700 012604
6757 035702 062716 000002
6758 035706 000207
6759
6760
6761
6762
6763
6764
6765
6766 035710 021427 177777
6767 035714 001405
6768 035716 022437 007360
6769 035722 001405
6770 035724 005724
6771 035726 000770
6772
6773 035730 062716 000002
6774 035734 000207
6775
6776 035736 022437 007350
6777 035742 001401
6778 035744 000761
6779
6780 035746 000207
6781
6782
6783
6784 035750 005037 001372
6785 035754 005737 007504
6786 035760 001004
6787 035762 012777 000100 143336
6788 035770 000207
6789 035772 012777 177777 143322
6790 036000 012777 000135 143312
6791 036006 000207
6792
6793
6794
6795 036010 005037 001372
6796 036014 005337 001366
6797 036020 001010
6798 036022 013737 001364 001366
6799 036030 005337 001370
6800 036034 001002
6801 036036 005237 001372
6802 036042 000002
    
```

```

2$:  MOV    #BSE20H+8.,R4
      JSR   PC,TERR1      ;SEE IF ON HARDWARE DETECTED TABLE
      BR   3$             ;RETURN HERE IF YES

      MOV    #BSE20S+8.,R4 ;ELSE RETURN HERE
      JSR   PC,TERR1      ;SEE IF ON SOFTWARE DETECTED TABLE
      BR   3$             ;RETURN HERE IF YES
      BR   1$             ;RETURN HERE IF NO

3$:  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
; THE BSE TABLE FOR THE ABOVE SUBROUTINE.
; RETURN IF FOUND ON TABLE
; RETURN+2 IF NOT FOUND

TERR1: CMP    (R4),#-1     ;SEE IF ALL 1'S
      BEQ   1$            ;BR IF YES, NOT ON TABLE
      CMP   (R4)+,HDC     ;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 UP, 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
      RTS

1$:  RTI

; SORRY, TIME IS UP
    
```

# L10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR610.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 127  
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0127

```

6803
6804
6805
6806 036044 005737 007504
6807 036050 001003
6808 036052 005077 143250
6809 036056 000207
6810 036060 005077 143234
6811 036064 000207
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823 036066 010046
6824 036070 010146
6825 036072 012700 000021
6826 036076 005001
6827 036100 000241
6828
6829 036102 006137 007376
6830 036106 103001
6831 036110 005201
6832 036112 005300
6833 036114 001372
6834
6835 036116 032701 000001
6836 036122 001003
6837 036124 052737 100000 007376
6838 036132 012601
6839 036134 012600
6840 036136 000207
6841
6842
6843
6844
6845
6846
6847 036140 032777 001000 142772
6848 036146 001406
6849 036150 105737 001103
6850 036154 001403
6851 036156 013716 001110
6852 036162 000002
6853
6854 036164 011637 001110
6855 036170 000002
6856
6857
6858

;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

;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MESSAGE
;ENTER WITH THE EXPECTED WORD IN TEMP1
;TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
;R1 IS INCREMENTED. AT THE END OF 17 ROTATES (TEMP1 BACK TO ORIG),
;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
;THE PARITY BIT IS NOT SET IN B
;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
;SET IN TEMP1
SBPAR: MOV RO,-(SP) ;SAVE RO
        MOV R1,-(SP) ;SAVE R1
        MOV #17,R0 ;SHIFT COUNTER
        CLR R1 ;COUNT # OF 1'S IN TEMP1
        CLC ;CLEAR CARRY
1$: ROL TEMP1
    BCC 2$ ;BR IF CARRY CLEAR
    INC R1 ;COUNT # OF 1'S
2$: DEC R0 ;SHIFT COUNTER
    BNE 1$
3$: BIT #BIT0,R1
    BNE 3$ ;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 $LPERR SET BY OTHER THAN SCOPE ROUTINE
;IE: MY LOOP MACRO
SCOPE1$: BIT #SW9,$SWR ;LOOP ON ERROR?
        BEQ 1$ ;BR IF NO
        TSTB $ERFLG ;HAD ERROR?
        BEQ 1$ ;BR IF NO
        MOV $LPERR,(SP)
        RTI
1$: MOV (SP),$LPERR ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
    RTI

;THIS ROUTINE IS ENTERED BY TYPING A CONTROL-C.
;IT IS USED TO ALLOW THE OPERATOR TO HLAT THE CPU WHILE INSURING

```

# M10

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 123  
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0128

```

6859 ; THAT HEADS ARE LOADED & FORMATTING IS VALID BEFORE ACTUALLY HALTING
6860 ; THE CPU.
6861
6862 036172 022626 STOP: CMP (SP)+,(SP)+ ; RESTORE STACK FROM INTERRUPT
6863
6864 036174 004737 034140 JSR PC,SUBCLR
6865 036200 104024 ERROR 24 ; CERR AFTER
6866
6867 036202 005737 007322 TST UNLD ; SEE IF HEADS UNLOADED
6868 036206 001434 BEQ 3$ ; BR IF NO
6869 036210 005737 000042 TST 42 ; SEE IF MANUAL OR AUTO MODE
6870 036214 001403 BEQ 1$ ; BR IF MANUAL MODE
6871 036216 104401 045720 TYPE ,MSG74 ; PGM ABORT PENDING
6872 036222 000402 BR 2$
6373 036224 104401 045766 1$: TYPE ,MSG75 ; HALT PENDING
6874 036230 2$:
6875
6876 036230 004737 034140 JSR PC,SUBCLR
6877 036234 104024 ERROR 24 ; CERR AFTER SCLR
6878
6879 036236 012765 000011 000000 MOV #SRTSPL,RKCS1(R5) ; START SPINDLE CMD
6880 036244 013737 001406 007376 MOV T10,TEMP1 ; SET TIMEOUT
6881 036252 004737 032230 JSR PC,FRDY ; FIND RDY
6882 036256 104121 ERROR 121 ; RDY NOT SET AFTER ST SPIN CMD.
6883
6884 036260 013737 001414 007400 MOV T100,TEMP2 ; SETUP TIMEOUT
6885 036266 004737 032544 JSR PC,FATT1 ; FIND ATTN
6886 036272 104074 ERROR 74 ; NO ATTN AFTER ST SPIN CMD.
6887
6888 036274 005037 007322 CLR UNLD
6889
6890
6891 036300 005737 007324 3$: TST BADHDR ; SEE IF HEADERS VALID
6892 036304 001466 BEQ 4$ ; BR IF YES
6893 036306 005237 007326 INC HPEND
6894
6895 036312 012765 100000 000000 MOV #CCLR,RKCS1(R5)
6896 036320 013765 001222 000010 MOV $UNIT,RKCS2(R5)
6897 036326 012765 000013 000000 MOV #RECAL,RKCS1(R5) ; RECAL CMD
6898 ; RESET CYL DIFF/OFFSET & CYL ADDR REG
6899 ; IN RKMR2 & RKMR3 RESP.
6900 036334 013737 001406 007376 MOV T10,TEMP1 ; SETUP TIMEOUT
6901 036342 004737 032230 JSR PC,FRDY ; FIND RDY
6902 036346 104124 ERROR 124 ; RDY NOT SET AFTER RECAL CMD
6903
6904 036350 012765 000001 000026 MOV #1,RKMR1(R5) ; SELECT WORD 1
6905 036356 004737 033566 JSR PC,GSTAT
6906 036362 032737 020000 007366 BIT #D.RTZ,HMR2
6907 036370 001001 BNE 64$
6908 036372 104244 ERROR 244 ; RTZ NOT SET DURING RECAL CMD
6909 036374 013737 001406 007400 64$: MOV T10,TEMP2 ; SETUP TIMEOUT
6910 036402 004737 032544 JSR PC,FATT1 ; FIND ATTN
6911 036406 104055 ERROR 55 ; NO ATTN AFTER RECAL CMD
6912
6913 036410 012765 100000 000000 MOV #CCLR,RKCS1(R5)
6914 036416 013765 001222 000010 MOV $UNIT,RKCS2(R5) ; DRIVE#

```

```

6915 036424 012765 000005 000000      MOV      #CLEAR,RKCS1(R5)      ;DRIVE CLEAR CMD
6916 036432 013737 001406 007376      MOV      T10,TEMP1           ;SETUP TIMEOUT
6917 036440 004737 032230      JSR      PC,FRDY             ;FIND RDY
6918 036444 104151      ERROR   151                 ;NO RDY AFTER DRIVE CLEAR CMD
6919 036446 004737 032512      JSR      PC,TSTATN          ;TEST FOR ATTN
6920 036452 000401      BR      65$
6921 036454 104154      ERROR   154                 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
6922 036456      65$:
6923
6924
6925 036456 000137 031226      JMP      FORM                ;WRITE VALID FORMATS
6926
6927 036462 005737 000042      4$:  TST      42              ;SEE IF MANUAL OR AUTO MODE
6928 036466 001410      BEQ     5$                  ;BR IF MANUAL MODE
6929 036470 104401 046023      TYPE   ,MSG76              ;PGM ABORTED
6930 036474 005037 031576      CLR     $EOPCT             ;SET UP EOP TO EXIT TO MONITOR
6931 036500 005037 001176      CLR     $ESCAPE
6932 036504 000137 031550      JMP     $EOP1              ;ABORT PGM
6933
6934 036510 104401 046045      5$:  TYPE   ,MSG77              ;CPU HALTED
6935 036514 000000      HALT
6936 036516 000137 013340      JMP     ST5                ;START OVER IF CONTINUE PRESSED
6937
6938
6939
6940      .SBTTL  UNEXPECTED TIMEOUT HANDLER
6941
6942      ;
6943      ;THIS ROUTINE IS ENTERED IF THERE IS
6944      ;A. NON EXISTANT MEMORY (NO SSYN)
6945      ;B. BOUNDARY ERROR
6946      ;C. STACK OVERFLOW
6947      ;
6948      ;
6949      BADTMO: MOV      (SP),RO      ;SAVE PC WHERE TIMEOUT OCCURRED.
6950      TST      -(RO)          ;GET PC BEFORE UPDATE
6951      BIT      #SW13,@SWR     ;INHIBIT ERR TYP0UT?
6952      BNE     1$              ;YES, DON'T TYPE
6953      TYPE   ,EM3           ;ABORT TESTS,UNEXP T.O. @ PC=
6954      MOV     RO,-(SP)       ;SAVE RO FOR TYPEOUT
6955      TYPE   PC              ;TYPE PC
6956      036544 104403      TYPOS      ;GO TYPE--OCTAL ASCII
6957      036546      006        ;TYPE 6 DIGIT(S)
6958      036547      000        ;SUPPRESS LEADING ZEROS
6959      036550 032777 001000 142362 1$:  BIT      #SW9,@SWR        ;LOOP ON ERROR?
6960      036556 001403      BEQ     2$                ;NO BRANCH
6961      036560 022626      CMP     (SP)+,(SP)+      ;YES, RESTORE STACK
6962      036562 000177 142320      JMP     @SLPADR          ;GO TO STARTING ADDR OF TEST
6963      ;THAT GAVE BAD TIMEOUT
6964      036566 032777 040000 142344 2$:  BIT      #SW14,@SWR      ;LOOP ON TEST?
6965      036574 001401      BEQ     3$                ;NO BRANCH
6966      036576 000002      RTI
6967      ;YES
6968      036600 000000      3$:  HALT
6969
6970      ;UNEXPECTED TIME OUT OCCURRED
      ;AS INDICATED. YOU CAN LOOP ON
      ;ERROR, LOOP ON TEST OR INHIBIT

```

```

6971                                     ;ERROR TYPEOUT BY SETTING THOSE
6972                                     ;SWITCHES.
6973
6974 036602 022626                       CMP      (SP)+,(SP)+      ;RESTORE STACK
6975 036604 000137 031550                JMP      $EOP1          ;ABORT TESTS
6976
6977 .SBTTL MEMORY CHECK ENABLE TRAP
6978
6979 036610 012737 036624 001176 MEMERR: MOV      #1$,$ESCAPE
6980 036616 011637 001334                MOV      (SP),TRAPPC    ;STORE PC
6981 036622 104041                        ERROR    41             ;UNEXP MEM PARITY TRAP
6982 036624 005037 001176 1$:          CLR      $ESCAPE
6983 036630 032777 001000 142302        BIT      #SW9,$SWR      ;CHECK IF LOOP ON ERROR
6984 036636 001001                        BNE     2$             ;YES, FORCE STACK AND TRY AGAIN
6985 036640 000002                        RTI
6986
6987 036642 012706 001100 2$:          MOV      #STACK,$SP    ;INIT STACK
6988 036646 000177 142236                JMP      $SLPERA       ;LOOP ON ERROR
6989
6990 .SBTTL RK06 INTERRUPT HANDLER
6991
6992 INTER:  NOP
6993 036652 000240                        NOP
6994 036654 000240                        NOP
6995 036656 000240                        MOV      (SP),RO       ;SAVE PC WHERE INT OCCURRED.
6996 036660 011600                        TST     -(RO)          ;GET PC BEFORE UPDATE.
6997 036662 005740                        TYPE    MSG6           ;INT AT PC=
6998 036664 104401 044473                MOV     RO,-(SP)       ;SAVE RO FOR TYPEOUT
6999 036670 010046                        ;TYPE PC
7000                                     ;GO TYPE--OCTAL ASCII
7001 036672 104403                        .BYTE   6             ;TYPE 6 DIGIT(S)
7002 036674           006                                     ;SUPPRESS LEADING ZEROS
7003 036675           000
7004 036676 000000                        HALT
7005 036700 000240                        NOP
7006 036702 000240                        NOP
7007 036704 000002                        RTI
7008
7009 .SBTTL POWER DOWN AND UP ROUTINES
7010
7011 ;POWER DOWN ROUTINE
7012
7013 036706 012737 036720 000024 $PWRDN: MOV     #SPWRUP,PWRVEC ;SET UP VECTOR
7014 036714 000000                        HALT
7015 036716 000776                        BR      -2             ;HANG UP.
7016
7017 ;POWER UP ROUTINE
7018
7019 036720 005037 036772 $PWRUP: CLR     $PWRCT   ;WAIT LOOP FOR TTY
7020 036724 005237 036772 1$:          INC     $PWRCT        ;WAIT FOR THE INCR
7021 036730 001375                        BNE     1$            ;OF WORD
7022 036732 012737 036706 000024        MOV     #SPWRDN,PWRVEC ;SET POWER DOWN VECTOR
7023 036740 012737 000340 000026        MOV     #PR7,PWRVEC+2 ;PRIORITY 7
7024 036746 012737 000340 000036        MOV     #PR7,TRAPVEC+2 ;LOCKOUT ALL INTERRUPTS FOR TRAPS
7025 036754 012706 001100                MOV     #STACK,$SP    ;INITIALIZE STACK
7026 036760 104401 044663                TYPE    ,MSG11        ;REPORT POWER FAIL

```

```

7027 036764 000005          RESET
7028 036766 000137 015122      JMP      PFSRT
7029
7030 036772 000000          SPWRCT: 0          ;WAIT COUNT FOR TTY
7031
7032
7033          ;DIVISION UTILITY ROUTINE
7034          ;R0-R1-R2-R3=DIVIDEND
7035          ;R4-R5=DIVISOR
7036          ;R0-R1=REMAINDER AFTER DIVISION
7037          ;R2-R3=QUOTIENT AFTER DIVISION
7038          ;ENTER WITH JSR PC,M.DPID
7039
7040
7041 036774 012746 000040      M.DPID: MOV      #40,-(SP)      ;COUNTER FOR DIVISION CYCLES
7042 037000 010446          MOV      R4,-(SP)      ;HI ORDER
7043 037002 010546          MOV      R5,-(SP)      ;LO ORDER TO THE STACK
7044 037004 005466 000002      NEG      2(SP)          ;FORM NEGATIVE
7045 037010 005416          NEG      @SP            ;VERSION OF DIVISOR
7046 037012 005666 000002      SBC      2(SP)
7047 037016 061601          ADD      @SP,R1
7048 037020 005500          ADC      R0            ;PERFORM INIT SUBT.
7049 037022 066600 000002      ADD      2(SP),R0
7050 037026 103445          BCS      M.DP50        ;IF CARRY THEN OVERFLOW HAS OCCURRED
7051 037030 005046          CLR      -(SP)        ;THIS IS A LONGER LASTING CARRY BIT
7052 037032 006103      M.DP40: ROL      R3
7053 037034 006102          ROL      R2
7054 037036 006101          ROL      R1
7055 037040 006100          ROL      R0
7056 037042 005716          TST      @SP            ;TEST CARRY INDICATOR
7057 037044 001410          BEQ      M.DP41        ;IF TO CARRY THEN ADD, ELSE SUBT.
7058 037046 005016          CLR      @SP            ;CLEAR UP FOR NEXT TIME
7059 037050 066601 000002      ADD      2(SP),R1
7060 037054 005500          ADC      R0            ;ADD -(DIVISOR)
7061 037056 005516          ADC      @SP            ;SET CARRY
7062 037060 066600 000004      ADD      4(SP),R0
7063 037064 000404          BR       M.DP42
7064
7065 037066 060501      M.DP41: ADD      R5,R1
7066 037070 005500          ADC      R0            ;ADD +(DIVISOR)
7067 037072 005516          ADC      @SP            ;SET CARRY
7068 037074 060400          ADD      R4,R0
7069 037076 005516      M.DP42: ADC      @SP
7070 037100 005716          TST      @SP            ;SET CARRY
7071 037102 001401          BEQ      .+4            ;TEST THE UPDATE INDICATOR
7072 037104 005203          INC      R3            ;IF 0 FORGET IT
7073 037106 005366 000006      DEC      6(SP)        ;NO CARRY POSSIBLE HERE
7074 037112 003347          BGT      M.DP40        ;DECREMENT CTR
7075 037114 006003          ROR      R3            ;BR IF MORE TO DO
7076 037116 103404          BCS      M.DP44
7077 037120 060501          ADD      R5,R1
7078 037122 005500          ADC      R0
7079 037124 060400          ADD      R4,R0
7080 037126 000241          CLC
7081
7082 037130 006103      M.DP44: ROL      R3

```

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 132  
POWER DOWN AND UP ROUTINES

SEQ 0132

7083 037132 062706 000010  
7084 037136 000242  
7085 037140 000207  
7086  
7087 037142 062706 000006  
7088 037146 000262  
7089 037150 000207  
7090

ADD #10,SP ;ADJUST STACK BY 4 WORDS  
CLV  
RTS PC  
M.DPSO: ADD #6,SP  
SEV  
RTS PC



```

7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105 037152
7106 037152 104407
7107 037154 032777 040000 141756
7108 037162 001114
7109
7110 037164 000416
7111
7112 037166 013746 000004
7113 037172 012737 037212 000004
7114 037200 005737 177060
7115 037204 012637 000004
7116 037210 000463
7117 037212 022626
7118 037214 012637 000004
7119 037220 000423
7120 037222
7121 037222 032777 000400 141710
7122 037230 001404
7123 037232 127737 141702 001102
7124 037240 001465
7125 037242 105737 001103
7126 037246 001421
7127 037250 123737 001115 001103
7128 037256 101015
7129 037260 032777 001000 141652
7130 037266 001404
7131 037270 013737 001110 001106
7132 037276 000446
7133 037300 105037 001103
7134 037304 005037 001174
7135 037310 000415
7136 037312 032777 004000 141620
7137 037320 001011
7138 037322 005737 001216
7139 037326 001406
7140 037330 005237 001104
7141 037334 023737 001174 001104
7142 037342 002024
7143 037344 012737 000001 001104
7144 037352 013737 037430 001174
7145 037360 105237 001102
7146 037364 113737 001102 001214

```

```

.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 (SERFLG) INTO DISPLAY<15:08>
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1 LOOP ON TEST
*SW11=1 INHIBIT ITERATIONS
*SW09=1 LOOP ON ERROR
*SW08=1 LOOP ON TEST IN SWR<7:0>
*CALL
* SCOPE ;;SCOPE=IOT

$SCOPE:
CKSWR
1$: BIT #BIT14,@SWR ;;TEST FOR CHANGE IN SOFT-SWR
BNE $OVER ;;LOOP ON PRESENT TEST?
;;YES IF SW14=1
*****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
THIS INSTRUCTION TO A "NOP" (NOP=240)
MOV @ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
MOV #5$,@ERRVEC ;;SET FOR TIMEOUT
TST @177060 ;;TIME OUT ON XOR?
MOV (SP)+,@ERRVEC ;;RESTORE THE ERROR VECTOR
BR $SVLAD ;;GO TO THE NEXT TEST
5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
MOV (SP)+,@ERRVEC ;;RESTORE THE ERROR VECTOR
BR 7$ ;;LOOP ON THE PRESENT TEST
6$; *****END OF CODE FOR THE XOR TESTER*****
BIT #BIT08,@SWR ;;LOOP ON SPEC. TEST?
BEQ 2$ ;;BR IF NO
CMPB @SWR,STSTNM ;;ON THE RIGHT TEST? SWR<7:0>
BEQ $OVER ;;BR IF YES
2$: TSTB SERFLG ;;HAS AN ERROR OCCURRED?
BEQ 3$ ;;BR IF NO
CMPB $ERMAX,SERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
BHI 3$ ;;BR IF NO
BIT #BIT09,@SWR ;;LOOP ON ERROR?
BEQ 4$ ;;BR IF NO
7$: MOV $LPERR,$LPAOR ;;SET LOOP ADDRESS TO LAST SCOPE
BR $OVER
4$: CLRB SERFLG ;;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
MOV $STSTNM,$STSTN ;;SET TEST NUMBER IN APT MAILBOX

```



```

7203 037616
7204 037616 022737 031636 000042
7205 037624 001001
7206 037626 000000
7207 037630
7208 037630 000002
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226 037632 105737 001157
7227 037636 100002
7228 037640 000000
7229 037642 000430
7230 037644 010046
7231 037646 017600 000002
7232 037652 122737 000001 001230
7233 037660 001011
7234 037662 132737 000100 001231
7235 037670 001405
7236 037672 010037 037702
7237 037676 004737 040346
7238 037702 000000
7239 037704 132737 000040 001231
7240 037712 001003
7241 037714 112046
7242 037716 001005
7243 037720 005726
7244 037722 012600
7245 037724 062716 000002
7246 037730 000002
7247 037732 122716 000011
7248 037736 001430
7249 037740 122716 000200
7250 037744 001006
7251 037746 005726
7252 037750 104401
7253 037752 001205
7254 037754 105037 040110
7255 037760 000755
7256 037762 004737 040044
7257 037766 123726 001156
7258 037772 001350

```

```

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

*****
*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
*
*CALL:
*1) USING A TRAP INSTRUCTION
* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
*OR
* TYPE
* MESADR
*

$TYPE: TSTB $TFPLG ;; IS THERE A TERMINAL?
        BPL 1$ ;;BR IF YES
        HALT HERE IF NO TERMINAL
        BR 3$ LEAVE
1$: MOV RO, -(SP) SAVE RO
   MOV 22(SP), RO GET ADDRESS OF ASCIZ STRING
   CMPB #APTENV, $ENV RUNNING IN APT MODE
   BNE 62$ NO, GO CHECK FOR APT CONSOLE
   BITB #APTSPool, $ENVM SPOOL MESSAGE TO APT
   BEQ 62$ NO, GO CHECK FOR CONSOLE
   MOV RO, 61$ SETUP MESSAGE ADDRESS FOR APT
   JSR PC, $ATY3 SPOOL MESSAGE TO APT
   .WORD 0 MESSAGE ADDRESS
61$: BITB #APTCSUP, $ENVM APT CONSOLE SUPPRESSED
62$: BNE 60$ YES, SKIP TYPE OUT
   MOVB (RO)+, -(SP) PUSH CHARACTER TO BE TYPED ONTO STACK
   BNE 4$ BR IF IT ISN'T THE TERMINATOR
   TST (SP)+ IF TERMINATOR POP IT OFF THE STACK
60$: MOV (SP)+, RO RESTORE RO
3$: ADD #2, (SP) ADJUST RETURN PC
   RTI RETURN
4$: CMPB #HT, (SP) ;;BRANCH IF <HT>
   BEQ 8$
   CMPB #CRLF, (SP) ;;BRANCH IF NOT <CRLF>
   BNE 5$
   TST (SP)+ ;;POP <CR><LF> EQUIV
   TYPE TYPE A CR AND LF

5$: JSR PC, $TYPEC ;;CLEAR CHARACTER COUNT
6$: CMPB $FILLC, (SP)+ ;;GET NEXT CHARACTER
   BNE 2$ ;;GO TYPE THIS CHARACTER
   ;; IS IT TIME FOR FILLER CHARS.?
   ;; IF NO GO GET NEXT CHAR.

```

```

7259 037774 013746 001154          MOV     $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED
7260                                     ;; AND THE NULL CHAR.
7261 040000 105366 000001 7$:    DECB    1(SP)          ;; DOES A NULL NEED TO BE TYPED?
7262 040004 002770                                     ;; BR IF NO--GO POP THE NULL OFF OF STACK
7263 040006 004737 040044          JSR     PC,$TYPEC       ;; GO TYPE A NULL
7264 040012 105337 040110          DECB    $CHARCNT        ;; DO NOT COUNT AS A COUNT
7265 040016 000770          BR      7$             ;; LOOP

```

;HORIZONTAL TAB PROCESSOR

```

7269 040020 112716 000040 8$:    MOVB    #' (SP)        ;; REPLACE TAB WITH SPACE
7270 040024 004737 040044 9$:    JSR     PC,$TYPEC       ;; TYPE A SPACE
7271 040030 132737 000007 040110 BITB    #',$CHARCNT     ;; BRANCH IF NOT AT
7272 040036 001372                                     ;; TAB STOP
7273 040040 005726          TST     (SP)+          ;; POP SPACE OFF STACK
7274 040042 000724          BR      2$            ;; GET NEXT CHARACTER
7275 040044 105777 141100 $TYPEC: TSTB    2$TPS        ;; WAIT UNTIL PRINTER IS READY
7276 040050 100375          BPL     $TYPEC         ;;
7277 040052 116677 000002 141072 MOVB    2(SP),2$TPB     ;; LOAD CHAR TO BE TYPED INTO DATA REG.
7278 040060 122766 000015 000002 CMPB    #CR,2(SP)       ;; IS CHARACTER A CARRIAGE RETURN?
7279 040066 001003          BNE     1$            ;; BRANCH IF NO
7280 040070 105037 040110          CLRB    $CHARCNT      ;; YES--CLEAR CHARACTER COUNT
7281 040074 000406          BR      $TYPEX        ;; EXIT
7282 040076 122766 000012 000002 1$:    CMPB    #LF,2(SP)     ;; IS CHARACTER A LINE FEED?
7283 040104 001402          BEQ     $TYPEX        ;; BRANCH IF YES
7284 040106 105227          INCB    (PC)+         ;; COUNT THE CHARACTER
7285 040110 000000          $CHARCNT: .WORD    0  ;; CHARACTER COUNT STORAGE
7286 040112 000207          $TYPEX:  RTS         PC

```

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7300
7301
7302
7303
7304
7305
7306
7307
7308
7309
7310
7311
7312
7313
7314

```

;\*\*\*\*\*  
;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT  
;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE  
;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED  
;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE  
;REPLACED WITH SPACES.  
;CALL:  
; \* MOV NUM,-(SP) ;; PUT THE BINARY NUMBER ON THE STACK  
; \* TYPDS ;; GO TO THE ROUTINE

```

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

```

```

7315 040164 016001 040320      MOV      $DTBL(R0),R1      ;; GET THE CONSTANT
7316 040170 160105      3$: SUB      R1,R5          ;; FORM THIS BCD DIGIT
7317 040172 002402      BLT      4$              ;; BR IF DONE
7318 040174 005202      INC      R2              ;; INCREASE THE BCD DIGIT BY 1
7319 040176 000774      BR       3$
7320 040200 060105      4$: ADD      R1,R5          ;; ADD BACK THE CONSTANT
7321 040202 005702      TST      R2              ;; CHECK IF BCD DIGIT=0
7322 040204 001002      BNE      5$              ;; FALL THROUGH IF 0
7323 040206 105716      TSTB     (SP)            ;; STILL DOING LEADING 0'S?
7324 040210 100407      BMI      7$              ;; BR IF YES
7325 040212 106316      5$: ASLB     (SP)          ;; MSD?
7326 040214 103003      BCC      6$              ;; BR IF NO
7327 040216 116663 000001 177777      MOVB     1(SP),-1(R3)     ;; YES--SET THE SIGN
7328 040224 052702 000060      6$: BIS      #'0,R2       ;; MAKE THE BCD DIGIT ASCII
7329 040230 052702 000040      7$: BIS      #' ,R2       ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
7330 040234 110223      MOVB     R2,(R3)+        ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
7331 040236 005720      TST      (R0)+          ;; JUST INCREMENTING
7332 040240 020027 000010      CMP      R0,#10         ;; CHECK THE TABLE INDEX
7333 040244 002746      BLT      2$              ;; GO DO THE NEXT DIGIT
7334 040246 003002      BGT      8$              ;; GO TO EXIT
7335 040250 010502      MOV      R5,R2          ;; GET THE LSD
7336 040252 000764      BR       6$              ;; GO CHANGE TO ASCII
7337 040254 105726      8$: TSTB     (SP)+        ;; WAS THE LSD THE FIRST NON-ZERO?
7338 040256 100003      BPL      9$              ;; BR IF NO
7339 040260 116663 177777 177776      9$: MOVB     -1(SP),-2(R3) ;; YES--SET THE SIGN FOR TYPING
7340 040266 105013      CLRB     (R3)           ;; SET THE TERMINATOR
7341 040270 012605      MOV      (SP)+,R5        ;; POP STACK INTO R5
7342 040272 012603      MOV      (SP)+,R3        ;; POP STACK INTO R3
7343 040274 012602      MOV      (SP)+,R2        ;; POP STACK INTO R2
7344 040276 012601      MOV      (SP)+,R1        ;; POP STACK INTO R1
7345 040300 012600      MOV      (SP)+,R0        ;; POP STACK INTO R0
7346 040302 104401 040330      TYPE     $DBLK           ;; NOW TYPE THE NUMBER
7347 040306 016666 000002 000004      MOV      2(SP),4(SP)     ;; ADJUST THE STACK
7348 040314 012616      MOV      (SP)+,(SP)
7349 040316 000002      RTI
7350 040320 023420      $DTBL: 10000.           ;; RETURN TO USER
7351 040322 001750      1000.
7352 040324 000144      100.
7353 040326 000012      10.
7354 040330 000004      $DBLK: .BLKW 4
7355      .SBTTL APT COMMUNICATIONS ROUTINE
7356
7357      ;; *****
7358 040340 112737 000001 040604 $ATY1: MOVB     #1,$FFLG     ;; TO REPORT FATAL ERROR
7359 040346 112737 000001 040602 $ATY3: MOVB     #1,$MFLG     ;; TO TYPE A MESSAGE
7360 040354 000403      BR       $ATYC
7361 040356 112737 000001 040604 $ATY4: MOVB     #1,$FFLG     ;; TO ONLY REPORT FATAL ERROR
7362 040364      $ATYC:
7363 040364 010046      MOV      R0,-(SP)        ;; PUSH R0 ON STACK
7364 040366 010146      MOV      R1,-(SP)        ;; PUSH R1 ON STACK
7365 040370 105737 040602      TSTB     $MFLG          ;; SHOULD TYPE A MESSAGE?
7366 040374 001450      BEQ      5$              ;; IF NOT: BR
7367 040376 122737 000001 001230      CMPB     #APTENV,$ENV    ;; OPERATING UNDER APT?
7368 040404 001031      BNE      3$              ;; IF NOT: BR
7369 040406 132737 000100 001231      BITB     #APTPOOL,$ENVM  ;; SHOULD SPOOL MESSAGES?
7370 040414 001425      BEQ      3$              ;; IF NOT: BR

```

```

7371 040416 017600 000004      MOV      24(SP),RO      ;; GET MESSAGE ADDR.
7372 040422 062766 000002 000004      ADD      #2,4(SP)      ;; BUMP RETURN ADDR.
7373 040430 005737 001210      1S:     TST      $MSGTYPE      ;; SEE IF DONE W/ LAST XMISSION?
7374 040434 001375      BNE      1S              ;; IF NOT: WAIT
7375 040436 010037 001224      MOV      RO,$MSGAD      ;; PUT ADDR IN MAILBOX
7376 040442 105720      2S:     TSTB     (RO)+      ;; FIND END OF MESSAGE
7377 040444 001375      BNE      2S              ;;
7378 040446 161375 001224      SUB      $MSGAD,RO      ;; SUB START OF MESSAGE
7379 040452 006200      ASR      RO              ;; GET MESSAGE LNGTH IN WORDS
7380 040454 010037 001226      MOV      RO,$MSGLG      ;; PUT LENGTH IN MAILBOX
7381 040460 012737 000004 001210      MOV      #4,$MSGTYPE      ;; TELL APT TO TAKE MSG.
7382 040466 000413      BR       5S              ;;
7383 040470 017637 000004 040514 3S:     MOV      24(SP),4S      ;; PUT MSG ADDR IN JSR LINKAGE
7384 040476 062766 000002 000004      ADD      #2,4(SP)      ;; BUMP RETURN ADDRESS
7385 040504 013746 177776      MOV      177776,-(SP)   ;; PUSH 177776 ON STACK
7386 040510 004737 037632      JSR      PC,$TYPE      ;; CALL TYPE MACRO
7387 040514 000000      4S:     .WORD    0
7388 040516      5S:
7389 040516 105737 040604      10S:    TSTB     $FFLG      ;; SHOULD REPORT FATAL ERROR?
7390 040522 001416      BEQ      12S            ;; IF NOT: BR
7391 040524 005737 001230      TST      $ENV          ;; RUNNING UNDER APT?
7392 040530 001413      BEQ      12S            ;; IF NOT: BR
7393 040532 005737 001210      11S:   TST      $MSGTYPE      ;; FINISHED LAST MESSAGE?
7394 040536 001375      BNE      11S           ;; IF NOT: WAIT
7395 040540 017637 000004 001212      MOV      24(SP),$FATAL  ;; GET ERROR #
7396 040546 062766 000002 000004      ADD      #2,4(SP)      ;; BUMP RETURN ADDR.
7397 040554 005237 001210      INC      $MSGTYPE      ;; TELL APT TO TAKE ERROR
7398 040560 105037 040604      12S:   CLRB     $FFLG      ;; CLEAR FATAL FLAG
7399 040564 105037 040603      CLRB     $LFLG        ;; CLEAR LOG FLAG
7400 040570 105037 040602      CLRB     $MFLG        ;; CLEAR MESSAGE FLAG
7401 040574 012601      MOV      (SP)+,R1      ;; POP STACK INTO R1
7402 040576 012600      MOV      (SP)+,RO      ;; POP STACK INTO RO
7403 040600 000207      RTS      PC            ;; RETURN
7404 040602 000      $MFLG: .BYTE    0      ;; MESSG. FLAG
7405 040603 000      $LFLG: .BYTE    0      ;; LOG FLAG
7406 040604 000      $FFLG: .BYTE    0      ;; FATAL FLAG
7407      .EVEN
7408      APTSIZE=200
7409      APTENV=001
7410      APTSPool=100
7411      APTCSUP=040
7412      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
7413
7414      ;*****
7415      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
7416      ;OCTAL (ASCII) NUMBER AND TYPE IT.
7417      ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
7418      ;CALL:
7419      ;      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
7420      ;      TYPOS      ;; CALL FOR TYPEOUT
7421      ;      .BYTE    N              ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
7422      ;      .BYTE    M              ;; M=1 OR 0
7423      ;      ;; 1=TYPE LEADING ZEROS
7424      ;      ;; 0=SUPPRESS LEADING ZEROS
7425
7426      ;$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST

```

```

7427      ;*STYPOS OR STYPOC
7428      ;*CALL:
7429      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
7430      ;*      TYPON      ;;CALL FOR TYPEOUT
7431      ;*
7432      ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
7433      ;*CALL:
7434      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
7435      ;*      TYPOC      ;;CALL FOR TYPEOUT
7436
7437 040606 017646 000000      STYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
7438 040612 116637 000001 041031      MOVVB 1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
7439 040620 112637 041033      MOVVB (SP)+,SOMODE+1      ;;NUMBER OF DIGITS TO TYPE
7440 040624 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
7441 040630 000406      BR      STYPON
7442 040632 112737 000001 041031      STYPOC: MOVVB #1,SOFILL      ;;SET THE ZERO FILL SWITCH
7443 040640 112737 000006 041033      MOVVB #6,SOMODE+1      ;;SET FOR SIX(6) DIGITS
7444 040646 112737 000005 041030      STYPON: MOVVB #5,SOCNT      ;;SET THE ITERATION COUNT
7445 040654 010346      MOV      R3,-(SP)      ;;SAVE R3
7446 040656 010446      MOV      R4,-(SP)      ;;SAVE R4
7447 040660 010546      MOV      R5,-(SP)      ;;SAVE R5
7448 040662 113704 041033      MOVVB  $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
7449 040666 005404      NEG      R4
7450 040670 062704 000006      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
7451 040674 110437 041032      MOVVB  R4,SOMODE      ;;SAVE IT FOR USE
7452 040700 113704 041031      MOVVB  $OFILL,R4      ;;GET THE ZERO FILL SWITCH
7453 040704 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
7454 040710 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
7455 040712 006105      1$: ROL      R5      ;;ROTATE MSB INTO "C"
7456 040714 000404      BR      3$      ;;GO DO MSB
7457 040716 006105      2$: ROL      R5      ;;FORM THIS DIGIT
7458 040720 006105      ROL      R5
7459 040722 006105      ROL      R5
7460 040724 010503      MOV      R5,R3
7461 040726 006103      3$: ROL      R3      ;;GET LSB OF THIS DIGIT
7462 040730 105337 041032      DECB  $OMODE      ;;TYPE THIS DIGIT?
7463 040734 100016      BPL      7$      ;;BR IF NO
7464 040736 047703 177770      BIC      #177770,R3      ;;GET RID OF JUNK
7465 040742 001002      BNE      4$      ;;TEST FOR 0
7466 040744 075704      TST      R4      ;;SUPPRESS THIS 0?
7467 040746 001403      BEQ      5$      ;;BR IF YES
7468 040750 005204      4$: INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
7469 040752 052703 000060      BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
7470 040756 052703 000040      5$: BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY
7471 040762 110337 041026      MOVVB  R3,#$      ;;SAVE FOR TYPING
7472 040766 104401 041026      TYPE      8$      ;;GO TYPE THIS DIGIT
7473 040772 105337 041030      7$: DECB  $OCNT      ;;COUNT BY 1
7474 040776 003347      BGT      2$      ;;BR IF MORE TO DO
7475 041000 002402      BLT      6$      ;;BR IF DONE
7476 041002 005204      INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
7477 041004 000744      BR      2$      ;;GO DO THE LAST DIGIT
7478 041006 012605      6$: MOV      (SP)+,R5      ;;RESTORE R5
7479 041010 012604      MOV      (SP)+,R4      ;;RESTORE R4
7480 041012 012603      MOV      (SP)+,R3      ;;RESTORE R3
7481 041014 016666 000002 000004      MOV      2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
7482 041022 012616      MOV      (SP)+,(SP)

```

```

7483 041024 000002
7484 041026 000
7485 041027 000
7486 041030 000
7487 041031 000
7488 041032 000000
7489
7490
7491
7492
7493 041034 000000
7494 041036 000000
7495 041040 000000
7496 041042 000001
7497 041043
7498 041044
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508 041044 005037 041034
7509 041050 012737 041042 041036
7510 041056 013737 041036 041040
7511 041064 012737 041114 000060
7512 041072 012737 000200 000062
7513 041100 005777 140042
7514 041104 012777 000100 140032
7515 041112 000207
7516
7517
7518
7519
7520
7521
7522
7523
7524 041114 117746 140026
7525 041120 042716 177600
7526 041124 021627 000003
7527 041130 001007
7528 041132 104401 042242
7529 041136 004737 041044
7530 041142 005726
7531 041144 000137 036172
7532 041150 021627 000007
7533 041154 001004
7534 041156 022737 000176 001140
7535 041164 001500
7536
7537 041166
7538 041166 022737 000001 041034

```

```

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

;*****
.ENABL LSB
STKCNT: .WORD 0 ;; NUMBER OF ITEMS IN QUEUE
STKQIN: .WORD 0 ;; INPUT POINTER
STKQOUT: .WORD 0 ;; OUTPUT POINTER
STKQSRV: .BLKB 1 ;; TTY KEYBOARD QUEUE
STKQEND=.
.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,STKINT
; * RETURN
STKINT: CLR STKCNT ;; CLEAR COUNT OF ITEMS IN QUEUE
        MOV #STKQSRV,STKQIN ;; MOVE THE STARTING ADDRESS OF THE
        MOV STKQIN,STKQOUT ;; QUEUE INTO THE INPUT & OUTPUT POINTERS.
        MOV #STKSRV,@STKVEC ;; INITIALIZE THE KEYBOARD VECTOR
        MOV #200,@STKVEC+2 ;; "BR" LEVEL 4
        TST @STKB ;; CLEAR DONE FLAG
        MOV #100,@STKS ;; ENABLE TTY KEYBOARD INTERRUPT
        RTS PC ;; RETURN TO CALLER

; *TK SERVICE ROUTINE
; *THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
; *BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
; *IT IN THE QUEUE.
; *IF THE CHARACTER IS A "CONTROL-C" (↑C) STKINT IS CALLED AND
; *UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
STKSRV: MOVB @STKB,-(SP) ;; PICKUP THE CHARACTER
        BIC #↑C177,(SP) ;; STRIP THE JUNK
        CMP (SP),#3 ;; IS IT A CONTROL C?
        BNE 1$ ;; BRANCH IF NO
        TYPE ,SCNTLC ;; TYPE A CONTROL-C (↑C)
        JSR PC,STKINT ;; 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,STKCNT ;; IS THE QUEUE FULL?

```



```

7539 041174 001004      BNE      3$          ;: BRANCH IF NO
7540 041176 104401 001200  TYPE     $BELL      ;: RING THE TTY BELL
7541 041202 005726      TST     (SP)+       ;: CLEAN CHARACTER OFF OF STACK
7542 041204 000451      BR       5$          ;: EXIT
7543 041206 021627 000023  3$:     CMP     (SP),#23 ;: IS IT A CONTROL-S?
7544 041212 001021      BNE     32$         ;: BRANCH IF NO
7545 041214 005077 137724  CLR     @STKS      ;: DISABLE TTY KEYBOARD INTERRUPTS
7546 041220 005726      TST     (SP)+       ;: CLEAN CHAR OFF STACK
7547 041222 105777 137716  31$:   TSTB    @STKS      ;: WAIT FOR A CHAR
7548 041226 100375      BPL     31$         ;: LOOP UNTIL ITS THERE
7549 041230 117746 137712  MOVB   @STKB,-(SP) ;: GET THE CHARACTER
7550 041234 042716 177600  BIC     #1C177,(SP) ;: MAKE IT 7-BIT ASCII
7551 041240 022627 000021  CMP     (SP)+,#21  ;: IS IT A CONTROL-Q?
7552 041244 001366      BNE     31$         ;: BRANCH IF NO
7553 041246 012777 000100 137670  MOV     #100,@STKS ;: REENABLE TTY KEYBOARD INTERRUPTS
7554 041254 000002      RTI                    ;: RETURN
7555 041256 005237 041034  32$:   INC     $TKCNT     ;: COUNT THIS CHARACTER
7556 041262 021627 000140  CMP     (SP),#140  ;: IS IT UPPER CASE?
7557 041266 002405      BLT     4$          ;: BRANCH IF YES
7558 041270 021627 000175  CMP     (SP),#175  ;: IS IT A SPECIAL CHAR?
7559 041274 003002      BGT     4$          ;: BRANCH IF YES
7560 041276 042716 000040  BIC     #40,(SP)    ;: MAKE IT UPPER CASE
7561 041302 112677 177530  4$:   MOVB   (SP)+,@STKQIN ;: AND PUT IT IN QUEUE
7562 041306 005237 041036  INC     $TKQIN      ;: UPDATE THE POINTER
7563 041312 023727 041036 041043  CMP     $TKQIN,$STKQEND ;: GO OFF THE END?
7564 041320 001003      BNE     5$          ;: BRANCH IF NO
7565 041322 012737 041042 041036  MOV     @STKQSRST,$TKQIN ;: RESET THE POINTER
7566 041330 000002  5$:   RTI                    ;: RETURN

```

```

7567
7568 ;:*****
7569 ;: *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
7570 ;: *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
7571 ;: *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
7572 ;: *CALL WHEN OPERATING IN TTY INTERRUPT MODE.

```

```

7573 041332 022737 000176 001140 $CKSWR: CMP     $SWREG,$SWR ;: IS THE SOFT-SWR SELECTED
7574 041340 001124      BNE     15$         ;: EXIT IF NOT
7575 041342 105777 137576  TSTB   @STKS      ;: IS A CHAR WAITING?
7576 041346 100121      BPL     15$         ;: IF NOT, EXIT
7577 041350 117746 137572  MOVB   @STKB,-(SP) ;: YES
7578 041354 042716 177600  BIC     #1C177,(SP) ;: MAKE IT 7-BIT ASCII
7579 041360 021627 000007  CMP     (SP),#7    ;: IS IT A CONTROL-G?
7580 041364 001300      BNE     2$          ;: IF NOT, PUT IT IN THE TTY QUEUE
7581 ;: AND EXIT

```

```

7582
7583 ;:*****
7584 ;: *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
7585 ;: *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
7586 ;: *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.

```

```

7587 041366 123727 001134 000001 6$:   CMPB   $AUTOB,#1 ;: ARE WE RUNNING IN AUTO-MODE?
7588 041374 001674      BEQ     2$          ;: BRANCH IF YES
7589 041376 005726      TST     (SP)+       ;: CLEAR CONTROL-G OFF STACK
7590 041400 004737 041044  JSR     PC,$TKINT   ;: FLUSH THE TTY INPUT QUEUE
7591 041404 005077 137534  CLR     @STKS      ;: DISABLE TTY KEYBOARD INTERRUPTS
7592 041410 112737 000001 001135  MOVB   #1,$INTAG   ;: SET INTERRUPT MODE INDICATOR
7593
7594 041416 104401 042254  TYPE     ,SCNTLG    ;: ECHO THE CONTROL-G (+G)

```

7595	041422	104401	042261		\$GTSWR: TYPE	\$MSWR	:: TYPE CURRENT CONTENTS
7596	041426	013746	000176		MOV	\$WREG, -(SP)	:: SAVE SWREG FOR TYPEOUT
7597	041432	104402			TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
7598	041434	104401	042272		TYPE	, \$MNEW	:: PROMPT FOR NEW SWR
7599	041440	005046		19\$:	CLR	-(SP)	:: CLEAR COUNTER
7600	041442	005046			CLR	-(SP)	:: THE NEW SWR
7601	041444	105777	137474	7\$:	TSTB	\$STKS	:: CHAR THERE?
7602	041450	100375			BPL	7\$	:: IF NOT TRY AGAIN
7603							
7604	041452	117746	137470		MOVB	\$STKB, -(SP)	:: PICK UP CHAR
7605	041456	042716	177600		BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
7606							
7607	041462	021627	000003		CMP	(SP), #3	:: IS IT A CONTROL-C?
7608	041466	001015			BNE	9\$	:: BRANCH IF NOT
7609	041470	104401	042242		TYPE	\$CNTLC	:: YES, ECHO CONTROL-C (↑C)
7610	041474	062706	000006		ADD	#6, SP	:: CLEAN UP STACK
7611	041500	123727	001135	000001	CMPB	\$INTAG, #1	:: REENABLE TTY KEYBOARD INTERRUPTS?
7612	041506	001003			BNE	8\$	:: BRANCH IF NO
7613	041510	012777	000100	137425	MOV	#100, \$STKS	:: ALLOW TTY KEYBOARD INTERRUPTS
7614	041516	000137	036172	8\$:	JMP	STOP	:: CONTROL-C RESTART
7615							
7616							
7617	041522	021627	000025	9\$:	CMP	(SP), #25	:: IS IT A CONTROL-U?
7618	041526	001005			BNE	10\$	:: BRANCH IF NOT
7619	041530	104401	042247		TYPE	\$CNTLU	:: YES, ECHO CONTROL-U (↑U)
7620	041534	062706	000006	20\$:	ADD	#6, SP	:: IGNORE PREVIOUS INPUT
7621	041540	000737			BR	19\$	:: LET'S TRY IT AGAIN
7622							
7623							
7624	041542	021627	000015	10\$:	CMP	(SP), #15	:: IS IT A <CR>?
7625	041546	001022			BNE	16\$	:: BRANCH IF NO
7626	041550	005766	000004		TST	4(SP)	:: YES, IS IT THE FIRST CHAR?
7627	041554	001403			BEQ	11\$	:: BRANCH IF YES
7628	041556	016677	000002	137354	MOV	2(SP), \$SWR	:: SAVE NEW SWR
7629	041564	062706	000006	11\$:	ADD	#6, SP	:: CLEAN UP STACK
7630	041570	104401	001205	14\$:	TYPE	\$CRLF	:: ECHO <CR> AND <LF>
7631	041574	123727	001135	000001	CMPB	\$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS?
7632	041602	001003			BNE	15\$	:: BRANCH IF NOT
7633	041604	012777	000100	137332	MOV	#100, \$STKS	:: RE-ENABLE TTY KBD INTERRUPTS
7634	041612	000002		15\$:	RTI		:: RETURN
7635	041614	004737	040044	16\$:	JSR	PC, \$TYPEC	:: ECHO CHAR
7636	041620	021627	000060		CMP	(SP), #60	:: CHAR < 0?
7637	041624	002420			BLT	18\$	:: BRANCH IF YES
7638	041626	021627	000067		CMP	(SP), #67	:: CHAR > 7?
7639	041632	003015			BGT	18\$	:: BRANCH IF YES
7640	041634	042726	000060		BIC	#60, (SP)+	:: STRIP-OFF ASCII
7641	041640	005766	000002		TST	2(SP)	:: IS THIS THE FIRST CHAR
7642	041644	001403			BEQ	17\$	:: BRANCH IF YES
7643	041646	006316			ASL	(SP)	:: NO, SHIFT PRESENT
7644	041650	006316			ASL	(SP)	:: CHAR OVER TO MAKE
7645	041652	006316			ASL	(SP)	:: ROOM FOR NEW ONE.
7646	041654	005266	000002	17\$:	INC	2(SP)	:: KEEP COUNT OF CHAR
7647	041660	056616	177776		BIS	-2(SP), (SP)	:: SET IN NEW CHAR
7648	041664	000667			BR	7\$	:: GET THE NEXT ONE
7649	041666	104401	001204	18\$:	TYPE	\$QUES	:: TYPE ?<CR><LF>
7650	041672	000720			BR	20\$	:: SIMULATE CONTROL-U

```

7651
7652
7653
7654
7655
7656
7657
7658
7659
7660
7661
7662 041674 011646
7663 041676 016666 000004 000002
7664 041704 005066 000004
7665 041710 005046
7666 041712 012746 041720
7667 041716 000002
7668 041720
7669 041720 005737 041034
7670 041724 001775
7671 041726 005337 041034
7672 041732 117766 177102 000004
7673 041740 005237 041040
7674 041744 023727 041040 041043
7675 041752 001003
7676 041754 012737 041042 041040
7677 041762 000002
7678
7679
7680
7681
7682
7683
7684
7685 041764 010346
7686 041766 005046
7687 041770 012703 042220
7688 04 774 022703 042242
7689 042000 101456
7690 042002 104410
7691 042004 112613
7692 042006 122713 000177
7693 042012 001022
7694 042014 005716
7695 042016 001007
7696 042020 112737 000134 042216
7697 042026 104401 042216
7698 042032 012716 177777
7699 042036 005303
7700 042040 020327 042220
7701 042044 103434
7702 042046 111337 042216
7703 042052 104401 042216
7704 042056 000746
7705 042060 005716
7706 042062 001406

```

```

.DSABL LSB

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

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

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

2$:
*****
*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
*CALL:
*   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

SRDLIN: 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     1-1, (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 TYPE
5$:     TST     (SP)           ;; GO READ ANOTHER CHAR.
        BEQ     7$              ;; RUBOUT KEY SET?
        BR     10$             ;; BR IF NO

```



```

7763 042320 010246          MOV      R2,-(SP)          ;; PUSH R2 ON STACK
7764 042322 104411          1$:     RDLIN           ;; READ AN ASCIZ LINE
7765 042324 012600          MOV      (SP)+,R0        ;; GET ADDRESS OF 1ST CHARACTER
7766 042326 010037 042432    MOV      R0,$$          ;; AND SAVE IT
7767 042332 005001          CLR      R1             ;; CLEAR DATA WORD
7768 042334 005002          CLR      R2
7769 042336 112046          2$:     MOVB      (R0)+,-(SP) ;; PICKUP THIS CHARACTER
7770 042340 001420          BEQ      3$             ;; IF ZERO GET OUT
7771 042342 122716 000060      CMPB     #'0,(SP)       ;; MAKE SURE THIS CHARACTER
7772 042346 003026          BGT      4$             ;; IS AN OCTAL DIGIT
7773 042350 122716 000067      CMPB     #'7,(SP)
7774 042354 002423          BLT      4$
7775 042356 006301          ASL      R1             ;; *2
7776 042360 006102          ROL      R2
7777 042362 006301          ASL      R1             ;; *4
7778 042364 006102          ROL      R2
7779 042366 006301          ASL      R1             ;; *8
7780 042370 006102          ROL      R2
7781 042372 042716 177770      BIC      #'C7,(SP)      ;; STRIP THE ASCII JUNK
7782 042376 062601          ADD      (SP)+,R1       ;; ADD IN THIS DIGIT
7783 042400 000756          BR       2$             ;; LOOP
7784 042402 005726          3$:     TST      (SP)+     ;; CLEAN TERMINATOR FROM STACK
7785 042404 010166 000012      MOV      R1,12(SP)      ;; SAVE THE RESULT
7786 042410 010237 042442      MOV      R2,$HIOCT
7787 042414 012602          MOV      (SP)+,R2       ;; POP STACK INTO R2
7788 042416 012601          MOV      (SP)+,R1       ;; POP STACK INTO R1
7789 042420 012600          MOV      (SP)+,R0       ;; POP STACK INTO R0
7790 042422 000002          RTI
7791 042424 005726          4$:     TST      (SP)+     ;; CLEAN PARTIAL FROM STACK
7792 042426 105010          CLRB     (R0)           ;; SET A TERMINATOR
7793 042430 104401          TYPE
7794 042432 000000          5$:     .WORD     0         ;; TYPE UP THRU THE BAD CHAR.
7795 042434 104401 001204      TYPE     $QUES         ;; "?" "CR" & "LF"
7796 042440 000730          BR       1$             ;; TRY AGAIN
7797 042442 000000          $HIOCT: .WORD     0     ;; HIGH ORDER BITS GO HERE
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809 042444 104413          $SBTTL DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
7810 042446 016601 000002      ;; *****
7811 042452 012705 042563      ;; *THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
7812 042456 012704 000014      ;; *UNSIGNED OCTAL ASCIZ NUMBER.
7813 042462 012703 177770      ;; *CALL
7814 042466 012100          ;; * MOV      #PNTR,-(SP)   ;; POINTER TO LOW WORD OF BINARY NUMBER
7815 042470 012101          ;; * JSR      PC,$SDB20    ;; CALL THE ROUTINE
7816 042472 005002          ;; * RETURN           ;; THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
7817 042474 110245          $SDB20: SAVREG
7818 042476 010002          MOV      2(SP),R1       ;; SAVE ALL REGISTERS
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
7901
7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000
8001
8002
8003
8004
8005
8006
8007
8008
8009
8010
8011
8012
8013
8014
8015
8016
8017
8018
8019
8020
8021
8022
8023
8024
8025
8026
8027
8028
8029
8030
8031
8032
8033
8034
8035
8036
8037
8038
8039
8040
8041
8042
8043
8044
8045
8046
8047
8048
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081
8082
8083
8084
8085
8086
8087
8088
8089
8090
8091
8092
8093
8094
8095
8096
8097
8098
8099
8100
8101
8102
8103
8104
8105
8106
8107
8108
8109
8110
8111
8112
8113
8114
8115
8116
8117
8118
8119
8120
8121
8122
8123
8124
8125
8126
8127
8128
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
8143
8144
8145
8146
8147
8148
8149
8150
8151
8152
8153
8154
8155
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169
8170
8171
8172
8173
8174
8175
8176
8177
8178
8179
8180
8181
8182
8183
8184
8185
8186
8187
8188
8189
8190
8191
8192
8193
8194
8195
8196
8197
8198
8199
8200
8201
8202
8203
8204
8205
8206
8207
8208
8209
8210
8211
8212
8213
8214
8215
8216
8217
8218
8219
8220
8221
8222
8223
8224
8225
8226
8227
8228
8229
8230
8231
8232
8233
8234
8235
8236
8237
8238
8239
8240
8241
8242
8243
8244
8245
8246
8247
8248
8249
8250
8251
8252
8253
8254
8255
8256
8257
8258
8259
8260
8261
8262
8263
8264
8265
8266
8267
8268
8269
8270
8271
8272
8273
8274
8275
8276
8277
8278
8279
8280
8281
8282
8283
8284
8285
8286
8287
8288
8289
8290
8291
8292
8293
8294
8295
8296
8297
8298
8299
8300
8301
8302
8303
8304
8305
8306
8307
8308
8309
8310
8311
8312
8313
8314
8315
8316
8317
8318
8319
8320
8321
8322
8323
8324
8325
8326
8327
8328
8329
8330
8331
8332
8333
8334
8335
8336
8337
8338
8339
8340
8341
8342
8343
8344
8345
8346
8347
8348
8349
8350
8351
8352
8353
8354
8355
8356
8357
8358
8359
8360
8361
8362
8363
8364
8365
8366
8367
8368
8369
8370
8371
8372
8373
8374
8375
8376
8377
8378
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399
8400
8401
8402
8403
8404
8405
8406
8407
8408
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8420
8421
8422
8423
8424
8425
8426
8427
8428
8429
8430
8431
8432
8433
8434
8435
8436
8437
8438
8439
8440
8441
8442
8443
8444
8445
8446
8447
8448
8449
8450
8451
8452
8453
8454
8455
8456
8457
8458
8459
8460
8461
8462
8463
8464
8465
8466
8467
8468
8469
8470
8471
8472
8473
8474
8475
8476
8477
8478
8479
8480
8481
8482
8483
8484
8485
8486
8487
8488
8489
8490
8491
8492
8493
8494
8495
8496
8497
8498
8499
8500
8501
8502
8503
8504
8505
8506
8507
8508
8509
8510
8511
8512
8513
8514
8515
8516
8517
8518
8519
8520
8521
8522
8523
8524
8525
8526
8527
8528
8529
8530
8531
8532
8533
8534
8535
8536
8537
8538
8539
8540
8541
8542
8543
8544
8545
8546
8547
8548
8549
8550
8551
8552
8553
8554
8555
8556
8557
8558
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569
8570
8571
8572
8573
8574
8575
8576
8577
8578
8579
8580
8581
8582
8583
8584
8585
8586
8587
8588
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8599
8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8610
8611
8612
8613
8614
8615
8616
8617
8618
8619
8620
8621
8622
8623
8624
8625
8626
8627
8628
8629
8630
8631
8632
8633
8634
8635
8636
8637
8638
8639
8640
8641
8642
8643
8644
8645
8646
8647
8648
8649
8650
8651
8652
8653
8654
8655
8656
8657
8658
8659
8660
8661
8662
8663
8664
8665
8666
8667
8668
8669
8670
8671
8672
8673
8674
8675
8676
8677
8678
8679
8680
8681
8682
8683
8684
8685
8686
8687
8688
8689
8690
8691
8692
8693
8694
8695
8696
8697
8698
8699
8700
8701
8702
8703
8704
8705
8706
8707
8708
8709
8710
8711
8712
8713
8714
8715
8716
8717
8718
8719
8720
8721
8722
8723
8724
8725
8726
8727
8728
8729
8730
8731
8732
8733
8734
8735
8736
8737
8738
8739
8740
8741
8742
8743
8744
8745
8746
8747
8748
8749
8750
8751
8752
8753
8754
8755
8756
8757
8758
8759
8760
8761
8762
8763
8764
8765
8766
8767
8768
8769
8770
8771
8772
8773
8774
8775
8776
8777
8778
8779
8780
8781
8782
8783
8784
8785
8786
8787
8788
8789
8790
8791
8792
8793
8794
8795
8796
8797
8798
8799
8800
8801
8802
8803
8804
8805
8806
8807
8808
8809
8810
8811
8812
8813
8814
8815
8816
8817
8818
8819
8820
8821
8822
8823
8824
8825
8826
8827
8828
8829
8830
8831
8832
8833
8834
8835
8836
8837
8838
8839
8840
8841
8842
8843
8844
8845
8846
8847
8848
8849
8850
8851
8852
8853
8854
8855
8856
8857
8858
8859
8860
8861
8862
8863
8864
8865
8866
8867
8868
8869
8870
8871
8872
8873
8874
8875
8876
8877
8878
8879
8880
8881
8882
8883
8884
8885
8886
8887
8888
8889
8890
8891
8892
8893
8894
8895
8896
8897
8898
8899
8900
8901
8902
8903
8904
8905
8906
8907
8908
8909
8910
8911
8912
8913
8914
8915
8916
8917
8918
8919
8920
8921
8922
8923
8924
8925
8926
8927
8928
8929
8930
8931
8932
8933
8934
8935
8936
8937
8938
8939
8940
8941
8942
8943
8944
8945
8946
8947
8948
8949
8950
8951
8952
8953
8954
8955
8956
8957
8958
8959
8960
8961
8962
8963
8964
8965
8966
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978
8979
8980
8981
8982
8983
8984
8985
8986
8987
8988
8989
8990
8991
8992
8993
8994
8995
8996
8997
8998
8999
9000

```

```

7819 042500 005304          DEC      R4          ;; COUNT THIS CHARACTER
7820 042502 003007          BGT     3$          ;; BR IF NOT THE LAST DIGIT
7821 042504 001405          BEQ     2$          ;; BR IF IT IS THE LAST DIGIT
7822 042506 005205          INC     R5          ;; ALL DIGITS DONE-ADJUST POINTER FOR FIRST
7823 042510 010566 000002    MOV     R5,2(SP)   ;; ASCII CHAR. & PUT IT ON THE STACK
7824 042514 104414          RESREG          ;; RESTORE ALL REGISTERS
7825 042516 000207          RTS     PC          ;; RETURN TO USER
7826 042520 006203          2$:    ASR     R3          ;; POSITION THE MASK FOR THE LAST DIGIT
7827 042522 006001          3$:    ROR     R1          ;; POSITION THE BINARY NUMBER FOR
7828 042524 006000          ROR     R0          ;; THE NEXT OCTAL DIGIT
7829 042526 006001          ROR     R1
7830 042530 006000          ROR     R0
7831 042532 006001          ROR     R1
7832 042534 006000          ROR     R0
7833 042536 040302          BIC     R3,R2      ;; MASK OUT ALL JUNK
7834 042540 062702 000060    ADD     #'0,R2     ;; MAKE THIS CHAR. ASCII
7835 042544 000753          BR      1$          ;; GO PUT IT IN THE DATA TABLE
7836 042546 000016          .S:    .BLKB 14     ;; RESERVE DATA TABLE
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850 042564 104413          $DB2D: SAVREG      ;; SAVE REGISTERS
7851 042566 016602 000002    MOV     2(SP),R2   ;; PICKUP THE DATA POINTER
7852 042572 012700 042744    MOV     #'$DECVL,R0 ;; GET ADDRESS OF "$DECVL" STRING
7853 042576 010066 000002    MOV     R0,2(SP)   ;; PUT ADDRESS OF ASCII STRING ON STACK
7854 042602 012201          MOV     (R2)+,R1   ;; PICKUP THE BINARY NUMBER
7855 042604 012202          MOV     (R2)+,R2
7856 042606 012737 000012 042662    MOV     #10,R3     ;; SET UP TO DO 10 CONVERSIONS
7857 042614 012704 042674    MOV     #'$TNPWR,R4 ;; ADDRESS OF TEN POWER
7858 042620 012705 042676    MOV     #'$TNPWR+2,R5
7859 042624 005003          1$:    CLR     R3          ;; CLEAR PARTIAL
7860 042626 161401          2$:    SUB     (R4),R1   ;; SUBTRACT TEN POWER
7861 042630 005602          SBC     R2
7862 042632 161502          SUB     (R5),R2
7863 042634 002402          BLT     3$          ;; BR IF TEN POWER TO LARGE
7864 042636 005203          INC     R3          ;; ADD 1 TO PARTIAL
7865 042640 000772          BR      2$          ;; LOOP
7866 042642 062401          3$:    ADD     (R4)+,R1   ;; RESTORE SUBTRACTED VALUE
7867 042644 005502          ADC     R2
7868 042646 062402          ADD     (R4)+,R2
7869 042650 022525          CMP     (R5)+,(R5)+ ;; MOVE TO NEXT TEN POWER
7870 042652 052703 000060    BIS     #'0,R3     ;; CHANGE PARTIAL TO ASCII
7871 042656 110320          MOVVB  R3,(R0)+   ;; SAVE IT
7872 042660 005327          DEC     (PC)+     ;; DONE?
7873 042662 000000          4$:    .WORD 0
7874 042664 001357          BNE     1$          ;; BR IF NO

```

```

*****
; THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
; DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
; POSITIVE.
; CALL
; * MOV #PNTR, -(SP) ;; POINTER TO LOW WORD OF BINARY NUMBER
; * JSR PC, #'$DB2D ;; THE FIRST ADDRESS OF ASCII
; * RETURN ;; IS ON THE STACK

```

```

7875 042666 105020          CLR      (RO)+          ;; TERMINATOR
7876 042670 104414          RESREG          ;; RESTORE REGISTERS
7877 042672 000207          RTS           PC          ;; RETURN
7878 042674 145000          STNPWR: 145000          ;; 1.0E09
7879 042676 035632          35632
7880 042700 160400          160400          ;; 1.0E08
7881 042702 002765          2765
7882 042704 113200          113200          ;; 1.0E07
7883 042706 000230          230
7884 042710 041100          041100          ;; 1.0E06
7885 042712 000017          17
7886 042714 103240          103240          ;; 1.0E05
7887 042716 000001          1
7888 042720 023420          23420          ;; 1.0E04
7889 042722 000000          0
7890 042724 001750          1750          ;; 1.0E03
7891 042726 000000          0
7892 042730 000144          144          ;; 1.0E02
7893 042732 000000          0
7894 042734 000012          12          ;; 1.0E01
7895 042736 000000          0
7896 042740 000001          1          ;; 1.0E00
7897 042742 000000          0
7898 042744 000014          SDECLV: .BLKB 12.          ;; RESERVE STORAGE FOR ASCII STRING
7899          .SBTTL SINGLE LENGTH BINARY TO DECIMAL ASCII ROUTINE
7900
7901          ;; *****
7902          ;; *THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
7903          ;; *UNSIGNED DECIMAL ASCII NUMBER.
7904          ;; *CALL
7905          ;; *   MOV      NUMBER, -(SP)          ;; PUT BINARY NUMBER ON THE STACK
7906          ;; *   JSR      PC, @#SSB2D          ;; CALL
7907          ;; *   RETURN          ;; ADDRESS OF THE 1ST ASCII CHAR. IS ON THE STACK
7908
7909
7910 042760 016637 000002 043010 SSB2D: MOV      2(SP), 1$          ;; SAVE BINARY NUMBER
7911 042766 012746 043010          MOV      #1$, -(SP)          ;; SET POINTER
7912 042772 004737 042564          JSR      PC, @#SSB2D          ;; CALL DOUBLE LENGTH CONVERT
7913 042776 062716 000005          ADD      #5, (SP)          ;; ONLY ALLOW FIVE CHARACTERS
7914 043002 012666 000002          MOV      (SP)+, 2(SP)          ;; PICKUP POINTER
7915 043006 000207          RTS           PC          ;; RETURN
7916 043010 000000 000000          1$: .WORD 0,0
7917          .SBTTL TYPE NUMERICAL ASCII STRING SUPPRESS LEADING ZEROS
7918
7919          ;; *****
7920          ;; *THIS ROUTINE IS USED TO TYPE AN ASCII NUMBER SUPPRESSING THE
7921          ;; *LEADING NUMBERS.
7922          ;; *CALL
7923          ;; *   MOV      #NUMADR, -(SP)          ;; FIRST ADDRESS OF ASCII STRING
7924          ;; *   JSR      PC, @#SSUPRS
7925
7926
7927 043014 010046          SSUPRS: MOV      RO, -(SP)          ;; SAVE RO
7928 043016 016600 000004          MOV      4(SP), RO          ;; PICKUP THE POINTER
7929 043022 105710          1$: TSTB      (RO)          ;; TERMINATOR?
7930 043024 001403          BEQ      2$          ;; BR IF YES
  
```

```

7931 043026 122720 000060
7932 043032 001773
7933 043034 005300
7934 043036 010037 043044
7935 043042 104401
7936 043044 000000
7937 043046 012600
7938 043050 012616
7939 043052 000207
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954 043054
7955 043054 010046
7956 043056 010146
7957 043060 010246
7958 043062 005046
7959 043064 016601 000012
7960 043070 100002
7961 043072 005216
7962 043074 005401
7963 043076 016602 000014
7964 043102 100002
7965 043104 005316
7966 043106 005402
7967 043110 012746 000021
7968 043114 005000
7969 043116 103001
7970 043120 060200
7971 043122 006000
7972 043124 006001
7973 043126 005316
7974 043130 001372
7975 043132 022616
7976 043134 001403
7977 043136 005400
7978 043140 005401
7979 043142 005600
7980 043144 005726
7981 043146 010066 000012
7982 043152 010166 000010
7983 043156 012602
7984 043160 012601
7985 043162 012600
7986 043164 000207

```

```

CMPB #'0,(RO)+ ;; IS THIS AN ASCII "0" ?
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
.SBTTL 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$: ROR RO ;; POSITION THE PARTIAL PRODUCT AND
ROR R1 ;; THE MULTIPLICAND
DEC (SP) ;; HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
BNE 3$ ;; BR IF NO
CMP (SP)+,(SP) ;; SHOULD PRODUCT BE NEGATIVE?
BEQ 5$ ;; GO TO EXIT IF NO
NEG RO ;; YES--SO MAKE IT SO
NEG 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

```



.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

```

```

7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000
8001
8002
8003
8004 043166
8005 043166 010046
8006 043170 010146
8007 043172 010246
8008 043174 010346
8009 043176 010446
8010 043200 010546
8011 043202 016646 000022
8012 043206 016646 000022
8013 043212 016646 000022
8014 043216 016646 000022
8015 043222 000002
8016
8017
8018
8019
8020 043224
8021 043224 012666 000022
8022 043230 012666 000022
8023 043234 012666 000022
8024 043240 012666 000022
8025 043244 012605
8026 043246 012604
8027 043250 012603
8028 043252 012602
8029 043254 012601
8030 043256 012600
8031 043260 000002
8032
8033
8034
8035
8036
8037
8038
8039
8040 043262 010046
8041 043264 016600 000002
8042 043270 005740

```

```

8043 043272 111000          MOVB  (R0),R0          ;;GET RIGHT BYTE OF TRAP
8044 043274 006300          ASL   R0              ;;POSITION FOR INDEXING
8045 043276 016000 043316  MOV   $TRPAD(R0),R0   ;;INDEX TO TABLE
8046 043302 000200          RTS   R0              ;;GO TO ROUTINE
8047
8048
8049
8050
8051 043304 011646          $TRAP2: MOV  (SP),-(SP) ;;MOVE THE PC DOWN
8052 043306 016666 000004 000002  MOV  4(SP),2(SP)      ;;MOVE THE PSW DOWN
8053 043314 000002          RTI                    ;;RESTORE THE PSW
8054
8055 .SBTTL  TRAP TABLE
8056
8057 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
8058 ;*BY THE "TRAP" INSTRUCTION.
8059
8060 ;
8061 ; ROUTINE
8062 ;-----
8062 043316 043304  $TRPAD: .WORD  $TRAP2
8063 043320 037632  $TYPE  ;;CALL=TYPE   TRAP+1(104401) TTY TYPEOUT ROUTINE
8064 043322 040632  $TYPOC ;;CALL=TYPOC  TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
8065 043324 040606  $TYPOS ;;CALL=TYPOS  TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
8066 043326 040646  $TYPON ;;CALL=TYPON   TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
8067 043330 040114  $TYPDS ;;CALL=TYPDS   TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
8068
8069 043332 041422  $GTSWR ;;CALL=GTSWR   TRAP+6(104406) GET SOFT-SWR SETTING
8070
8071 043334 041332  $CKSWR ;;CALL=CKSWR   TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
8072 043336 041674  $RDCHR ;;CALL=RDCHR   TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
8073 043340 041764  $RDLIN ;;CALL=RDLIN   TRAP+11(104411) TTY TYPEIN STRING ROUTINE
8074 043342 042304  $RDOCT ;;CALL=RDOCT   TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
8075 043344 043166  $$SAVREG ;;CALL=SAVREG  TRAP+13(104413) SAVE R0-R5 ROUTINE
8076 043346 043224  $RESREG ;;CALL=RESREG  TRAP+14(104414) RESTORE R0-R5 ROUTINE
8077 043350 036140  $SCOPI$ ;;CALL=SCOPI$  TRAP+15(104415) INTERNAL LOOP ON ERROR
8078
  
```

8079				
8080				
8081				
8082				
8083	043352	005015	047125	041111
8084	043360	051525	051040	030113
8085	043366	020066	051104	053111
8086	043374	020105	044504	043501
8087	043402	047516	052123	041511
8088	043410	005015	040515	047111
8089	043416	042504	026503	030461
8090	043424	042055	051132	044466
8091	043432	042055	050055	006502
8092	043440	012		
8093	043441	015	004412	025052
8094	043446	025052	020052	040503
8095	043454	052125	047511	020116
8096	043462	025052	025052	006452
8097	043470	012		
8098	043471	015	052012	044510
8099	043476	020123	051120	043517
8100	043504	040522	020115	044123
8101	043512	052517	042114	020040
8102	043520	042502	044040	046101
8103	043526	042524	020104	047117
8104	043534	054514	041040	020131
8105	043542	054524	044520	043516
8106	043550	041440	047117	051124
8107	043556	046117	041455	
8108	043562	005015	052117	042510
8109	043570	053522	051511	026105
8110	043576	041440	051101	051124
8111	043604	042111	042507	043040
8112	043612	051117	040515	052124
8113	043620	047111	020107	047101
8114	043626	026104	047440	020122
8115	043634	044124	020105	051104
8116	043642	053111	105	
8117	043645	015	046412	054501
8118	043652	041040	020105	042514
8119	043660	052106	044440	020116
8120	043666	047101	052440	042116
8121	043674	052105	051105	044515
8122	043702	042516	020104	052123
8123	043710	052101	105	
8124	043713	015	044412	044516
8125	043720	044524	046101	054514
8126	043726	020054	051104	053111
8127	043734	051505	052040	020117
8128	043742	042502	052040	051505
8129	043750	042524	020104	044123
8130	043756	052517	042114	044040
8131	043764	053101	035105	005015
8132	043772	005015	027101	044040
8133	044000	040505	051504	046440
8134	044006	047101	040525	046114

.SBTTL SERVICE MESSAGES

MSG1: .ASCII <CR><LF>/UNIBUS RK06 DRIVE DIAGNOSTIC/

.ASCII <CR><LF>/MAINDEC-11-DZR6I-D-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 DRIVE/

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

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

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

8135	044014	020131	047514	042101	
8136	044022	042105			
8137	044024	005015	027102	041440	.ASCII <CR><LF>/B. CORRECT PORT SELECTED/
8138	044032	051117	042522	052103	
8139	044040	050040	051117	020124	
8140	044046	042523	042514	052103	
8141	044054	042105			
8142	044056	005015	027103	053440	.ASCII <CR><LF>/C. WRITE LOCK DISABLED/
8143	044064	044522	042524	046040	
8144	044072	041517	020113	044504	
8145	044106	040523	046102	042105	
8146	044106	005015	027104	042040	.ASCII <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
8147	044114	044522	042526	051040	
8148	044122	040505	054504	044440	
8149	044130	042116	041511	052101	
8150	044136	051117	047440	006516	
8151	044144	012			
8152	044145	015	042012	044522	.ASCII <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
8153	044152	042526	020123	047516	
8154	044160	020124	047524	041040	
8155	044166	020105	042524	052123	
8156	044174	042105	046440	051525	
8157	044202	020124	040510	042526	
8158	044210	005015	047502	044124	.ASCIZ <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
8159	044216	050040	051117	051524	
8160	044224	042040	051505	046105	
8161	044232	041505	042524	006504	
8162	044240	000012			
8163					
8164	044242	005015	042502	051440	MSG2: .ASCIZ <CR><LF>/BE SURE TO PUT SCRATCH PACK IN DRIVE 0/
8165	044250	051125	020105	047524	
8166	044256	050040	052125	051440	
8167	044264	051103	052101	044103	
8168	044272	050040	041501	020113	
8169	044300	047111	042040	044522	
8170	044306	042526	030040	000	
8171	044313	015	042012	044522	MSG3: .ASCIZ <CR><LF>/DRIVE(S) TO BE TESTED: /
8172	044320	042526	051450	020051	
8173	044326	047524	041040	020105	
8174	044334	042524	052123	042105	
8175	044342	020072	000		
8176	044345	015	052012	050131	MSG4: .ASCIZ <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440: /
8177	044352	020105	052502	051523	
8178	044360	040440	042104	042522	
8179	044366	051523	044440	020106	
8180	044374	047516	020124	033461	
8181	044402	032067	030064	020072	
8182	044410	000040			
8183	044412	005015	054524	042520	MSG5: .ASCIZ <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210: /
8184	044420	041440	047117	051124	
8185	044426	046117	042514	020122	
8186	044434	047111	042524	051122	
8187	044442	050125	020124	042526	
8188	044450	052103	051117	044440	
8189	044456	020106	047516	020124	
8190	044464	030462	035060	020040	

8191	044472	000			
8192	044473	015	044412	052116	MSG6: .ASCIZ <CR><LF>/INTERRUPT OCCURRED AT PC=/ 
8193	044500	051105	052522	052120	
8194	044506	047440	041503	051125	
8195	044514	042522	020104	052101	
8196	044522	050040	036503	000	
8197	044527	015	042012	044522	MSG7: .ASCIZ <CR><LF>/DRIVE 0 WILL NOT BE TESTED/ 
8198	044534	042526	030040	053440	
8199	044542	046111	020114	047516	
8200	044550	020124	042502	052040	
8201	044556	051505	042524	000104	
8202	044564	005015	042522	042101	MSG8: .ASCIZ <CR><LF>/READ DATA WITH OFFSET TEST/<CR><LF> 
8203	044572	042040	052101	020101	
8204	044600	044527	044124	047440	
8205	044606	043106	042523	020124	
8206	044614	042524	052123	005015	
8207	044622	000			
8208	044623	015	044012	040505	MSG9: .ASCIZ <CR><LF>/HEAD NO./ 
8209	044630	020104	047516	000056	
8210	044636	005015	053412	046111	MSG10: .ASCIZ <CR><LF><LF>/WILL TEST DRIVES:/ 
8211	044644	020114	042524	052123	
8212	044652	042040	044522	042526	
8213	044660	035123	000		
8214	044663	015	005012	047520	MSG11: .ASCIZ <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF> 
8215	044670	042527	020122	050125	
8216	044676	051040	051505	040524	
8217	044704	052122	052040	020117	
8218	044712	042524	052123	030440	
8219	044720	005015	000		
8220	044723	015	005012	044124	MSG12: .ASCII <CR><LF><LF>/THE ABOVE OFFSET FAILURES ARE NOT ERRORS/ 
8221	044730	020105	041101	053117	
8222	044736	020105	043117	051506	
8223	044744	052105	043040	044501	
8224	044752	052514	042522	020123	
8225	044760	051101	020105	047516	
8226	044766	020124	051105	047522	
8227	044774	051522			
8228	044776	005015	052502	020114	.ASCIZ <CR><LF>/BUT INDICATORS OF SURFACE, HEAD, & ELECTRONICS QUALITY/<CR><LF> 
8229	045004	047111	044504	040503	
8230	045012	047524	051522	047440	
8231	045020	020106	052523	043122	
8232	045026	041501	026105	042510	
8233	045034	042101	020054	020046	
8234	045042	046105	041505	051124	
8235	045050	047117	041511	020123	
8236	045056	052521	046101	052111	
8237	045064	006531	000012		
8238	045070	005015	047516	046040	MSG13: .ASCII <CR><LF>/NO L OR P CLOCKS PRESENT/ 
8239	045076	047440	020122	020120	
8240	045104	046103	041517	051513	
8241	045112	050040	042522	042523	
8242	045120	052116			
8243	045122	005015	042510	042101	.ASCIZ <CR><LF>/HEAD SWITCHING TIME TEST BYPASSED/ 
8244	045130	051440	044527	041524	
8245	045136	044510	043516	052040	
8246	045144	046511	020105	042524	

8247	045152	052	041040	050131	
8248	045160	051	042523	000104	
8249	045166	005015	054502	040520	MSG14: .ASCIZ <CR><LF>/BYPASSING DRIVE /
8250	045174	051523	047111	020107	
8251	045202	051104	053111	020105	
8252	045210	000			
8253	045211	015	005012	051104	MSG15: .ASCIZ <CR><LF><LF>/DRIVE /
8254	045216	053111	020105	000	
8255	045223	015	042012	044522	MSG16: .ASCIZ <CR><LF>/DRIVE SERIAL NO. /
8256	045230	042526	051440	051105	
8257	045236	040511	020114	047516	
8258	045244	020056	000		
8259	045247	015	041412	051101	MSG17: .ASCIZ <CR><LF>/CARTRIDGE SERIAL NO. /
8260	045254	051124	042111	042507	
8261	045262	051440	051105	040511	
8262	045270	020114	047516	020056	
8263	045276	000			
8264	045277	015	005012	041101	MSG26: .ASCIZ <CR><LF><LF>/ABORTING BALANCE OF TESTS ON THIS DRIVE/<CR><LF><LF>
8265	045304	051117	044524	043516	
8266	045312	041040	046101	047101	
8267	045320	042503	047440	020106	
8268	045326	042524	052123	020123	
8269	045334	047117	052040	044510	
8270	045342	020123	051104	053111	
8271	045350	006505	005012	000	
8272	045355	015	005012	046101	MSG27: .ASCIZ <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
8273	045362	020114	051104	053111	
8274	045370	051505	052040	051505	
8275	045376	042524	006504	005012	
8276	045404	000			
8277	045405	015	047012	020117	MSG37: .ASCIZ <CR><LF>/NO WRITE CHECK ERROR AT MAX POSITIVE OFFSET/
8278	045412	051127	052111	020105	
8279	045420	044103	041505	020113	
8280	045426	051105	047522	020122	
8281	045434	052101	046440	054101	
8282	045442	050040	051517	052111	
8283	045450	053111	020105	043117	
8284	045456	051506	052105	000	
8285	045463	015	047012	020117	MSG38: .ASCIZ <CR><LF>/NO WRITE CHECK ERROR AT MAX NEGATIVE OFFSET/<CR><LF>
8286	045470	051127	052111	020105	
8287	045476	044103	041505	020113	
8288	045504	051105	047522	020122	
8289	045512	052101	046440	054101	
8290	045520	047040	043505	052101	
8291	045526	053111	020105	043117	
8292	045534	051506	052105	005015	
8293	045542	000			
8294	045543	015	053412	044522	MSG39: .ASCIZ <CR><LF>/WRITE CHECK FAILURE AT OFFSET =/
8295	045550	042524	041440	042510	
8296	045556	045503	043040	044501	
8297	045564	052514	042522	040440	
8298	045572	020124	043117	051506	
8299	045600	052105	036440	000	
8300	045605	015	041412	052517	MSG40: .ASCII <CR><LF>/COULD NOT READ BAD SECTOR INFO ON CYL 410/
8301	045612	042114	047040	052117	
8302	045620	051040	040505	020104	

8303	045626	040502	020104	042523	
8304	045634	052103	051117	044440	
8305	045642	043116	020117	047117	
8306	045650	041440	046131	032040	
8307	045656	030061			
8308	045660	005015	051117	040440	.ASCIZ <CR><LF>/OR ALIGNMENT CARTRIDGE USED/<CR><LF>
8309	045666	044514	047107	042515	
8310	045674	052116	041440	051101	
8311	045702	051124	042111	042507	
8312	045710	052440	042523	006504	
8313	045716	000012			
8314	045720	005015	051120	043517	MSG74: .ASCIZ <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/
8315	045726	040522	020115	041101	
8316	045734	051117	020124	042520	
8317	045742	042116	047111	027107	
8318	045750	027056	046120	040505	
8319	045756	042523	053440	044501	
8320	045764	000124			
8321	045766	005015	040510	052114	MSG75: .ASCIZ <CR><LF>/HALT PENDING...PLEASE WAIT/
8322	045774	050040	047105	044504	
8323	046002	043516	027056	050056	
8324	046010	042514	051501	020105	
8325	046016	040527	052111	000	
8326	046023	015	050012	047522	MSG76: .ASCIZ <CR><LF>/PROGRAM ABORTED/
8327	046030	051107	046501	040440	
8328	046036	047502	052122	042105	
8329	046044	000			
8330	046045	015	041412	052520	MSG77: .ASCIZ <CR><LF>/CPU HALTED/
8331	046052	044040	046101	042524	
8332	046060	000104			
8333					
8334					.SBTTL ERROR MESSAGES
8335					
8336	046062	005015	051105	047522	EM1: .ASCIZ <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>
8337	046070	026122	047440	046116	
8338	046076	020131	020060	044124	
8339	046104	052522	033440	040440	
8340	046112	046114	053517	042105	
8341	046120	020054	051124	020131	
8342	046126	043501	044501	006516	
8343	046134	000012			
8344	046136	051104	053111	020105	EM2: .ASCIZ /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKM2/
8345	046144	020043	047111	051040	
8346	046152	041513	031123	041440	
8347	046160	047101	047516	020124	
8348	046166	042502	051040	040505	
8349	046174	020104	040502	045503	
8350	046202	041440	051117	042522	
8351	046210	052103	054514	044440	
8352	046216	020116	045522	051115	
8353	046224	000062			
8354	046226	005015	041101	051117	EM3: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/
8355	046234	020124	042524	052123	
8356	046242	027123	027056	047125	
8357	046250	054105	042520	052103	
8358	046256	042105	052040	046511	

8359	046264	020105	052517	020124	
8360	046272	052101	050040	036503	
8361	046300	000			
8362	046301	015	040412	047502	EM4: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=/ 
8363	046306	052122	052040	051505	
8364	046314	051524	027056	052456	
8365	046322	042516	050130	041505	
8366	046330	042524	020104	047111	
8367	046336	042524	051122	050125	
8368	046344	020124	052101	050040	
8369	046352	036503	000		
8370	046355	115	051504	051440	EMS: .ASCIZ /MDS SET IN RKCS2/ 
8371	046362	052105	044440	020116	
8372	046370	045522	051503	000062	
8373	046376	043125	020105	042523	EM6: .ASCIZ /UFE SET IN RKCS2/ 
8374	046404	020124	047111	051040	
8375	046412	041513	031123	000	
8376	046417	104	040522	044440	EM7: .ASCIZ /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/ 
8377	046424	020116	045522	051504	
8378	046432	023040	047040	042105	
8379	046440	044440	020116	045522	
8380	046446	051503	027062	042522	
8381	046454	042523	035524	053440	
8382	046462	047522	043516	050040	
8383	046470	051117	020124	042523	
8384	046476	042514	052103	042105	
8385	046504	000077			
8386	046506	051104	053111	020105	EM8: .ASCIZ /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/ 
8387	046514	051120	051505	047105	
8388	046522	020124	052502	020124	
8389	046530	047516	020124	050123	
8390	046536	041505	043111	042511	
8391	046544	020104	054502	047440	
8392	046552	042520	040522	047524	
8393	046560	000122			
8394	046562	051104	053111	020105	EM9: .ASCIZ /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/ 
8395	046570	047516	020124	051120	
8396	046576	051505	047105	020124	
8397	046604	052502	020124	050123	
8398	046612	041505	043111	042511	
8399	046620	020104	054502	047440	
8400	046626	042520	040522	047524	
8401	046634	000122			
8402	046636	041101	051117	020124	EM10: .ASCIZ /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/ 
8403	046644	042524	052123	027123	
8404	046652	027056	040503	047116	
8405	046660	052117	051040	043105	
8406	046666	051105	047105	042503	
8407	046674	041440	047117	051124	
8408	046702	046117	042514	020122	
8409	046710	042522	044507	052123	
8410	046716	051105	000		
8411	046721	104	040522	044440	EM11: .ASCIZ /DRA IN RKDS & NED IN RKCS2 BOTH SET/ 
8412	046726	020116	045522	051504	
8413	046734	023040	047040	042105	
8414	046742	044440	020116	045522	



8415	046750	051503	020062	047502	
8416	046756	044124	051440	052105	
8417	046764	000			
8418	046765	103	047117	051124	EM12: .ASCIZ /CONTROLLER NOT READY IN RKCS1/
8419	046772	046117	042514	020122	
8420	047000	047516	020124	042522	
8421	047006	042101	020131	047111	
8422	047014	051040	041513	030523	
8423	047022	000			
8424	047023	116	020117	052101	EM13: .ASCIZ /NO ATTN IN RKASOF/
8425	047030	047124	044440	020116	
8426	047036	045522	051501	043117	
8427	047044	000			
8428	047045	125	042516	050130	EM14: .ASCIZ /UNEXPECTED MEMORY PARITY TRAP/
8429	047052	041505	042524	020104	
8430	047060	042515	047515	054522	
8431	047066	050040	051101	052111	
8432	047074	020131	051124	050101	
8433	047102	000			
8434	047103	122	042113	020103	EM15: .ASCII /RKDC & RKDA INDICATE THAT WCE OCCURRED AT/
8435	047110	020046	045522	040504	
8436	047116	044440	042116	041511	
8437	047124	052101	020105	044124	
8438	047132	052101	053440	042503	
8439	047140	047440	041503	051125	
8440	047146	042522	020104	052101	
8441	047154	005015	054503	020114	.ASCIZ <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
8442	047162	030464	026061	052040	
8443	047170	040522	045503	031040	
8444	047176	020054	042523	052103	
8445	047204	051117	031040	000061	
8446	047212	040503	047116	052117	EM16: .ASCIZ /CANNOT READ BAD SECTOR INFORMATION/
8447	047220	051040	040505	020104	
8448	047226	040502	020104	042523	
8449	047234	052103	051117	044440	
8450	047242	043116	051117	040515	
8451	047250	044524	047117	000	
8452	047255	115	051505	040523	EM17: .ASCIZ /MESSAGE A0 ERROR/
8453	047262	042507	040440	020060	
8454	047270	051105	047522	000122	
8455	047276	042515	051523	043501	EM18: .ASCIZ /MESSAGE B0 ERROR/
8456	047304	020105	030102	042440	
8457	047312	051122	051117	000	
8458	047317	115	051505	040523	EM19: .ASCIZ /MESSAGE A1 ERROR/
8459	047324	042507	040440	020061	
8460	047332	051105	047522	000122	
8461	047340	042515	051523	043501	EM20: .ASCIZ /MESSAGE B1 ERROR/
8462	047346	020105	030502	042440	
8463	047354	051122	051117	000	
8464	047361	103	051105	020122	EM21: .ASCIZ /CERR SET IN RKCS1/
8465	047366	042523	020124	047111	
8466	047374	051040	041513	030523	
8467	047402	000			
8468	047403	116	020117	051104	EM22: .ASCII /NO DRIVES FOUND IN DEVICE MAP (\$DEVN)/<CR><LF>
8469	047410	053111	051505	043040	
8470	047416	052517	042116	044440	

8471	047424	020116	042504	044526
8472	047432	042503	046440	050101
8473	047440	024040	042044	053105
8474	047446	024515	005015	
8475	047452	042523	052524	020120
8476	047460	047503	051122	041505
8477	047466	046124	020131	047101
8478	047474	020104	042522	052123
8479	047502	051101	006524	000012
8480	047510	047516	042040	044522
8481	047516	042526	020123	047506
8482	047524	047125	020104	047117
8483	047532	041040	051525	006523
8484	047540	012		
8485	047541	123	052105	050125
8486	047546	041440	051117	042522
8487	047554	052103	054514	040440
8488	047562	042116	050040	042522
8489	047570	051523	023440	047503
8490	047576	052116	047111	042525
8491	047604	006447	000012	
8492	047610	047526	020114	040526
8493	047616	044514	020104	047516
8494	047624	020124	042523	020124
8495	047632	047111	051040	046513
8496	047640	031122	000	
8497	047643	015	042012	052105
8498	047650	041505	042524	020104
8499	047656	030061	041040	042101
8500	047664	051440	041505	047524
8501	047672	051522	027056	040456
8502	047700	047502	052122	047111
8503	047706	020107	042524	052123
8504	047714	000		
8505	047715	104	052105	041505
8506	047722	042524	020104	051502
8507	047730	020105	052502	020124
8508	047736	047516	020124	044514
8509	047744	052123	042105	044440
8510	047752	020116	040502	020104
8511	047760	042523	052103	051117
8512	047766	043040	046111	000105
8513	047774	042504	042524	052103
8514	050002	042105	041040	042523
8515	050010	044440	020116	042522
8516	050016	042101	041440	046517
8517	050024	040515	042116	
8518	050030	005015	052502	020124
8519	050036	047516	020124	047111
8520	050044	050040	042522	044526
8521	050052	052517	020123	051127
8522	050060	052111	020105	047503
8523	050066	046515	047101	020104
8524	050074	047524	051440	046501
8525	050102	020105	042523	052103
8526	050110	051117	000	

.ASCIZ /SETUP CORRECTLY AND RESTART/<CR><LF>

EM23: .ASCII /NO DRIVES FOUND ON BUSS/<CR><LF>

.ASCIZ /SETUP CORRECTLY AND PRESS 'CONTINUE'/<CR><LF>

EM24: .ASCIZ /VOL VALID NOT SET IN RKMR2/

EM25: .ASCIZ <CR><LF>/DETECTED 10 BAD SECTORS...ABORTING TEST/

EM26: .ASCIZ /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/

EM27: .ASCII /DETECTED BSE IN READ COMMAND/

.ASCIZ <CR><LF>/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/

8527	050113	103	046131	040440	FM36:	.ASCIZ /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
8528	050120	042104	020122	047111		
8529	050126	051040	046513	031522		
8530	050134	047040	052117	051440		
8531	050142	046501	020105	051501		
8532	050150	051040	042113	000103		
8533	050156	054503	020114	044504	EM39:	.ASCIZ /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
8534	050164	043106	023040	047440		
8535	050172	043106	042523	020124		
8536	050200	047111	051040	046513		
8537	050206	031122	047040	052117		
8538	050214	041440	042514	051101		
8539	050222	042105	000			
8540	050225	103	046131	040440	EM40:	.ASCIZ /CYL ADDR IN RKMR3 NOT CLEARED/
8541	050232	042104	020122	047.11		
8542	050240	051040	046513	031522		
8543	050246	047040	052117	041440		
8544	050254	042514	051101	042105		
8545	050262	000				
8546	050263	103	046131	040440	EM41:	.ASCIZ /CYL ADDR IN B2 DID NOT REMAIN CLEARED/
8547	050270	042104	020122	047111		
8548	050276	041040	020062	044504		
8549	050304	020104	047516	020124		
8550	050312	042522	040515	047111		
8551	050320	041440	042514	051101		
8552	050326	042105	000			
8553	050331	101	052124	020116	EM55:	.ASCIZ /ATTN NOT CLEARED IN RKASOF/
8554	050336	047516	020124	046103		
8555	050344	040505	042522	020104		
8556	050352	047111	051040	040513		
8557	050360	047523	000106			
8558	050364	046104	020124	042523	EM63:	.ASCIZ /DLT SET IN RKCS2/
8559	050372	020124	047111	051040		
8560	050400	041513	031123	000		
8561	050405	122	040505	020104	EM65:	.ASCIZ /READ HEADER ERROR/
8562	050412	042510	042101	051105		
8563	050420	042440	051122	051117		
8564	050426	000				
8565	050427	101	044514	047107	EM69:	.ASCIZ /ALIGNMENT CARTRIDGE USED/
8566	050434	042515	052116	041440		
8567	050442	051101	051124	042111		
8568	050450	042507	052440	042523		
8569	050456	000104				
8570	050460	052103	020117	042523	EM73:	.ASCIZ /CTO SET IN RKCS1/
8571	050466	020124	047111	051040		
8572	050474	041513	030523	000		
8573	050501	122	055124	047040	EM74:	.ASCIZ /RTZ NOT SET IN RKMR2/
8574	050506	052117	051440	052105		
8575	050514	044440	020116	045522		
8576	050522	051115	000062			
8577	050526	042516	020104	042523	EM79:	.ASCIZ /NED SET IN RKCS2/
8578	050534	020124	047111	051040		
8579	050542	041513	031123	000		
8580	050547	127	044522	042524	EM80:	.ASCIZ /WRITE CHECK ERROR SET IN RKCS2/
8581	050554	041440	042510	045503		
8582	050562	042440	051122	051117		

8583	050570	051440	052105	044440	
8584	050576	020116	045522	051503	
8585	050604	000062			
8586	050606	051127	052111	020105	EM81: .ASCIZ /WRITE CHECK COMMAND NOT FUNCTIONING/
8587	050614	044103	041505	020113	
8588	050622	047503	046515	047101	
8589	050630	020104	047516	020124	
8590	050636	052506	041516	044524	
8591	050644	047117	047111	000107	
8592	050652	042522	042101	042040	EM82: .ASCIZ /READ DATA DID NOT COMPARE WITH WRITE DATA/
8593	050660	052101	020101	044504	
8594	050666	020104	047516	020124	
8595	050674	047503	050115	051101	
8596	050702	020105	044527	044124	
8597	050710	053440	044522	042524	
8598	050716	042040	052101	000101	
8599	050724	040504	040524	041440	EM83: .ASCIZ /DATA CHECK ERROR SET IN RKER/
8600	050732	042510	045503	042440	
8601	050740	051122	051117	051440	
8602	050746	052105	044440	020116	
8603	050754	045522	051105	000	
8604	050761	127	044510	042514	EM84: .ASCIZ /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
8605	050766	053440	044501	044524	
8606	050774	043516	043040	051117	
8607	051002	041440	047117	051124	
8608	051010	051040	040505	054504	
8609	051016	047440	020122	043101	
8610	051024	042524	020122	047503	
8611	051032	052116	020122	042522	
8612	051040	042101	020131	042522	
8613	051046	023503	000104		
8614	051052	043117	051506	052105	EM85: .ASCIZ /OFFSET STATUS BIT IN RKMR2 CLEARED/
8615	051060	051440	040524	052524	
8616	051066	020123	044502	020124	
8617	051074	047111	051040	046513	
8618	051102	031122	041440	042514	
8619	051110	051101	042105	000	
8620	051115	117	043106	042523	EM86: .ASCIZ /OFFSET REG IN A2 NOT = RKASOF/
8621	051122	020124	042522	020107	
8622	051130	047111	040440	020062	
8623	051136	047516	020124	020075	
8624	051144	045522	051501	043117	
8625	051152	000			
8626	051153	104	042111	047040	EM88: .ASCIZ /DID * FIND SECTOR 0 FROM INDEX/
8627	051160	052117	043040	047111	
8628	051166	020104	042523	052103	
8629	051174	051117	030040	043040	
8630	051202	047522	020115	047111	
8631	051210	042504	000130		
8632	051214	042522	042101	047111	EM93: .ASCIZ /READING WRONG CYLINDER # IN HEADER...MISPOSITION/
8633	051222	020107	051127	047117	
8634	051230	020107	054503	044514	
8635	051236	042116	051105	021440	
8636	051244	044440	020116	042510	
8637	051252	042101	051105	027056	
8638	051260	046456	051511	047520	

8639	051266	044523	044524	047117	
8640	051274	000			
8641	051275	117	043106	042523	EM94: .ASCIZ /OFFSET IT IN A2 NOT CLEARED/
8642	051302	020124	052111	044440	
8643	051310	020116	031101	047040	
8644	051316	052117	041440	042514	
8645	051324	051101	042105	000	
8646	051331	106	051117	040515	EM95: .ASCIZ /FORMAT BIT NOT SET IN RKMR2/
8647	051336	020124	044502	020124	
8648	051344	047516	020124	042523	
8649	051352	020124	047111	051040	
8650	051360	046513	031122	000	
8651	051365	103	047101	047516	EM96: .ASCIZ /CANNOT FIND SECTOR 23(B)/
8652	051372	020124	044506	042116	
8653	051400	051440	041505	047524	
8654	051406	020122	031462	034050	
8655	051414	000051			
8656	051416	042510	042101	051440	EM97: .ASCIZ /HEAD SWITCHING REQ'D ANOTHER FULL REVOLUTION OF DISK/
8657	051424	044527	041524	044510	
8658	051432	043515	051040	050505	
8659	051440	042047	040440	047516	
8660	051446	044124	051105	043040	
8661	051454	046125	020114	042522	
8662	051462	047526	052514	044524	
8663	051470	047117	047440	020106	
8664	051476	044504	045523	000	
8665	051503	104	044522	042526	EM100: .ASCIZ /DRIVE OFF TRACK SET IN RKMR3/
8666	051510	047440	043106	052040	
8667	051516	040522	045507	051440	
8668	051524	052105	044440	020116	
8669	051532	045522	051115	000063	
8670	051540	044504	020104	047516	EM101: .ASCIZ /DID NOT GO TO CYLINDER 10/
8671	051546	020124	047507	052040	
8672	051554	020117	054503	044514	
8673	051562	042116	051105	030440	
8674	051570	000060			
8675					
8676					.SBTTL DATA HEADERS
8677					
8678	051572	042524	052123	047040	DH1: .ASCIZ /TEST NO. PC/
8679	051600	027117	020040	041520	
8680	051606	000			
8681	051607	122	046513	030522	DH2: .ASCIZ /RKMR1 RKMR2 RKMR3 RKER RKDS RKCS1 RKCS2/
8682	051614	051011	046513	031122	
8683	051622	051011	046513	031522	
8684	051630	051011	042513	004522	
8685	051636	045522	051504	051011	
8686	051644	041513	030523	051011	
8687	051652	041513	031123	000	
8688	051657	122	053513	004503	DH3: .ASCIZ /RKWC RKBA RKDA RKASOF RKDC RKECPS RKECPT/
8689	051664	045522	040502	051011	
8690	051672	042113	004501	045522	
8691	051700	051501	043117	051011	
8692	051706	042113	004503	045522	
8693	051714	041505	051520	051011	
8694	051722	042513	050103	000124	

# H13

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR61D.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 162  
DATA HEADERS

SEQ 0162

8695	051730	051106	046517	041440	DH6:	.ASCIZ /FROM CYL TO CYL CYL DIFF/
8696	051736	046131	020040	047524		
8697	051744	041440	046131	020040		
8698	051752	054503	020114	044504		
8699	051760	043106	000			
8700	051763	124	051505	020124	DH8:	.ASCIZ /TEST NO. TRAP PC/
8701	051770	047516	004456	051124		
8702	051776	050101	050040	000103		
8703	052004	043101	042524	020122	DH9:	.ASCIZ /AFTER START SPINDLE COMMAND REC'D BY DRIVE/
8704	052012	052123	051101	020124		
8705	052020	050123	047111	046104		
8706	052026	020105	047503	046515		
8707	052034	047101	020104	042522		
8708	052042	023503	020104	054502		
8709	052050	042040	044522	042526		
8710	052056	000				
8711	052057	101	020124	047105	DH10:	.ASCIZ /AT END OF HEAD LOADING/
8712	052064	020104	043117	044040		
8713	052072	040505	020104	047514		
8714	052100	042101	047111	000107		
8715	052106	054105	042520	052103	DH11:	.ASCIZ /EXPECTED WAS/
8716	052114	042105	053411	051501		
8717	052122	000				
8718	052123	117	020116	042523	DH13:	.ASCIZ /ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/
8719	052130	052103	051117	020123		
8720	052136	030061	020054	031061		
8721	052144	020054	032061	020054		
8722	052152	033061	020054	034061		
8723	052160	047440	020122	030062		
8724	052166	041440	046131	032040		
8725	052174	030061	052040	040522		
8726	052202	045503	031040	000		
8727	052207	117	020116	042523	DH14:	.ASCIZ /ON SECTORS 11, 13, 15, 17, 19 OR 21 CYL 410 TRACK 2/
8728	052214	052103	051117	020123		
8729	052222	030461	020054	031461		
8730	052230	020054	032461	020054		
8731	052236	033461	020054	034461		
8732	052244	047440	020122	030462		
8733	052252	041440	046131	032040		
8734	052260	030061	052040	040522		
8735	052266	045503	031040	000		
8736	052273	101	052106	051105	DH17:	.ASCIZ /AFTER RECAL COMMAND/
8737	052300	051040	041505	046101		
8738	052306	041440	046517	040515		
8739	052314	042116	000			
8740	052317	101	052106	051105	DH19:	.ASCIZ /AFTER PACK COMMAND/
8741	052324	050040	041501	020113		
8742	052332	047503	046515	047101		
8743	052340	000104				
8744	052342	043101	042524	020122	DH20:	.ASCIZ /AFTER SELECT DRIVE COMMAND/
8745	052350	042523	042514	052103		
8746	052356	042040	044522	042526		
8747	052364	041440	046517	040515		
8748	052372	042116	000			
8749	052375	101	052106	051105	DH21:	.ASCIZ /AFTER SUBSYSTEM CLEAR/
8750	052402	051440	041125	054523		



8807	053050	042116	000		
8808	053053	117	020116	042523	DH42: .ASCIZ /ON SECTORS 0,2,4,6 OR 8 CYL 410 TRACK 2/
8809	053060	052103	051117	020123	
8810	053066	026060	026062	026064	
8811	053074	020066	051117	034040	
8812	053102	020040	054503	020114	
8813	053110	030464	020060	051124	
8814	053116	041501	020113	000062	
8815	053124	047117	051440	041505	DH43: .ASCIZ /ON SECTORS 1,3,5,7 OR 9 CYL 410 TRACK 2/
8816	053132	047524	051522	030440	
8817	053140	031454	032454	033454	
8818	053146	047440	020122	020071	
8819	053154	041440	046131	032040	
8820	053162	030061	052040	040522	
8821	053170	045503	031040	000	
8822	053175	106	051117	040515	DH44: .ASCIZ /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
8823	053202	020124	020046	046101	
8824	053210	020114	042522	042101	
8825	053216	053455	044522	042524	
8826	053224	052040	051505	051524	
8827	053232	053440	046111	020114	
8828	053240	042502	041040	050131	
8829	053246	051501	042523	000104	
8830	053254	043101	042524	020122	DH47: .ASCIZ /AFTER READ HEADER COMMAND WITH MOVEMENT/
8831	053262	042522	042101	044040	
8832	053270	040505	042504	020122	
8833	053276	047503	046515	047101	
8834	053304	020104	044527	044124	
8835	053312	046440	053117	046505	
8836	053320	047105	000124		
8837	053324	051515	020107	020101	DH49: .ASCIZ /MSG A & B IN RKMR2 & RKMR3 RESP. ARE INVALID/
8838	053332	020046	020102	047111	
8839	053340	051040	046513	031122	
8840	053346	023040	051040	046513	
8841	053354	031522	051040	051505	
8842	053362	027120	040440	042522	
8843	053370	044440	053116	046101	
8844	053376	042111	000		
8845	053401	101	052106	051105	DH51: .ASCIZ /AFTER SEEK TO SELF COMMAND/
8846	053406	051440	042505	020113	
8847	053414	047524	051440	046105	
8848	053422	020106	047503	046515	
8849	053430	047101	000104		
8850	053434	044527	044124	044440	DH52: .ASCIZ /WITH INTENTIONAL MISCOMPARE/
8851	053442	052116	047105	044524	
8852	053450	047117	046101	046440	
8853	053456	051511	047503	050115	
8854	053464	051101	000105		
8855	053470	052504	044522	043516	DH53: .ASCIZ /DURING OFFSET COMMAND/
8856	053476	047440	043106	042523	
8857	053504	020124	047503	046515	
8858	053512	047101	000104		
8859	053516	043101	042524	020122	DH54: .ASCIZ /AFTER FORMAT CHANGE AND CONTR READY REC'D/
8860	053524	047506	046522	052101	
8861	053532	041440	040510	043516	
8862	053540	020105	047101	020104	



8863	053546	047503	052116	020122	
8864	053554	042522	042101	020131	
8865	053562	042522	023503	000104	
8866	053570	054503	020114	004443	DH56: .ASCIZ /CYL # HEADER WORD 0/
8867	053576	042510	042101	051105	
8868	053604	053440	051117	020104	
8869	053612	000060			
8870	053614	043101	042524	020122	DH57: .ASCIZ /AFTER WRITE COMMAND WITH OFFSET/
8871	053622	051127	052111	020105	
8872	053630	047503	046515	047101	
8873	053636	020104	044527	044124	
8874	053644	047440	043106	042523	
8875	053652	000124			
8876					
8877					.SBTTL ERROR OUTPUT DATA
8878					
8879					
8880	053654	001214	001116		DT1: .EVEN
8881	053660	007364	007366	007370	\$TESTN,\$ERRPC
8882	053666	007354	007352	007340	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8883	053674	007342			
8884	053676	007344	007346	007350	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8885	053704	007356	007360	007372	
8886	053712	007374			
8887	053714	001214	001334		DT3: \$TESTN,TRAPPC
8888	053720	001214	001116	001344	DT4: \$TESTN,\$ERRPC,FRCYL,TOCYL,CALDIF
8889	053726	001346	001354		
8890	053732	007364	007366	007370	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8891	053740	007354	007352	007340	
8892	053746	007342			
8893	053750	007344	007346	007350	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8894	053756	007356	007360	007372	
8895	053764	007374			
8896	053766	001214	001116	001440	DT6: \$TESTN,\$ERRPC,WD2,WD1
8897	053774	001436			
8898	053776	007364	007366	007370	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8899	054004	007354	007352	007340	
8900	054012	007342			
8901	054014	007344	007346	007350	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8902	054022	007356	007360	007372	
8903	054030	007374			
8904	054032	001214	001116	001456	DT7: \$TESTN,\$ERRPC,WDcnt,HDWD,TEMP1
8905	054040	001474	007376		
8906	054044	007364	007366	007370	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8907	054052	007354	007352	007340	
8908	054060	007342			
8909	054062	007344	007346	007350	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8910	054070	007356	007360	007372	
8911	054076	007374			
8912	054100	001214	001116	001346	DT8: \$TESTN,\$ERRPC,TOCYL,FRCYL,CALDIF
8913	054106	001344	001354		
8914	054112	007364	007366	007370	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8915	054120	007354	007352	007340	
8916	054126	007342			
8917	054130	007344	007346	007350	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8918	054136	007356	007360	007372	

8919	054144	007374				
8920	054146	001214	001116	001346	DT9:	\$TESTN,\$ERRPC,TOCYL,RHTAB
8921	054154	001712				
8922	054156	007364	007366	007370		HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8923	054164	007354	007352	007340		
8924	054172	007342				
8925	054174	007344	007346	007350		HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8926	054202	007356	007360	007372		
8927	054210	007374				
8928	054212	001214	001116	007430	DT13:	\$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,H.A0,H.B0,H.A1,H.B1
8929	054220	007432	007434	007436		
8930	054226	007410	007412	007414		
8931	054234	007416				
8932	054236	007364	007366	007370		HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8933	054244	007354	007352	007340		
8934	054252	007342				
8935	054254	007344	007346	007350		HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8936	054262	007356	007360	007372		
8937	054270	007374				
8938	054272	001214	001116	007430	DT14:	\$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2
8939	054300	007432	007434	007436		
8940	054306	007440	007442			
8941	054312	007410	007412	007414		H.A0,H.B0,H.A1,H.B1,H.A2,H.B2
8942	054320	007416	007420	007422		
8943	054326	007364	007366	007370		HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8944	054334	007354	007352	007340		
8945	054342	007342				
8946	054344	007344	007346	007350		HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8947	054352	007356	007360	007372		
8948	054360	007374				
8949	054362	001214	001116	007430	DT15:	\$TESTN,\$ERRPC,E.A0,E.B0,E.A1,E.B1,E.A2,E.B2,E.B3
8950	054370	007432	007434	007436		
8951	054376	007440	007442	007446		
8952	054404	007410	007412	007414		H.A0,H.B0,H.A1,H.B1,H.A2,H.B2,H.B3
8953	054412	007416	007420	007422		
8954	054420	007426				
8955	054422	007364	007366	007370		HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8956	054430	007354	007352	007340		
8957	054436	007342				
8958	054440	007344	007346	007350		HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8959	054446	007356	007360	007372		
8960	054454	007374				
8961						
8962					.SBTTL	ERROR DATA FORMATS
8963						
8964	054456	000003			DF1:	3
8965	054460	002	000			.BYTE 2,0
8966	054462	051607				DH2
8967	054464	007	000			.BYTE 7,0
8968	054466	051657				DH3
8969	054470	007	000			.BYTE 7,0
8970						
8971	054472	000001			DF2:	1
8972	054474	002	000			.BYTE 2,0
8973						
8974	054476	000005			DF3:	5

8975	054500	000	000	.BYTE	0,0
8976	054502	051572		DH1	
8977	054504	002	000	.BYTE	2,0
8978	054506	052106		DH11	
8979	054510	002	000	.BYTE	2,0
8980	054512	051607		DH2	
8981	054514	007	000	.BYTE	7,0
8982	054516	051657		DH3	
8983	054520	007	000	.BYTE	7,0
8984					
8985	054522	000003		DF4:	3
8986	054524	002	000	.BYTE	2,0
8987	054526	051607		DH2	
8988	054530	007	000	.BYTE	7,0
8989	054532	051657		DH3	
8990	054534	007	000	.BYTE	7,0
8991					
8992	054536	000005		DF5:	5
8993	054540	000	000	.BYTE	0,0
8994	054542	053324		DH49	
8995	054544	000	000	.BYTE	0,0
8996	054546	051572		DH1	
8997	054550	002	000	.BYTE	2,0
8998	054552	051607		DH2	
8999	054554	007	000	.BYTE	7,0
9000	054556	051657		DH3	
9001	054560	007	000	.BYTE	7,0
9002					
9003	054562	000005		DF6:	5
9004	054564	000	000	.BYTE	0,0
9005	054566	051572		DH1	
9006	054570	002	000	.BYTE	2,0
9007	054572	051730		DH6	
9008	054574	003	000	.BYTE	3,0
9009	054576	051607		DH2	
9010	054600	007	000	.BYTE	7,0
9011	054602	051657		DH3	
9012	054604	007	000	.BYTE	7,0
9013					
9014					
9015	054606	000004		DF10:	4
9016	054610	000	000	.BYTE	0,0
9017	054612	051572		DH1	
9018	054614	002	000	.BYTE	2,0
9019	054616	051607		DH2	
9020	054620	007	000	.BYTE	7,0
9021	054622	051657		DH3	
9022	054624	007	000	.BYTE	7,0
9023					
9024	054626	000004		DF14:	4
9025	054630	002	000	.BYTE	2,0
9026	054632	052772		DH40	
9027	054634	003	000	.BYTE	3,0
9028	054636	051607		DH2	
9029	054640	007	000	.BYTE	7,0
9030	054642	051657		DH3	

9031	054644	007	000		.BYTE	7,0
9032						
9033						
9034	054646	000004		DF15:	4	
9035	054650	000	000		.BYTE	0,0
9036	054652	051572			DH1	
9037	054654	002	000		.BYTE	2,0
9038	054656	051607			DH2	
9039	054660	007	000		.BYTE	7,0
9040	054662	051657			DH3	
9041	054664	007	000		.BYTE	7,0
9042						
9043	054666	000005		DF17:	5	
9044	054670	000	000		.BYTE	0,0
9045	054672	053175			DH44	
9046	054674	000	000		.BYTE	0,0
9047	054676	051572			DH1	
9048	054700	002	000		.BYTE	2,0
9049	054702	051607			DH2	
9050	054704	007	000		.BYTE	7,0
9051	054706	051657			DH3	
9052	054710	007	000		.BYTE	7,0
9053	054712	000005		DF20:	5	
9054	054714	000	000		.BYTE	0,0
9055	054716	051572			DH1	
9056	054720	002	000		.BYTE	2,0
9057	054722	053570			DH56	
9058	054724	002	000		.BYTE	2,0
9059	054726	051607			DH2	
9060	054730	007	000		.BYTE	7,0
9061	054732	051657			DH3	
9062	054734	007	000		.BYTE	7,0
9063						
9064	054736	000007		DF21:	7	
9065	054740	000	000		.BYTE	0,0
9066	054742	051572			DH1	
9067	054744	002	000		.BYTE	2,0
9068	054746	052606			DH28	
9069	054750	000	000		.BYTE	0,0
9070	054752	052660			DH31	
9071	054754	004	000		.BYTE	4,0
9072	054756	052616			DH29	
9073	054760	004	000		.BYTE	4,0
9074	054762	051607			DH2	
9075	054764	002	000		.BYTE	7,0
9076	054766	051657			DH3	
9077	054770	007	000		.BYTE	7,0
9078						
9079	054772	000007		DF22:	7	
9080	054774	000	000		.BYTE	0,0
9081	054776	051572			DH1	
9082	055000	002	000		.BYTE	2,0
9083	055002	052606			DH28	
9084	055004	000	000		.BYTE	0,0
9085	055006	052660			DH31	
9086	055010	006	000		.BYTE	6,0

9087	055012	052616	
9088	055014	006	000
9089	055016	051607	
9090	055020	007	000
9091	055022	051657	
9092	055024	007	000
9093			
9094	055026	000007	
9095	055030	000	000
9096	055032	051572	
9097	055034	002	000
9098	055036	052606	
9099	055040	000	000
9100	055042	052660	
9101	055044	007	000
9102	055046	052616	
9103	055050	007	000
9104	055052	051607	
9105	055054	007	000
9106	055056	051657	
9107	055060	007	000
9108			
9109			
9110			
9111			
9112			
9113			
9114			
9115			
9116			
9117			
9118			
9119	055062	104413	
9120	055064	113700	001114
9121	055070	042700	177400
9122	055074	005300	
9123	055076	006300	
9124	055100	006300	
9125	055102	006300	
9126	055104	062700	007512
9127	055110	012037	055124
9128	055114	001404	
9129	055116	104401	001205
9130	055122	104401	
9131	055124	000000	
9132	055126	012037	055142
9133	055132	001404	
9134	055134	104401	001205
9135	055140	104401	
9136	055142	000000	
9137	055144	012001	
9138	055146	001455	
9139	055150	005004	
9140	055152	012000	
9141	055154	012002	
9142	055156	001446	

	DH29	
	.BYTE	6,0
	DH2	
	.BYTE	7,0
	DH3	
	.BYTE	7,0
DF23:	7	
	.BYTE	0,0
	DH1	
	.BYTE	2,0
	DH28	
	.BYTE	0,0
	DH31	
	.BYTE	7,0
	DH29	
	.BYTE	7,0
	DH2	
	.BYTE	7,0
	DH3	
	.BYTE	7,0

```

.EVEN
:*****
:SBTTL TYPE ERROR ROUTINE
:*ENTRY JSR PC,TYP ERR
:*RETURN RTS PC
:*
:*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
:*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
:*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
:*THE ERROR.
:*****
TYPERR: SAVREG
      MOVB $ITEMB,RO ;ENTER ERROR NUMBER
      BIC #177400,RO ;CLEAR SIGN EXTENSION
      DEC RO ;FORM INDEX FOR ERROR TABLE
      ASL RO
      ASL RO
      ASL RO
15:   ADD #ERRTB,RO ;FORM ADDRESS OF ERROR ENTRY
      MOV (RO)+,2$ ;GET EM POINTER
      BEQ 3$ ;BRANCH IF THERE ISN'T ONE
      TYPE ,SCLF ;TYPE CARRIAGE RETURN LINE FEED
      TYPE ;TYPE ERROR MESSAGE (EM)
25:   .WORD 0 ;EM POINTER GOES HERE
35:   MOV (RO)+,4$ ;GET DH POINTER
      BEQ 5$ ;BRANCH IF THERE ISN'T ONE
      TYPE ,SCLF ;TYPE CR-LF
      TYPE ;TYPE DATA HEADER
45:   .WORD 0 ;DH POINTER GOES HERE
55:   MOV (RO)+,R1 ;GET DT POINTER
      BEQ 20$ ;BRANCH IF THERE ARE NONE
      CLR R4 ;SET INDENT SWITCH
      MOV (RO)+,RO ;GET DF POINTER
      MOV (RO)+,R2 ;STORE NUMBER OF DH'S
      BEQ 17$ ;DH NUM IS 0-BRANCH

```

```

9143 055160 005104          COM      R4          ;NO INDENT
9144 055162 104401 001205    TYPE     SCRLF
9145 055166 112003          10$:  MOVB   (R0)+,R3    ;GET & STORE NUMBER OF DATA WORDS
9146 055170 105720          TSTB   (R0)+        ;BUMP PAST FORMAT WORD
9147 055172 005703          TST    R3           ;TEST IF ANY DATA FOR THIS HEADER
9148 055174 001407          BEQ    14$          ;NO - SKIP DATA PRINT
9149 055176 013146          11$:  MOV    2(R1)+,-(SP) ;PUT FIRST DATA WORD ON STACK
9150 055200 104402          TYPOC
9151 055202 005303          DEC    R3           ;MORE DATA WORDS
9152 055204 001403          BEQ    14$          ;NO-BRANCH
9153 055206 104401 055336    TYPE     SPACE2     ;TYPE SEPARATORS
9154 055212 000771          BR     11$          ;LOOP
9155 055214 005302          14$:  DEC    R2           ;MORE DH'S?
9156 055216 003431          BLE    20$          ;NO-BRANCH
9157 055220 104401 001205    TYPE     SCRLF
9158 055224 005760 000002    TST    2(R0)        ;ONLY A DH IN THIS REQUEST?
9159 055230 001404          BEQ    15$          ;YES-BRANCH BYPASS INDENT
9160 055232 005104          COM      R4          ;INDENT?
9161 055234 001002          BNE    15$          ;NO-BRANCH
9162 055236 104401 055336    TYPE     SPACE2     ;YES-TYPE SPACES
9163 055242 012037 055250    15$:  MOV    (R0)+,16$   ;GET NEXT DH POINTER
9164 055246 104401          TYPE     DH         ;TYPE DH
9165 055250 000000          .WORD   0           ;DH POINTER GOES HERE
9166 055252 105710          TSTB   (R0)         ;TYPE A DT?
9167 055254 001003          BNE    21$          ;YES-BRANCH
9168 055256 062700 000002    ADD    #2,R0        ;INCREMENT OF POINTER
9169 055262 000754          BR     14$          ;SEE IF END OF DF BLOCK
9170 055264 104401 001205    21$:  TYPE     SCRLF
9171 055270 005704          TST    R4           ;INDENT?
9172 055272 001335          BNE    10$          ;NO-BRANCH
9173 055274 104401 055336    17$:  TYPE     SPACE2     ;YES-TYPE SPACES
9174 055300 000732          BR     10$          ;LOOP
9175 055302 104414          20$:  RESREG
9176
9177 055304 032777 010000 123626  BIT     #SW12,2SWR   ;SEE IF ABORT DRV AFTER 20 ERRORS
9178 055312 001410          BEQ    25$          ;BR IF NO
9179 055314 023727 001103 000024  CMP    SERFLG,#20.  ;ELSE SEE IF HAVE 20 ERRORS
9180 055322 001004          BNE    25$          ;BR IF NO
9181 055324 012706 001100          MOV    #STACK,SP   ;ELSE RESTORE STACK PTR
9182 055330 000137 031514          JMP    $EOP        ;AND GO TO NEXT DRV
9183
9184 055334 000207          25$:  RTS     PC
9185 055336 020040 000          SPACE2: .ASCIZ/ /   ;2 SPACES
; ODT-11 -- VOOSA
; DEC-11-UODPA-A-LA
; COPYRIGHT 1969,1970,1972
; DIGITAL EQUIPMENT CORPORATION
; MAYNARD, MASSACHUSETTS 01754
9193          .ENABL ABS,AMA
9194          .EVEN
9195          .=. +60
9196          R0      =      %0      ; REGISTER
9197          R1      =      %1      ; NAMING
9198          R2      =      %2      ; CONVENTIONS

```

```

9199      000003      R3      =      %3
9200      000004      R4      =      %4
9201      000005      R5      =      %5
9202      000006      SP      =      %6
9203      000007      PC      =      %7
9204      177776      ST      =      177776      ;STATUS REGISTER
9205
9206      000014      O.TVEC =      14      ;TRT VECTOR LOCATION
9207      000340      O.STM  =      340     ;PRIORITY MASK - STATUS REGISTER
9208      000020      O.TBT  =      20     ;T-BIT MASK - STATUS REGISTER
9209      000003      TRT    =      000003 ;TRT INSTRUCTION
9210      000006      RTT    =      000006 ;RTT INSTRUCTION
9211
9212      ; RS IS USUALLY CONSIDERED SAFE, THE CURRENT ADDRESS WORD
9213      ; RESIDES IN IT. AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
9214      ; OPERATIONS LEAVE IT RANDOMLY FILLED. OTHERWISE, IT SHOULD NOT
9215      ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAD).
9216
9217      177562      O.RDB  =      177562 ;R DATA BUFFER
9218      177560      O.RCSR =      177560 ;R C/SR
9219      177566      O.TDB  =      177566 ;T DATA BUFFER
9220      177564      O.TCSR =      177564 ;T C/SR
9221
9222      ;
9223      ; INITIALIZE OOT
9224      ; USE O.OOT FOR A NORMAL ENTRY
9225      ; USE O.OOT+2 TO RESTART OOT - WIPING OUT ALL BREAKPOINTS
9226      ; USE O.OOT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
9227
9228      055422      000413      O.OOT: BR      O.STRT      ;NORMAL ENTRY
9229      055424      000417      BR      O.RST        ;RESTART
9230      055426      013737      177776      055402      O.ENTR: MOV     ST,O.UST    ;RE-ENTER -- SAVE STATUS
9231      055434      013737      000016      177776      MOV     O.TVEC+2,ST      ;SET UP LOCAL STATUS
Z 9232      055442      010737      055400      MOV     PC,O.UPC        ;FAKE THE PC
9233      055446      000137      056600      JMP     O.BK1
9234
9235      055452      012706      055362      O.STRT: MOV     #O.URD,SP  ;SET UP STACK
9236      055456      010637      055376      MOV     SP,O.USP        ;FAKE THE SAVED STACK
9237      055462      000414      BR      O.RST1         ;CLEAR BREAKPOINT TABLES
9238      055464      004037      057006      O.RST:  JSR     O,O.SVR   ;SAVE REGISTERS
9239      055470      013777      055420      177716      MOV     O.UIN,O.ADR1    ;REMOVE THE BREAKPOINT
9240      055476      113704      055404      MOV     O.PRI,R4        ;GET OOT PRIORITY
9241      055502      106004      RORB    R4              ;SHIFT
9242      055504      106004      RORB    R4              ; INTO
9243      055506      106004      RORB    R4              ; POSITION
9244      055510      110437      177776      MOV     R4,ST           ;STORE IN STATUS
Z 9245      055514      000127      O.RST1: JMP     (PC)+
9246      055516      000403      BR      O.45
9247      055520      012737      000002      056510      MOV     #RTI,O.RTIT     ;SET TO RTI IF 11/20 OR /05
9248      055526      105037      057427      O.45:  CLRB   O.P           ;DISALLOW PROCEED
9249      055532      012737      000340      000016      MOV     #O.STM,O.TVEC+2 ;STATUS WORD TO TRT VECTOR + 2
9250      055540      012737      056570      000014      MOV     #O.BRK,O.TVEC   ;PC TO TRT VECTOR
9251      055546      000447      BR      O.RALL         ;CLEAR BREAKPOINT TABLES
9252
9253      ; SPECIAL NAME HANDLER
9254      ; DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES O.TL AND O.URD

```

```

9255
9256 055550 004537 057230
9257 055554 012704 057453
9258 055560 120024
9259 055562 001413
9260 055564 022704 057461
9261 055570 101373
9262 055572 042700 177770
9263 055576 010004
9264 055600 006304
9265 055602 062704 055362
9266 055606 005202
9267 055610 000444
9268 055612 162704 057444
9269 055616 000770
9270
9271
9272
9273 055620 004737 057354
9274 055624 010502
9275 055626 061202
9276 055630 006202
9277 055632 103421
9278 055634 006302
9279 055636 005722
9280 055640 010205
9281 055642 000137 056114
9282
9283
9284
9285 055646 005702
9286 055650 001406
9287 055652 006204
9288 055654 103410
9289 055656 006304
9290 055660 010437 055414
9291 055664 000412
9292 055666 012737 057470 055414
9293 055674 000406
9294
9295
9296
9297
9298
9299
9300 055676 052705 000001
9301 055702 012700 000077
9302 055706 004537 057306
9303 055712 004537 057406
9304 055716 005004
9305 055720 005002
9306 055722 004537 057230
9307 055726 022700 000060
9308 055732 101013
9309 055734 022700 000067
9310 055740 103410

```

```

;
O.REGT: JSR 5,O.GET ;SPECIAL NAME, GET ONE MORE CHARACTER
MOV #0,TL,R4 ;TABLE START ADDRESS
O.RSP: CMPB R0,(R4)+ ;IS THIS THE CORRECT CHARACTER?
BEQ 0,SP ;JUMP IF YES
CMP #0,TL+0.LG,R4 ;IS THE SEARCH DONE?
BHI 0,RSP ;BRANCH IF NOT
BIC #177770,R0 ;MASK OFF OCTAL
MOV R0,R4
O.SP1: ASL R4
ADD #0,URO,R4 ;GENERATE ADDRESS
INC R2 ;SET FOUND FLAG
BR 0,SCAN ;GO FIND NEXT CHARACTER
O.SP: SUB #0,TL-7,R4 ;CORRECT CONSTANT
BR 0,SP1
;
; * HANDLER - OPEN INDEXED ON THE PC
;
O.ORPC: JSR PC,O.TCLS
MOV R5,R2 ;CURRENT ADDRESS IN R2
ADD #R2,R2 ;COMPUTE
ASR R2 ;MOVE ONE BIT TO CARRY
BCS 0,ERR ;ERROR IF ODD NUMBER
ASL R2 ;RESTORE WORD
TST (R2)+ ;AND INCREMENT BY TWO
MOV R2,R5 ;UPDATE CAD
JMP 0,OP2 ;GO FINISH UP
;
; B HANDLER - SET AND REMOVE BREAKPOINTS
;
O.BKPT: TST R2 ;IF NO NUMBER TYPED
BEQ 0,RALL ;REMOVE BREAKPOINT
ASR R4 ;CHECK IF ODD
BCS 0,ERR ;JUMP IF ODD
ASL R4 ;RESTORE ONE BIT
MOV R4,O.ADR1 ;SET A BREAKPOINT
BR 0,DCD
O.RALL: MOV #0,TRTC,O.ADR1 ;CLEAR BREAKPOINT
BR 0,DCD
;
; COMMAND DECODER - ODT11
;
; REGISTERS R0-R4 MAY BE USED,
; REGISTER R5 WILL BE CONSIDERED SAFE
;
O.ERR: BIS #1,R5 ;CLOSE EVERYTHING
MOV #',R0 ;' TO BE TYPED
JSR 5,O.FTYP ;OUTPUT ?
O.DCD: JSR 5,O.CRLS ;TYPE <CR><LF>*
O.DCD1: CLR R4 ;R4 CONTAINS THE CONVERTED OCTAL
CLR R2 ;R2 IS THE NUMBER FOUND FLAG
O.SCAN: JSR 5,O.GET ;GET A CHAR, RETURN IN R0
CMP #0,R0 ;COMPARE WITH ASCII 0
BHI 0,CLGL ;CHECK LEGALITY IF NON-NUMERIC
CMP #7,R0 ;COMPARE WITH ASCII 7
BLO 0,CLGL ;CHECK LEGALITY IF NOT OCTAL

```



```

9311 055742 042700 177770          BIC      #177770,R0      ; CONVERT TO BCD
9312 055746 006304          ASL      R4            ; MAKE ROOM
9313 055750 006304          ASL      R4            ; IN
9314 055752 006304          ASL      R4            ; R4
9315 055754 060004          ADD      R0,R4        ; PACK THREE BITS IN R4
9316 055756 005202          INC      R2            ; R2 HAS NUMERIC FLAG
9317 055760 000760          BR       0.SCAN       ; AND TRY AGAIN
9318 055762 005001          0.CLGL: CLR      R1      ; CLEAR INDEX
9319 055764 120061 057437          0.LGL1: CMPB     R0,0.LGCH(R1) ; DO THE CODES MATCH?
9320 055770 001405          BEQ      0.LGL2       ; JUMP IF YES
9321 055772 005201          INC      R1            ; SET INDEX FOR NEXT SEARCH
9322 055774 020127 000014          CMP      R1,#0.CLGT   ; IS THE SEARCH DONE?
9323 056000 103336          BHS      0.ERR        ; OOPS!
9324 056002 000770          BR       0.LGL1       ; RE-LOOP
9325 056004 006301          0.LGL2: ASL      R1      ; MULTIPLY BY TWO
9326 056006 000171 056012          JMP      @0.LGDR(R1)  ; GO TO PROPER ROUTINE
9327
9328 056012 056042          0.LGDR: 0.WRD      ; / OPEN WORD
9329 056014 056074          0.CRET  ; CARRIAGE RETURN CLOSE
9330 056016 055550          0.REGT  ; $ REGISTER OPS
9331 056020 056404          0.GO    ; G GO TO ADDRESS K
9332 056022 056106          0.OP1   ; <LF> MODIFY, CLOSE, OPEN NEXT
9333 056024 055620          0.ORPC  ; + OPEN RELATED, INDEX - PC
9334 056026 056140          0.BACK  ; ↑ OPEN PREVIOUS
9335 056030 056150          0.OFST  ; 0 OFFSET
9336 056032 056226          0.WSCH  ; W SEARCH WORD
9337 056034 056222          0.EFF   ; E SEARCH EFFECTIVE ADDRESS
9338 056036 055646          0.BKPT  ; B BREAKPOINTS
9339 056040 056512          0.PROC  ; P PROCEED
9340
9341          0.LGL   =      -0.LGDR      ; LGL MUST EQUAL 2X CHLGT ALWAYS
9342
9343          ; PROCESS / - OPEN WORD
9344
9344 056042 005702          0.WRD:  TST      R2      ; GET VALUE IF R2 IS NON-ZERO
9345 056044 001410          BEQ      0.WRDA       ; SKIP OTHERWISE
9346 056046 010405          MOV      R4,R5        ; PUT VALUE IN CAD
9347 056050 006205          0.WRD1: ASR      R5      ; MOVE ONE BIT TO CARRY
9348 056052 103711          0.ERR2: BCS      0.ERR   ; JUMP IF ODD ADDRESS
9349 056054 006305          ASL      R5            ; RESTORE THE CARRY BIT
9350 056056 011500          MOV      @R5,R0       ; GET CONTENTS OF WORD
9351 056060 004537 057144          JSR      5,0.CADV     ; GO GET AND TYPE OUT @CAD
9352 056064 000714          BR       0.DCD1       ; GO BACK TO DECODER
9353 056066 042705 000001          0.WRDA: BIC      #1,R5   ; CLEAR CLOSED BIT
9354 056072 000766          BR       0.WRD1       ; GO BACK TO MAIN-LINE
9355
9356          ; PROCESS CARRIAGE RETURN
9357
9358 056074 004737 057354          0.CRET: JSR      PC,0.TCLS ; CLOSE LOCATION
9359 056100 052705 000001          BIS      #1,R5        ; CLOSE EVERYTHING
9360 056104 000702          BR       0.DCD        ; RETURN TO DECODER
9361
9362          ; PROCESS <LF>, OPEN NEXT WORD
9363
9364 056106 004737 057354          0.OP1:  JSR      PC,0.TCLS ; CLOSE PRESENT CELL
9365 056112 005725          TST      (R5)+        ; GENERATE NEW ADDRESS
9366 056114 004537 057400          0.OP2:  JSR      5,0.CRLF ; <CR><LF>

```

9367	056120	010500		MOV	R5,RO		;NUMBER TO TYPE
9368	056122	004537	057144	JSR	S,0.CADV		;TYPE OUT ADDRESS
9369	056126	012700	000057	MOV	#',RO		;TYPE A /
9370	056132	004537	057306	JSR	S,0.FTYP		
9371	056136	000744		BR	0.WRD1		;GO PROCESS IT
9372							
9373							
9374							
9375	056140	004737	057354	0.BACK:	JSR	PC,0.TCLS	
9376	056144	005745			TST	-(R5)	;GENERATE NEW ADDRESS
9377	056146	000762			BR	0.OP2	;GO DO THE REST
9378							
9379							
9380							
9381	056150	006205		0.OFST:	ASR	R5	;GET LOW ORDER BIT
9382	056152	103737			BCS	0.ERR2	;ERROR IF CLOSED
9383	056154	006305			ASL	R5	;RESTORE WORD
9384	056156	012700	000040		MOV	#',RO	;TYPE ONE BLANK
9385	056162	004537	057306		JSR	S,0.FTYP	;AS A SEPARATOR
9386	056166	160504			SUB	R5,R4	;COMPUTE
9387	056170	005304			DEC	R4	
9388	056172	005304			DEC	R4	;16 BIT OFFSET
9389	056174	010400			MOV	R4,RO	;TYPE A
9390	056176	010402			MOV	R4,R2	;SAVE R4
9391	056200	004537	057144		JSR	S,0.CADV	;NUMBER IN RO - WORD MODE
9392	056204	010200			MOV	R2,RO	
9393	056206	006200			ASR	RO	;DIVIDE BY TWO
9394	056210	103402			BCS	0.OF1	;BRANCH IF ODD
9395	056212	004537	057144		JSR	S,0.CADV	;NUMBER IN RO - BYTE MODE
9396	056216	000137	055716	0.OF1:	JMP	0.DCD1	;ALL DONE
9397							
9398							
9399							
9400							

SEARCHES - \$MSK HAS THE MASK  
\$MSK+2 HAS THE FWA  
\$MSK+4 HAS THE LWA

9401				0.EFF:	INC	R1		;SET EFFECTIVE SEARCH
9402	056222	005201			BR	0.WDS		
9403	056224	000401		0.WSCH:	CLR	R1		;SET WORD SEARCH
9404	056226	005001		0.WDS:	TST	R2		;CHECK FOR OBJECT FOUND
9405	056230	005702		0.ERR1:	BEQ	0.ERR		;ERROR IF NO OBJECT
9406	056232	001621			MOV	0.MSK+2,R2		;SET ORIGIN
9407	056234	013702	055410		MOV	0.MSK,R5		;SET MASK
9408	056240	013705	055406		COM	R5		;AND COMPLEMENT IT
9409	056244	005105		0.WDS2:	CMP	R2,0.MSK+4		; IS THE SEARCH ALL DONE?
9410	056246	020237	055412		BHI	0.DCD		; YES
9411	056252	101217			MOV	R2,R0		; GET OBJECT
9412	056254	011200			TST	R1		;NO
9413	056256	005701			BNE	0.EFF1		;BRANCH IF EFFECTIVE SEARCH
9414	056260	001027			MOV	R0,-(SP)		
9415	056262	010046			MOV	R4,R3		;EXCLUSIVE OR
9416	056264	010403			BIC	R4,R0		; IS DONE
9417	056266	040400			BIC	(SP)+,R3		; IN A VERY
9418	056270	042603			BIS	R0,R3		; FANCY MANNER HERE
9419	056272	050003			BIC	R5,R3		; AND RESULT WITH MASK
9420	056274	040503		0.WDS3:	BNE	0.WDS4		;RE-LOOP IF NO MATCH
9421	056276	001016			MOV	R4,-(SP)		;REGISTERS R2,R4, AND R5 ARE SAFE
9422	056300	010446			JSR	5,0.CRLF		;TYPE (CR,LF)
9423	056302	004537	057400		MOV	R2,R0		;GET READY TO TYPE
9424	056306	010700			JSR	5,0.CADV		;TYPE ADDRESS
9425	056310	004537	057144		MOV	#1/,R0		;SLASH TO R0
9426	056314	012700	000057		JSR	5,0.FTYP		;TYPE IT
9427	056320	004537	057306		MOV	R2,R0		;GET CONTENTS
9428	056324	011200			JSR	5,0.CADV		;TYPE CONTENTS
9429	056326	004537	057144		MOV	(SP)+,R4		;RESTORE R4
9430	056332	012604		0.WDS4:	TST	(R2)+		;INCREMENT TO NEXT CELL AND
9431	056334	005722			BR	0.WDS2		;RETURN
9432	056336	000743		0.EFF1:	CMP	R0,R4		; IS (X)=K?
9433	056340	020004			BEQ	0.WDS3		;TYPE IF EQUAL
9434	056342	001755			MOV	R0,R3		; (X) TO R3
9435	056344	010003			ADD	R2,R3		; (X)+X
9436	056346	060203			INC	R3		; (X)+X+2
9437	056350	005203			INC	R3		; IS (X)+X+2=K?
9438	056352	005203			CMP	R3,R4		;BRANCH IF EQUAL
9439	056354	020304			BEQ	0.WDS3		;WIPE OUT EXTRANEIOUS BITS
9440	056356	001747			BIC	#177400,R0		;EXTEND SIGN
9441	056360	042700	177400		MOVB	R0,R0		
9442	056364	110000			CCC			
9443	056366	000257			ASL	R0		;MULTIPLY BY TWO
9444	056370	006300			INC	R0		;ADD TWO
9445	056372	005200			INC	R0		
9446	056374	005200			ADD	R2,R0		;ADD PC
9447	056376	060200			CMP	R0,R4		; IS THE RESULT A PROPER REL. BRANCH?
9448	056400	020004			BR	0.WDS3		
9449	056402	000735						
9450								
9451								
9452								
9453	056404	105037	057427		0.GO:	CLRB	0.P	;DISALLOW PROCEED
9454	056410	006204				ASR	R4	;CHECK LOW ORDER BIT
9455	056412	103617				BCS	0.ERR2	;ERROR IF ODD NUMBER
9456	056414	006304				ASL	R4	;RESTORE WORD

PROCESS G - GO

9457	056416	010437	055400		MOV	R4,0.UPC	:SET UP NEW PC
9458	056422	112737	000340	177776	MOV	#0.STM,ST	:SET HIGH PRIORITY
9459	056430	004537	057076		JSR	5,0.RST	:RESTORE TELETYPE
9460	056434	105037	057426		0.TBIT: CLR	0.T	:CLEAR BOTH
9461	056440	042737	000020	055402	BIC	#0.TBT,0.UST	:T-BIT FLAGS
9462	056446	017737	176742	055420	MOV	20.ADR1,0.UIN	:SAVE INSTRUCTION
9463	056454	013777	057470	176732	MOV	0.TRTC,20.ADR1	:REPLACE WITH TRAP
9464	056462	012600			0.G02: MOV	(SP)+,R0	:RESTORE
9465	056464	012601			MOV	(SP)+,R1	:R0
9466	056466	012602			MOV	(SP)+,R2	:THRU
9467	056470	012603			MOV	(SP)+,R3	
9468	056472	012604			MOV	(SP)+,R4	
9469	056474	012605			MOV	(SP)+,R5	:R5
9470	056476	012606			MOV	(SP)+,SP	:AND SP
9471	056500	013746	055402		MOV	0.UST,-(SP)	:AND STATUS
9472	056504	013746	055400		MOV	0.UPC,-(SP)	:AND PC
9473	056510	000006			0.RTIT: RTT		:CHANGED TO RTI FOR 11/20 AND /05
9474							
9475							
9476							
9477							
9478	056512	105737	057427		0.PROC: TSTB	0.P	:CHECK LEGALITY OF PROCEED
9479	056516	001645			BEQ	0.ERR1	:NOT LEGAL
9480	056520	105037	057427		CLR	0.P	:CLEAR PROCEED FLAG
9481	056524	005702			TST	R2	:WAS COUNT SPECIFIED?
9482	056526	001402			BEQ	0.PRI	:NO
9483	056530	010437	055416		MOV	R4,0.CT	:YES, PUT AWAY COUNT
9484	056534	112737	000340	177776	0.PRI: MOV	#0.STM,ST	:FORCE HIGH PRIORITY
9485	056542	004537	057076		JSR	5,0.RST	:RESTORE TTY
9486	056546	112737	000340	177776	0.C1: MOV	#0.STM,ST	:SET HIGH PRIORITY
9487	056554	105237	057426		INCB	0.T	:SET T-BIT FLAG
9488	056560	052737	000020	055402	BIS	#0.TBT,0.UST	:SET T-BIT
9489	056566	000735			BR	0.G02	
9490							
9491							
9492							
9493							
9494							
9495							
9496	056570	012637	055400		0.BRK: MOV	(SP)+,0.UPC	:PRIORITY IS 7 UPON ENTRY
9497	056574	012637	055402		MOV	(SP)+,0.UST	:SAVE STATUS AND PC
9498	056600	004037	057006		0.BK1: JSR	0,0.SVR	:SAVE VARIOUS REGISTERS
9499	056604	105737	057426		TSTB	0.T	:CHECK FOR T-BIT SET
9500	056610	001311			BNE	0.TBIT	:JUMP IF SET
9501	056612	013777	055420	176574	MOV	0.UIN,20.ADR1	:REMOVE BREAKPOINTS
9502	056620	105737	055404		TSTB	0.PRI	:CHECK IF PRIORITY
9503	056624	100003			BPL	0.BK2	:IS AS SAME AS USER PGM
9504	056626	113705	055402		MOV	0.UST,R5	:PICK UP USER UST IF SO
9505	056632	000407			BR	0.BK3	:AND DON'T COMPUTE THE PRIORITY
9506	056634	113705	055404		0.BK2: MOV	0.PRI,R5	:OTHERWISE PICK UP ACTUAL PRIORITY
9507	056640	000257			CCC		:CLEAR CARRY
9508	056642	106005			RORB	R5	:SHIFT LOW ORDER BITS
9509	056644	106005			RORB	R5	:INTO
9510	056646	106005			RORB	R5	:HIGH ORDER
9511	056650	106005			RORB	R5	:POSITION
9512	056652	110537	177776		0.BK3: MOV	R5,ST	:PUT THE STATUS AWAY WHERE IT BELONGS

```

9513 056656 013705 055400      MOV      0.UPC,R5      ;GET PC, IT POINTS TO THE TRT
9514 056662 005745              TST      -(R5)        ;SUBTRACT TWO
9515 056664 010537 055400      MOV      R5,0.UPC     ;FROM THE USER'S PC
9516 056670 020537 055414      CMP      R5,0.ADR1    ;COMPARE WITH LIST
9517 056674 001417              BEQ      0.B2         ;JUMP IF FOUND
9518 056676 004537 057044      JSR      5,0.SVTT     ;SAVE TELETYPE STATUS
9519 056702 004537 057400      JSR      5,0.CRLF     ;
9520 056706 012704 057432      MOV      #0,BD,R4     ;ERROR, NOTHING FOUND
9521 056712 012703 057433      MOV      #0,BD+1,R3
9522 056716 004537 057272      JSR      5,0.TYPE     ;OUTPUT "BE" FOR BAD ENTRY
9523 056722 010500              MOV      R5,RO
9524 056724 042737 000020 055402      BIC      #0,TBT,0.UST ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
9525 056732 000420              BR       0.B3         ; AND CONTINUE
9526 056734 005337 055416      0.B2:   DEC      0.CT
9527 056740 003302              BGT      0.C1         ; JUMP IF REPEAT
9528 056742 012737 000001 055416      MOV      #1,0.CT     ; RESET COUNT TO 1
9529 056750 105237 057427      INCB    0.P           ; ALLOW PROCEED
9530 056754 004537 057044      JSR      5,0.SVTT     ;SAVE TELETYPE STATUS, R4 IS SAFE
9531 056760 012700 000102      MOV      #1,B,RO
9532 056764 004537 057306      JSR      5,0.FTYP     ;TYPE "B"
9533 056770 013700 055414      MOV      0.ADR1,RO   ;GET ADDRESS OF BREAK
9534 056774 004537 057144      0.B3:   JSR      5,0.CADV    ;TYPE ADDRESS
9535 057000 005005              CLR      R5          ;CLEAR CAD
9536 057002 000137 055712      JMP      0.DCD       ;GO TO DECODER
9537
9538 ;
9539 ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
9540
9540 057006 012637 057424      0.SVR:  MOV      (SP)+0,XXX ;PICK REGISTER FROM STACK AND SAVE
9541 057012 010637 055376      MOV      SP,0.USR    ;SAVE USER STACK ADDRESS
9542 057016 012706 055376      MOV      #0,USP,SP   ;SET TO INTERNAL STACK
9543 057022 010546              MOV      R5,-(SP)    ;SAVE
9544 057024 010446              MOV      R4,-(SP)    ;REGISTERS
9545 057026 010346              MOV      R3,-(SP)    ;1
9546 057030 010246              MOV      R2,-(SP)    ;THRU
9547 057032 010146              MOV      R1,-(SP)    ;5
9548 057034 013746 057424      MOV      0.XXX,-(SP) ;PUT SAVED REGISTER ON STACK
9549 057040 005746              TST      -(SP)
9550 057042 000200              RTS      RO
9551
9552 ;
9553 ; SAVE TELETYPE STATUS
9554
9554 057044 113737 177560 057430      0.SVTT: MOVB    0.RCSR,0.CSR1 ;SAVE R C/SR
9555 057052 113737 177564 057431      MOVB    0.TCSR,0.CSR2 ;SAVE T C/SR
9556 057060 105037 177560      CLRB    0.RCSR       ;CLEAR ENABLE AND MAINTENANCE
9557 057064 105037 177564      CLRB    0.TCSR       ;BITS IN BOTH C/SR
9558 057070 004537 057400      JSR      5,0.CRLF     ;TYPE <CR,LF>
9559 057074 000205      RTS      R5
9560
9561 ;
9562 ; RESTORE TELETYPE STATUS
9563
9563 057076 004537 057400      0.RSTT: JSR      5,0.CRLF ;<CR,LF> BEFORE RESTORING
9564 057102 105737 177564      TSTB    0.TCSR       ;WAIT READY ON PRINTER
9565 057106 100375              BPL     -4
9566 057110 032737 004000 177560      BIT     #4000,0.RCSR  ;CHECK BUSY FLAG ON READER
9567 057116 001403              BEQ     0.RSE1       ;SKIP READY LOOP IF NOT BUSY
9568 057120 105737 177560      TSTB    0.RCSR       ;WAIT READY

```

```

9569 057124 100375
9570 057126 113737 057430 177560 0.RSE1: BPL -4 ; ON READER
9571 057134 113737 057431 177564 MOVB 0.CSR1,0.RCSR ; RESTORE
9572 057142 000205 MOVB 0.CSR2,0.TCSR ; THE STATUS REGISTERS
9573 RTS R5
9574 ;
9575 ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
9576 ; WORD IS IN RO
9577 057144 010246 0.CADV: MO R2,-(SP) ; SAVE R2
9578 057146 012704 057467 MOV #0.BUF+6,R4 ; BUFFER START ADDRESS
9579 057152 012746 000060 MOV #0,-(SP) ; CONSTANT ASCII 0
9580 057156 010002 0.SPC: MOV RO,R2 ; GET
9581 057160 042702 177770 BIC #177770,R2 ; OCTAL CHARACTER
9582 057164 061602 ADD #SP,R2 ; CONVERT TO ASCII
9583 057166 110244 MOVB R2,-(R4) ; STORE IN BUFFER
9584 057170 006200 ASR RO ; SHIFT THIS MESS
9585 057172 006200 ASR RO ; RIGHT
9586 057174 006200 ASR RO ; THREE WHOLE PLACES
9587 057176 020427 057462 CMP R4,#0.BUF+1 ; DONE?
9588 057202 101365 BHI 0.SPC ; NO
9589 057204 042700 177776 BIC #177776,RO ; GET LAST BIT
9590 057210 062600 ADD (SP)+,RO ; CONVERT TO ASCII
9591 057212 110044 MOVB RO,-(R4) ; AND PUT IT AWAY
9592 057214 012703 057467 MOV #0.BUF+6,R3 ; LWA
9593 057220 004537 057272 JSR 5,0.FTYP ; TYPE WHOLE STRING OF CHARACTERS
9594 057224 012602 MOV (SP)+,R2 ; RESTORE R2
9595 057226 000205 RTS R5
9596 ;
9597 ; GENERAL CHARACTER INPUT ROUTINE
9598 ; CHARACTER INPUT GOES TO RO
9599 ;
9600 057230 105737 177560 0.GET: TSTB 0.RCSR ; WAIT FOR
9601 057234 100375 BPL -4 ; INPUT FROM KEYBOARD
9602 057236 113700 177562 MOVB 0.RDB,RO ; GET A CHARACTER
9603 057242 004537 057306 JSR 5,0.FTYP ; ECHO CHARACTER
9604 057246 042700 177600 BIC #177600,RO ; STRIP OFF PARITY FROM CHARACTER
9605 057252 001766 BEQ 0.GET ; IGNORE NULLS
9606 057254 122700 000040 CMPB #40,RO ; CHECK FOR SPACES
9607 057260 001763 BEQ 0.GET ; IGNORE NULLS
9608 057262 122700 000073 CMPB #'',RO ; CHECK FOR SEMI-COLON
9609 057266 001760 BEQ 0.GET ; IGNORE THEM IF FOUND
9610 057270 000205 RTS R5
9611 ;
9612 ; GENERAL CHARACTER OUTPUT ROUTINE
9613 ; ADDRESS OF FIRST BYTE IN R4,
9614 ; ADDRESS OF LAST BYTE IN R3. (R3)>(R4)
9615 ;
9616 057272 020304 0.TYPE: CMP R3,R4 ; CHECK FOR COMPLETION
9617 057274 103426 BLO 0.TYP1 ; EXIT WHEN DONE
9618 057276 112400 MOVB (R4)+,RO ; GET A CHARACTER
9619 057300 004537 057306 JSR 5,0.FTYP ; TYPE ONE CHARACTER
9620 057304 000772 BR 0.TYPE ; LOOP UNTIL DONE
9621 ;
9622 ; TYPE ONLY ONE CHARACTER (CONTAINED IN RO)
9623 ;
9624 057306 105737 177564 0.FTYP: TSTB 0.TCSR ; CHECK STATUS

```

9625	057312	100375		BPL	-4				;WAIT UNTIL READY
9626	057314	110037	177566	MOV8	R0,0.TDB				;TYPE ONE CHARACTER
9627	057320	120037	000045	CMP8	R0,0#45				;IS CHAR TO BE FILLED?
9628	057324	001012		BNE	0.TYP1				;NO
9629	057326	113746	000044	MOV8	0#44,-(SP)				;YES, INIT THE COUNT
9630	057332	105737	177564	O.TYP2:	TST8	0.TCSR			
9631	057336	100375		BPL	0.TYP2				
9632	057340	105037	177566	CLR8	0.TDB				;GENERATE NULL FILLER
9633	057344	105316		DECB	0.SP				
9634	057346	003371		BGT	0.TYP2				
9635	057350	005726		TST	(SP)+				;POP STACK
9636	057352	000205		O.TYP1:	RTS	R5			
9637									
9638									
9639									
9640									
9641	057354	006205		O.TCLS:	ASR	R5			;GET LOW ORDER BIT
9642	057356	103405			BCS	0.TC			;JUMP IF ALREADY CLOSED
9643	057360	006305			ASL	R5			
9644	057362	005702			TST	R2			;IF NO NUMBER WAS TYPED THERE IS
9645	057364	001401			BEQ	0.CLS1			;NO CHANGE TO THE OPEN CELL
9646	057366	010415			MOV	R4,0R5			;STORE WORD
9647	057370	000207		O.CLS1:	RTS	PC			
9648	057372	005746		O.TC:	TST	-(SP)			;POP EXTRA CELL FROM STACK
9649	057374	000137	055676		JMP	0.ERR			;AND SCREAM BLOODY MURDER
9650									
9651									
9652									
9653									
9654	057400	012703	057435	O.CRLF:	MOV	#0.CR+1,R3			;LWA <CR,LF>
9655	057404	000402			BR	0.CRS			
9656	057406	012703	057436	O.CRLS:	MOV	#0.CR+2,R3			;LWA <CR,LF>*
9657	057412	012704	057434	O.CRS:	MOV	#0.CR,R4			;FWA
9658	057416	004537	057272		JSR	5.0.TYPE			;TYPE SOMETHING
9659	057422	000205			RTS	R5			
9660									
9661	057424	0000C		O.XXX:	.WORD	0			;TEMPORARY STORAGE
9662	057426	000		O.T:	.BYTE	0			;T-BIT FLAG
9663	057427	000		O.P:	.BYTE	0			;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
9664									= 1 IF PROCEED ALLOWED
9665	057430	000		O.CSR1:	.BYTE	0			;SAVE CELL - R C/SR
9666	057431	000		O.CSR2:	.BYTE	0			;SAVE CELL - Y C/SR
9667									
9668									
9669	057432	042502		O.BD:	.EVEN				
9670					.WORD	"BE			
9671	057434	015		O.CR:	.BYTE	015			<CR>
9672	057435	012			.BYTE	012			<LF>
9673	057436	052			.BYTE	'*			*
9674									
9675	057437	057		O.LGCH:	.BYTE	'/'			/
9676	057440	015			.BYTE	015			CARRIAGE RETURN
9677	057441	044			.BYTE	'\$			\$
9678	057442	107			.BYTE	'G			G
9679	057443	012			.BYTE	012			<LF>
9680	057444	137			.BYTE	'+			+







# B15

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 183  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0182

BADTMO = 036522	3382	6949*																	
BAI = 000020	1203*	4123	4274	4416	4489	4898	4959	5087	5256	5331	5451	5505	5630						
	5762	5808																	
BA16 = 000400	1189*																		
BA17 = 001000	1190*																		
BIT0 = 000001	1132*	1186	1218	1237	2116	3421	6835												
BIT00 = 000001	1122*	1132																	
BIT01 = 000002	1121*	1131																	
BIT02 = 000004	1120*	1130																	
BIT03 = 000010	1119*	1129																	
BIT04 = 000020	1118*	1128																	
BIT05 = 000040	1117*	1127																	
BIT06 = 000100	1116*	1126																	
BIT07 = 000200	1115*	1125																	
BIT08 = 000400	1114*	1124	7121																
BIT09 = 001000	1113*	1123	7129	7197															
BIT1 = 000002	1131*	1219	2117																
BIT10 = 002000	1112*	1191	1209	1228	1260	1274	1288	1301	1315	7174									
BIT11 = 004000	1111*	1192	1210	1229	1246	1261	1275	1289	1302	1316	7136								
BIT12 = 010000	1110*	1193	1211	1230	1262	1276	1290	1303	1317										
BIT13 = 020000	1109*	1194	1212	1231	1247	1263	1277	1291	1304	1318	7181								
BIT14 = 040000	1108*	1195	1213	1232	1248	1264	1278	1292	1305	1319	1332	6637	6648						
	7107																		
BIT15 = 100000	1107*	1196	1197	1214	1233	1249	1265	1335	5694	5857	6633	6644							
BIT2 = 000004	1130*	1220	1239	2118															
BIT3 = 000010	1129*	1202	1221	1240															
BIT4 = 000020	1128*	1203	1222	1241	1254	1282	1309												
BIT5 = 000040	1127*	1204	1223	1242	1255	1269	1283	1296	1310										
BIT6 = 000100	1126*	1187	1205	1224	1243	1256	1270	1284	1297	1311									
BIT7 = 000200	1125*	1188	1206	1225	1244	1257	1271	1285	1298	1312	3411	4654	4698						
	4750	4849	5112																
BIT8 = 000400	1124*	1189	1207	1226	1245	1258	1272	1286	1299	1313									
BIT9 = 001000	1123*	1190	1208	1227	1259	1273	1287	1300	1314										
BPTVEC = 000014	1139*																		
BSE = 000200	1225*	4025	4134	4199	4431	4911	5267	5462											
BSEERR = 001476	2032*	3886*	3891*	3969															
BSE20H = 002322	2041*	3907	6642	6747															
BSE20S = 004322	2043*	3916	6646	6751															
BSE22H = 003322	2042*	3826	3888	3925	6631	6736													
BSE22S = 005322	2044*	3898	6635	6740															
BYP = 032736	3491	3514	3518	3533	3585	3593	3611	3615	3619	3623	6213*								
BYP CER = 001502	2034*	3408*	3637*	6327															
CALADD = 001362	1982*	4094*	4386*	4564*	5866*	6604	6610	6612	6666										
CALDIF = 001354	1979*	5755*	8888	8912															
CCLR = 100000	1197*	3787	3805	4035	4144	4441	4668	4679	4921	5075	5145	5163	5241						
	5277	5392	5410	5563	5581	6368	6378	6388	6398	6895	6913								
CCYL = 001350	1977*																		
CDT = 002000	1191*																		
CERR = 100000	1196*	3471	3571	3838	3944	3997	4022	4105	4131	4197	4284	4400	4428						
	4500	4575	4721	4778	4842	4908	4968	5264	5340	5459	5513	5645	5718						
	5741	5775	5821	5877	5902	6329	6424												
CFMT = 010000	1193*	4342	4396	4420	4424	4463	4492	4496	4529	6733									
CHKFLG = 001504	2035*	6273*	6305	6308	6312														
CHKMSG = 032752	3850	7	4066	4156	4175	4223	4240	4303	4322	4357	4453	4481	4519						
	4547	119	4640	4690	4742	4805	4933	4952	4987	5006	5046	5067	5212						
	5233	289	5308	5359	5378	5477	5496	5532	5551	5657	5789	5835	6223*						











# H15

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 189  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0188

E.B2	007442	2110#	3847*	4044*	4063*	4153*	4172*	4220*	4237*	4300*	4319*	4354*	4450*	4478*
		4516#	4544*	4637*	4739*	4802*	4930*	4949*	4984*	5003*	5064*	5230*	5286*	5305*
E.B3	007446	5356#	5375*	5474*	5493*	5529*	5548*	5654*	5786*	5832*	6254	6256*	8938	8949
		2112#	3848*	4045*	4064*	4154*	4173*	4221*	4238*	4301*	4320*	4355*	4451*	4479*
		4517#	4545*	4638*	4740*	4803*	4931*	4950*	4985*	5004*	5065*	5231*	5287*	5306*
		5357#	5376*	5475*	5494*	5530*	5549*	5655*	5787*	5833*	6258	6260*	8949	
FATT1	032544	3802	5160	5407	5578	6157#	6885	6910						
FATT2	032640	3941	4628	4718	4839	5055	5221	5738	5899	6184#				
FHDHM	035000	6557#	6565											
FHDTAB	035122	4097	4389	4567	5869	6590#								
FLGTST	035402	6632	6636	6643	6647	6659#								
FLOAD	035054	6574#												
FMT#	= 000020	1222#												
FMT1	001454	2020#	6600*	6601*	6602*	6607								
FORM	031226	5747	5853	5856#	6925									
FORMAT	001452	2019#	4096*	4388*	4566*	5868*	6600	6629						
FRCYL	001344	1975#	5750*	8888	8912									
FRDY	032230	3450	3464	3469	3564	3569	3747	3793	3809	3835	3937	3994	4019	4102
		4128	4194	4281	4340	4394	4398	4422	4426	4465	4494	4498	4531	4572
		4611	4672	4714	4776	4835	4905	4965	5038	5079	5092	5151	5167	5204
		5245	5261	5337	5398	5414	5456	5510	5569	5585	5640	5715	5734	5767
		5813	5874	5895	6086#	6089	6358	6419	6881	6901	6917			
FRDY1	032276	6100#	6103	6373	6383	6393	6402							
FSEC23	034710	5625	6533#											
FTITLE	001340	1971#	5979	5981*										
G8A	032132	3283	6046#											
GDRVS	031772	3281	6001#											
GINT	032160	3285	6059#											
GMS	= ***** U	1343	8063	8064	8065	8066	8067	8069	8071	8072	8073	8074	8075	8076
		8077												
GO	= 000001	1186#												
GSTAT	033566	3700	3797	3837	3996	4021	4104	4130	4196	4283	4342	4574	4680	4791
		4907	4967	5094	5155	5263	5339	5402	5458	5512	5573	5642	5717	5770
		5816	5876	6167	6171	6191	6195	6354#	6423	6434	6446	6479	6559	6576
		6905												
GSTAT1	033630	6225	6366#											
GTSWR	= 104406	5990	8069#											
HASOF	007356	2076#	6125*	6141	8884	8893	8901	8909	8917	8925	8935	8946	8958	
H8A	007346	2072#	6121*	8884	8893	8901	8909	8917	8925	8935	8946	8958		
HCS1	007340	2069#	3471	3571	3838	3944	3997	4022	4105	4131	4197	4284	4400	4428
		4500	4575	4721	4778	4842	4908	4968	5264	5340	5459	5513	5645	5718
		5741	5775	5821	5877	5902	6118*	6329	6333	6424	6733	8881	8890	8898
		8906	8914	8922	8932	8943	8955							
HCS2	007342	2070#	3504	3506	3510	3522	3597	3599	3603	3606	4286	4343	4502	4970
		5095	5342	5515	6119*	6338	6342	8881	8890	8898	8906	8914	8922	8932
		8943	8955											
H8A	007350	2073#	6122*	6776	8884	8893	8901	8909	8917	8925	8935	8946	8958	
H8B	007362	2079#												
H8C	007360	2077#	6126*	6768	8884	8893	8901	8909	8917	8925	8935	8946	8958	
H8S	007352	2074#	3508	3601	6123*	8881	8890	8898	8906	8914	8922	8932	8943	8955
HDTAB	001506	2037#	3980	3986	3989	4090	4380	4559	5699	5706	5709	5862	6592	
H8WD	001474	2030#	8904											
H81	001450	2018#	6594*	6595*	6596*	6597*	6598*	6599*	6606					
HEAD	001444	2016#	4095*	4387*	4565*	5867*	6594	6671						
HEAD8	001446	2017#	6523*	6524*	6525*	6526*								
HER	007354	2075#	4025	4134	4199	4206	443.	4911	5267	5462	6124*	8881	8890	8898







0.B2	056734	9517	9526#						
0.B3	056774	9525	9534#						
0.CADV	057144	9351	9368	9391	9395	9425	9429	9534	9577#
0.CLGL	055762	9308	9310	9318#					
0.CLGT=	000014	9322	9687#						
0.CLS1	057370	9645	9647#						
0.CR	057434	9654	9656	9657	9671#				
0.CRET	056074	9329	9358#						
0.CRLF	057400	9366	9423	9519	9558	9563	9654#		
0.CRLS	057406	9303	9656#						
0.CRS	057412	9655	9657#						
0.CSR1	057430	9554*	9570	9665#					
0.CSR2	057431	9555*	9571	9666#					
0.CT	055416	9483*	9526*	9528*	9725#				
0.C1	056546	9486#	9527						
0.DCD	055712	9291	9293	9303#	9360	9411	9536		
0.DCD1	055716	9304#	9352	9396					
0.EFF	056222	9337	9402#						
0.EFF1	056340	9414	9433#						
0.ENTR	055426	9230#							
0.ERR	055676	9277	9288	9300#	9323	9348	9406	9649	
0.ERR1	056232	9406#	9479						
0.ERR2	056052	9348#	9382	9455					
0.FTYP	057306	9302	9370	9385	9427	9532	9603	9619	9624#
0.GET	057230	9256	9306	9600#	9605	9607	9609		
0.GO	056404	9331	9453#						
0.GO2	056462	9464#	9489						
0.LG =	000006	9260	9695#						
0.LGCH	057437	9319	9675#	9687					
0.LGDR	056012	9326	9328#	9340					
0.LGL =	000030	9340#							
0.LGL1	055764	9319#	9324						
0.LGL2	056004	9320	9325#						
0.MSK	055406	9407	9408	9410	9717#				
0.ODT	055422	1352	9228#	9706					
0.OFST	056150	9335	9381#						
0.OF1	056216	9394	9396#						
0.OP1	056106	9332	9364#						
0.OP2	056114	9281	9366#	9377					
0.ORPC	055620	9273#	9333						
0.P	057427	9248*	9453*	9478	9480*	9529*	9663#		
0.PRI	055404	9240	9502	9506	9716#				
0.PROC	056512	9339	9478#						
0.PR1	056534	9482	9484#						
0.RALL	055666	9251	9286	9292#					
0.RCSR=	177560	9218#	9554	9556*	9566	9568	9570*	9600	
0.ROB =	177562	9217#	9602						
0.REGT	055550	9256#	9330						
0.RSE1	057126	9567	9570#						
0.RSP	055560	9258#	9261						
0.RST	055464	9229	9238#						
0.RSTT	057076	9459	9485	9563#					
0.RST1	055514	9237	9245#						
0.RTIT	056510	9247*	9473#						
0.SCAN	055722	9267	9306#	9317					
0.SP	055612	9259	9268#						

0.SPC	057156	9580#	9588						
0.SP1	055600	9264#	9269						
0.STM =	000340	9207#	9249	9458	9484	9486			
0.STRT	055452	9228	9235#						
0.SVR	057006	9238	9498	9540#					
0.SVTT	057044	9518	9530	9554#					
0.T	057426	9460*	9487*	9499	9662#				
0.TBIT	056434	9460#	9500						
0.TBT =	000020	9208#	9461	9488	9524				
0.TC	057372	9642	9648#						
0.TCLS	057354	9273	9358	9364	9375	9641#			
0.TCSR=	177564	9220#	9555	9557*	9564	9571*	9624	9630	
0.TDB =	177566	9219#	9626*	9632*					
0.TL	057453	9257	9260	9268	9689#	9695			
0.TRTC	057470	9292	9463	9702#					
0.TVEC=	000014	9206#	9231	9249*	9250*				
0.TYPE	057272	9522	9593	9616#	9620	9658			
0.TYP1	057352	9617	9628	9636#					
0.TYP2	057332	9630#	9631	9634					
0.UIN	055420	9239	9462*	9501	9726#				
0.UPC	055400	9232*	9457*	9472	9496*	9513	9515*	9714#	
0.URD	055362	9235	9265	9707#					
0.USP	055376	9236*	9541*	9542	9713#				
0.UST	055402	9230*	9461*	9471	9488*	9497*	9504	9524*	9715#
0.WDS	056230	9403	9405#						
0.WDS2	056246	9410#	9432						
0.WDS3	056276	9421#	9434	9440	9449				
0.WDS4	056334	9421	9431#						
0.WRD	056042	9328	9344#						
0.WROA	056066	9345	9353#						
0.WRD1	056050	9347#	9354	9371					
0.WSCH	056226	9336	9404#						
0.XXX	057424	9540*	9548	9661#					
0.45	055526	9246	9248#						
PACK =	000003	1173#	3745						
PARAM	001336	1970#	3165*	3168*	3277				
PARSRT	012532	1349	3165#						
PAT =	000020	1254#							
PCA =	004000	1261#							
PCD =	010000	1262#							
PCLKF	007504	2143#	3321*	3328*	6785	6806			
PCVEC	001332	1964#	3322	3329					
PCYL	001352	1978#							
PF SRT	015122	3681#	7028						
PGE =	002000	1209#							
PIP =	020000	1247#							
PIRQ =	177772	1052#							
PIRQVE=	000240	1146#							
PKRB	001324	1960#							
PKS	001320	1958#	3320	3327	6790*	6810*			
PKSB	001322	1959#	6789*						
PPTP	007456	2127#	3268*						
PRGSRT	012546	3166	3169#	6033					
PRO =	000000	1069#	3171	3359					
PR1 =	000040	1070#							
PR2 =	000100	1071#							























J16

UNIBUS RK06 DRIVE DIAGNOSTIC PART 2  
DZR6ID.P11 31-JAN-77 15:13

MACY11 27(1006) 01-FEB-77 04:10 PAGE 205  
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0203

.\$SCOP 996# 7091  
.\$SUPR 996# 7917  
.\$TRAP 996# 8032  
.\$STYPD 996# 7288  
.\$STYPE 996# 7209  
.\$STYPO 996# 7412

. ABS. 057472 000

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

DZR6ID,DZR6ID.SEQ/SOL/CRF/NL:TOC/DOC=DZR6ID.P11  
RUN-TIME: 24 22 2 SECONDS  
RUN-TIME RATIO: 72/49=1.4  
CORE USED: 31K (62 PAGES)

DOCUMENT PAGES: 203

EOF1DZR6IDSEQ

00010000

770323

PDP10 411