

RP04

FNCTNL CONT TEST 1 STAT  
MD-11-DERPU-B

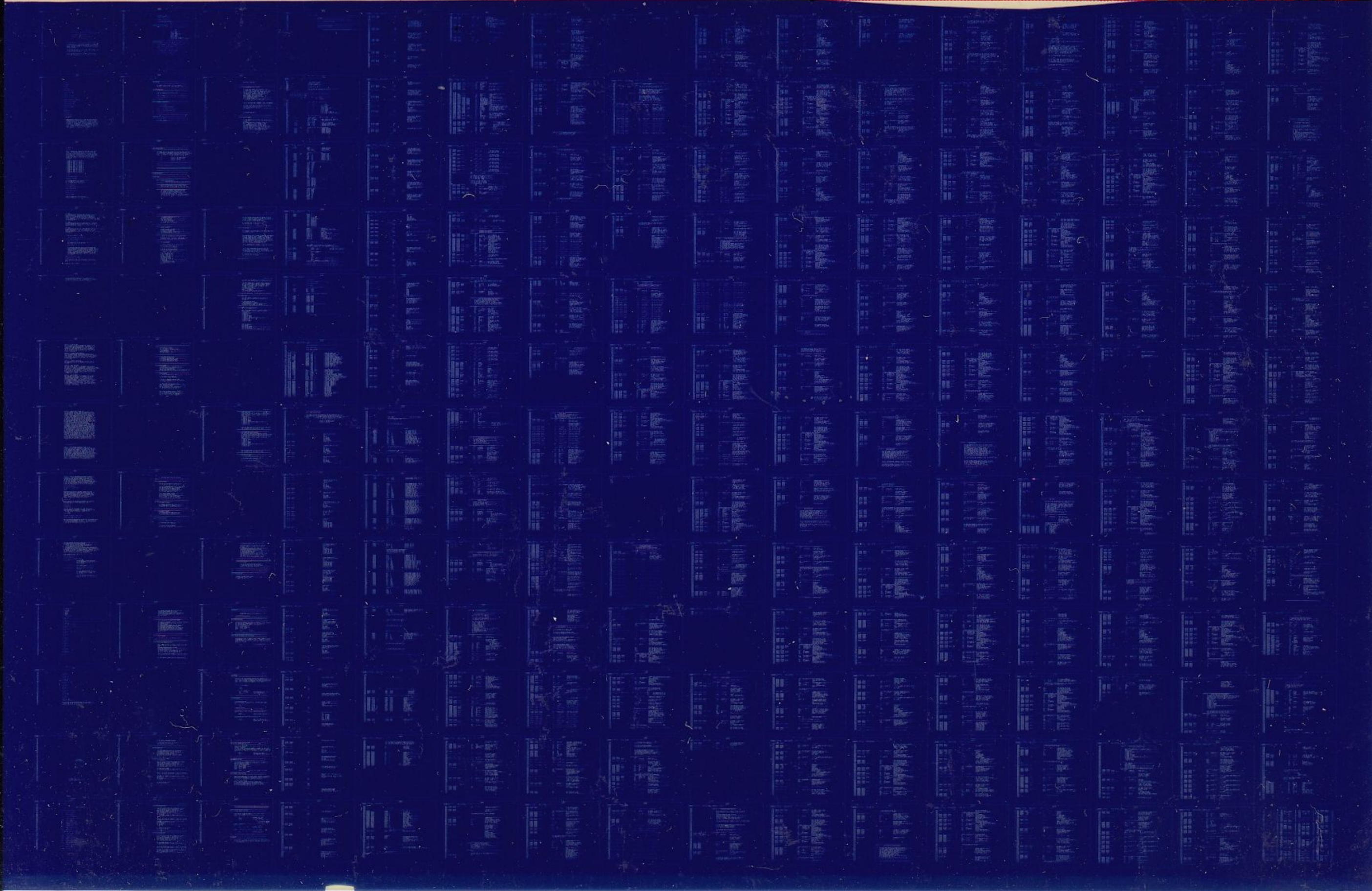
EP-DERPU-B-DL-A  
COPYRIGHT © 1976

NOV 1976

digital

FICHE 1 OF 2

MADE IN USA



















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

200 START  
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS ALL THE RPO4S ON THE SYSTEM WILL BE TESTED ONE AT A TIME BEFORE "END PASS" IS PRINTED OUT. TESTING WILL START WITH THE LOWEST UNIT NUMBER DRIVE THAT IS POWERED UP (THAT IS THE LOWEST UNIT NUMBER RHAS REGISTER THAT RESPONDS) THEN GO ON TO THE NEXT HIGHER UNIT NUMBER THAT IS POWERED UP.

210 START  
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS THE CONSOLE TELETYPE WILL ASK FOR THE UNIT NUMBER TO BE TESTED. THEN ONLY THAT UNIT WILL BE TESTED FOR EACH PASS OF THE PROGRAM.

220 START  
ALL SWITCHES MUST BE DOWN FOR WORST CASE RUN. WITH THIS STARTING ADDRESS THE PROGRAM WILL NOT RUN THOSE TESTS THAT NEED MANUAL INTERVENTION. THIS IS RECOMMENDED ONLY FOR DEBUGGING WHERE THE ERROR IS NOT IN A TEST THAT REQUIRES MANUAL INTERVENTION

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY.
2. SET STARTING ADDRESS ON THE SWITCH REGISTER
3. PRESS "LOAD ADDRESS".
4. SET "OPERATIONAL SWITCH SETTINGS" (SEE SECTION 5.1) WORST CASE IS ALL SWITCHES DOWN.
5. PRESS "START".
6. FOR THE FIRST PASS EACH TEST WILL BE EXECUTED ONCE ON THE DRIVES PRESENT OR DRIVE SELECTED BEFORE "END PASS" IS PRINTED. THE FIRST PASS WILL REQUIRE OPERATOR INTERVENTION IF THE PROGRAM IS NOT RUN UNDER AN "ACT-11" MONITOR. THE SECOND AND SUBSEQUENT PASSES WILL EXECUTE EACH TEST FOUR TIMES ON EACH DRIVES PRESENT OR DRIVE SELECTED BEFORE "END PASS" IS PRINTED. THE SECOND AND SUBSEQUENT PASSED DO NOT NEED ANY OPERATOR INTERVENTION.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SWITCH DEFINITIONS ARE GIVEN IN SECTION 9 "OPERATIONAL SWITCH SETTINGS" HOWEVER THE DETAIL DESCRIPTION ARE GIVEN HERE.

SWITCH 15 - HALT ON ERROR  
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR



F01

MAINDEC-11-DERPU-B  
DERPUB.P11

MACY11 27(732) 17-SEP-76 13:23 PAGE 5

210  
111  
111  
111  
111

THEN THE APPROPRIATE INFORMATION WILL BE PRINTED OUT  
AND THEN THE PROGRAM WILL HALT. AFTER THIS HALT, PRESSING  
"CONTINUE" WILL CONTINUE WITH THE PROGRAM TILL THE NEXT  
ERROR IS FOUND WHEN THE SAME THING WILL HAPPEN.



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

SWITCH 14 - LOOP ON TEST  
WHEN THIS SWITCH IS SET THE PROGRAM WILL BEGIN TO LOOP ON THE CURRENT TEST BEING EXECUTED. FOR EXAMPLE IF THIS SWITCH IS SET WHEN THE PROGRAM IS IN TEST 10 THEN THE PROGRAM WILL KEEP EXECUTING ALL OF TEST 10 REPEATEDLY. ONE WAY TO BE SURE THAT THE PROGRAM IS IN THE EXPECTED TEST IS TO SET THIS SWITCH DURING AN ERROR PRINTOUT OR DURING A PROGRAM HALT.

SWITCH 13 - INHIBIT ERROR TYPEOUTS  
WHEN THIS SWITCH IS SET FURTHER ERROR PRINTOUTS WILL CEASE. HOWEVER OPERATOR INSTRUCTIONS SUCH AS "STOP DRIVE X" WILL CONTINUE. AT THE END OF PASS "TOTAL NUMBER OF ERRORS ON THIS PASS ON DRIVE X" WILL BE TRUE, THAT IS, ALTHOUGH PRINTOUTS WERE INHIBITED IF THAT PASS FOUND 6 ERRORS, IT WILL SAY SO.

SWITCH 11 - INHIBIT ITERATIONS  
WHEN THIS SWITCH IS SET THE PROGRAM ON SECOND PASS WILL NOT REPEAT EACH TEST FOUR TIMES BUT WILL DO EACH TEST ONCE ONLY.

SWITCH 10 - BELL ON ERROR  
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR THE "BELL" OR "ALARM" WILL BE SOUNDED. THIS SWITCH IS USEFUL WHEN SWITCH 11 IS SET YET INFORMATION IS NEEDED WHEN ANY ERROR IS DETECTED. TAKE THE EXAMPLE OF A PROGRAM LOOPING ON A TEST WITH SWITCH 11 SET TO HELP SCOPING. THEN IF THIS SWITCH IS SET AND THE BELL OR ALARM SOUNDS IT MEANS THAT THE ERROR IS PRESENT BUT IF THE BELL OR ALARM STOPS IT MEANS THAT THE ERROR IS NOT PRESENT.

SWITCH 9 - LOOP ON ERROR  
WHEN THIS SWITCH IS SET, IF THE PROGRAM FINDS AN ERROR THEN GENERALLY THE PROGRAM WILL LOOP BACK TO THE LAST EXECUTED "SCOPE" STATEMENT. IF ON THE SECOND TIME THROUGH AN ERROR IS FOUND IT WILL AGAIN LOOP BACK TO THAT "SCOPE" STATEMENT. THIS LOOPING WILL CONTINUE AS LONG AS THE ERROR IS PRESENT AND THIS SWITCH IS SET. HOWEVER IF THE ERROR IS NOT PRESENT AT ANY TIME THEN IT WILL CONTINUE NORMALLY WITH THE PROGRAM. EACH TIME THE ERROR IS ENCOUNTERED PRINTOUT WILL TAKE PLACE UNLESS SWITCH 11 IS ALSO SET. DURING BEGUG, USING A SCOPE, IT IS RECOMMENDED THAT SWITCH 11 IS ALSO SET.

NOTE: SEE SECTION 8.3



2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799  
2800  
2801  
2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816

SWITCH 8 - LOOP ON TEST IN SWR <7:0>  
THIS IS A SPECIAL SWITCH. WHEN SET SWITCHES 0 THRU 7  
HAVE ONE MEANING AND WHEN RESET SWITCHES 0 THRU 7 HAVE  
ANOTHER MEANING. THIS MEANS THAT ANY SETTING OF SWITCH  
0 THRU 7 MUST BE DONE WITH SWITCH 8 IN THE APPROPRIATE  
POSITION. WHEN THIS SWITCH IS SET THEN SWITCHES 0 THRU  
7 GIVE THE TEST NUMBER TO BE LOOPED ON. FOR EXAMPLE  
WITH SWITCH 8 SET AND SWITCH 3 SET THE PROGRAM WILL LOOP  
ON TEST 10. HOWEVER THIS SETTING MUST BE DONE AT THE  
BEGINNING OF THE PROGRAM THEN ALL THE TESTS FROM 1 TO 10  
WILL BE EXECUTED AND THEN TEST 10 WILL BE REPEATED OVER  
AND OVER AGAIN. WHEN THIS SWITCH IS NOT SET THEN SWITCHES  
0 THRU 7 HAVE THE MEANING ITS NAME INDICATES.  
FOR EXAMPLE SWITCH 7 IS "STOP FURTHER COMPARES: THAT IS  
IF SWITCH 8 IS NOT SET AND SWITCH 7 IS SET THEN WHEN A  
DATA ERROR IS DETECTED NO FURTHER COMPARES WILL BE DONE.  
FOR EXAMPLE IN A 256 WORD BUFFER IF ALL THE WORDS ARE IN  
ERROR THEN AFTER SEEING THE PRINTOUT FOR THE FIRST FEW  
WORDS SETTING SWITCH 7 ONLY WILL STOP FURTHER PRINTOUTS  
OF THIS ERROR AND GO ON WITH THE TEST RATHER THAN PRINT  
ALL THE 256 WORDS. HOWEVER IF THIS WAS DONE WITH SWITCH 11  
THEN THE NEXT ERROR THAT THE PROGRAM DETECTS IN A SUB-  
SEQUENT TEST WILL ALSO BE LOST. BUT WITH SWITCH 7, ONLY  
THIS GROUP OF DATA ERRORS ARE NOT PRINTED OUT. ANOTHER  
EXAMPLE OF SWITCH 8 BEING LOW IS WITH SWITCH 6, WHICH  
IS "ECC TEST-COMPARE END RESULT ONLY". THAT IS IF SWITCH  
8 IS NOT SET AND SWITCH 6 IS SET THEN ON ECC TESTS (TEST  
120 THRU TEST 134) INSTEAD OF COMPARING CONTENTS OF THE  
POSITION REGISTER AND PATTERN REGISTER AFTER EVERY CLOCK,  
COMPARES WILL ONLY BE DONE AT THE END OF ALL THE CLOCKS.

NOTE: SEE SECTION 8.3

SWITCH 7 - STOP FURTHER COMPARES IF SW08 IS LOW.  
IF SWITCH 8 IS SET AND THIS SWITCH IS ALSO SET THEN THIS  
SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS INDICATED  
IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS NOT SET  
AND THIS SWITCH IS SET THEN THE PROGRAM WILL DO AS THE  
NAME INDICATES. FOR EXAMPLE IN A 256 WORD BUFFER IF ALL  
THE WORDS ARE IN ERROR THEN AFTER SEEING THE ERROR PRINTOUTS  
FOR THE FIRST FEW WORDS THEN SETTING SWITCH 7 WITH SWITCH 8  
NOT SET WILL STOP THE PRINTOUT OF ALL 256 WORDS BUT WILL  
NOT STOP THE PRINTOUT OF ANOTHER ERROR IN ANY SUBSEQUENT  
TEST. IT IS EXPECTED THAT SWITCH 7 AFTER BEING SET FOR  
A WHILE TO STOP PRINTING ALL THE 256 WORDS WILL BE RESET  
AGAIN TO ENABLE THE PRINTING OF OTHER DATA ERRORS.



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

SWITCH 6 - TYPE ALL REGISTERS WITH ERROR IF SW08 IS LOW  
IF SWITCH 9 IS SET AND THIS SWITCH IS ALSO SET THEN  
THIS SWITCH GIVES THE TEST NUMBER TO BE LOOPED ON AS  
INDICATED IN THE DESCRIPTION OF SWITCH 8. IF SWITCH 8 IS  
NOT SET AND THIS SWITCH IS SET THEN THE PROGRAM WILL DO  
AS THE NAME INDICATES. THAT IS ON FINDING AN ERROR INSTEAD  
OF ONLY GIVING THE ERROR MESSAGE AND RELEVANT REGISTERS  
AS WILL BE DONE IF SWITCH 11 IS NOT SET BUT WILL ALSO  
GIVE ALL THE REGISTER CONTENTS (EXCEPT "DATA BUFFER" RHDB).

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES".

5.0 ERRORS

ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS AND OTHER  
PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE.  
THIS INFORMATION MAY BE THE CONTENTS OF RELEVANT RPO4  
REGISTERS OR GOOD/RECEIVED DATA. IF THE ERROR OCCURRED IN A  
SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO  
GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED  
ADDRESS TO DETERMINE THE CAUSE OF THE ERROR.

7.0 RESTRICTIONS

BEFORE STARTING THE PROGRAM THE OPERATOR MUST HAVE THE  
DRIVE PORT SWITCH LOCKED EITHER ON PORT A OR PORT B BUT  
MUST NEVER LEAVE IT IN THE PROGRAMMABLE STATE.

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL TAKE APPROXIMATELY 1  
MINUTES PROVIDED AN OPERATOR IS PRESENT TO CARRY OUT THE  
TYPED INSTRUCTIONS IMMEDIATELY. SUBSEQUENT PASSES WILL TAKE  
30 SECONDS WHETHER AN OPERATOR IS THERE OR NOT.

8.2 STACK POINTER

THE STACK IS INITIALLY SET TO 1000

8.3 OPERATOR SELECTABLE SCOPE LOOPS

HERE IS A DETAILED EXPLANATION OF HOW THE LOOP ON ERROR WORKS.  
FOR INSTRUCTIONS REGARDING THE USAGE OF THIS TECHNIQUE, HIT ↑  
ANY TIME WHILE THE PROGRAM IS RUNNING. ON HITTING AN ERROR  
IF THE LOOP ON ERROR SWITCH IS SET, THE PROGRAM GOES BACK -  
USUALLY BACK TO THE BEGINNING OF THE TEST.



368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406

WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT THE PROGRAM GOES BACK TO CAN BE CHANGED.  
THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -  
1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION  
2. LOOP ON ERROR SWITCH MUST BE SET  
3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION  
IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT COMES TO THE END OF THE TEST UNDER CONSIDERATION.

AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN NORMAL OPERATION WILL CONTINUE.

9.0 PROGRAM DESCRIPTION

9.1 LOGIC DIVISION IN HARDWARE MODULES

REGISTER BOARD (RG) - ERROR REGISTER 1 STATUS REGISTERS  
MUX FOR REGISTERS GO HANDLING REGISTER  
DECODE COMMAND DECODE EXECUTION OF  
MECH. COMMANDS

SYNC. DATA BOARD (SN) - DATA CONTROL PARALLEL TO SERIAL  
SYNC. BYTE DETECT.

SEEK AND SEARCH (SS) - SEEK LOGIC SEARCH LOGIC HEADER  
HANDLING.

ERROR CORRECTION (EC) - ECC LOGIC ERROR REGISTER 2 & 3  
MUX FOR ERROR REG. 2 & 3 LOOK AHEAD  
REG. SECTOR COUNTER DATA FORMATION  
RING COUNTER.

DUAL PORT (DP) - DUAL PORT ARBITRATION ATTENTION LOGIC  
SERIAL NO REGISTER MASS BUS REGISTER  
STORAGE

40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58

9.2 DISK SURFACE USAGE

SYMBOLS USED

C = CYLINDER

T = TRACK

S = SECTOR

W = WRITE

R = READ

TT = TEST NUMBER

C0, T0, S0

TT22-W,R, TT23-R, TT24-W,R, TT25-W,R, TT26-W,R, TT35-W,R, TT37-W, TT50-W, TT51-W

C0, T0, S1

TT27-W,R, TT37-W,R, TT40-R, TT41-W,R, TT42-W,R, TT43-W,R

C0, T18, S21

TT30-W, TT31-W,R

C1, T0, S0

TT30-W,R, TT31-W,R, TT53-W,R, TT54-W,R

C1, T18, S21

TT31-W

C2, T0, S0

TT31-W,R

C2, T18, S21

TT31-W

C3, T0, S0

TT31-W,R

C3, T18, S21

TT31-W

C4, T0, S0

TT31-W,R

C4, T18, S21

TT31-W

C5, T0, S0

TT31-W,R

C5, T7, S4

TT33-W,R, TT34-W,R

C5, T18, S21

TT31-W

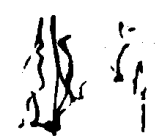


459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493

C6, TO, SO  
TT31-W,R  
  
C6, T18, S21  
TT31-W  
  
C7, TO, SO  
TT31-W,R  
  
C7, T18, S18  
TT31-W  
  
C8, TO, SO  
TT31-W,R  
  
C8, T18, S21  
TT31-W  
  
C9, TO, SO  
TT31-W  
  
C9, T18, S21  
TT31-W, TT32-R  
  
C10, TO, SO  
TT31-W,R  
  
C410, T18, S21  
TT36-W,R, TT50-W,R

9.3

THE FOLLOWING SECTION DESCRIBES EACH TEST AND SUBROUTINES  
IN DETAIL AND CAN BE USED AS AN INDEX TO THE LISTING.  
THE LEFT MOST COLUMN IS THE LINE NUMBER WITHIN THE LISTING  
WHERE THAT ITEM WILL BE FOUND.



MO1

MAINDEC-11-DERPU-B  
DERPUB 011

MACY11 27(732) 17-SEP-76 13:23 PAGE 12

MAINDEC-11-DERPU-B

DECDOC VER 00.04

494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600

DOCUMENT  
\*\*\*\*\*  
MAINDEC-11-DERPU-B  
\*\*\*\*\*

COPYRIGHT 1974, 1976  
DIGITAL EQUIPMENT CORPORATION  
MAYNARD, MASS. 01754



TABLE OF CONTENTS  
\*\*\*\*\*

22	OPERATIONAL SWITCH SETTINGS
36	BASIC DEFINITIONS
142	TRAP CATCHER
149	STARTING ADDRESS(ES)
165	MEMORY MANAGEMENT DEFINITIONS
204	COMMON TAGS
260	ERROR POINTER TABLE
1197	REGISTER ADDRESSES
1360	REGISTER TEST
8771	END OF PASS ROUTINE
8813	SUBROUTINES
9548	SCOPE HANDLER ROUTINE
9611	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
9679	TYPE ROUTINE
9726	TTY INPUT ROUTINE
9832	READ AN OCTAL NUMBER FROM THE TTY
9886	ERROR HANDLER ROUTINE
9933	ERROR MESSAGE TYPEOUT ROUTINE
10163	BINARY TO OCTAL (ASCII) AND TYPE
10241	TRAP DECODER
10256	TRAP TABLE
10278	POWER DOWN AND UP ROUTINES

541  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000

MAINDEC-11-DEPUB-8

DECDOC VER 00.04

2

COPYRIGHT (C) 1974  
DIGITAL EQUIPMENT CORP.  
MAYNARD, MASS. 01754

PROGRAM BY SUB MALLICK

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
PACKAGE (MAINDEC-11-DZGAC-A3).

22

\*\*\*\*\*  
OPERATIONAL SWITCH SETTINGS  
\*\*\*\*\*

23

SWITCH

JSE

SWITCH	JSE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
11	INHIBIT ITERATIONS
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SWR(7:0)
7	STOP FURTHER COMPARES IF SW08 IS LOW
6	TYPE ALL REG. WITH ERROR IF SW8 LOW

35

\*\*\*\*\*  
BASIC DEFINITIONS  
\*\*\*\*\*

- 38 INITIAL ADDRESS OF THE STACK POINTER \*\*\* 1000 \*\*\*
- 49 GENERAL PURPOSE REGISTER DEFINITIONS
- 61 PRIORITY LEVEL DEFINITIONS
- 71 "SWITCH REGISTER" SWITCH DEFINITIONS
- 99 DATA BIT DEFINITIONS (BIT00 TO BIT15)
- 127 BASIC "CPU" TRAP VECTOR ADDRESSES

MACY11-27(732) 17-SEP-76 13:23 PAGE 14





69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
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

```

*****
260 ERROR POINTER TABLE
*****

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

268          EM          ;;POINTS TO THE ERROR MESSAGE
          CH          ;;POINTS TO THE DATA HEADER
          DT          ;;POINTS TO THE DATA
          DF          ;;POINTS TO THE DATA FORMAT

277 *****
977 *****
983 *****
1017 *****
1192 *****

1197 *****
REGISTER ADDRESSES
*****

1360 *****
REGISTER TEST
*****

1479 *****
TEST 1 REFERENCE EACH REGISTER
          REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
*****

1517 *****
TEST 2 PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT
          CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES
*****

1537 *****
TEST 3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
          THE NUMBER OF RPO4 DRIVES PRESENT ARE FOUND
          BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
          IN RHCS2 INCREMENTED FROM ZERO TO SEVEN
          THEN THE SET BITS IN RHAS WILL GIVE DRIVES PRESENT
          THE DRIVE TYPE IS CHECKED TO HAVE 2020 OR 24020 AND THEN
          UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'

```



MAINDEC-11-DERPU-B

DECDOC VER 00.04

739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794

```

1546 *****
1657 *****
TEST 4 TYPE SERIAL NUMBER AND DRIVE TYPE
      SET APPROPRIATE ATTENTION BIT OF UNIT UNDER TEST IN 'ATT
      TYPE UNIT UNDER TEST
      READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
      TYPE IT OUT AND PROCEED
      TO LOOP HERE SET SWITCH 8, AND THIS TEST NUMBER ON
      SWITCHES 0 THRU 7 AND RESTART

1666 *****
1709 *****
TEST 5 PROGRAM INTERRUPT

1712 PROGRAM INTERRUPT IS TESTED BY SETTING RDY AND IE
      IN RHCS1 AT THE SAME TIME
      THIS SHOULD INTERRUPT THROUGH LOCATION 254
      THE PROCESSOR PRIORITY IS SET TO 4
*****

1751 *****
TEST 6 INTERRUPT AT PROCESSOR AND DISK PRIORITY SAME

1754 PROCESSOR PRIORITY IS SET AT 5 (SAME AS THE DISK)
      IE AND RDY IS SET. THIS SHOULD NOT INTERRUPT
*****

1789 *****
TEST 7 PACK ACKNOWLEDGE

1792 IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PE
      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
      THEN THIS TEST IS NOT PERFORMED

      IF NO ACT-11 MONITOR IS PRESENT
      THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE

      THIS TESTS THE ACKNOWLEDGE COMMAND=44
      VV BIT - RHDS1 BIT #6
      MOL BIT - RHDS1 BIT #12
      DVA BIT - RHCS1 BIT #11
      THE DRIVE IS STOPED MOL IS CHECKED TO BE 0
      AND DVA SHOULD BE 0
      THE DRIVE IS POWERED UP
      VV SHOULD BE 0, MOL SHOULD BE 1, DVA SHOULD BE 1
      GO SHOULD BE 0
      ALL REGISTERS EXCEPT RHDB, RHLA AND RHCC ARE STORED
      PACK ACKNOWLEDGE IS ISSUED
      ALL STORED REGISTERS SHOULD BE UNCHANGED
      EXCEPT VV

```

F02

MAINDEC-11-DERPU-B  
DERFUB.P11

MACY11 27(732) 17-SEP-76 13:23 PAGE 19

795

MAINDEC-11-DERPU-B

DECDOC VER 00.04

0000  
0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051

```

1815 *****
1993 *****
TEST 10 SET VV BIT #6 IN RHDS1
      THIS TEST SETS VV IN RHDS1 INCASE
      ACT-11 MONITOR IS PRESENT AND THE PREVIOUS TEST
      IS NOT PERFORMED
      THERE IS A RESET AT THE BEGINING OF THIS TEST
      FOR ERROR RECOVERY ONLY.
2001 *****
2119 *****
TEST 11 NO OPERATION FUNCTION TEST
      ALL POSSIBLE REGISTERS ARE CLEARED THEN A "NOP"=0
      IS GIVEN NO CHANGE SHOULD HAPPEN
      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A "NOP"
      IS GIVEN NO CHANGE SHOULD HAPPEN
2126 *****
2323 *****
TEST 12 DRIVE CLEAR
      ALL WRITE BITS OF ALL REGISTERS ARE FILLED
      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
      THEN A DRIVE CLEAR IS GIVEN
      THEN ALL REGISTERS ARE CHECKED TO HAVE APPROIPIAVE VALUE
*****
2513 *****
TEST 13 READ-IN-PRESET
      IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PE
      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
      THEN THIS TEST IS NOT PERFORMED
      IF NO ACT-11 MONITOR IS PRESENT THEN
      THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
      THIS TESTS THE READ-IN-PRESET COMMAND
      FIRST THE DRIVE IS POWERED DOWN AND UP IN ORDER TO
      RESET VV-BIT #6 IN RHDS1
      THEN ALL WRITE BITS OF ALL REGISTERS ARE FILLED
      WITH ONES EXCEPT GO,CLR,IE,PAT,MCPE,UPE
      ATA FOR DRIVE UNDER TEST IS MADE = 0
      THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN
      THEN ALL REGISTERS ARE TESTED
      THE FOLLOWING SHOULD BE THE REGISTER CONTENTS
      RHCA = 0, RHDST = 0, RHOF SHOULD HAVE
      FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1

```



MAINDEC-11-DERPU-B  
DERPUB.P11

MACY11 27(732) 17-SEP-76 13:23 PAGE 20

H02

852

MAINDEC-11-DERPU-B

DECDOC VER 00.04

853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908

ALL OTHER REGISTERS SHOULD BE UNCHANGED

2540 \*\*\*\*\*

2724 \*\*\*\*\*  
TEST 14 READ-IN-PRESET

2727 THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT  
THAT IT DOES NOT TEST THE SETTING OF VV

THIS TEST IS HERE BECAUSE IF ACT-11 MONITOR IS PRESENT  
THEN THE PREVIOUS TEST WILL NOT BE PERFORMED  
THIS TESTS THE READ-IN-PRESET COMMAND  
ALL WRITE BITS OF ALL REGISTERS ARE FILLED  
WITH ONES EXCEPT GO, CLR, IE, PAT, MCPE, UPE  
ATA FOR DRIVE UNDER TEST IS MADE = 0

THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN  
THEN ALL REGISTERS ARE TESTED  
THE FOLLOWING SHOULD BE THE REGISTER CONTENTS  
RHCA = 0, RHDST = 0, RHOF SHOULD HAVE  
FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1  
ALL OTHER REGISTERS SHOULD BE UNCHANGED

2744 \*\*\*\*\*

2872 \*\*\*\*\*  
TEST 15 RECALIBRATE COMMAND

2875 ALL POSSIBLE REGISTERS ARE FILLED WITH ONES  
THEN A RECALIBRATE = 6 COMMAND IS GIVEN  
NO REGISTERS SHOULD CHANGE EXCEPT RHCC = 0

2879 \*\*\*\*\*

3047 \*\*\*\*\*  
TEST 16 RECALIBRATE COMMAND  
ALL POSSIBLE REGISTERS ARE FILLED WITH 0  
THEN A RECALIBRATE =6 COMMAND IS GIVEN  
NO REGISTERS SHOULD CHANGE EXCEPT RHCC=0

3053 \*\*\*\*\*

3210 \*\*\*\*\*  
TEST 17 UNLOAD COMMAND

3213 IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PE  
IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
THEN THIS TEST IS NOT PERFORMED  
IF NO ACT-11 MONITOR IS PRESENT  
THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS  
ON SUBSEQUENT PASSES THIS TEST IS NOT DONE

MAINDEC-11-DERPU-B  
DERFUS.P!!

MACY11 27(732) 17-SEP-76 13:23 PAGE 22

J02

909



910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965

ALL POSSIBLE REGISTERS ARE FILLED WITH ONES  
THEN AN UNLOAD COMMAND =2 IS GIVEN  
NO REGISTERS SHOULD CHANGE EXCEPT MOL SHOULD=0  
THEN THE DRIVE IS POWERED UP BY OPERATOR  
AND A PACK ACKNOWLEDGE COMMAND (ALREADY TESTED)  
SETS VV-IN RHDSI

3230 \*\*\*\*\*

3517 \*\*\*\*\*  
TEST 20 OFFSET AND RETURN TO CENTER LINE COMMAND

3520 THIS TESTS TWO COMMANDS (1)OFFSET (2)RETURN-TO-CENTER-LI  
ALL POSSIBLE REGISTERS ARE FILLED WITH ONES (EXCEPT RHOF  
AN OFFSET IS GIVEN  
ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET  
ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED  
THEN A RETURN-TO-CENTER-LINE IS GIVEN  
ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET  
AND RHOF SHOULD CLEAR (EXCEPT HCI,ECI,FMT22)  
ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED  
THE ABOVE PROCESS IS REPEATED FOR OFFSET REGISTER  
VALUE 1 TO 377

3532 \*\*\*\*\*

3836 \*\*\*\*\*  
TEST 21 OFFSET COMMAND

3839 THIS TEST WILL ONLY GIVE REPEATED OFFSETS

3841 \*\*\*\*\*

3901 \*\*\*\*\*  
TEST 22 WRITE/READ HEADER AND DATA

3904 WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WOR  
TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS 256 WORDS  
OF 0  
THEN READ HEADER AND DATA FOR ABOVE.  
WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
1000,0,0,0, AND 256 OF 0  
THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCE  
THE GO BIT, AND ALL THE REGISTERS ARE SAVED  
THEN GO 'S GIVEN FOR WRITE HEADER AND DATA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CH  
THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CH

NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED  
WITH ALL ONES, COMMAND IS LOADED INTO REGISTERS EXCEPT

MA:NDCC-11-DERPU-B  
DERPUB.P:1

MACY11 27(732) 17-SEP-76 13:23 PAGE 24

L02

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

GO BIT AND ALL REGISTERS ARE SAVED  
GO IS GIVEN FOR THE READ COMMAND

ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE  
THEN THE READ DATA IS COMPARED.

3926 \*\*\*\*\*

4232 \*\*\*\*\*  
TEST 23 READ DATA

4235 THIS TEST READS DATA WRITTEN BY THE PREVIOUS TEST  
THE WRITE FROM BUFFER IS FILLED WITH ALL OS  
THE COMMAND IS FILLED, THEN ALL REGISTERS SAVED  
THEN READ DATA COMMAND IS GIVEN  
ALL REGISTERS ARE COMPARED FOR PROPER VALUES  
READ DATA INTO 'READ INTO' BUFFER IS CHECKED

4242 \*\*\*\*\*

4378 \*\*\*\*\*  
TEST 24 WRITE/READ DATA

4381 THIS TEST GIVES A WRITE DATA COMMAND FRO CYLINDER 0  
TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF ALL ONES  
THIS SECTOR IS FORMATED BY PREVIOUS TEST  
THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN  
SAME CYLINDER, TRACK, SECTOR, KEYS  
WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH

4387 200 ONES AND 56 125252  
THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE  
ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER  
CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

THEN READ INTO BUFFER IS FILLED WITH 200 OF ZEROS  
AND 56 OF 377, WRITE FROM BUFFER IS FILLED WITH 200 ONES  
AND 56 OF 377.  
THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED  
GO IS GIVEN

ALL REGISTERS ARE CHECKED  
READ DATA IS CHECKED

4405 \*\*\*\*\*

MAINDEC-11-DERPU-B

DECDOC VER 00.04

1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074

4716

\*\*\*\*\*  
TEST 25 WRITE/READ DATA

4719

THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0  
TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 125252  
THIS SECTOR IS FORMATED BY PREVIOUS TEST  
THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN  
SAME CYLINDER, TRACK, SECTOR, KEYS  
WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
256 OF 125252  
THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE  
ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER  
CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

THEN READ INTO BUFFER IS FILLED WITH 256 OF ZEROS  
WRITE FROM BUFFER IS FILLED WITH  
256 OF 125252  
THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED  
GO IS GIVEN

ALL REGISTERS ARE CHECKED  
READ DATA IS CHECKED

4743

\*\*\*\*\*

5031

\*\*\*\*\*  
TEST 26 WRITE/READ DATA

5034

THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0  
TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525  
THIS SECTOR IS FORMATED BY PREVIOUS TEST  
THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN  
SAME CYLINDER, TRACK, SECTOR, KEYS  
WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH

5040

200 OF 052525 AND 56 OF 377  
THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE  
ALL REGISTERS ARE SAVED AND THEN GO IS TO WRITE DATA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER  
CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS  
AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200  
WORDS OF 52525 AND 56 WORDS OF 0  
THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED  
GO IS GIVEN



115

MAINDEC-11-DERPU-B

DECDOC VER 00.04

1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200

ALL REGISTER ARE CHECKED  
READ DATA IS CHECKED

5058 \*\*\*\*\*

5359 \*\*\*\*\*

TEST 27 WRITE/READ DATA USING UNIBUS B

5372

THIS TEST USES UNIBUS B IF CONNECTED TO THE RH  
IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFO  
THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0  
TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525  
THIS SECTOR IS FORMATED BY PREVIOUS TEST  
THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN  
SAME CYLINDER, TRACK, SECTOR, KEYS  
THESE COMMANDS USE UNIBUS B FOR DATA  
WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
200 OF 052525 AND 56 OF 377  
THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE  
ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER  
CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS  
AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200  
WORDS OF 52525 AND 56 WORDS OF 0  
THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED  
GO IS GIVEN

ALL REGISTER ARE CHECKED  
READ DATA IS CHECKED

5399 \*\*\*\*\*

5751 \*\*\*\*\*

TEST 30 IMPLIED SEARCH

5754

ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEARCH  
A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN ONE SECTO  
OF WORDS  
WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WOR  
TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS  
266 (4 HEADER 256 DATA=0 4 HEADER 2 DATA=1  
THEN READ HEADER AND DATA FOR ABOVE SECTOR ! ONLY  
WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
10000,0,0,0 AND 256 OF 0, 10000,1,0,0, AND 2 OF 1  
THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCE  
THE GO BIT, AND ALL THE REGISTERS ARE SAVED  
THEN GO IS GIVEN FOR WRITE HEADER AND DATA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CH

003

MAINDEC-11-DERPU-B  
DERPUB.P!!

MAY11 27(732) 17-SEP-76 13:23 PAGE 29

1132

MAINDEC-1.-DERPU-B

DECDOC VER 00.04

1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184

THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CH

NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED WITH 377, WRITE FROM BUFFER IS FILLED WITH 10000 1 0 0 0 AND 254 OF ZEROS COMMAND IS LOADED INTO REGISTERS EXCEPT GO AND IE THEN ALL REGISTERS ARE SAVED GO IS GIVEN FOR THE READ COMMAND 256 WORDS

ALL REGISTERS ARE CHECKED FOR ! .. CHANGE THEN THE READ DATA IS COMPARED

5780 \*\*\*\*\*

6134 \*\*\*\*\*  
TEST 31 IMPLIED SEEK

6137 ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEEK FROM 0 TO CYLINDER 1, A WRITE HEADER AND DATA IS GIVEN FOR MORE ONE SECTOR OF WORDS

WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD TRACK 18, SECTOR 21, KEYS=0, NUMBER OF WORDS 266 WORDS (4 HEADER, 256 DATA=1125, 4 HEADER 2 DATA = 2000 THEN READ HEADER AND DATA FOR ABOVE TRACK 0, SECTOR 0, WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH 10000 0 0 0 AND 256 OF 1125, 10001 0 0 0 AND 2 OF 2000 THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT THE GO BIT, AND ALL THE REGISTERS ARE SAVED THEN GO IS GIVEN FOR WRITE HEADER AND DATA

THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CH THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CH

NOW FOR READ COMMAND READ INTO BUFFER IS FILLED WITH ALL ONES, WRITE FROM BUFFER IS LOADED WITH 10001 0 0 0 AND 2 OF 2000, 254 OF ZEROS COMMAND IS LOADED INTO REGISTERS EXCEPT GO AND IE ALL REGISTERS ARE SAVED GO IS GIVEN FOR THE READ COMMAND

ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE THEN THE READ DATA IS COMPARED.

6165 \*\*\*\*\*

6524 \*\*\*\*\*  
TEST 32 SEEK TEST



MAINDEC-11-DERPU-B

DECD0C VER 00.04

1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240

6527

THIS TEST GETS THE HEADS OUT TO CYLINDER 10 NOT BY ONE  
SEEK BUT BY TEN IMPLIED SEEKS ONE CYLINDER AT A TIME

THIS TEST STARTS WITH A (ALREADY TESTED) RECALIBRATE  
THAT IS CYLINDER ZERO. THEN ON CYLINDER 0 SECTOR  
#21 TRACK #18 IT WRITES 266 WORDS THERE BY GETTING  
THE HEAD TO CYLINDER 1 THEN IT WRITES 266 WORDS ON  
CYLINDER 1 SECTOR #21 TRACK #18 THERE BY GETTING  
THE HEADS TO CYLINDER 2  
THIS IS REPEATED 10 TIMES GETTING THE  
HEADS TO CYLINDER 10  
THEN A SEEK COMMAND IS GIVEN TO CYLINDER 0  
AND DATA ALREADY WRITTEN IS CHECKED

6541

\*\*\*\*\*

7135

\*\*\*\*\*  
TEST 33 SEEK TEST

7139

THIS TEST DEPENDS ON HEADER AND DATA WRITTEN BY THE  
PREVIOUS TEST. AT THIS POINT THE HEADS ARE ON  
CYLINDER 0

ALL REGISTERS ARE SAVED  
A SEEK IS GIVEN TO CYLINDER 9  
ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE  
DATA WRITTEN ON CYLINDER 9 IS CHECKED

\*\*\*\*\*

7383

\*\*\*\*\*  
TEST 34 WRITE CHECK HEADER AND DATA

7386

WRITE CHECK HEADER AND DATA CYLINDER 5 FORMAT 16 BITS PE  
TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 266  
CONSISTING OF  
10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SE  
10 WORDS OF 177777  
10 WORDS OF 0  
10 WORDS OF 052525  
10 WORDS OF 125252  
16 WORDS OF LEFT ROTATING ZERO (EG 177776,177775)  
16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)  
174 WORDS OF 377  
4 WORDS OF HEADER  
2 WORDS OF 12345

FIRST THE ABOVE DATA IS WRITTEN BY A WRITE HEADER AND  
DATA COMMAND  
CHECK FOR NO ERRORS  
THEN THE ABOVE DATA IS READ BY A READ HEADER AND DATA  
CHECK FOR NO ERRORS  
THEN THE ABOVE WRITE CHECK HEADER AND DATA IS GIVEN

\*\*\*\*\*

G03

MA:NDCC-11-DERPU-B  
DERPUB.P:1

MACY11 27(732) 17-SEP-76 13:23 PAGE 32

1241

29 9/0

EX  
22

29 7

MAINDEC-11-DERPU-B

DECDOC VER 00.04

1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297

7843

\*\*\*\*\*  
 TEST 35 WRITE CHECK DATA  
 THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TE  
 WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS  
 TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258  
 CONSISTING OF  
 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SE  
 10 WORDS OF 177777  
 10 WORDS OF 0  
 10 WORDS OF 052525  
 10 WORD OF 125252  
 16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)  
 16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)  
 174 WORDS OF 377  
 2 WORDS OF 12345

FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER  
THEN THE ABOVE WRITE CHECK DATA IS GIVEN

9058

\*\*\*\*\*  
 TEST 36 WRITE CHECK DATA USING UNIBUS B  
 THIS TEST USES UNIBUS B IF CONNECTED TO THE RH  
 IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFO  
 THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TE  
 WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS  
 TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258  
 CONSISTING OF  
 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SE  
 10 WORDS OF 177777  
 10 WORDS OF 0  
 10 WORDS OF 052525  
 10 WORD OF 125252  
 16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)  
 16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)  
 174 WORDS OF 377  
 2 WORDS OF 12345

FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER  
THEN THE ABOVE WRITE CHECK DATA IS GIVEN

8085

\*\*\*\*\*

8310

\*\*\*\*\*  
 TEST 37 WRITE PROTECT BUTTON  
 IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PE  
  
 IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
 THEN THIS TEST IS NOT PERFORMED  
  
 IF NO ACT-11 MONITOR IS PRESENT  
 THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS

Handwritten signature or initials at the bottom right of the page.

8

1-  
34



MAINDEC-11-DERPU-B

DECDOC VER 00.04

1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340

ON SUBSEQUENT PASSES THIS TEST IS NOT DONE

WRITE FROM BUFFER IS FILLED WITH ALL ONES AND  
SECTOR 0, TRACK 0, CYLINDER 0 IS FILLED WITH  
ALL ONES

ALL REGISTERS ARE SAVED THEN WRITE LOCK BUTTON IS  
PRESSED AND ALL REGISTERS ARE CHECKED.

WRITE FROM BUFFER IS FILLED WITH 377 AND A WRITE IS  
ATTEMPTED TO SECTOR 0, TRACK 0, CYLINDER 0 70. WORDS  
ALL REGISTERS ARE CHECKED

THE SAME SECTOR IS READ AND DATA COMPARED TO SEE  
THAT NOTHING GOT DESTROYED (READ DATA SHOULD BE ALL  
ONES AND NCT 377)

THEN WRITE LOCK BUTTON IS PRESSED TO UNLOCK  
WRITE LOCKS AND ALL REGISTERS ARE COMPARED

8335 \*\*\*\*\*

8724 \*\*\*\*\*  
\*\*\*\*\*  
TEST 40 END OF DRIVE

8728 THIS IS THE END OF TEST FOR ONE DRIVE  
IF THERE ARE MORE DRIVES THEN THE PROGRAM  
JUMPS TO TEST 5 FOR NEXT DRIVE TEST  
END PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE

8733 \*\*\*\*\*

8769 \*\*\*\*\*

8771 \*\*\*\*\*  
END OF PASS ROUTINE  
\*\*\*\*\*

8773 INCREMENT THE PASS NUMBER (\$PASS)  
TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)  
IF THERES A MONITOR GO TO IT  
IF THERE ISN'T JUMP TO TSTi

MAINDEC-11-DERPU-B

DECDOC VER 00.04

1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384

8813

\*\*\*\*\*  
SUBROUTINES  
\*\*\*\*\*

9478

THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE  
ADDRESS FROM 176700 TO ANY TYPED VALUE

9545

\*\*\*\*\*  
\*\*\*\*\*

9548

\*\*\*\*\*  
SCOPE HANDLER ROUTINE  
\*\*\*\*\*

9550

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

SCOPE                    ::SCOPE=IOT

9609

\*\*\*\*\*

9611

\*\*\*\*\*  
CONVERT BINARY TO DECIMAL AND TYPE ROUTINE  
\*\*\*\*\*

9613

THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIG  
SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER  
NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE T  
BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS  
REPLACED WITH SPACES.  
CALL:

MOV        NUM,-(SP)            ::PUT THE BINARY NUMBER ON THE S  
TYPDS                            ::GO TO THE ROUTINE

9677

\*\*\*\*\*

1385  
1386  
1387  
1388  
1389  
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

```

9679 *****
TYPE ROUTINE
*****

9681 ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 9
THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE
NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
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
OR
   TYPE
   MESADR

2) USING A JSR INSTRUCTION
   MOV      PS,-(SP)      ;;PUSH PROCESSOR STATUS WORD ON
   JSR      PC,$TYPE      ;;CALL TYPE ROUTINE
   MESADDR      ;;FIRST ADDRESS OF MESSAGE

9724 *****

9726 *****
TTY INPUT ROUTINE
*****

9735 TK INITIALIZE ROUTINE
THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
CALL
   JSR      PC,$TKINT
   RETURN

9752 TK SERVICE ROUTINE
THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT

9774 *****
THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
CALL:
   RDCHR      ;;INPUT A SINGLE CHARACTER FROM
   RETURN HERE      ;;CHARACTER IS ON THE STACK

9794 *****
THIS ROUTINE WILL INPUT A STRING FROM THE TTY
CALL:
   RDLIN      ;;INPUT A STRING FROM THE TTY
   RETURN HERE      ;;ADDRESS OF FIRST CHARACTER WILL
   ;;TERMINATOR WILL BE A BYTE OF A

```

MAINDEC-11-DERPU-B

DECDOC VER 00.04

1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489

```

9830 *****
*****
9832 READ AN OCTAL NUMBER FROM THE TTY
*****
9834 THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
CHANGE IT TO BINARY.
THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPE
FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUS
THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RE
CALL:
          RDOCT                ;;READ AN OCTAL NUMBER
          RETURN HERE          ;;LOW ORDER BITS ARE ON TOP OF T
                               ;;HIGH ORDER BITS ARE IN $HIOCT
9884 *****
*****
9886 ERROR HANDLER ROUTINE
*****
9888 THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
AND GO TO $ERRTYP ON ERROR
THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
SW15=1 HALT ON ERROR
SW13=1 INHIBIT ERROR TYPEOUTS
SW10=1 BELL ON ERROR
SW09=1 LOOP ON ERROR
CALL          ERROR  N          ;;ERROR=EMT AND N=ERROR ITEM NUMBER
9933 *****
*****
9934 THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE
ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE
AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
9938 WITH ONLY MINOR CHANGES
FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
AND NOT THE ERROR MESSAGE AND HEADER.

```

MAINDEC-11-DERPU-B

DECDOC VER 00.04

1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532

10159 \*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

10163 \*\*\*\*\*  
BINARY TO OCTAL (ASCII) AND TYPE  
\*\*\*\*\*

10165 THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIG  
OCTAL (ASCII) NUMBER AND TYPE IT.  
\$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS  
CALL:  
MOV NUM,-(SP) ;:NUMBER TO BE TYPED  
TYPOS ;:CALL FOR TYPEOUT  
.BYTE N ;:N=1 TO 6 FOR NUMBER OF DIGITS  
.BYTE M ;:M=1 OR 0  
;:1=TYPE LEADING ZEROS  
;:0=SUPPRESS LEADING ZER

\$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE  
\$TYPOS OR \$TYPOC  
CALL:  
MOV NUM,-(SP) ;:NUMBER TO BE TYPED  
TYPON ;:CALL FOR TYPEOUT

\$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER  
CALL:  
MOV NUM,-(SP) ;:NUMBER TO BE TYPED  
TYPOC ;:CALL FOR TYPEOUT

10239 \*\*\*\*\*

10241 \*\*\*\*\*  
TRAP DECODER  
\*\*\*\*\*

10243 THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTIO  
AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDR  
OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL  
GO TO THAT ROUTINE.







```

1606 J00006
1607 000007
1608
1609
1610
1611
1612 000000
1613 000040
1614 000100
1615 000140
1616 000200
1617 000240
1618 000300
1619 000340
1620
1621 100000
1622 040000
1623 020000
1624 010000
1625 004000
1626 002000
1627 001000
1628 000400
1629 000200
1630 000100
1631 000040
1632 000020
1633 000010
1634 000004
1635 000002
1636 000001
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650 100000
1651 040000
1652 020000
1653 010000
1654 004000
1655 002000
1656 001000
1657 000400
1658 000200
1659 000100
1660 000040
1661 000020

```

```

R6= %6 :: GENERAL REGISTER
R7= %7 :: GENERAL REGISTER
.EQUIV R6,SP :: STACK POINTER
.EQUIV R7,PC :: 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
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

```

1662 000010  
1663 000004  
1664 000002  
1665 000001  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678 000004  
1679 000010  
1680 000014  
1681 000014  
1682 000014  
1683 000020  
1684 000024  
1685 000030  
1686 000034  
1687 000060  
1688 000054  
1689 000240  
1690  
1691  
1692  
1693  
1694 000000  
1695  
1696  
1697  
1698 000174  
1699 000174 000000  
1700 000176 000000  
1701  
1702  
1703 000200 000137 004712  
1704 000046  
1705 000046 034346  
1706 000052  
1707 000052 040000  
1708 000210  
1709 000210 000137 004676  
1710 000220  
1711 000220 000137 004662  
1712  
1713  
1714  
1715  
1716  
1717

BIT03= 10  
BIT02= 4  
BIT01= 2  
BIT00= 1  
.EQUIV BIT09,BIT9  
.EQUIV BIT08,BIT8  
.EQUIV BIT07,BIT7  
.EQUIV BIT06,BIT6  
.EQUIV BIT05,BIT5  
.EQUIV BIT04,BIT4  
.EQUIV BIT03,BIT3  
.EQUIV BIT02,BIT2  
.EQUIV BIT01,BIT1  
.EQUIV BIT00,BIT0

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

.SBTTL TRAP CATCHER

. =0  
:\*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"  
:\*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS  
:\*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

. =174  
DISPREG: .WORD 0 ;: SOFTWARE DISPLAY REGISTER  
SWREG: .WORD 0 ;: SOFTWARE SWITCH REGISTER

.SBTTL STARTING ADDRESS(ES)

JMP @#BEGIN ;: JUMP TO STARTING ADDRESS OF PROGRAM  
. =46  
\$ENDAD  
. =52  
40000  
. =210  
JMP @#BEGIN2 ; JUMP SELECT TEST  
. =220  
JMP @#BEGIN1 ; JUMP TO NO OPERATOR TESTS

.\*STARTING ADDRESS 200 FOR NORMAL STARTS  
.\*THIS WILL TEST ALL RPO4'S ON THE SYSTEM A SINGLE DRIVE AT A TIME  
.\*  
.\*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE  
.\*  
.\*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR

1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
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

000250

177572  
177574  
177576  
172516

172300  
172302  
172304  
172306  
172310  
172312  
172314  
172316

172340  
172342  
172344  
172346  
172350  
172352  
172354  
172356

001110

;\*AT THE DRIVE

.SBTTL MEMORY MANAGEMENT DEFINITIONS

;\*KT11 VECTOR ADDRESS

MMVEC= 250

;\*KT11 STATUS REGISTER ADDRESSES

SR0= 177572  
SR1= 177574  
SR2= 177576  
SR3= 172516

;\*KERNAL "I" PAGE DESCRIPTOR REGISTERS

KIPDR0= 172300  
KIPDR1= 172302  
KIPDR2= 172304  
KIPDR3= 172306  
KIPDR4= 172310  
KIPDR5= 172312  
KIPDR6= 172314  
KIPDR7= 172316

;\*KERNAL "I" PAGE ADDRESS REGISTERS

KIPAR0= 172340  
KIPAR1= 172342  
KIPAR2= 172344  
KIPAR3= 172346  
KIPAR4= 172350  
KIPAR5= 172352  
KIPAR6= 172354  
KIPAR7= 172356

\*\*\*\*\*  
.=1110

1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764 001100  
1765 001100 000000  
1766 001100 000000  
1767 001102 000  
1768 001103 000  
1769 001104 000000  
1770 001106 000000  
1771 001110 000000  
1772 001112 000000  
1773 001114 000  
1774 001115 001  
1775 001116 000000  
1776 001120 000000  
1777 001122 000000  
1778 001124 000000  
1779 001126 000000  
1780 001130 000000  
1781 001132 000000  
1782 001134 000000  
1783 001136 177570  
1784 001140 177570  
1785 001142 177560  
1786 001144 177562  
1787 001146 177564  
1788 001150 177566  
1789 001152 000  
1790 001153 002  
1791 001154 012  
1792 001155 000  
1793 001156 000000  
1794  
1795 001160 000000  
1796 001162 000000  
1797 001164 000000  
1798 001166 000000  
1799 001170 000000  
1800 001172 000000  
1801 001174 000000  
1802 001176 000000  
1803 001200 000000  
1804 001202 000000  
1805 001204 000000  
1806 001206 000000  
1807 001210 000000  
1808 001212 000000  
1809 001214 177607 000377  
1810 001220 077  
1811 001221 015  
1812 001222 000012

:\*\*\*\*\*

.SBTTL COMMON TAGS

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

.=1100

\$CMTAG: .WORD 0  
\$PASS: .WORD 0  
\$STNM: .BYTE 00  
\$ERFLG: .BYTE 00  
\$ICNT: .WORD 00  
\$LPADR: .WORD 00  
\$LPERR: .WORD 00  
\$ERTTL: .WORD 00  
\$ITEMB: .BYTE 0  
\$ERMAX: .BYTE 1  
\$ERRPC: .WORD 0  
\$GDADR: .WORD 00  
\$BDADR: .WORD 00  
\$GDDAT: .WORD 00  
\$BDDAT: .WORD 00  
\$SWR: .WORD DSWR  
\$DISPLAY: .WORD DDISP  
\$TKS: 177560  
\$TKB: 177562  
\$TPS: 177564  
\$TPB: 177566  
\$NULL: .BYTE 0  
\$FILLS: .BYTE 2  
\$FILLC: .BYTE 12  
\$TPFLG: .BYTE 0  
\$REGAD: .WORD 0  
\$REGO: .WORD 0  
\$REG1: .WORD 0  
\$REG2: .WORD 0  
\$REG3: .WORD 0  
\$REG4: .WORD 0  
\$REG5: .WORD 0  
\$TMP0: .WORD 0  
\$TMP1: .WORD 0  
\$TMP2: .WORD 0  
\$TMP3: .WORD 0  
\$TMP4: .WORD 0  
\$TMP5: .WORD 0  
\$TIMES: 0  
\$ESCAPE: 0  
\$BELL: .ASCIZ <207><377><377>  
\$QUES: .ASCII /?/  
\$CRLF: .ASCII <15>  
\$LF: .ASCIZ <12>

:: START OF COMMON TAGS  
:: CONTAINS PASS COUNT  
:: CONTAINS THE TEST NUMBER  
:: CONTAINS ERROR FLAG  
:: CONTAINS SUBTEST ITERATION COUNT  
:: CONTAINS SCOPE LOOP ADDRESS  
:: CONTAINS SCOPE RETURN FOR ERRORS  
:: CONTAINS TOTAL ERRORS DETECTED  
:: CONTAINS ITEM CONTROL BYTE  
:: CONTAINS MAX. ERRORS PER TEST  
:: CONTAINS PC OF LAST ERROR INSTRUCTION  
:: CONTAINS ADDRESS OF 'GOOD' DATA  
:: CON INS ADDRESS OF 'BAD' DATA  
:: CONTAINS 'GOOD' DATA  
:: CONTAINS 'BAD' DATA  
:: RESERVED--NOT TO BE USED  
:: ADDRESS OF SWITCH REGISTER  
:: ADDRESS OF DISPLAY REGISTER  
:: TTY KBD STATUS  
:: TTY KBD BUFFER  
:: TTY PRINTER STATUS REG. ADDRESS  
:: TTY PRINTER BUFFER REG. ADDRESS  
:: CONTAINS NULL CHARACTER FOR FILLS  
:: CONTAINS # OF FILLER CHARACTERS REQUIRED  
:: INSERT FILL CHARS. AFTER A "LINE FEED"  
:: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)  
:: CONTAINS THE ADDRESS FROM WHICH (\$REGO) WAS OBTAINED  
:: CONTAINS ((\$REGAD)+0)  
:: CONTAINS ((\$REGAD)+2)  
:: CONTAINS ((\$REGAD)+4)  
:: CONTAINS ((\$REGAD)+6)  
:: CONTAINS ((\$REGAD)+10)  
:: CONTAINS ((\$REGAD)+12)  
:: USER DEFINED  
:: USER DEFINED  
:: USER DEFINED  
:: USER DEFINED  
:: USER DEFINED  
:: USER DEFINED  
:: MAX. NUMBER OF ITERATIONS  
:: ESCAPE ON ERROR ADDRESS  
:: CODE FOR BELL  
:: QUESTION MARK  
:: CARRIAGE RETURN  
:: LINE FEED



1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868

001224  
  
  
  
  
001224 043172  
001226 060100  
  
  
  
001230 062022  
001232 062262  
  
001234 043221  
001236 060100  
  
  
001240 062022  
001242 062262  
  
001244 043310  
001246 060100  
  
001250 062022

```
*****
.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:

*****
;
; ITEM1
      EM1      :RPO4 DID NOT INTERRUPT
              :WAITED ON BIT DID NOT OCCUR
              :PC
              :WAT PC
              :BIT WAITED
              :REG ADDRESS
              :REG CONTENTS
              :RHCSI CONTENTS
              :$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CSI
              :0,0,0,0,0,0
      DF1
; ITEM2
      EM2      : INTERRUPT ENABLE BIT DOWN BUT
              : WAITED ON BIT DID NOT OCCUR
              : PC
              : WAT PC
              : BIT WAITED
              : REG ADDRESS
              : REG CONTENTS
              : RHCSI CONTENTS
              : $ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CSI
              : 0,0,0,0,0,0
      DF1
; ITEM3
      EM3      :RPO4 DID NOT INTERRUPT WHEN
              :WAITED ON BIT DID SET
              :PC
              :WAT PC
              :BIT WAITED
              :REG ADDRESS
              :RHCSI CONTENTS
              :$ERRPC,WAITPC,WAITBT,WAITRE,$BDDAT,CSI
      DT1
```

1859	001252	062262	DF1		;0,0,0,0,0,0
1870					
1871				:ITEM4	
1872	001254	043371	EM4		;WAITED ON BIT DID SET BUT ;TIME IS IN ERROR ;TIME IS GIVEN IN 10 MICRO SEC. ;(DECIMAL)
1873					
1874					
1875					
1876	001256	060240	DH4		;PC ;WAT PC ;BIT WAITED ;REG ADDRESS ;TIME IN 10 MSEC ;\$ERRPC, WAITPC, WAITBT, WAITRE, \$BDDAT, WAITIM
1877					
1878					
1879					
1880					
1881	001250	062040	DT4		
1882	001262	062270	DF4		;0,0,0,0,0,1
1883					
1884				:ITEM5	
1885	001264	043500	EM5		;RHAS DOES NOT CLEAR BY ;MOVING IN ALL ONES
1886					
1887	001266	060363	DH5		;PC ;REG. ADDR. ;GOOD DATA ;RECEIVED DATA ;\$ERRPC, REGADR, \$GDDAT, \$BDDAT
1888					
1889					
1890					
1891	001270	062056	DT5		
1892	001272	062276	DF5		;0,0,0,0
1893					
1894				:ITEM6	
1895	001274	043552	EM6		;LOADING RHER1 FOR ALL ;UNITS DID NOT SET ANY BITS ;IN RHAS-NO UNITS PRESENT
1896					
1897					
1898	001276	060462	DH6		;PC ;REG ADDR ;RECEIVED DATA ;\$ERRPC, REGADR, \$BDDAT
1899					
1900					
1901	001300	062070	DT6		
1902	001302	062302	DF6		;0,0,0
1903					
1904				:ITEM7	
1905	001304	043640	EM7		;SPECIFIED REGISTER NONEXISTANT ;SO ABORT PROGRAM
1906					
1907	001306	060541	DH7		;PC ;ADDR. OF REG. ;\$ERRPC, TEMP1
1908					
1909	001310	062100	DT7		
1910	001312	062305	DF7		;0,0
1911					
1912				:ITEM10	
1913	001314	043722	EM10		;STOPED DRIVE HAS MOL BIT ;IN RHDS1 = 1
1914					
1915	001316	060571	DH10		;PC ;TEST NO ;FAILING REG ADDR ;CONTENTS OF RHCS1 ;CONTENTS OF RHCS2 ;CONTENTS OF RHDS1 ;CONTENTS OF RHER1
1916					
1917					
1918					
1919					
1920					
1921					
1922	001320	062106	DT10		\$ERRPC, \$STNM, \$BDADR, CS1, CS2, DS1, ER1
1923	001322	062307	DF10		;0,0,0,0,0,0,0
1924					

1925			:ITEM11		
1926	001324	043771		EM11	:WITH SPINDLE POWERED DOWN
1927					:RHCS2 SHOULD HAVE ONL?
1928					:UNIT NUMBER AND IR HIGH
1929	001226	060571		DH10	:PC
1930					:TEST NO
1931					:FAILING REG. ADR
1932					:CONTENTS OF RHCS1
1933					:CONTENTS OF RHCS2
1934					:CONTENTS OF RHDS1
1935					:CONTENTS OF RHER1
1936	001330	062106		DT10	:SERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
1937	001332	062307		DF10	:0,0,0,0,0,0,0
1938					
1939			:ITEM12		
1940	001334	044076		EM12	:AFTER A POWER UP WITH
1941					:NO PACK ACKNOWLEDGE COMMAND
1942					:RHDS1 SHOULD HAVE MOL=1, VV=0
1943	001336	060571		DH10	:PC
1944					:TEST NO
1945					:FAILING REGISTER ADDR.
1946					:CONTENTS OF RHCS1
1947					:CONTENTS OF RHCS2
1948					:CONTENTS OF RHDS1
1949					:CONTENTS OF RHER1
1950	001340	062106		DT10	:SERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
1951	001342	062307		DF10	:0,0,0,0,0,0,0
1952					
1953			:ITEM13		
1954	001344	044204		EM13	:AFTER A POWER UP WITHOUT
1955					:ANY INIT RHCS1 SHOULD
1956					:HAVE GO=0, DVA=1, RDY=1
1957					:IE=0, DISREGARD
1958					:ALL OTHER BITS
1959	001346	060571		DH10	:PC
1960					:TEST NO
1961					:FAILING REGISTER ADDR.
1962					:CONTENTS OF RHCS1
1963					:CONTENTS OF RHCS2
1964					:CONTENTS OF RHDS1
1965					:CONTENTS OF RHER1
1966	001350	062106		DT10	:SERRPC,\$TSTNM,\$BDADR,CS1,CS2,DS1,ER1
1967	001352	062307		DF10	:0,0,0,0,0,0,0
1968					
1969			:ITEM14		
1970	001354	044323		EM14	:AFTER POWER UP RHCC
1971					:SHOULD BE=0
1972	001356	060363		DH5	:PC
1973					:REG. ADDR.
1974					:GOOD DATA
1975					:RECEIVED DATA
1976	001360	062056		DT5	:SERRPC,REGADR,\$GDDAT,\$BDDAT
1977	001362	062276		DF5	:0,0,0,0
1978					
1979			:ITEM15		
1980	001364	044375		EM15	:PACK ACKNOWLEDGE CAUSED

1981					; AN ERROR
1982					; GOOD DATA IS BEFORE COMMAND
1983					; RECEIVED DATA IS AFTER COMMAND
1984	001366	060363	DHS		; PC
1985					; REG. ADDR.
1986					; GOOD DATA
1987					; RECEIVED DATA
1988	001370	062056	DTS		; \$ERRPC, REGADR, \$GDDAT, \$BDDAT
1989	001372	062276	DFS		; 0,0,0,0
1990					
1991				: ITEM16	
1992	001374	044536	EM16		; GIVING A NO-OP COMMAND CAUSED
1993					; AN ERROR
1994					; GOOD DATA GIVES REGISTER
1995					; CONTENTS BEFORE COMMAND
1996					; RECEIVED DATA GIVES REGISTER
1997					; CONTENTS AFTER COMMAND
1998	001376	060363	DHS		; PC
1999					; REG. ADDR.
2000					; GOOD DATA
2001					; RECEIVED DATA
2002	001400	062056	DTS		; \$ERRPC, REGADR, \$GDDAT, \$BDDAT
2003	001402	062276	DFS		; 0,0,0,0
2004					
2005				: ITEM17	
2006	001404	044664	EM17		; DRIVE CLEAR COMMAND
2007					; CAUSED AN ERROR
2008					; GOOD DATA GIVES WHAT SHOULD
2009					; BE THERE
2010					; RECEIVED DATA GIVES WHAT WAS
2011					; THERE AFTER COMMAND
2012	001406	060363	DHS		; PC
2013					; REG. ADDR.
2014					; GOOD DATA
2015					; RECEIVED DATA
2016	001410	062056	DTS		; \$ERRPC, REGADR, \$GDDAT, \$BDDAT
2017	001412	062276	DFS		; 0,0,0,0
2018					
2019				: ITEM20	
2020	001414	045021	EM20		; READ-IN COMMAND GAVE AN ERROR
2021					; GOOD DATA HAS WHAT SHOULD BE THERE
2022					; RECEIVED DATA HAS WHAT WAS
2023					; AFTER COMMAND
2024	001416	060363	DHS		; PC
2025					; REG. ADDR.
2026					; GOOD DATA
2027					; RECEIVED DATA
2028	001420	062056	DTS		; \$ERRPC, REGADR, \$GDDAT, \$BDDAT
2029	001422	062276	DFS		; 0,0,0,0
2030					
2031					
2032				: ITEM 21	
2033	001424	045170	EM21		; RHCSI CONTENTS DURING
2034					; COMMAND WAS IN ERROR
2035	001426	060363	DHS		
2036	001430	062056	DTS		

2037	001432	062276	DFS	
2038				
2039				
2040	001434	045243	: ITEM 22 EM22	: RHDS1 CONTENTS DURING : COMM ANS WAS IN ERROR
2041				
2042	001436	060363	DHS	
2043	001440	062056	DTS	
2044	001442	062276	DFS	
2045				
2046				
2047	001444	045316	: ITEM 23 EM23	: UNLOAD COMMAND GAVE AN ERROR : GOOD DATA GIVES WHAT SHOULD : BE THERE : RECEIVED DATA GIVES WHAT WAS : THERE AFTER COMMAND
2048				
2049				
2050				
2051				
2052	001446	060363	DHS	
2053	001450	062056	DTS	
2054	001452	062276	DFS	
2055				
2056				
2057	001454	045465	: ITEM 24 EM24	: OFFSET COMMAND CAUSED AN ERROR : GOOD DATA IS WHAT SHOULD BE THERE : RECEIVED DATA GIVES WHAT WAS THERE : AFTER AN OFFSET COMMAND
2058				
2059				
2060				
2061	001456	060363	DHS	
2062	001460	062056	DTS	
2063	001462	062276	DFS	
2064				
2065				
2066	001464	045630	: ITEM 25 EM25	: RETURN TO CENTER LINE COMMAND : CAUSED AN ERROR : GOOD DATA GIVES WHAT SHOULD BE : THERE : RECEIVED DATA GIVES WHAT WAS : THERE AFTER COMMAND
2067				
2068				
2069				
2070				
2071				
2072	001466	060363	DHS	
2073	001470	062056	DTS	
2074	001472	062276	DFS	
2075				
2076				
2077	001474	046012	: ITEM 26 EM26	: 500 OFFSETS CAUSED AN ERROR
2078	001476	060750	DH26	: PC : CONT. OF RHCS1 : CONT. OF RHCS2 : CONT. OF RHDS1 : CONT. OF RHER1 : CONT. OF RHER2 : CONT. OF RHER3 : \$ERRPC, CS1, CS2, DS1, ER1, ER2, ER3 : 0,0,0,0,0,0,0
2079				
2080				
2081				
2082				
2083				
2084				
2085	001500	062126	DT26	
2086	001502	062316	DF26	
2087				
2088				
2089	001504	046102	: ITEM 27 EM27	: WRITE HEADER AND DATA : CAUSED IMPROPER REGISTER CHANGE : GOOD DATA GIVES WHAT : SHOULD BE THERE
2090				
2091				
2092				

2093					:RECEIVED DATA GIVES WHAT
2094					:WAS THERE AFTER COMMAND
2095	001506	060363		DH5	
2096	001510	062056		DT5	
2097	001512	062276		DF5	
2098					
2099					
2100	001514	046320		EM30	:ITEM 30
2101					:WRITE HEADER AND DATA
2102	001516	061127		DH30	:CHANGED WRITE FROM BUFFER
2103					:PC
2104					:WORD NO
2105					:GOOD DATA
2106	001520	062146		DT30	:BAD DATA
2107	001522	062325		DF30	:SERRFC, ERWORD, \$GDDAT, \$BDDAT
2108					:0,0,0,0
2109					
2110	001524	046400		EM31	:ITEM 31
2111					:READ HEADER AND DATA CAUSED
2112					:IMPROPER REGISTER CHANGE
2113					:GOOD DATA HAS WHAT SHOULD
2114					:BE THERE
2115					:RECEIVED DATA GIVES WHAT
2116	001526	060363		DH5	:WAS THERE AFTER COMMAND
2117	001530	062056		DT5	
2118	001532	062276		DF5	
2119					
2120					
2121	001534	046615		EM32	:ITEM 32
2122					:WRITE HEADER AND DATA FOLLOWED
2123					:BY A READ HEADER AND DATA
2124	001536	061127		DH30	:CAUSED A READ/WRITE ERROR
2125	001540	062146		DT30	
2126	001542	062325		DF30	
2127					
2128					
2129	001544	046722		EM33	:ITEM 33
2130					:READ DATA CAUSED IMPROPER REGISTER
2131					:CHANGE
2132					:GOOD DATA GIVES WHAT SHOULD BE THERE
2133					:RECEIVED DATA GIVES WHAT WAS THERE AFTER
2134	001546	060363		DH5	:COMMAND
2135	001550	062056		DT5	
2136	001552	062276		DF5	
2137					
2138					
2139	001554	047124		EM34	:ITEM 34
2140	001556	061127		DH30	:READ DATA INCORRECT
2141	001560	062146		DT30	
2142	001562	062325		DF30	
2143					
2144					
2145	001564	047150		EM35	:ITEM 35
2146					:WRITE DATA COMMAND CAUSED
2147					:IMPROPER REGISTER CHANGE
2148					:GOOD DATA GIVES WHAT SHOULD BE THERE
					:RECEIVED DATA GIVES REGISTER



2149					;CONTENTS AFTER WRITE DATA
2150	001566	060363		DH5	
2151	001570	062056		DTS	
2152	001572	062276		DF5	
2153					
2154					
2155	001574	047366	;ITEM 36	EM36	;WRITE DATA COMMAND CHANGED
2156					;WRITE FROM BUFFER
2157	001576	061127		DH30	
2158	001600	062146		DT30	
2159	001602	062325		DF30	
2160					
2161			;ITEM 37		
2162	001604	047443		EM37	;SEEK COMMAND CAUSED AN
2163					;ERROR
2164					;GOOD DATA GIVES WHAT SHOULD
2165					;BE THERE
2166					;RECEIVED DATA GIVES WHAT
2167					;WAS THERE AFTER SEEK COMMAND
2168	001606	060363		DH5	
2169	001610	062056		DTS	
2170	001612	062276		DF5	
2171					
2172			;ITEM 40		
2173	001614	047660		EM40	;WRITE CHECK CAUSED AN
2174					;IMPROPER REGISTER CHANGE
2175					;GOOD DATA GIVES WHAT SHOULD
2176					;BE THERE
2177					;RECEIVED DATA GIVES WHAT WAS
2178					;THERE AFTER COMMAND
2179	001616	060363		DH5	
2180	001620	062056		DTS	
2181	001622	062276		DF5	
2182					
2183			;ITEM 41		
2184	001624	050067		EM41	;LOCKING OUT WRITES BY WRITE
2185					;LOCK BUTTON CAUSED IMPROPER
2186					;REGISTER CHANGE
2187					;GOOD DATA GIVES WHAT SHOULD
2188					;BE THERE
2189					;RECEIVED DATA GIVES WHAT
2190					;WAS THERE AFTER WRITES
2191					;WERE LOCKED OUT BY
2192					;BUTTON
2193	001626	060363		DH5	
2194	001630	062056		DTS	
2195	001632	062276		DF5	
2196					
2197			;ITEM 42		
2198	001634	050350		EM42	;ATTEMPTING TO WRITE WITH WRITE
2199					;LOCKED OUT CAUSED IMPROPER
2200					;REGISTER CHANGE
2201					;GOOD DATA GIVES WHAT SHOULD
2202					;BE THERE
2203					;RECEIVED DATA GIVES WHAT WAS
2204					;THERE AFTER ATTEMPT

001656	050363	DH30	
001657	050364	DT3C	
001658	050365	DF30	
001674	050626		: ITEM 43
			EM43
001676	051127	DH30	
001700	052146	DT3C	
001702	052325	DF30	
001704	052065		: ITEM 47
			EM47

```

: WRITING WITH WRITE LOCKED
: BUT CHANGED DISK DATA
: GOOD DATA GIVES WHAT WAS
: ON DISK BEFORE WRITE LOCK
: WRITE LOCK WAS ATTEMPTED
: RECEIVED DATA GIVES WHAT WAS
: THERE AFTER WRITE LOCK
: WRITE LOCK WAS ATTEMPTED

```

```

: ENABLING WRITES BY WRITE LOCK
: BUTTON CAUSED AN ERROR
: GOOD DATA GIVES WHAT SHOULD
: BE THERE
: RECEIVED DATA GIVES WHAT WAS
: THERE AFTER WRITE LOCK
: BUTTON ENABLED WRITES

```

```

: TRANSFERRING ON LAST BLOCK IE. CYLINDER
: 410, SECTOR 21, TRACK 19
: CAUSED IMPROPER REGISTER
: CHANGE
: GOOD DATA GIVES WHAT SHOULD
: BE THERE
: RECEIVED DATA GIVES WHAT WAS
: THERE AFTER TRANSFER

```

```

: DATA READ FROM LAST
: BLOCK IE. CYLINDER 410
: SECTOR 21, TRACK 18 IS IN
: ERROR

```

```

: TRANSFERRING FROM NONEXISTANT
: SECTOR CAUSED IMPROPER
: REGISTER CHANGE
: GOOD DATA GIVES WHAT SHOULD

```

2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354  
2355  
2356  
2357  
2358  
2359  
2360  
2361  
2362  
2363  
2364  
2365  
2366  
2367  
2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400

001706 060363  
001710 062056  
001712 062276

DH5  
DT5  
DF5

:BE THERE  
:RECEIVED DATA GIVES WHAT WAS  
:THERE AFTER ATTEMPTED  
:TRANSFER

001714 052347

:ITEM 50  
EM50

:TRANSFERRING FROM NONEXISTANT  
:SECTOR CAUSED DATA ERROR  
:GOOD DATA GIVES WHAT  
:SHOULD BE IN BUFFER  
:RECEIVED DATA GIVES WHAT WAS  
:IN BUFFER AFTER TRANSFER

001716 061127  
001720 062146  
001722 062325

DH30  
DT30  
DF30

:ITEM 51  
EM51

:GIVING ILLEGAL FUNCTION CAUSED  
:IMPROPER REGISTER CHANGE  
:GOOD DATA GIVES WHAT SHOULD BE  
:THERE  
:RECEIVED DATA GIVES REGISTER  
:CONTENTS AFTER ILLEGAL FUNCTION

001724 052566

001726 061221

DH51

:PC  
:REG. ADDR.  
:GOOD DATA  
:RECEIVED DATA  
:ILLEGAL FUNCTION  
:SERRPC,REGADR,SGDDAT,SBDDAT,ILLEG  
:0,0,0,0,0

001730 062160  
001732 062331

DT51  
DF51

:ITEM 52  
EM52

:WRITE DATA ON NONEXISTANT  
:SECTOR CAUSED IMPROPER  
:REGISTER CHANGE  
:GOOD DATA GIVES WHAT SHOULD  
:BE THERE  
:RECEIVED DATA GIVES WHAT  
:WAS THERE AFTER ATTEMPTED  
:WRITE DATA

001736 060363  
001740 062056  
001742 062276

DH5  
DT5  
DF5

:ITEM 53  
EM53

:READ HEADER AND DATA AFTER  
:A SEARCH CAUSED AN ERROR

001744 053304  
001746 061127  
001750 062146  
001752 062325

DH30  
DT30  
DF30

:ITEM 54

237

2317	001754	053372	EMS4	: ATTEMPTED OPERATION WITH
2318				: INVALID ADDRESS CAUSED
2319				: IMPROPER REGISTER CHANGE
2320				: GOOD DATA GIVES WHAT SHOULD
2321				: BE THERE
2322				: RECEIVED DATA GIVES WHAT WAS
2323				: THERE AFTER OPERATION
2324	001756	060363	DH5	
2325	001760	062056	DT5	
2326	001762	062276	DF5	
2327				
2328				
2329				
2330				
2331				
2332				
2333				
2334				
2335				
2336				
2337	001766	060363	DH5	
2338	001770	062056	DT5	
2339	001772	062276	DF5	
2340				
2341				
2342				
2343				
2344				
2345				
2346				
2347				
2348	001776	061127	DH30	
2349	002000	062146	DT30	
2350	002002	062325	DF30	
2351				
2352				
2353				
2354				
2355				
2356				
2357				
2358				
2359				
2360				
2361	002006	060363	DH5	
2362	002010	062056	DT5	
2363	002012	062276	DF5	
2364				
2365				
2366				
2367	002014	054627	EM60	
2368				
2369				
2370				
2371				
2372				

: ITEM 55  
EM55

: WRITING/READING WITH EXPECTED  
: ADDRESS OVERFLOW ERROR CAUSED  
: IMPROPER REGISTER CHANGE  
: GOOD DATA GIVES WHAT SHOULD  
: BE THERE  
: RECEIVED DATA GIVES WHAT  
: WAS THERE AFTER OPERATION

: ITEM 56  
EM56

: DATA READ WITH AN EXPECTED  
: ADDRESS OVERFLOW ERROR IS  
: INCORRECT  
: WORD NO 1 TO 260 SHOULD  
: BE READ  
: WORD NOS 261 TO 266 SHOULD  
: NOT CHANGE DUE TO READ

: ITEM 57  
EM57

: ATTEMPTING DATA COMMAND  
: WITH WRONG FORMAT BIT CAUSED  
: IMPROPER REGISTER CHANGE  
: GOOD DATA GIVES WHAT SHOULD BE  
: THERE  
: RECEIVED DATA GIVES WHAT WAS  
: THERE AFTER ATTEMPTED DATA  
: TRANSFER

: ITEM 60  
EM60

: ATTEMPTING TO MODIFY REGISTER  
: DURING AN OPERATION CAUSED  
: IMPROPER REGISTER CHANGE  
: GOOD DATA GIVES WHAT SHOULD  
: BE THERE  
: RECEIVED DATA GIVES WHAT WAS  
: THERE AFTER OPERATION

237	002016	061340			: WAS COMPLETE
238				DH60	: PC
239					: REG. ADDR.
240					: GOOD DATA
241					: RECEIVED DATA
242					: MODIFYING REGISTER
243	002020	062174		DT60	: \$ERRPC, REGADR, \$GDDAT, \$BDDAT, \$BDADR
244	002022	062336		DF60	: 0,0,0,0
245					
246					: ITEM 61
247	002024	055236		EM61	: DEVICE NOT AVAILBLE BEFOR COMMAND WAS TO BE GIVEN
248	002026	061455		DH61	: PC
249					: PC OF JSR
250					: RHDS1
251	002030	062210		DT61	: \$ERRPC, PCJSR, \$BDADR
252	002032	062343		DF61	: 0,0,0
253					
254					: ITEM 62
255	002034	055236		EM61	: DEVICE NOT AVAILBLE BEFOR COMMAND WAS TO BE GIVEN
256	002036	061540		DH62	: PC
257					: PC OF JSR
258					: RHCSI WAS
259	002040	062220		DT62	: \$ERRPC, PCJSR, \$BDADR
260	002042	062346		DF62	: 0,0,0
261					
262					: ITEM 63
263	002044	055322		EM63	: RHDS1 CONTENTS DURING
264					: COMMAND WAS IN ERROR
265	002046	060363		DH5	
266	002050	062056		DT5	
267	002052	062276		DF5	
268					
269					: ITEM 64
270	002054	055375		EM64	: RECALIBRATE COMMAND CAUSED
271					: IMPROPER REGISTER CHANGE.
272					: GOOD DATA GIVES WHAT SHOULD BE
273					: THERE.
274					: RECEIVED DATA GIVES WHAT WAS THERE
275					: AFTER COMMAND
276	002056	060363		DH5	
277	002060	062056		DT5	
278	002062	062276		DF5	
279					
280					: ITEM 65
281	002064	055614		EM65	: INTERRUPT FAILING
282	002066	061613		DH65	: PC
283					: TEST NO
284					: CONTENTS OF RHCSI
285					: CONTENTS OF RHAS
286					: CONTENTS OF RHDS1
287	002070	062230		DT65	: \$ERRPC, TSTNM, CS1, AS, DS1
288	002072	062351		DF65	: 0,0,0,0,0

2429  
2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483  
2484

002074	055636	: ITEM66	EM66
002076	061734		DH66
002100	062244		DT66
002102	062356		DF66
002104	056047	: ITEM67	EM67
002106	061127		DH30
002110	062146		DT30
002112	062325		DF30
002114	056323	: ITEM70	EM70
002116	061127		DH30
002120	062146		DT30
002122	062325		DF30
002124	056531	: ITEM 71	EM71
002126	060363		DH5
002130	062056		DT5
002132	062276		DF5
002134	057274	: ITEM 74	EM74

```

: HEADER AND DATA COMMAND
: FOR HEAD SELECTION TEST
: CAUSED AN ERROR
: RHDST GIVES WHAT TRACK
: WAS BEING WRITTEN ON CYLINDER 0
: SECTOR 0
: PC
: RHDST
: RHER1
: RHER2
: RHER3
: RHCS1
: RHCS2
: $ERRPC, DST, ER1, ER2, ER3, CS1, CS2
: 0,0,0,0,0,0,0

: READ HEADER AND DATA ERROR
: IN HEAD SELECTION TEST
: FIRST FOUR WORDS GIVE HEAD EP
: NEXT WORDS ARE DATA
: GOOD DATA WORDS GIVE
: THE TRACK NUMBER IN
: BITS 4,5,6,7,8

: READ HEADER AND DATA ERROR
: IN DIFFERENCE LINE TEST
: WORD NOS. 1-4 GIVE
: HEADER
: WORD NOS. 5-260 GIVE DATA
: WHICH IS THE CYLINDER
: ADDRESS

: FORCIN OPI CAUSED IMPROPER REGISTER
: CHANGE
: GOOD DATA GIVES WHAT SHOULD
: BE THERE
: RECEIVED DATA GIVES WHAT WAS
: THERE AFTER 3 INDEX PULSES
: PC
: REG. ADDR.
: GOOD DATA
: RECEIVED DATA
: $ERRPC, REGADR, $GDDAT, $BDDAT
: 0,0,0,0

: WHILE USING UNIBUS B

```

2485					: READ DATA CAUSED IMPROPER REGISTER
2486					: CHANGE
2487					: GOOD DATA GIVES WHAT SHOULD BE THERE
2488					: RECEIVED DATA GIVES WHAT WAS THERE AFTER
2489					: COMMAND
2490	002136	060363		DH5	
2491	002140	062056		DT5	
2492	002142	062276		DF5	
2493					
2494			: ITEM 73		
2495	002144	057222		EM73	: WHILE USING UNIBUS B
2496					: READ DATA INCORRECT
2497	002146	061127		DH30	
2498	002150	062146		DT30	
2499	002152	062325		DF30	
2500					
2501			: ITEM 74		
2502	002154	057274		EM74	: WHILE USING UNIBUS B
2503					: WRITE DATA COMMAND CAUSED
2504					: IMPROPER REGISTER CHANGE
2505					: GOOD DATA GIVES WHAT SHOULD BE THERE
2506					: RECEIVED DATA GIVES REGISTER
2507					: CONTENTS AFTER WRITE DATA
2508	002156	060363		DH5	
2509	002160	062056		DT5	
2510	002162	062276		DF5	
2511					
2512			: ITEM 75		
2513	002164	057540		EM75	: WHILE USING UNIBUS B
2514					: WRITE DATA COMMAND CHANGED
2515					: WRITE FROM BUFFER
2516	002166	061127		DH30	
2517	002170	062146		DT30	
2518	002172	062325		DF30	
2519					
2520			: ITEM 76		
2521	002174	057643		EM76	: WHILE USING UNIBUS B
2522					: WRITE CHECK CAUSED AN
2523					: IMPROPER REGISTER CHANGE
2524					: GOOD DATA GIVES WHAT SHOULD
2525					: BE THERE
2526					: RECEIVED DATA GIVES WHAT WAS
2527					: THERE AFTER COMMAND
2528	002176	060363		DH5	
2529	002200	062056		DT5	
2530	002202	062276		DF5	







2643 000010  
 2644 000020  
 2645 000040  
 2646 000100  
 2647 000200  
 2648  
 2649  
 2650  
 2651  
 2652  
 2653  
 2654  
 2655 000001  
 2656 000002  
 2657 000004  
 2658 000010  
 2659 000020  
 2660 000040  
 2661 000100  
 2662 000200  
 2663 000400  
 2664 001000  
 2665 002000  
 2666 004000  
 2667 010000  
 2668 020000  
 2669 040000  
 2670 100000  
 2671  
 2672  
 2673  
 2674 000001  
 2675 000002  
 2676 000004  
 2677 000010  
 2678 000020  
 2679 000040  
 2680 000100  
 2681 000200  
 2682 000400  
 2683 001000  
 2684 002000  
 2685 004000  
 2686 010000  
 2687 020000  
 2688 100000  
 2689  
 2690  
 2691  
 2692 000001  
 2693 000002  
 2694 000004  
 2695 000010  
 2696 000020  
 2697 000040  
 2698

AT3= 10  
 AT4= 20  
 AT5= 40  
 AT6= 100  
 AT7= 200

;DEVICE 3 (BIT #3)  
 ;DEVICE 4 (BIT #4)  
 ;DEVICE 5 (BIT #5)  
 ;DEVICE 6 (BIT #6)  
 ;DEVICE 7 (BIT #7)

;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDS) (#1)  
 ;EACH BIT IS CALLED BY BIT NUMBER  
 ;DRIVE TYPE REGISTER (RHDT) (#6)  
 ;EACH BIT IS CALLED BY BIT NUMBER  
 ;LOOK-AHEAD REGISTER (RHLA) (#7)

EXT1= 1  
 EXT2= 2  
 EXT4= 4  
 EXT10= 10  
 EXT20= 20  
 EXT40= 40  
 SC1= 100  
 SC2= 200  
 SC4= 400  
 SC10= 1000  
 SC20= 2000  
 TRK1= 4000  
 TRK2= 10000  
 TRK4= 20000  
 TRK10= 40000  
 TRK20= 100000

;EXTENSION 1 (BIT #0)  
 ;EXTENSION 2 (BIT #1)  
 ;EXTENSION 3 (BIT #2)  
 ;EXTENSION 4 (BIT #3)  
 ;EXTENSION 5 (BIT #4)  
 ;EXTENSION 6 (BIT #5)  
 ;SECTOR COUNT FIELD 0 (BIT #6)  
 ;SECTOR COUNT FIELD 1 (BIT #7)  
 ;SECTOR COUNT FIELD 2 (BIT #8)  
 ;SECTOR COUNT FIELD 3 (BIT #9)  
 ;SECTOR COUNT FIELD 4 (BIT #10)  
 ;TRACK FIELD 1 (BIT #11)  
 ;TRACK FIELD 2 (BIT #12)  
 ;TRACK FIELD 3 (BIT #13)  
 ;TRACK FIELD 4 (BIT #14)  
 ;TRACK FIELD 5 (BIT #15)

;ERROR REGISTER #2 (RHER2) (#10)

WCU= 1  
 CSF= 2  
 WSU= 4  
 CSU= 10  
 MSE= 20  
 TDF= 40  
 TUF= 100  
 FEN= 200  
 WRU= 400  
 MHS= 1000  
 NHS= 2000  
 IXE= 4000  
 VU30= 10000  
 PLU= 20000  
 ACU= 100000

;WRITE CURRENT UNSAFE (BIT #0)  
 ;CURRENT SINK FAILURE (BIT #1)  
 ;WRITE SELECT UNSAFE (BIT #2)  
 ;CURRENT SWITCH UNSAFE (BIT #3)  
 ;MOTOR SEQUENCE ERROR (BIT #4)  
 ;TRANSITIONS DETECTOR FAILURE (BIT #5)  
 ;TRANSITIONS UNSAFE (BIT #6)  
 ;FAILSAFE ENABLED (BIT #7)  
 ;WRITE READY UNSAFE (BIT #8)  
 ;MULTIPLE HEAD SELECT (BIT #9)  
 ;NO HEAD SELECTION (BIT #10)  
 ;INDEX ERROR (BIT #11)  
 ;30VOLT UNSAFE (BIT #12)  
 ;PLO UNSAFE (BIT #13)  
 ;ACUNSAFE (BIT #15)

;OFFSET REGISTER (RHOF) (#11)

OF25= 1  
 OF50= 2  
 OF100= 4  
 OF200= 10  
 OF400= 20  
 OF800= 40

;OFFSET 25 MICRO INCHES (BIT #0)  
 ;OFFSET 50 MICRO INCHES (BIT #1)  
 ;OFFSET 100 MICRO INCHES (BIT #2)  
 ;OFFSET 200 MICRO INCHES (BIT #3)  
 ;OFFSET 400 MICRO INCHES (BIT #4)  
 ;OFFSET 800 MICRO INCHES (BIT #5)

2699	000200	OFREV= 200	;OFFSET NEGATIVE (REVERSE) (BIT #5)
2700	002000	HCI= 2000	;HEADER COMPARE INHIBIT (BIT #10)
2701	004000	ECI= 4000	;ERROR CORRECTION CODE INHIBIT (BIT #11)
2702	010000	FMT22= 10000	;FORMAT BIT (BIT #12)
2703			
2704		;DESIRED CYLINDER ADDRESS (RHCA) (#12)	
2705		;EACH BIT IS CALLED BY BIT NUMBER.	
2706			
2707			
2708			
2709			
2710		;CURRENT CYLINDER ADDRESS (RHCC) (#13)	
2711		;EACH BIT IS CALLED BY BIT NUMBER	
2712			
2713			
2714			
2715			
2716		;SERIAL NUMBER REGISTER (RHSN) (#14)	
2717		;EACH IS CALLED BY BIT NUMBER	
2718			
2719			
2720			
2721			
2722		;ERROR REGISTER #03 (RHER3) (#15)	
2723			
2724	000001	PSU= 1	;PACK SPEED UNSAFE (BIT #0)
2725	000002	VUF= 2	;VELOCITY UNSAFE (BIT #1)
2726	000010	UWR= 10	;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
2727	000020	PRE= 20	;DISK PACK ROTATION ERROR (BIT #4)
2728	000040	ACL= 40	;AC LOW (BIT #5)
2729	000100	DCL= 100	;DC LOW (BIT #6)
2730	040000	SKI= 40000	;SEEK INCCMPLETE (BIT #14)
2731	100000	OCYL= 100000	;OFF CYLINDER (BIT #15)
2732			
2733			
2734			
2735		;ECC POSITION REGISTER (RHEC1) (#16)	
2736		;EACH BIT IS CALLED BY BIT NUMBER	
2737			
2738			
2739			
2740			
2741		;ECC PATTERN REGISTER (RHEC2) (#17)	
2742		;EACH BIT IS CALLED BY BIT NUMBER	
2743			
2744			
2745			
2746			
2747		;*****	
2748			

2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799

.SBTTL REGISTER ADDRESSES

;RPO4 DISK I/O REGISTERS LOCATED IN THE RH11 CONTROLLER

002206 176722  
 002210 176702  
 002212 176704  
 002214 176710

RHDB: 176722 ;DATA BUFFER  
 RHWC: 176702 ;WORD COUNT  
 RHBA: 176704 ;BUS ADDRESS  
 RHCS2: 176710 ;CONTROL AND STATUS 2

;RPO4 DISK I/O REGISTERS LOCATED IN THE RPO4 DEVICE LOGIC

002216 176700  
 002220 176714  
 002222 176706  
 002224 176740  
 002226 176732  
 002230 176734  
 002232 176742  
 002234 176716  
 002236 176724  
 002240 176712  
 002242 176726  
 002244 176730  
 002246 176744  
 002250 176746  
 002252 176736  
 002254 176720

RHCS1: 176700 ;CONTROL AND STATUS 1  
 RHER1: 176714 ;ERROR #1  
 RHDST: 176706 ;DESIRED SECTOR/TRACK ADDRESS  
 RHER2: 176740 ;ERROR #2  
 RHOF: 176732 ;OFFSET  
 RHCA: 176734 ;DESIRED CYLINDER ADDRESS  
 RHER3: 176742 ;ERROR #3  
 RHAS: 176716 ;ATTENTION SUMMARY  
 RHMR: 176724 ;MAINTAINABILITY  
 RHDS1: 176712 ;DRIVE STATUS  
 RHDT: 176726 ;DRIVE TYPE  
 RHSN: 176730 ;SERIAL NUMBER  
 RHEC1: 176744 ;ECC POSITION  
 RHEC2: 176746 ;ECC PATTERN  
 RHCC: 176736 ;CURRENT CYLINDER ADDRESS  
 RHLA: 176720 ;LOOK-AHEAD

;P-CLOCK (KW11-P) I/O REGISTERS

002256 172540  
 002260 172542  
 002262 172544

PCLCSR: 172540 ;CONTROL AND STATUS REGISTERS  
 PCLBUF: 172542 ;COUNT SET BUFFER  
 PCLCTR: 172544 ;COUNTER

```

2800
2801
2802
2803
2804
2805
2806 002264 000000
2807 002266 000000
2808 002270 000000
2809 002272 000000
2810
2811
2812 002274 000000
2813 002276 000000
2814 002300 000000
2815 002302 000000
2816 002304 000000
2817 002306 000000
2818 002310 000000
2819 002312 000000
2820 002314 000000
2821 002316 000000
2822 002320 000000
2823 002322 000000
2824 002324 000000
2825 002326 000000
2826 002330 000000
2827 002332 000000
2828
2829
2830

```

;THE FOLLOWING LOCATIONS ARE RESERVED FOR REGISTERS  
;ANY TIME THERE IS AN ERROR ALL THESE WILL BE FILLED  
;ONLY SOME MAY BE PRINTED BUT ALL WILL BE FILLED TRUE  
;FOR THE TIME JUST AFTER THE "ERROR" ERROR COMMAND

```

DB: 0 ;DATA BUFFER
WC: 0 ;WORD COUNT
BA: 0 ;BUS ADDRESS
CS2: 0 ;CONTROL AND STATUS 2

CS1: 0 ;CONTROL AND STATUS 1
ER1: 0 ;ERROR #1
DST: 0 ;DESIRED SECTOR/TRACK ADDRESS
ER2: 0 ;ERROR #2
OF: 0 ;OFFSET
CA: 0 ;DESIRED CYLINDER ADDRESS
ER3: 0 ;ERROR #3
AS: 0 ;ATTENTION SUMMARY
MR: 0 ;MAINTAINABILITY
DS1: 0 ;DRIVE STATUS
DT: 0 ;DRIVE TYPE
SN: 0 ;SERIAL NUMBER
EC1: 0 ;ECC POSITION
EC2: 0 ;ECC PATTERN
CC: 0 ;CURRENT CYLINDER ADDRESS
LA: 0 ;LOCK-AHEAD

```

```

2831
2832
2833 ;FUNCTION EQUATES
2834
2835 ;TABLE OF FUNCTIONS FOR RHCSI THEN "GO" BIT HAS TO BE SET
2836 002334 FUTABL:
2837 002334 000000 NOPERA: 0 ;NO OPERATION
2838 002336 000002 UNLOAD: 2 ;UNLOAD (STAND BY)
2839 002340 000006 RECALI: 6 ;RECALIBRATE
2840 002342 000010 DCLEAR: 10 ;DRIVE CLEAR
2841 002344 000012 RELEAS: 12 ;RELEASE (DUAL-PORT OPERATION)
2842 002346 000030 SERCH: 30 ;SEARCH COMMAND
2843 002350 000050 WRCHK: 50 ;WRITE CHECK DATA
2844 002352 000052 WRCHD*: 52 ;WRITE CHECK HEADER AND DATA
2845 002354 000060 WRIDAT: 60 ;WRITE DATA
2846 002356 000062 WRIFOR: 62 ;WRITE HEADER AND DATA (FORMAT)
2847 002360 000070 READAT: 70 ;READ DATA
2848 002362 000072 REFOR: 72 ;READ HEADER AND DATA
2849 002364 000004 SEECOM: 4 ;SEEK COMMAND
2850 002366 000014 OFSETC: 14 ;OFFSET COMMAND
2851 002370 000016 RETCL: 16 ;RETURN TO CENTERLINE
2852 002372 000022 PKACK: 22 ;PACK ACKNOWLEDGE
2853 002374 000020 READIN: 20 ;READ IN
2854 002376 000000 ILLEGL: .WORD 0 ;COMPUTED ILLEGAL FUNCTION
2855
2856 ;DATA BUFFER FOR READ WRITE
2857 002400 000422 WRFROM: .BLKW 274. ;WRITE FROM THIS BUFFER
2858 003444 000422 REINTO: .BLKW 274. ;READ INTO THIS BUFFER
2859
2860
2861
2862 ;RESERVED LOCATIONS
2863 004510 000000 REGADR: 0 ;SAVE REGISTER ADDRESS HERE
2864 004512 000000 ERWORD: 0 ;SAVE ERROR WORD NUMBER HERE
2865 004514 000000 TSTNM: 0 ;TEST NUMBER
2866 004516 000000 RP4VEC: 0 ;CONTAINS ADDRESS OF LOCATION
2867 ;WHERE AN RPO4 INTERRUPT IS TO VECTOR TO
2868 ;THIS MUST BE MOVED INTO 'RPVEC' TO BE
2869 ;EFFECTIVE.
2870
2871 004520 000000 OFSTVL: 0 ;OFFSET VALUE USED IN OFFSET TEST
2872
2873
2874 004522 000024 SAVERE: .BLKW 20. ;BLOCK TO SAVE REGISTERS
2875 004572 000000 FINALA: 0 ;SAVE LOOK AHEAD REGISTER AT END OF OPERATION
2876 004574 000000 FINACC: 0 ;SAVE CURRENT CYLINDER REGISTER AT END OF OPERATION
2877
2878
2879 ;TABLE FOR ATTENTION BITS
2880 ;ATTENTION TABLE
2881 004576 001 002 004
2882 004601 010 020 040
2883 004604 100 200
2884
2885
2886
  
```





```

2915 .SBTTL REGISTER TEST
2916 004662 012737 177777 004634 BEGIN1: MOV # -1, @#NCPUSH ; JUMP OVER OPERATOR REQUIRED TESTS
2917 004670 005037 004636 CLR @#SELECT ; DO NOT SELECT UNIT
2918 004674 000412 BR START
2919 004676 012737 177777 004636 BEGIN2: MOV # -1, @#SELECT ; SELECT UNIT
2920 004704 005037 004634 CLR @#NCPUSH ; DO NOT JUMP OVER ANY TEST
2921 004710 000404 BR START
2922 004712 005037 004636 BEGIN: CLR @#SELECT ; DO NOT SELECT UNIT
2923 004715 005037 004634 CLR @#NCPUSH ; DO NOT JUMP OVER ANY TEST
2924 ; NORMAL RUN
2925
2926 START:
2927 004722 0127 001100 MOV # $CMTAG, R6 ; FIRST LOCATION TO BE CLEARED
2928 004726 00502 CLR (R6)+ ; CLEAR MEMORY LOCATION
2929 004730 022706 001126 CMP # $BDDAT, R6 ; DONE?
2930 004734 001374 BNE .-6 ; LOOP BACK IF NO
2931 004736 012706 001000 MOV # $STACK, SP ; SETUP THE STACK POINTER
2932 004742 012737 037560 000020 MOV # $SCOPE, @#IOTVEC ; IOT VECTOR FOR SCOPE ROUTINE
2933 004750 012737 000340 000022 MOV # 340, @#IOTVEC+2 ; LEVEL 7
2934 004756 012737 041242 000030 MOV # $ERROR, @#EMTVEC ; EMT VECTOR FOR ERROR ROUTINE
2935 004764 012737 000340 000032 MOV # 340, @#EMTVEC+2 ; LEVEL 7
2936 004772 012737 042762 000034 MOV # $TRAP, @#TRAPVEC ; TRAP VECTOR FOR TRAP CALLS
2937 005000 012737 000340 000036 MOV # 340, @#TRAPVEC+2 ; LEVEL 7
2938 005006 012737 043026 000024 MOV # $PWRON, @#PWRVEC ; POWER FAILURE VECTOR
2939 005014 012737 000340 000026 MOV # 340, @#PWRVEC+2 ; LEVEL 7
2940 005022 005067 174162 CLR $TIMES ; INITIALIZE NUMBER OF ITERATIONS
2941 005026 005067 174160 CLR $ESCAPE ; CLEAR THE ESCAPE ON ERROR ADDRESS
2942 005032 112767 000001 174055 MOV # 1, $ERRMAX ; ALLOW ONE ERROR PER TEST
2943 005040 012767 005340 174040 MOV # ., $LPADR ; INITIALIZE THE LOOP ADDRESS FOR SCOPE
2944 005046 012767 005046 174034 MOV # ., $LPERR ; SETUP THE ERROR LOOP ADDRESS
2945 005054 013746 000004 MOV @#4, -(SP) ; SAVE ERROR VECTOR
2946 005060 013746 000006 MOV @#6, -(SP)
2947 005064 012767 005100 172712 MOV # 64$, 4 ; SET UP TIME OUT VECTOR
2948 005072 005777 174040 TST @#SWR ; TRY TO REFERENCE HARDWARE SWR
2949 005076 000407 BR 65$ ; BRANCH IF NO TIMEOUT TRAP OCCURS
2950 005100 012767 000176 174030 64$: MOV # $SWREG, SWR ; POINT TO SOFTWARE SWR
2951 005106 012767 000174 174024 MOV # $DISPREG, DISPLAY ; POINT TO SOFTWARE DISPLAY REG
2952 005114 022626 CMP (SP)+, (SP)+ ; RESTORE STACK
2953 005116 012637 000006 65$: MOV (SP)+, @#6 ; RESTORE ERROR VECTOR
2954 005122 012637 000004 MOV (SP)+, @#4
2955
2956
2957 005126 012767 000000 172642 MOV # 0, PS ; SET PROCESSOR STATUS TO 0
2958 005134 012737 000200 000036 MOV # 200, @#TRAPVEC+2 ; TRAP PRIORITY = 4
2959 005142 013700 002204 MOV @#RPVEC, R0 ; GET RP VECTOR ADDRESS
2960 005146 012720 037516 MOV # RPVECT, (R0)+ ; THIS IS FOR UNTIMELY INTERRUPTS
2961 005152 012710 000340 MOV # 340, (R0) ; RPO4 INTERRUPT SERVICE ROUTINE
2962 ; PRIORITY = .
2963
2964 005156 004737 040510 JSR PC, @#$TKINT ; INITILIZE THE TK
2965 005162 005737 004646 TST @#FIRST ; IS THIS FIRST TIME ROUND
2966 005166 001001 BNE 1$ ; BRANCH IF NOT
2967 005170 000402 BR 2$
2968 005172 000137 005776 1$: JMP @#$NO1
2969 005176 104400 005204 2$: TYPE .+4 ; TYPE ASCIZ STRING
2970 005202 000440 BR 66$ ; GET OVER THE ASCIZ

```

```

2971      005304      104400      005312      665:      ::.ASCIZ      <15><12>/RPO4 FUNCTIONAL CONTROLLER TEST PART I (STATIC 2A) - DERP
2972      005304      104400      005312      TYPE      ::TYPE ASCIZ STRING
2973      005310      000435      BR      675+4      ::GET OVER THE ASCIZ
2974      005310      000435      ::.ASCIZ      <15><12>/MAKE SURE PORT SWITCH ON DRIVE IS LOCKED ON EITHER PORT/
2975      005404      104400      005412      675:      TYPE      ::TYPE ASCIZ STRING
2976      005404      104400      005412      BR      685+4      ::GET OVER THE ASCIZ
2977      005410      000440      ::.ASCIZ      <15><12>/IF CHANGE IS REQUIRED ON PORT SWITCH POWER DRIVE DOWN AND
2978      005512      104400      005520      685:      TYPE      ::TYPE ASCIZ STRING
2979      005512      104400      005520      BR      695+4      ::GET OVER THE ASCIZ
2980      005516      000416      ::.ASCIZ      <15><12>/AGAIN AFTER SWITCH CHANGE
2981      005554      104400      005562      695:      TYPE      ::TYPE ASCIZ STRING
2982      005554      104400      005562      BR      705+4      ::GET OVER THE ASCIZ
2983      005560      000432      ::.ASCIZ      <15><12>/ALL DCL UNDER TEST MUST BE LOCKED ON CORRECT PORT/
2984      005646      104400      005654      705:      TYPE      ::TYPE ASCIZ STRING
2985      005646      104400      005654      BR      715+4      ::GET OVER THE ASCIZ
2986      005652      000427      ::.ASCIZ      <15><12>/ALL DCL NOT UNDER TEST MUST BE SWITCHED OFF/
2987      005732      104400      005740      715:      TYPE      ::TYPE ASCIZ STRING
2988      005732      104400      005740      BR      725+4      ::GET OVER THE ASCIZ
2989      005736      000417      ::.ASCIZ      <15><12>/OR LOCKED ON THE OTHER PORT/
2990      005776      012737      177777      004646      725:      SND1:      MOV      #-1,2#FIRST      ;NEXT TIME DO NOT GIVE HEADER
2991      005776      012737      177777      004646      ;IS THERE A P-CLOCK (KW11-P) ON THE SYSTEM
2992      005776      012737      177777      004646      ;IF SO MAKE 'WAT' TRAPS GO TO 'WAIT.P'
2993      005776      012737      177777      004646      ;IF SO MAKE RPO4 INTERRUPTS GO TO 'TIME 1'
2994      005776      012737      177777      004646      ;IF NOT MAKE 'WAT' TRAPS GO TO 'WAIT.T'
2995      005776      012737      177777      004646      ;IF NOT MAKE RPO4 INTERRUPTS GO TO 'TIME 2'
2996      005776      012737      177777      004646      ;THE NEXT LINE IS TO BE ADDED LATER
2997      005776      012737      177777      004646      ;AND THE JUMP AND NOP REMOVED
2998      005776      012737      177777      004646      ;FOR NOW NO CLOCK WILL BE USED
2999      005776      012737      177777      004646      ;MOV2#15,2#ERRVEC;SET TIME-OUT VECCTR
3000      006004      000137      006034      JMP      2#15      ;DO NOT USE CLOCK
3001      006010      000240      NOP
3002      006012      005737      002256      TST      2#PCLCSR      ;REFERENCE P-CLOCK STATUS REGISTER
3003      006016      012737      035010      043024      MOV      #WAIT.P,2#STRPAD+20 ;ADDRESS = 172540
3004      006024      012737      034750      004516      MOV      #TIME1,2#RP4VEC ;THERE IS A P-CLOCK
3005      006032      000406      BR      25      ;THERE IS A P CLOCK SO
3006      006034      012737      035250      043024      15:      MOV      #WAIT.T,2#STRPAD+20 ;VECTOR TO TIME1
3007      006042      012737      034762      004516      MOV      #TIME2,2#RP4VEC ;THERE IS NO P-CLOCK
3008      006050      012737      177777      041404      25:      MOV      #-1,2#PRITEM      ;CLEAR PREVIOUS ITEM NUMBER
3009      006056      005737      004636      TST      2#SELECT      ;WAS IT A 200 START
3010      006062      001442      BEQ      .TST!      ;BRANCH IF STARTING FROM 200
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026

```

```

3027 006064 104400 006072          TYPE          +4          ;;TYPE ASCIZ STRING
3028 006070 000424          BR          64$          ;;GET OVER THE ASCIZ
3029          ;;.ASCIZ          <15><12>/SELECT UNIT NUMBER TO BE TESTED ? //15 12
3030 006142          64$:
3031 006142 104407          RDOCT
3032 006144 042716 177770          BIC          #177770,(SP)          ;ONLY KEEP LAST 3 BITS
3033 006150 011637 004626          MOV          (SP),@#UNIT          ;SAVE UNIT TO BE TESTED
3034 006154 012637 004640          MOV          (SP)+,@#UNITSL      ;SAVE UNIT TO BE TESTED
3035
3036
3037
3038
3039 006160 001403          BEQ          TST1          ;BRANCH IF STARTING FROM 200
3040
3041 006162 013737 004640 004626          MOV          @#UNITSL,@#UNIT      ;SET UNIT NUMBER
3042
3043
3044          ;*****
3045          ;*TEST 1          REFERENCE EACH REGISTER
3046          ;*          REFERENCE EACH REGISTER BY A MOVE INSTRUCTION
3047          ;*****
3048 006170 000004          TST1: SCOPE
3049 006172 012767 000001 173010          MOV          #1,$TIMES          ;;DO 1 ITERATION
3050 006200 012706 001000          MOV          #STACK,$SP          ;SET UP STACK POINTER
3051 006204 012737 041250 000030          MOV          #REGSA1,@#EMTVEC    ;ERROR VECTOR SO THAT
3052          ;NO REGISTERS ARE SAVED
3053 006212 012737 006240 000004          MOV          #2$, @#ERRVEC      ;SET UP FOR - 5 TIMEOUT
3054 006220 012700 000024          MOV          #24, $R0          ;THERE ARE 24 REG TO TEST
3055 006224 012701 002206          MOV          #RHD8, $R1        ;R1 NOW HAS ADDR OF ADDR OF FIRST REG.
3056 006230 013102          1$: MOV          @($R1)+,$R2      ;READ HARDWARE REG.
3057 006232 005300          DEC          $R0          ;COUNT DOWN
3058 006234 001375          BNE          1$          ;BRANCH IF 24 NOT DONE
3059 006236 000471          BR          3$          ;BRANCH IF 24 DONE
3060 006240 012737 000006 000004          2$: MOV          #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
3061 006246 022626          CMP          ($SP)+,($SP)+      ;CLEAN STACK
3062 006250 016167 177776 172720          MOV          -2($R1),$TMP1      ;STORE FAILING REG ADDR
3063 006256 104007          ERROR          7          ;REGISTER NON EXISTANT
3064 006260 032737 020000 001136          BIT          #SW13,@#SWR        ;INHIBIT ERROR PRINTOUT ?
3065 006266 001053          BNE          4$          ;BRANCH IF YES
3066 006270 104400 006276          TYPE          +4          ;;TYPE ASCIZ STRING
3067 006274 000431          BR          64$          ;;GET OVER THE ASCIZ
3068          ;;.ASCIZ          <15><12>/IF BASE ADDRESS IS TO BE CHANGED HALT PROGRAM
3069 006360          64$:
3070 006360 104400 006366          TYPE          +4          ;;TYPE ASCIZ STRING
3071 006364 000411          BR          65$          ;;GET OVER THE ASCIZ
3072          ;;.ASCIZ          <15><12>/AND RESTART AT /
3073          65$:
3074          MOV          #BASECH,-($SP) ;GET READY TO TYPE STARTING ADDRESS
3075          ;OF "CHANGE OF BASE ADDRESS" ROUTINE
3076          TYPOC
3077 006414 104401          4$: JMP          @#SEOP          ;GO TO END OF PROGRAM
3078 006416 000137 034260          3$: MOV          #SEORR,@#EMTVEC ;RESTORE ERROR VECTOR
3079 006422 012737 041242 000030          ;SO THAT REGISTERS ARE SAVED
3080          MOV          #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
3081
3082          ;*****
          ;*TEST 2          PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT

```

# F06

MAINDEC-11-DERPU-B  
DERPUB.P11 T2

MACY11 27(732) 17-SEP-76 13:23 PAGE 70  
PARTIAL TEST FOR RHAS FOR UNIT NUMBERS PRESENT

```

3093          : * CHECK THAT RHAS CAN BE CLEARED BY MOVING ALL ONES
3094          : *****
3095 006436 000004          TST2: SCOPE
3096 006440 012767 000001 172542  MOV #1, $TIMES          ;; DO 1 ITERATION
3097 006446 012706 001000  MOV #STA, SP          ;; SET STACK POINTER
3098 006452 013701 002234  MOV @R1, R1          ;; R1 HAS ADDRESS OF RHAS
3099 006456 012711 177777  MOV #-1, @R1          ;; THIS CLEARS RHAS (SURPRISED!)
3090 006462 105711  TSTB @R1
3091
3092 006464 001407  BEQ TST3          ; BRANCH IF GOOD
3093
3094 006466 011137 001125  MOV @R1, @#$BDDAT          ; BAD DATA
3095 006472 005037 001124  CLR @#$GDDAT          ; GOOD DATA
3096 006476 010137 004510  MOV PI, @#REGADR          ; FAILING REG. RHAS
3097 006502 104005  ERROR 5          ; RHAS DOES NOT CLEAR
3098          ; WITH ONES
3099
3100
3101          : *****
3102          : * TEST 3 TEST FOR DRIVES PRESENT USING RHAS AND RHCS2
3103          : * THE NUMBER OF RPO4 DRIVES PRESENT ARE FOUND
3104          : * BY MOVING ALL ONES INTO RHER1 WITH UNIT NUMBER
3105          : * IN RHCS2 INCREMENTED FROM ZERO TO SEVEN
3106          : * THEN THE SET BITS IN RHAS WILL GIVE DRIVES PRESENT
3107          : * THE DRIVE TYPE IS CHECKED TO HAVE 2020 OR 24020 AND THEN
3108          : * UNITS PRESENT ARE STORED IN A TABLE CALLED 'UNITS'
3109
3110          : *****
3111 006504 000004          TST3: SCOPE
3112 006506 012767 000001 172474  MOV #1, $TIMES          ;; DO 1 ITERATION
3113 006514 000005  RESET          ; START WITH AN INIT
3114 006516 004737 040510  JSR PC, @#$TKINT          ; INITILIZE TK
3115 006522 032737 020000 001136  BIT #SW13, @#SWR          ; INHIBIT ERROR TYPEOUT?
3116 006530 001030  BNE 4$          ; BRANCH IF YES
3117 006532 104400 006540  TYPE #+4          ; TYPE ASCIZ STRING
3118 006536 000425  BR 64$          ; GET OVER THE ASCIZ
3119          ;; .ASCIZ <15><12>/LOOKING AT RHAS - RPO4 DRIVES PRESENT
3120
3121 006512 013701 002234  64$: MOV @#RHAS, R1          ; R1 HAS ADDR. OF RHAS
3122 006616 013702 002214  4$: MOV @#RHCS2, R2          ; R2 HAS ADDR. OF RHCS2
3123 006622 005012  CLR @R2          ; CLEAR RHCS2
3124 006624 012700 000010  MOV #8, R0          ; COUNT
3125 006630 013704 002220  MOV @#RHER1, R4          ; R4 HAS ADDR. OF RHER1
3126 006634 012714 177777  1$: MOV #-1, @R4          ; MOVE ERRORS INTO RHER1
3127 006640 005212  INC @R2          ; INCREMENT UNIT NO.
3128 006642 005300  DEC R0          ; COUNT
3129 006644 001373  BNE 1$          ; BRANCH IF 8 NOT DONE
3130 006646 111137 004652  MOVB @R1, @#TOTALAT          ; SAVE TOTAL ATTENTION
3131          ; USED IN DRIVE CLEAR TEST
3132 006652 105037 004653  CLRB @#TOTALAT+1          ; CLEAR UPPER BYTE
3133 006656 105711  TSTB @R1          ; TEST FOR ANY DRIVES PRESENT
3134 006660 001402  BEQ 2$          ; IF SOME NOT THERE BRANCH
3135 006662 000167 000420  JMP XE2          ; NONE THERE
3136 006666 032737 020000 001136  2$: BIT #SW13, @#SWR          ; INHIBIT ERROR TYPE OUT?
3137 006674 001402  BEQ 3$          ; BRANCH IF NO
3138 006676 000167 000670  JMP TST4          ; OUT

```

3139	006702			35:		
3140	006702	104400	006710		TYPE +4	::TYPE ASCIZ STRING
3141	006706	000412			BR 655	::GET OVER THE ASCIZ
3142					::.ASCIZ	<15><12>/NO DRIVES-RHAS=0/
3143	006734			655:		
3144	006734	104400	006742		TYPE +4	::TYPE ASCIZ STRING
3145	006740	000436			9R 665	::GET OVER THE ASCIZ
3146					::.ASCIZ	<15><12>/WRITING ONES INTO ERROR REGISTER #1 FOR ALL UNIT NUMBERS/
3147	007036			665:		
3148	007036	104400	007044		TYPE +4	::TYPE ASCIZ STRING
3149	007042	000441			BR 675	::GET OVER THE ASCIZ
3150					::.ASCIZ	<15><12>/DOES NOT SET ANY BIT IN THE ATTENTION REGISTER SC ABCRT P
3151	007146			675:		
3152	007146	104400	007154		TYPE +4	::TYPE ASCIZ STRING
3153	007152	000442			BR 685	::GET OVER THE ASCIZ
3154					::.ASCIZ	<15><12>/TO LOOP ON THIS TEST WITHOUT PRINTOUT SET SWITCHES 13 9
3155	007260			585:		
3156	007260	000137	034260		JMP 2#SEOP	;GO OUT
3157	007264	005037	004606		CLR 2#UNITS	
3158	007270	012767	000001	175332	MOV #1,NOUNIT	;NO. UNITS PRESENT=1
3159	007276	005037	004626		CLR 2#UNIT	
3160	007302	000137	034260		JMP 2#SEOP	;JUMP OUT
3161	007306			XE2:		
3162	007306	012700	000010	25:	MOV #8,R0	;COUNTER
3163	007312	012703	004606		MOV #UNITS,R3	;POINTER
3164	007316	012723	177777	35:	MOV #-1,(R3)+	;PRESET BLOCK TO ALL ONES
3165	007322	005300			DEC R0	;COUNT
3166	007324	001374			BNE 35	;BRANCH IF 8 NOT DONE
3167	007326	012703	004606		MOV #UNITS,R3	;POINTER
3168	007332	005005			CLR R5	
3169	007334	005037	004630		CLR 2#NOUNIT	;NO. OF UNITS PRESENT
3170	007340	012700	000010		MOV #8,R0	;COUNTER
3171	007344	011137	001174		MOV 2R1,2#STMP0	;TEMPORARY STORAGE
3172	007350	006037	001174	45:	ROR 2#STMP0	;SET CARRY IF ONE IN 0 BIT
3173						
3174	007354	103065			BCC 55	
3175	007356	010577	172632		MOV R5,2RHCS2	;INSERT UNIT NUMBER
3176	007362	022777	024020	172652	CMP #24020,2RHDT	;IS THIS A DUAL PORT RPO4
3177	007370	001450			BEQ 65	;BRANCH IF YES
3178	007372	022777	023020	172642	CMP #20020,2RHDT	;IS THIS A SINGLE PORT RPO4
3179	007400	001444			BEQ 65	;BRANCH IF YES
3180	007402	104400	007410		TYPE +4	::TYPE ASCIZ STRING
3181	007406	000410			BR 645	::GET OVER THE ASCIZ
3182					::.ASCIZ	<15><12>/UNIT NUMBER /
3183	007430			645:		
3184	007430	010546			MOV R5,-(SP)	;GET READY TO TYPE UNIT NUMBER
3185	007432	104404			TYPDS	
3186	007434	104400	007442		TYPE +4	::TYPE ASCIZ STRING
3187	007436	000405			BR 655	::GET OVER THE ASCIZ
3188					::.ASCIZ	/, RHDT= /
3189	007454			655:		
3190	007454	017746	172562		MOV 2RHDT,-(SP)	;GET READY TO TYPE RHDT
3191	007460	104401			TYP0C	
3192	007462	104400	007470		TYPE +4	::TYPE ASCIZ STRING
3193	007466	000410			BR 665	::GET OVER THE ASCIZ
3194					::.ASCIZ	/ ---NOT AN RPO4/

21

1

1

1

```

3195 007510          56$:
3196 007510 000407   BR          5$          ;NO RPO4 FOUND SO BRANCH
3197 007512 010523   6$:      MOV          R5,(R3)+
3198 007514 104400 001221   TYPE      $CRLF
3199 007520 010546   MOV          R5,-(SP)
3200 007522 104404          TYPDS          ;TYPE DRIVE NO.
3201 007524 005237 004630   INC          2#NUNIT
3202 007530 005205   5$:      INC          R5
3203 007532 005300   CEC          R0
3204 007534 001305   BNE          4$
3205 007536 013737 004606 004626   MOV          2#UNITS,2#UNIT
3206 007544 013737 004630 004632   MOV          2#NUNIT,2#NUNIT ;SAVE NO. OF UNITS
3207 007552 005337 004632   DEC          2#NUNIT ;IF NUNIT = 0 THEN ONLY ONE UNIT
3208          ;IF NUNIT MORE THAN 0 THEN MORE THAN ONE UNIT
3209 007556 005737 004636   TST          2#SELECT ;STARTING ADDRESS 200
3210
3211 007562 001403   BEQ          TST4 ;BRANCH IF STARTING FROM 200
3212
3213 007564 013737 004640 004626   MOV          2#UNITS,2#UNIT ;SET UNIT NUMBER
3214
3215
3216
3217
3218
3219
3220
3221          ;*****
3222          ;*TEST 4 TYPE SERIAL NUMBER AND DRIVE TYPE
3223          ;* SET APPROPRIATE ATTENTION BIT OF UNIT UNDER TEST IN 'ATTE'T'
3224          ;* TYPE UNIT UNDER TEST
3225          ;* READ SERIAL NUMBER AND DRIVE TYPE REGISTERS
3226          ;* TYPE IT OUT AND PROCEED
3227          ;* TO LOOP HERE SET SWITCH 8 AND THIS TEST NUMBER ON
3228          ;* SWITCHES 0 THRU 7 AND RESTART
3229
3230          ;*****
3231 007572 000004   TST4:  SCOPE
3232 007574 012767 000001 171406   MOV          #1,$TIMES ;DO 1 ITERATION
3233 007602 012767 010016 171276   MOV          #1,$SLPADR ;SET SCOPE LOOP ADDRESS
3234 007610 012706 001000   MOV          #STACK,SP ;RESET STACK
3235 007614 012737 000004 004514   MOV          #4,2#STNM ;SAVE TEST NUMBER
3236
3237 007622 004737 034556   JSR          PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
3238          ;R3-RHDS1, R4-RHER1
3239          ;GIVE RH-11 INITIALIZE
3240          ;SETUP UNIT NUMBER
3241 007626 005037 004650   CLR          2#ATTENT ;CLEAR
3242 007632 013700 004626   MOV          2#UNIT,R0 ;R0 CONTAINS UNIT NO
3243 007636 116037 004576 004650   MOVB        ATABLE(R0),2#ATTENT ;SET APPROPRIATE ATTENTION BIT
3244 007644 104400 007652   TYPE          +4 ;TYPE ASCIZ STRING
3245 007650 000415   BR          64$ ;GET OVER THE ASCIZ
3246          ;;.ASCIZ <15><12> TESTING DRIVE NUMBER
3247
3248 007704 013746 004626   64$:      MOV          2#UNIT,-(SP) ;UNIT NO. TO STACK
3249 007710 104401   TYPOC          ;TYPE DRIVE NO.
3250 007712 104400 001221   TYPE          .$CRLF

```

Fig 1

MAINDEC-11-DERPU-8  
DERPUB.P11 T4

MACY11 27(732) 17-SEP-76 13:23 PAGE 73  
TYPE SERIAL NUMBER AND DRIVE TYPE

```

3251 007716 104400 007724      TYPE      +4      ;;TYPE ASCIZ STRING
3252 007722 000410              BR      65$      ;;GET OVER THE ASCIZ
3253              ;;ASCIZ      <15><12>/SERIAL NO. - /
3254 007744              65$: MOV      @RHSN,-(SP)      ;;SAVE @RHSN FOR TYPEOUT
3255 007744 017746 172274      TYPOC      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3256 007750 104401
3257 007752 104400 001221      TYPE      ,SCLF
3258 007756 104400 007764      TYPE      +4      ;;TYPE ASCIZ STRING
3259 007762 000410              BR      66$      ;;GET OVER THE ASCIZ
3260              ;;ASCIZ      <15><12>/DRIVE TYPE = /
3261 010004              66$: MOV      @RHDT,-(SP)      ;;SAVE @RHDT FOR TYPEOUT
3262 010004 017746 172232      TYPOC      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3263 010010 104401
3264 010012 104400 001221      TYPE      ,SCLF
3265 010016 005777 172222      1$: TST      @RHSN      ;READ SERIAL NO. AND DRIVE TYPE
3266 010022 005777 172214      TST      @RHDT      ;THESE TWO ARE TO HELP SCOPE LOOPS
3267 010026 017737 172212 002322      MOV      @RHSN,@#SN      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
3268 010034 017737 172202 002320      MOV      @RHDT,@#DT      ;SAVE TO CHECK IF DRIVE CLEAR CLEARS ANY BITS
3269
3270
3271
3272
3273              ;*****
3274              ;*TEST 5      PROGRAM INTERRUPT
3275
3276              ;*      PROGRAM INTERRUPT IS TESTED BY SETTING RDY AND IE
3277              ;*      IN RHCS1 AT THE SAME TIME
3278              ;*      THIS SHOULD INTERRUPT THROUGH LOCATION 254
3279              ;*      THE PROCESSOR PRIORITY IS SET TO 4
3280              ;*****
3281 010042 000004      †T5: SCOPE
3282
3283
3284 010044 012737 000005 004514      MOV      #6-1,@#TSTNM      ;THIS SAVES TEST NUMBER
3285
3286 010052 012706 001000      MOV      #STACK,SP      ;RESET STACK
3287 010056 004737 034556      JSR      PC,@#CLDISK      ;CLEAR DISK
3288 010062 013700 002204      MOV      @#RPVEC,RO      ;GET RP VECTOR ADDRESS
3289 010066 012720 010130      MOV      @#RTPR1,(RO)+      ;THIS IS FOR TIMELY INTERRUPTS
3290 010072 012710 000340      MOV      #340,(RO)      ;RPO4 INTERRUPT SERVICE ROUTINE
3291              ;PRIORITY = 7
3292
3293 010076 012767 000200 167672      MOV      #200,PS      ;SET PROCESSOR PRIORITY
3294 010104 012711 000300      MOV      #RDY!IE,@R1      ;RDY, IE IN RHCS1 SHOULD CAUSE INTERRUPT
3295 010110 013737 035246 001176      MOV      @#TIMCNT,@#STMP1 ;COUNTER
3296 010116 005337 001176      1$: DEC      @#STMP1      ;WAIT FOR INTERRUPT
3297 010122 001375      BNE      1$      ;BRANCH IF NOT ZERO
3298              ;BEFORE THIS IS ZERO INTERRUPT SHOULD
3299              ;OCCUR
3300 010124 104065      ERROR      65      ;INTERRUPT DID NOT OCCUR
3301
3302
3303 010126 000405      BR      TST6      ;BRANCH TO NEXT TEST
3304
3305
3306 010130 022626      RPTRP1: CMP      (SP)+,(SP)+      ;RESTORE STACK

```





```

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 010232 000004
3381 010234 012706 001000
3382 010240 012737 000007 004514
3383
3384 010246 004737 034556
3385
3386
3387
3388 010252 012767 000000 167516
3389 010260 005737 004634
3390 010264 001007
3391 010266 005737 000042
3392 010272 001004
3393 010274 005737 001100
3394 010300 001001
3395 010302 000402
3396 010304
3397 010304 000167 000506
3398 010310
3399 010310 104400 010316
3400 010314 000407
3401
3402 010334
3403 010334 013746 004626
3404 010340 104404
3405 010342 104400 010350
3406 010346 000413
3407
3408 010376

```

```

:*****
;TEST 7 PACK ACKNOWLEDGE

```

```

;* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
;*
;* IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
;* THEN THIS TEST IS NOT PERFORMED
;*
;* IF NO ACT-11 MONITOR IS PRESENT
;* THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
;* ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
;*
;* THIS TESTS THE ACKNOWLEDGE COMMAND=44
;* VV BIT - RHDS1 BIT #6
;* MOL BIT - RHDS1 BIT #12
;* DVA BIT - RHCS1 BIT #11
;* THE DRIVE IS STOPPED MOL IS CHECKED TO BE 0
;* AND DVA SHOULD BE 0
;* THE DRIVE IS POWERED UP
;* VV SHOULD BE 0, MOL SHOULD BE 1, DVA SHOULD BE 1
;* GO SHOULD BE 0
;* ALL REGISTERS EXCEPT RHDB, RHLA AND RHCC ARE STORED
;* PACK ACKNOWLEDGE IS ISSUED
;* ALL STORED REGISTERS SHOULD BE UNCHANGED
;* EXCEPT VV

```

```

:*****

```

```

TST7: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #7,J#TSTNM ;SAVE TEST NUMBER
JSR PC,#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
MOV #0,PS ;SET PROSESSER STATUS TO 0
TST #NOPUSH ;IS THIS 220 START
BNE 1$ ;BRANCH IF YES
TST #42 ;MONITOR (ACT-11) RETURN ADDRESS
BNE 1$ ;BRANCH IF MONITOR PRESENT
TST #SPASS ;IS THIS FIRST PASS
BNE 1$ ;BRANCH IF NOT FIRST PASS
BR 2$ ;BRANCH TO CONTINUE TEST
1$: JMP TST10 ;JUMP TO NEXT TEST
2$: TYPE #+4 ;;TYPE ASCIZ STRING
BR #64$ ;;GET OVER THE ASCIZ
;;.ASCIZ <15><12>/STOP DRIVE /
64$: MOV #UNIT,-(SP) ;GET UNIT UNDER TEST
TYPDS
TYPE #+4 ;;TYPE ASCIZ STRING
BR #65$ ;;GET OVER THE ASCIZ
;;.ASCIZ / THEN HIT CONTINUE/<15><12>
65$:

```

```

3409 010376 000000          HALT
3410 010400 004767 026066  JSR    PC,PUTREG      ;SAVE REG IN WC ON
3411 010404 032713 010000  BIT    #MOL,DR3      ;MOL IN RHDS1 SHOULD BE = 0
3412 010410 001403          BEQ    3$             ;BRANCH IF MOL=0
3413 010412 010337 001122  MOV    R3,DR3$BDADR  ;FAILING REGISTER ADDRESS-RHDS1
3414 010416 104010          ERROR  10           ;ON SPINPLE POWERED DOWN
3415          ;MOL SHOULD BE 0
3416 010420 013746 004626  3$:  MOV    DR3$UNIT, -(SP) ;UNIT NUMBER
3417 010424 052716 000100  BIS    #IR, (SP)     ;INCLUDE IR
3418 010430 022612          CMP    (SP)+, DR2    ;ONLY UNIT NO. AND IR SHOULD BE
3419          ;HIGH IN RHCS2
3420 010432 001403          BEQ    4$             ;BRANCH IF RHCS2 GOOD
3421 010434 010237 001122  MOV    R2,DR3$BDADR ;FAILING REGISTER ADDRESS-RHCS2
3422 010440 104011          ERROR  11           ;WITH SPINDLE POWERED DOWN
3423          ;ONLY UNIT NO. AND IR SHOULD BE
3424          ;HIGH
3425
3426 010442          4$:
3427
3428 010442 004737 034556  JSR    PC,DR3$CLDISK ;SET R1-RHCS1, R2-RHCS2
3429          ;R3-RHDS1, R4-RHER1
3430          ;GIVE RH-11 INITIALIZE
3431          ;SETUP UNIT NUBER
3432 010446 104400 010454  TYPE   ;+4
3433 010452 000410          BR     66$          ;TYPE ASCIZ STRING
3434          ;:ASCIZ <15><12>/START DRIVE /
3435          66$:
3436 010474 013746 004626  MOV    DR3$UNIT, -(SP) ;GET UNIT UNDER TEST
3437 010500 104404          TYPDS
3438 010502 104400 010510  TYPE   ;+4
3439 010506 000420          BR     67$          ;TYPE ASCIZ STRING
3440          ;:ASCIZ / AFTER HEAD LOAD HIT CONTINUE/<15><12>
3441          67$:
3442 010550 000000          HALT
3443 010552 004767 025714  JSR    PC,PUTREG      ;SAVE REG IN WC ON
3444 010556 032713 010000  BIT    #MOL,DR3      ;MOL IN RHDS1 SHOULD BE = 1
3445 010562 001411          BEQ    5$             ;BRANCH IF MOL = 0
3446 010564 032713 000400  BIT    #DPR,DR3      ;DPR IN RHDS1 SHOULD BE = 1
3447 010570 001406          BEQ    5$             ;BRANCH IF DPR = 0
3448 010572 032713 000200  BIT    #DRY,DR3      ;DRY IN RHDS1 SHOULD BE = 1
3449 010576 001403          BEQ    5$             ;BRANCH IF DRY = 0
3450 010600 032713 000100  BIT    #VV,DR3       ;VV IN RHDS1 SHOULD BE = 0
3451 010604 001403          BEQ    6$             ;BRANCH IF VV = 0 (GOOD)
3452 010606 010337 001122  5$:  MOV    R3,DR3$BDADR  ;FAILING REGISTER ADDRESS - RHDS1
3453 010612 104012          ERROR  12           ;WITH SPINDLE POWERED UP
3454          ;RHDS1 SHOULD HAVE VV = 0, MOL = 1
3455
3456 010614 011100          6$:  MOV    DR1,RO
3457 010616 042700 160076  BIC    #SC!TRE!MCPE!76,RO ;CLEAR SC,TRE,MCPE AND
3458          ;ALL FUNCTION BITS
3459 010622 022700 004200  CMP    #DVA!RDY,RO   ;RHCS1 SHOULD HAVE
3460          ;GO = 0, DVA = 1, RDY = 1
3461 010626 001403          BEQ    7$             ;BRANCH IF RHCS1 IS GOOD
3462 010630 010137 001122  MOV    R1,DR3$BDADR ;FAILING REGISTER RHCS1
3463 010634 104013          ERROR  13           ;AFTER A POWER UP WITHOUT ANY
3464          ;INIT RHCS1 SHOULD HAVE

```



```

3521 010752 004037 035440 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
3522 010756 002240 RHDS1 ;CHANGE RHDS1 REGISTER
3523
3524 010760 000001 1 ;1 BIT/BITS TO BE CHANGED
3525 010762 000001 1 ;NEW VALUE OF VV IS 1
3526 010764 000100 VV ;CHANGE VV BIT
3527
3528 ;NOW COMPARE REGISTERS SO THAT NO REGISTERS
3529 ;CHANGED EXCEPT VV IN RHDS1 AND IE IN RHDS1
3530
3531
3532 010766 004037 035546 JSR RO,3#COMREG ;COMPARE SAVED REGISTERS WITH
3533 ;PRESENT VALUE
3534 010772 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
3535 010774 002266 WC ;TEST DATA STARTING FROM 'RHWC'
3536 010776 000022 18. ;18. REGISTERS TO BE COMPARED
3537 011000 011004 11$ ;RETURN TO 11$ ON ERROR
3538 011002 011010 12$ ;RETURN TO 12$ ON NO ERROR
3539
3540 011004 104015 11$: ERROR 15 ;GIVING A PACK ACKNOWLEDGE
3541 011006 000207 RTS PC ;CAUSED AN ERROR
3542 ;PACK ACKNOWLEDGE SHOULD
3543 ;SET VV IN RHDS1
3544 ;INTERRUPT SHOULD MAKE
3545 ;IE = 0
3546 ;NO OTHER REGISTERS SHOULD
3547 ;CHANGE
3548 ;GOOD DATA GIVES
3549 ;CONTENTS OF REGISTER BEFORE
3550 ;PACK ACKNOWLEDGE
3551 ;RECEIVED DATA GIVES
3552 ;CONTENTS OF REGISTER
3553 ;AFTER PACK ACKNOWLEDGE
3554 011010 012737 177777 041404 12$: MOV #-1,2#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
3555
3556
3557
3558
3559
3560 ;*****
3561 ;*TEST 10 SET VV BIT #6 IN RHDS1
3562 ;* THIS TEST SETS VV IN RHDS1 INCASE
3563 ;* ACT-11 MONITOR IS PRESENT AND THE PREVIOUS TEST
3564 ;* IS NOT PERFORMED
3565 ;* THERE IS A RESET AT THE BEGINING OF THIS TEST
3566 ;* FOR ERROR RECOVERY ONLY.
3567 ;*****
3568 011016 000004 TST10: SCOPE
3569
3570 ;IN CASE THERE IS ANY DRIVE ERRORS DURING POWER UP
3571 ;OR POWER DOWN OR ANY PARITY ERRORS A RESET IS GIVEN
3572 011020 000005 RESET
3573 011022 004737 040510 JSR PC,2#STKINT ;INITILIZE TK
3574
3575
3576 011026 012706 001000 MOV #STACK,SP ;RESET STACK

```

MACY11 271732) 17-SEP-76 13:23 PAGE 79  
SET VV BIT #6 IN RHDS1

MACY11 271732) 17-SEP-76 13:23 PAGE 79  
SET VV BIT #6 IN RHDS1

```

011032 012737 000010 004514      MOV      #10, R0TSTNM      ;SAVE TEST NUMBER
011040 004737 004556              JSR      PC, R0CLDISK      ;SET R1-RMCS1, R2-RMCS2
                                ;R3-RHDS1, R4-RHER1
                                ;GIVE RM-11 INITIALIZE
                                ;SETUP UNIT NUMBER
011044 004737 004614      JSR      PC, R0CHECK      ;CHECK DVA, RC, MOL, OPP, DRY
011050 012777 002372 171140      MOV      R0PKACK, R0RCS1  ;GET READY FOR PKACK
                                ;PACK ACKNOWLEDGE WITH 22 IN RMCS1

                                ;NOW SAVE REGISTERS FOR COMPARISON AFTER PACK ACKNOWLEDGE
011058 004716      JSR      R0, R0SAVER      ;SAVE REGISTERS
                                ;R0C IS THE FIRST REGISTER SAVED
                                ;STARTING ADDRESS OF WHERE
                                ;THE REGISTERS ARE SAVED
                                ;NUMBER OF REGISTERS
                                ;SAVED = 18.
011070 012777 004516 171106      MOV      R0RPHVEC, R0PVEC ;SET R0P04 VECTOR ADDRESS
                                ;TO "TIME1" IF P-CLOCK IS PRESENT
                                ;OR "TIME2" IF P-CLOCK IS NOT PRESENT
                                ;"TIME" WILL ONLY SAVE
                                ;CURRENT CYLINDER ADDRESS
                                ;AND LOCK AHEAD REGISTERS
011076 012746 002372      MOV      R0PKACK, -(SP)   ;GET READY TO MOVE COMMAND
011102 052716 000001      BIS      #R0C, (SP)      ;GET READY TO SET GO
                                ;WITHOUT INTERRUPT ENABLE
011106 012677 171104      MOV      (SP)+, R0RCS1   ;GO WITH
                                ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
                                ;WITH INTERRUPT DISABLED
011112 011100      MOV      R0R1, R0C      ;SAVE RHCS1 DURING ABOVE OPERATION
011114 011305      MOV      R0R3, R0S      ;SAVE RHDS1 DURING ABOVE OPERATION
011116 104410      WAIT                      ;WAIT FOR VV BIT TO SET
011120 002240      RHDS1                    ;WAIT FOR RHDS1 REGISTER
011122 000100      VV                      ;WAIT FOR VV BIT IN RHDS1 REGISTER
011124 000001      I.                      ;ALLOW 10 MICRO SECONDS
011126 000001      I.                      ;VV MUST SET BETWEEN
                                ;00 AND 20 MICRO SECONDS
                                ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
                                ;R0 AND R5 IMMEDIATELY AFTER GO
011130 012746 002372      MOV      R0PKACK, -(SP)   ;SAVE COMMAND
011134 052716 004200      BIS      #DVA!RDY, (SP)  ;INCLUDE DVA!RDY
011140 011637 001124      MOV      (SP), R0$GDDAT  ;SAVE FOR PRINTOUT
011144 022600      CMP      (SP)+, R0      ;DURING ABOVE OPERATION ONLY DVA!RDY
                                ;AND COMMAND SHOULD BE SET
011146 001405      BEQ      645             ;BRANCH IF GOOD
011150 010027 001126      MOV      R0, R0$GDDAT    ;BAD DATA
011154 010137 004510      MOV      R0, R0$REGADR   ;FAILING REGISTER RHCS1

```

```

3633 011160 104021          ERROR 21          ; DURING ABOVE OPERATION ONLY
3634          ; COMMAND AND DVA:RDY SHOULD BE SET
3635 011162 012746 010700 64$: MOV #MOL:DPR:DRY:VV. -(SP) ; SAVE BITS SET DURING OPERATION IN RHDS1
3636 011165 011537 001124      MOV (SP),D#$GDDAT ; SAVE FOR PRINTOUT
3637 011172 022605      CMP (SP)+,R5 ; DURING ABOVE OPERATION ONLY MOL:DPR:DRY:VV
3638          ; SHOULD BE SET
3639 011174 001405      SEQ 66$ ; BRANCH IF GOOD
3640 011176 010537 001126      MOV R5,D#$BDDAT ; BAD DATA
3641 011202 010337 004510      MOV R3,D#$REGADR ; FAILING REGISTER RHDS!
3642 011206 104063      ERROR 63 ; DURING ABOVE OPERATION ONLY
3643          ; MOL:DPR:DRY:VV SHOULD BE SET
3644 011210          66$:
3645 011210 004037 035440      JSR RO,D#CHREG ; CHANGE BITS IN SAVED REGISTER
3646 011214 002240      RHDS1 ; CHANGE RHDS1 REGISTER
3647 011216 000001      1 ; 1 BIT/BITS TO BE CHANGED
3648 011220 000001      1 ; NEW VALUE OF VV IS 1
3649 011222 000100      VV ; CHANGE VV BIT
3650          ; NOW COMPARE REGISTERS BEFORE PACK ACKNOWLEDGE
3651          ; WITH AFTER PACK ACKNOWLEDGE
3652 011224 004037 035546      JSR RO,D#COMREG ; COMPARE SAVED REGISTERS WITH
3653          ; PRESENT VALUE
3654 011230 004522      SAVERE ; GOOD DATA SAVED IN 'SAVERE'
3655 011232 002266      WC ; TEST DATA STARTING FROM 'RHWC'
3656 011234 000022      18. ; 18. REGISTERS TO BE COMPARED
3657 011236 011242      1$ ; RETURN TO 1$ ON ERROR
3658 011240 011246      2$ ; RETURN TO 2$ ON NO ERROR
3659 011242 104015 15:      ERROR 15 ; GIVING A PACK ACKNOWLEDGE
3660 011244 000207      RTS PC ; CAUSED AN ERROR
3661          ; PACK ACKNOWLEDGE SHOULD
3662          ; SET VV IN RHDS1
3663          ; INTERRUPT SHOULD MAKE
3664          ; IE = 0
3665          ; NO OTHER REGISTERS SHOULD
3666          ; CHANGE
3667          ; GOOD DATA GIVES CONTENTS
3668          ; OF REGISTER BEFORE COMMAND
3669          ; RECEIVED DATA GIVES CONTENTS
3670          ; OF REGISTER AFTER COMMAND
3671 011246 2$:
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688

```

```

*****
*TEST 11 NO OPERATION FUNCTION TEST
* ALL POSSIBLE REGISTERS ARE CLEARED THEN A"NCP"=0
* IS GIVEN NO CHANGE SHOULD HAPPEN

```

```

3689          :*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES THEN A "NOP"
3690          :*      IS GIVEN NO CHANGE SHOULD HAPPEN
3691
3692          :*****
3693 011246 000004          TST11: SCOPE
3694 011250 012706 001000      MOV      #STACK,SP      ;RESET STACK
3695 011254 012737 000011 004514  MOV      #11,#TSTNM     ;SAVE TEST NUMBER
3696
3697 011262 004737 034556      JSR      PC,#CLDISK     ;SET R1-RHCS1, R2-RHCS2
3698                          ;R3-RHDS1, R4-RHER1
3699                          ;GIVE RH-11 INITIALIZE
3700
3701 011256 004737 034636      JSR      PC,#CHECKT     ;SETUP UNIT NUMBER
3702                          ;CHECK DVA,RDY,MOL,DPR,DRY,VV
3703
3704 011272 013777 002334 170716  MOV      @#NOPERA,@RHCS1 ;GET READY FOR NOPERA
3705                          ;NO OPERATION WITH 0 IN RHCS1
3706
3707                          ;NOW SAVE REGISTERS FOR COMPARISON AFTER NO OPERATION
3708
3709 011300 004037 034716      JSR      R0,@SAVER      ;SAVE REGISTERS
3710                          ;RHWC IS THE FIRST REGISTER SAVED
3711 011304 002210          RHWC
3712 011306 004522          SAVERE
3713                          ;STARTING ADDRESS OF WHERE
3714                          ;THE REGISTERS ARE SAVED
3715 011310 000022          18.
3716                          ;NUMBER OF REGISTERS
3717                          ;SAVED = 18.
3718
3719 011312 013777 004516 170664  MOV      @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
3720                          ;TO 'TIME1' IF P-CLOCK IS PRESENT
3721                          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3722                          ;'TIME' WILL ONLY SAVE
3723                          ;CURRENT CYLINDER ADDRESS
3724                          ;AND LOOK AHEAD REGISTERS
3725
3726 011320 013746 002334      MOV      @#NOPERA,-(SP) ;GET READY TO MOVE COMMAND
3727 011324 052716 000001      BIS      #GO,(SP)      ;SET READY TO SET GO
3728                          ;WITHOUT INTERRUPT ENABLE
3729 011330 012677 170662      MOV      (SP)+,@RHCS1  ;GO WITH
3730                          ;0 IN RHCS1 FOR NO-OPERATION
3731                          ;WITH INTERRUPT DISABLED
3732
3733
3734 011334 104410          WAT
3735 011336 002216          RHCS1
3736 011340 000200          RDY
3737 011342 000001          1.
3738 011344 000001          1.
3739                          ;WAIT FOR RDY BIT TO SET
3740                          ;WAIT FOR RHCS1 REGISTER
3741                          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
3742                          ;ALLOW 10 MICRO SECONDS
3743                          ;RDY MUST SET BETWEEN
3744                          ;00 AND 20 MICRO SECONDS
3745
3746                          ;NOW COMPARE REGISTERS BEFORE NO-OP COMMAND
3747                          ;WITH AFTER NO-OP COMMAND
3748 011346 004037 035546      JSR      R0,@COMREG     ;COMPARE SAVED REGISTERS WITH

```



```

3745 011352 004522          SAVERE          ;PRESENT VALUE
3746 011354 002266          WC             ;GOOD DATA SAVED IN 'SAVERE'
3747 011356 000022          18.           ;TEST DATA STARTING FROM 'RHWC'
3748 011360 011364          1$            ;18. REGISTERS TO BE COMPARED
3749 011362 011370          2$            ;RETURN TO 1$ ON ERROR
3750                                     ;RETURN TO 2$ ON NO ERROR
3751 011364 104016          1$:          ERROR 16          ;GIVING A NO-OP COMMAND
3752 011366 000207          RTS          PC          ;CAUSED AN ERROR
3753                                     ;NO REGISTERS SHOULD CHANGE
3754                                     ;GOOD DATA GIVES REGISTER
3755                                     ;CONTENTS BEFORE COMMAND
3756                                     ;RECEIVED DATA GIVES REGISTER
3757                                     ;CONTENTS AFTER COMMAND
3758
3759                                     ;NOW REPEAT TEST BY MOVING IN ALL POSSIBLE ONES
3760
3761 C11370          2$:
3762
3763 011370 004737 034556          JSR          PC, @CLDISK ;SET R1-RHCS1, R2-RHCS2
3764                                     ;R3-RHDS1, R4-RHER1
3765                                     ;GIVE RH-11 INITIALIZE
3766                                     ;SETUP UNIT NUMBER
3767 011374 012700 002210          MOV          #RHWC, R0    ;ADDR. OF ADDR OF RHWC IN R0
3768 011400 012730 177777          MOV          #177777, @R0+ ;LOAD 177777 INTO RHWC
3769
3770 011404 012730 177777          MOV          #177777, @R0+ ;LOAD 177777 INTO RHBA
3771
3772 011410 052730 043010          BIS          #43010, @R0+ ;LOAD 43010 INTO RHCS2
3773 011414 012730 001400          MOV          #1400, @R0+  ;LOAD 1400 INTO RHCS1
3774
3775 011420 012730 000000          MOV          #0, @R0+    ;LOAD 0 INTO RHER1
3776
3777 011424 012730 177777          MOV          #177777, @R0+ ;LOAD 177777 INTO RHDST
3778
3779 011430 012730 000000          MOV          #0, @R0+    ;LOAD 0 INTO RHER2
3780
3781 011434 012730 177777          MOV          #177777, @R0+ ;LOAD 177777 INTO RHOF
3782
3783 011440 012730 177777          MOV          #177777, @R0+ ;LOAD 177777 INTO RHCA
3784
3785 011444 012730 000000          MOV          #0, @R0+    ;LOAD 0 INTO RHER3
3786
3787                                     ;NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
3788
3789 011450 010046          MOV          R0, -(SP)    ;; PUSH R0 ON STACK
3790 011452 011446          MOV          @R4, -(SP)  ;SAVE RHER1 TO REINSTATE LATER
3791 011454 011246          MOV          @R2, -(SP)  ;SAVE RHCS2 TO BE REINSTATED
3792                                     ;AFTER ALL ATA BITS HAVE BEEN SET
3793 011456 013700 004652          MOV          @TOTALAT, R0 ;GET DRIVES PRESENT
3794 011462 005012          CLR          @R2         ;CLEAR RHCS2 AND CARRY
3795 011464 012705 000010          MOV          #8., R5     ;COUNTER
3796 011470 006000          87$:          ROR          R0         ;GET BIT INTO CARRY
3797 011472 103002          BCC          88$         ;BRANCH IF NO UNIT ON THIS BIT
3798 C11474 012714 177777          MOV          #-1, @R4    ;MOVE INTO ERROR REGISTER
3799                                     ;TO SET ATA BIT
3800 011500 005212          88$:          INC          @R2         ;INCREMENT RHCS2 TO NEXT UNIT

```



```

3857
3858 011616 043737 004650 004546      BIC      @*ATTENT,@*SAVERE+24      ;CLEAR APPROPRIATE ATA BITS
3859                                          ;FOR WORKING DRIVE IN SAVED PHAS
3860
3861 011624 004037 035440      JSR      RO,@*CHREG      ;CHANGE BITS IN SAVED REGISTER
3862 011630 002240      RHDS1    ;CHANGE RHDS1 REGISTER
3863
3864 011632 000001      I      ;1 BIT/BITS TO BE CHANGED
3865 011634 000000      C      ;NEW VALUE OF ATA IS 0
3866 011636 100000      ATA     ;CHANGE ATA BIT
3867
3868
3869
3870
3871
3872
3873 011640 004037 035546      JSR      RO,@*COMREG      ;COMPARE SAVED REGISTERS WITH
3874                                          ;PRESENT VALUE
3875 011644 004522      SAVERE  ;GOOD DATA SAVED IN 'SAVERE'
3876 011646 002266      WC      ;TEST DATA STARTING FROM 'RHWC'
3877 011650 000022      18.    ;18. REGISTERS TO BE COMPARED
3878 011652 011656      3$     ;RETURN TO 3$ ON ERROR
3879 011654 011662      4$     ;RETURN TO 4$ ON NO ERROR
3880
3881 011656 104016      3$:    ERROR 16      ;GIVING A NO-OP COMMAND
3882 011660 000207      RTS    PC          ;CAUSED AN ERROR
3883                                          ;NO REGISTERS SHOULD CHANGE
3884                                          ;GOOD DATA GIVES REGISTER
3885                                          ;CONTENTS BEFORE COMMAND
3886                                          ;RECEIVED DATA GIVES REGISTER
3887                                          ;CONTENTS AFTER COMMAND
3888 011662      4$:
3889
3890
3891

```

21

```

3892
3893
3894
3895
3896
3897
3898
3899 011662 000007
3900 011664 012706 001000
3901 011670 012737 000012 004514
3902
3903 011676 004737 034556
3904
3905
3906
3907 011702 004737 034636
3908
3909 011706 012700 002210
3910 011712 012730 177777
3911
3912 011716 012730 177776
3913
3914 011722 052730 043010
3915 011726 012730 001400
3916
3917 011732 012730 000000
3918
3919 011736 012730 177777
3920
3921 011742 012730 000000
3922
3923 011746 012730 177777
3924
3925 011752 012730 177777
3926
3927 011756 012730 000000
3928
3929
3930
3931 011762 010046
3932 011764 011446
3933 011766 011246
3934
3935 011770 013700 004652
3936 011774 005012
3937 011776 012705 000010
3938 012002 005000
3939 012004 103002
3940 012006 012714 177777
3941
3942 012012 005212
3943 012014 005305
3944 012016 001371
3945 012020 012612
3946 012022 012614
3947 012024 012600

```

```

*****
*TEST 12 DRIVE CLEAR
* ALL WRITE BITS OF ALL REGISTERS ARE FILLED
* WITH ONES EXCEPT GO, CLR, IE, PAT, MCPE, UPE
* THEN A DRIVE CLEAR IS GIVEN
* THEN ALL REGISTERS ARE CHECKED TO HAVE APPROPRIATE VALUE
*****
↑ST12: SCOPE
MOV #STACK, SP ;RESET STACK
MOV #12, @↑STNM ;SAVE TEST NUMBER
JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
JSR PC, @#CHECKT ;CHECK DVA, RDY, MOL, DPR, DR1, VV
MOV #RHC, RO ;ADDR. OF ADDR. OF RHC IN RO
MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHC
MOV #177776, @ (RO)+ ;LOAD 177776 INTO RHBA
BIS #43010, @ (RO)+ ;LOAD 43010 INTO RHCS2
MOV #1400, @ (RO)+ ;LOAD 1400 INTO RHCS1
MOV #0, @ (RO)+ ;LOAD 0 INTO RHER1
MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHDST
MOV #0, @ (RO)+ ;LOAD 0 INTO RHER2
MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHC5
MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHCA
MOV #0, @ (RO)+ ;LOAD 0 INTO RHER3
;NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
MOV RO, -(SP) ;PUSH RO ON STACK
MOV @R4, -(SP) ;SAVE RHER1 TO REINSTATE LATER
MOV @R2, -(SP) ;SAVE RHCS2 TO BE REINSTATED
;AFTER ALL ATA BITS HAVE BEEN SET
MOV @TOTALAT, RO ;GET DRIVES PRESENT
CLR @R2 ;CLEAR RHCS2 AND CARRY
MOV #8, R5 ;COUNTER
ROR RO ;GET BIT INTO CARRY
BCC 85$ ;BRANCH IF NO UNIT ON THIS BIT
MOV INTO ERROR REGISTER
;TO SET ATA BIT
85$: INC @R2 ;INCREMENT RHCS2 TO NEXT UNIT
DEC R5 ;COUNT
BNE 84$ ;BRANCH IF 8 NOT DONE
MOV (SP)+, @R2 ;REINSTATE RHCS2
MOV (SP)+, @R4 ;REINSTATE RHER1
MOV (SP)+, RO ;POP STACK INTO RO

```

```

3948 012026 005720          TST      (R0)+          ;GET OVER PHAS IN R0
3949
3950
3951 012030 012730 177776    MOV      #177776,(R0)+ ;LOAD 177776 INTO RHMR
3952
3953
3954 012034 017737 170212 004564  MOV      @RHCC,@#SAVERE+42 ;SAVE RHCC IN SAVERE TABLE
3955 012042 013777 004516 170134  MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
3956                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
3957                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
3958                                     ;'TIME' WILL ONLY SAVE
3959                                     ;CURRENT CYLINDER ADDRESS
3960                                     ;AND LOOK AHEAD REGISTERS
3961
3962
3963 012050 013746 002342    MOV      @#DCLEAR,-(SP) ;GET READY TO MOVE COMMAND
3964 012054 052716 000001    BIS      #GC,(SP)      ;GET READY TO SET GO
3965                                     ;WITHOUT INTERRUPT ENABLE
3966 012060 012677 170132    MOV      (SP)+,@RHCSI  ;GO WITH
3967                                     ;IO IN RHCSI FOR DRIVE CLEAR
3968                                     ;WITH INTERRUPT DISABLED
3969
3970
3971
3972 012064 104410          WAT                      ;WAIT FOR RDY BIT TO SET
3973 012066 002216          RHCSI                   ;WAIT FOR RHCSI REGISTER
3974 012070 000200          RDY                     ;WAIT FOR RDY BIT IN RHCSI REGISTER
3975 012072 000001          1.                      ;ALLOW 10 MICRO SECONDS
3976 012074 000001          1.                      ;RDY MUST SET BETWEEN
3977                                     ;00 AND 20 MICRO SECONDS
3978 012076 005737 004632    TST      @#NUNIT
3979 012102 001003          BNE
3980 012104 042737 100000 004530  BIC      #100000,@#SAVERE+6
3981
3982                                     ;NOW LOAD 'SAVERE' WITH EXPECTED VALUES
3983 012112 004037 034460    JSR      RO,@#FILLRE    ;MOV 177777 INTO SAVED RHWC
3984 012116 002210          RHJC                   ;SAVED REGISTER TO CHANGE
3985 012120 177777          177777                 ;DATA
3986 012122 004037 034460    JSR      RO,@#FILLRE    ;MOV 177776 INTO SAVED RHBA
3987 012126 002212          RHBA                   ;SAVED REGISTER TO CHANGE
3988 012130 177776          177776                 ;DATA
3989 012132 005037 004526    CLR      @#SAVERE+4     ;CLEAR LOCATION FOR RHCS2
3990 012136 053737 004626 004526  BIS      @#UNIT,@#SAVERE+4 ;PUT UNIT # BACK IN THE SAVED RHCS2
3991
3992 012144 004037 035440    JSR      RO,@#CHREG     ;CHANGE BITS IN SAVED REGISTER
3993 012150 002214          RHCS2                  ;CHANGE RHCS2 REGISTER
3994
3995 012152 000002          2                       ;2 BIT/BITS TO BE CHANGED
3996 012154 000001          1                       ;NEW VALUE OF IR IS 1
3997 012156 000100          IR                     ;CHANGE IR BIT
3998 012160 000001          1                       ;NEW VALUE OF BAI IS 1
3999 012162 000010          BAI                     ;CHANGE BAI BIT
4000 012164 004037 034460    JSR      RO,@#FILLRE    ;MOV 0 INTO SAVED RHER1
4001 012170 002220          RHER1                  ;SAVED REGISTER TO CHANGE
4002 012172 000000          0                       ;DATA
4003 012174 004037 034460    JSR      RO,@#FILLRE    ;MOV 17437 INTO SAVED RHOST

```

4004	012200	002222		RHDST		: SAVED REGISTER TO CHANGE
4005	012202	017437		17437		: DATA
4006	012204	004037	034460	JSR	RO, @#FILLRE	: MOV 0 INTO SAVED RHER2
4007	012210	002224		RHER2		: SAVED REGISTER TO CHANGE
4008	012212	000000		0		: DATA
4009	012214	004037	034460	JSR	RO, @#FILLRE	: MOV 116000 INTO SAVED RHOF
4010	012220	002226		RHOF		: SAVED REGISTER TO CHANGE
4011	012222	116000		116000		: DATA
4012	012224	004037	034450	JSR	RO, @#FILLRE	: MOV 1777 INTO SAVED RHCA
4013	012230	002230		RHCA		: SAVED REGISTER TO CHANGE
4014	012232	001777		1777		: DATA
4015	012234	004037	034460	JSR	RO, @#FILLRE	: MOV 0 INTO SAVED RHER3
4016	012240	002232		RHER3		: SAVED REGISTER TO CHANGE
4017	012242	000000		0		: DATA
4018	012244	013746	004652	MOV	@#TOTALAT, -(SP)	: GET ALL BITS OF DRIVE & PRESENT
4019						: IN RHAS
4020	012250	043716	004650	BIC	@#ATTENT, (SP)	: CLEAR WORKING DRIVE BIT
4021	012254	012637	004546	MOV	(SP)+, @#SAVERE+24	: MOVE THIS INTO RHAS POSITION
4022						: OF SAVERE TABLE
4023						
4024	012260	004037	034460	JSR	RO, @#FILLRE	: MOV 400 INTO SAVED RHMR
4025	012264	002236		RHMR		: SAVED REGISTER TO CHANGE
4026	012266	000400		400		: DATA
4027						
4028	012270					3\$:
4029	012270	004037	034460	JSR	RO, @#FILLRE	: MOV 10700 INTO SAVED RHDS1
4030	012274	002240		RHDS1		: SAVED REGISTER TO CHANGE
4031	012276	010700		10700		: DATA
4032						
4033	012300	013737	002320	004554	4\$:	MOV @#DT, @#SAVERE+32
4034						: MOVE DRIVE TYPE INTO RHD
4035	012306	013737	002322	004556		MOV @#SN, @#SAVERE+34
4036						: MOVE SERIAL NUMBER INTO RSN
4037						: POSITION OF SAVERE TABLE
4038	012314	004037	034460	JSR	RO, @#FILLRE	: MOV 0 INTO SAVED RHEC1
4039	012320	002246		RHEC1		: SAVED REGISTER TO CHANGE
4040	012322	000000		0		: DATA
4041	012324	004037	034460	JSR	RO, @#FILLRE	: MOV 0 INTO SAVED RHEC2
4042	012330	002250		RHEC2		: SAVED REGISTER TO CHANGE
4043	012332	000000		0		: DATA
4044						
4045	012334	004037	035440	JSR	RO, @#CHREG	: CHANGE BITS IN SAVED REGISTER
4046	012340	002216		RHCS1		: CHANGE RHCS1 REGISTER
4047						
4048	012342	000001		1		: 1 BIT/BITS TO BE CHANGED
4049	012344	000001		1		: NEW VALUE OF PAR IS 1
4050	012346	000010		PAR		: CHANGE PAR BIT
4051						
4052						: NOW THAT SAVERE TABLE HAS BEEN LOADED WITH
4053						: EXPECTED VALUES, THE REGISTERS WILL BE COMPARED
4054						: WITH SAVERE TABLE
4055						
4056						
4057	012350	004037	035546	JSR	RO, @#COMREG	: COMPARE SAVED REGISTERS WITH
4058						: PRESENT VALUE
4059	012354	004 72		SAVERE		: GOOD DATA SAVED IN 'SAVERE'

4060 012356 002266  
4061 012360 000022  
4062 012362 012366  
4063 012364 012372  
4064  
4065 012366 104017  
4066 012370 000207  
4067  
4068  
4069  
4070  
4071  
4072  
4073 012372  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091  
4092  
4093  
4094  
4095  
4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104 012372 000004  
4105 012374 012706 001000  
4106 012400 012737 000013 004514  
4107  
4108 012406 004737 034556  
4109  
4110  
4111  
4112 012412 005737 004634  
4113 012416 001402  
4114 012420 000167 000462  
4115 012424 005737 000042

WC ;TEST DATA STARTING FROM 'FHW' ;  
18. ;18. REGISTERS TO BE COMPARED ;  
5\$ ;RETURN TO 5\$ ON ERROR ;  
6\$ ;RETURN TO 6\$ ON NO ERROR ;  
5\$: ERROR 17 ;DRIVE CLEAR COMMAND ;  
RTS PC ;GAVE AN ERROR ;  
;GOOD DATA HAS WHAT SHOULD ;  
;BE IN REGISTER AFTER A ;  
;DRIVE CLEAR ;  
;RECEIVED DATA HAS WHAT ;  
;THE REGISTER ACTUALLY ;  
;CONTAINED ;

6\$:

\*\*\*\*\*

\*TEST 13 READ-IN-PRESET  
\* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED  
\*  
\* IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
\* THEN THIS TEST IS NOT PERFORMED  
\*  
\* IF NO ACT-11 MONITOR IS PRESENT THEN  
\* THIS TEST IS PERFORMED ONLY ON THE FIRST PASS  
\* ON SUBSEQUENT PASSES THIS TEST IS NOT DONE  
\*  
\* THIS TESTS THE READ-IN-PRESET COMMAND  
\* FIRST THE DRIVE IS POWERED DOWN AND UP IN ORDER TO  
\* RESET VV-BIT #6 IN RHDS1  
\* THEN ALL WRITE BITS OF ALL REGISTERS ARE FILLED  
\* WITH ONES EXCEPT GO, CLR, IE, PAT, MCPE, UPE  
\* ATA FOR DRIVE UNDER TEST IS MADE = 0  
\*  
\* THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN  
\* THEN ALL REGISTERS ARE TESTED  
\* THE FOLLOWING SHOULD BE THE REGISTER CONTENTS  
\* RHCA = 0, RHDST = 0, RHOF SHOULD HAVE  
\* FMT22 = 0, ECI = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1  
\* ALL OTHER REGISTERS SHOULD BE UNCHANGED  
\*

\*\*\*\*\*

TST13: SCOPE  
MOV #STACK, SP ;RESET STACK  
MOV #13, @#TSTNM ;SAVE TEST NUMBER  
JSR PC, @#CLODISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
TST @#NOPUSH ;IS THIS 220 START  
BEQ 7\$ ;BRANCH IF NO  
JMP TST14 ;JUMP TO NEXT TEST  
7\$: TST @#42 ;MONITOR (ACT-11) RETURN ADDRESS

```

4116 012430 001004      BNE      1$          ;BRANCH IF NO MONITOR
4117 012432 005737 001100  TST      2#$PASS    ;IS THIS FIRST PASS
4118 012436 001001      BNE      1$          ;BRANCH IF NOT FIRST PASS
4119 012440 000402      BR       2$          ;BRANCH TO CONTINUE TEST
4120 012442                1$:
4121 012442 000167 000440  JMP      TST14      ;JUMP TO NEXT TEST
4122 012446                2$:
4123 012446 104400 012454  TYPE     ,+4        ;:TYPE ASCIZ STRING
4124 012452 000407      BR       64$        ;:GET OVER THE ASCIZ
4125                ;:ASCIZ      <15><12>/STOP DRIVE /
4126 012472                64$:
4127 012472 013746 004626  MOV      2#UNIT,-(SP) ;GET UNIT UNDER TEST
4128 012476 104404      TYPDS
4129 012500 104400 001221  TYPE     ,$CRLF
4130 012504 032713 010000  3$:      BIT      #MOL,2R3    ;MOL WILL BE HIGH TILL STOP IS HIT
4131 012510 001375      BNE      3$          ;WAIT TILL STOP IS HIT
4132 012512 104400 012520  TYPE     ,+4        ;:TYPE ASCIZ STRING
4133 012516 000407      BR       65$        ;:GET OVER THE ASCIZ
4134                ;:ASCIZ      <15><12>/START DRIVE/
4135 012536                65$:
4136 012536 013746 004626  MOV      2#UNIT,-(SP) ;GET UNIT UNDER TEST
4137 012542 104404      TYPDS
4138 012544 104400 001221  TYPE     , $CRLF
4139 012550 032713 010000  4$:      BIT      #MOL,2R3    ;MOL WILL BE LOW TILL FILE READY
4140 012554 001775      BEQ     4$          ;WAIT TILL FILE READY
4141
4142 012556 004737 034556  JSR      PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
4143                ;R3-RHDS1, R4-RHER1
4144                ;GIVE RH-11 INITIALIZE
4145                ;SETUP UNIT NUMBER
4146 012562 004737 034614  .TSR     PC,2#CHECK  ;CHECK DVA, RDY, MOL, DPR, DRY
4147
4148 012566 012700 002210  MOV      #RHWC,RO    ;ADDR. OF ADDR. OF RHWC IN RO
4149
4150 012572 012730 177777  MOV      #177777,2(RO)+ ;LOAD 177777 INTO RHWC
4151
4152 012576 012730 177777  MOV      #177777,2(RO)+ ;LOAD 177777 INTO RHBA
4153
4154 012602 052730 043010  BIS      #43010,2(RO)+ ;LOAD 43010 INTO RHCS2
4155 012606 012730 001400  MOV      #1400,2(RO)+ ;LOAD 1400 INTO RHCS1
4156
4157 012612 012730 000000  MOV      #0,2(RO)+   ;LOAD 0 INTO RHER1
4158
4159 012616 012730 177777  MOV      #177777,2(RO)+ ;LOAD 177777 INTO RHDST
4160
4161 012622 012730 000000  MOV      #0,2(RO)+   ;LOAD 0 INTO RHER2
4162
4163 012626 012730 177777  MOV      #177777,2(RO)+ ;LOAD 177777 INTO RHOF
4164
4165 012632 012730 177777  MOV      #177777,2(RO)+ ;LOAD 177777 INTO RHCA
4166
4167 012636 012730 000000  MOV      #0,2(RO)+   ;LOAD 0 INTO RHER3
4168
4169                ;NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
4170
4171 012642 010046  MOV      RO,-(SP)    ;:PUSH RO ON STACK

```



4172	012644	011446		MOV	@R4,-(SP)	;SAVE RHER1 TO REINSTATE LATER
4173	012646	011246		MOV	@R2,-(SP)	;SAVE RHCS2 TO BE REINSTATED
4174						;AFTER ALL ATA BITS HAVE BEEN SET
4175	012650	013700	004652	MOV	@TOTALAT,R0	;GET DRIVES PRESENT
4176	012654	005012		CLR	@R2	;CLEAR RHCS2 AND CARRY
4177	012656	012705	000010	MOV	#8,R5	;COUNTER
4178	012662	006000		86\$: ROR	R0	;GET BIT INTO CARRY
4179	012664	103002		BCC	87\$	;BRANCH IF NO UNIT ON THIS BIT
4180	012666	012714	177777	MOV	#-1,@R4	;MOVE INTO ERROR REGISTER
4181						;TO SET ATA BIT
4182	012672	005212		87\$: INC	@R2	;INCREMENT RHCS2 TO NEXT UNIT
4183	012674	005305		DEC	R5	;COUNT
4184	012676	001371		BNE	86\$	;BRANCH IF 8 NOT DONE
4185	012700	012612		MOV	(SP)+,@R2	;REINSTATE RHCS2
4186	012702	012614		MOV	(SP)+,@R4	;REINSTATE RHER1
4187	012704	012600		MOV	(SP)+,R0	;POP STACK INTO R0
4188	012706	005720		TST	(R0)+	;GET OVER PHAS IN R0
4189						
4190						
4191	012710	012730	177776	MOV	#177776,@(R0)+	;LOAD 177776 INTO RHMR
4192						
4193						
4194	012714	053777	004650 167312	BIS	@ATTENT,@RHAS	;CLEAR WORKING DRIVE ATA
4195						;BIT IN RHAS
4196						
4197	012722	013777	002374 167266	MOV	@READIN,@RHCS1	;GET READY FOR READIN
4198						;READ IN WITH 20 IN RHCS1
4199						
4200						
4201						;NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
4202	012730	004037	034716	JSR	R0,@SAVER	;SAVE REGISTERS
4203	012734	002210		RHWC		;RHWC IS THE FIRST REGISTER SAVED
4204	012736	004522		SAVERE		;STARTING ADDRESS OF WHERE
4205						;THE REGISTERS ARE SAVED
4206	012740	000022		18.		;NUMBER OF REGISTERS
4207						;SAVED = 18.
4208						
4209	012742	013777	004516 167234	MOV	@RPO4VEC,@RPVEC	;SET RPO4 VECTOR ADDRESS
4210						;TO 'TIME1' IF P-CLOCK IS PRESENT
4211						;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4212						; 'TIME' WILL ONLY SAVE
4213						;CURRENT CYLINDER ADDRESS
4214						;AND LOOK AHEAD REGISTERS
4215						
4216	012750	013746	002374	MOV	@READIN,-(SP)	;GET READY TO MOVE COMMAND
4217	012754	052716	000001	BIS	#GO,(SP)	;GET READY TO SET GO
4218						;WITHOUT INTERRUPT ENABLE
4219	012760	012677	167232	MOV	(SP)+,@RHCS1	;GO WITH
4220						;20 IN RHCS1 FOR READ IN
4221						;WITH INTERRUPT DISABLED
4222						
4223						
4224						
4225	012764	104410		WAIT		;WAIT FOR VV BIT TO SET
4226	012766	002240		RHDS1		;WAIT FOR RHDS1 REGISTER
4227	012770	000100		VV		;WAIT FOR VV BIT IN RHDS1 REGISTER

```

4228 012772 000001      1.      ;ALLOW 10 MICRO SECONDS
4229 012774 000001      1.      ;VV MUST SET BETWEEN
4230                                     ;00 AND 20 MICRO SECONDS
4231
4232                                     ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
4233 012776 004037 034460 JSR      RO,@#FILLRE ;MOV 0 INTO SAVED RHCA
4234 013002 002230      RHCA      ;SAVED REGISTER TO CHANGE
4235 013004 000000      0          ;DATA
4236 013006 004037 034460 JSR      RO,@#FILLRE ;MOV 0 INTO SAVED RHDST
4237 013012 002222      RHDST     ;SAVED REGISTER TO CHANGE
4238 013014 000000      0          ;DATA
4239
4240 013016 004037 035440 JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
4241 013022 002226      RHOF      ;CHANGE RHOF REGISTER
4242
4243 013024 000003      3          ;3 BIT/BITS TO BE CHANGED
4244 013026 000000      0          ;NEW VALUE OF FMT22 IS 0
4245 013030 010000      FMT22    ;CHANGE FMT22 BIT
4246 013032 000000      0          ;NEW VALUE OF ECI IS 0
4247 013034 004000      ECI      ;CHANGE ECI BIT
4248 013036 000000      0          ;NEW VALUE OF HCI IS 0
4249 013040 002000      HCI      ;CHANGE HCI BIT
4250
4251 013042 004037 035440 JSR      RO,@#CHREG  ;CHANGE BITS IN SAVED REGISTER
4252 013046 002240      RHDS1    ;CHANGE RHDS1 REGISTER
4253
4254 013050 000001      1          ;1 BIT/BITS TO BE CHANGED
4255 013052 000001      1          ;NEW VALUE OF VV IS 1
4256 013054 000100      VV      ;CHANGE VV BIT
4257
4258                                     ;NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
4259                                     ;THE EXPECTED VALUE AFTER A READ-IN COMMAND
4260                                     ;COMPARISONS ARE MADE
4261
4262
4263 013056 004037 035546 JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
4264                                     ;PRESENT VALUE
4265 013062 004522      SAVERE   ;GOOD DATA SAVED IN 'SAVERE'
4266 013064 002266      WC      ;TEST DATA STARTING FROM 'RHWC'
4267 013066 000022      18.     ;18. REGISTERS TO BE COMPARED
4268 013070 013074      5$      ;RETURN TO 5$ ON ERROR
4269 013072 013100      6$      ;RETURN TO 6$ ON NO ERROR
4270
4271 013074 104020      5$:      ERROR   20      ;READ IN COMMAND GAVE AN
4272 013076 000207      RTS      PC      ;ERROR
4273                                     ;GOOD DATA HAS WHAT SHOULD
4274                                     ;BE IN REGISTER AFTER A
4275                                     ;READ-IN COMMAND
4276                                     ;RECEIVED DATA HAS WHAT
4277                                     ;THE REGISTER ACTUALLY CONTAINED
4278                                     ;THE FOLLOWING SHOULD
4279                                     ;BE THE REGISTER CONTENTS
4280                                     ;RHCA=0, RHDST = 0
4281                                     ;RHOF SHOULD HAVE FMT22 = 0.
4282                                     ;HCI = 0, ECI = 0.
4283                                     ;RHDS1 SHOULD HAVE VV = 1

```

- 172

MACY 11 27 1975 17-SEP-75 13:23 PAGE 92  
READ-IN-PRESET

MACY 11 27 1975  
READ-IN-PRESET

MACY 11 27 1975

018100 018727 177777 0440- 55: NO. 8-1.88PRITEM

: ALL OTHER BITS SHOULD  
: BE UNCHANGED  
: CLEAR PREVIOUS ITEM NUMBER

4298  
4299  
4300  
4301  
4302  
4303  
4304  
4305  
4306  
4307  
4308  
4309  
4310  
4311  
4312  
4313  
4314  
4315  
4316  
4317  
4318  
4319  
4320  
4321  
4322  
4323  
4324  
4325  
4326  
4327  
4328  
4329  
4330  
4331  
4332  
4333  
4334  
4335  
4336  
4337  
4338  
4339  
4340  
4341  
4342  
4343

\*\*\*\*\*  
:TEST 14 READ-IN-PRESET

:\* THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT  
:\* THAT IT DOES NOT TEST THE SETTING OF VV

:\* THIS TEST IS HERE BECAUSE IF ACT-11 MONITOR IS PRESENT  
:\* THEN THE PREVIOUS TEST WILL NOT BE PERFORMED  
:\* THIS TESTS THE READ-IN-PRESET COMMAND  
:\* ALL WRITE BITS OF ALL REGISTERS ARE FILLED  
:\* WITH ONES EXCEPT GO, CLR, IE, PAT, MCPE, UPE  
:\* ATA FOR DRIVE UNDER TEST IS MADE = 0

:\* THEN A READ-IN-PRESET COMMAND = 20 IS GIVEN  
:\* THEN ALL REGISTERS ARE TESTED  
:\* THE FOLLOWING SHOULD BE THE REGISTER CONTENTS  
:\* RHCA = 0, RHDST = 0, RHOF SHOULD HAVE  
:\* FMT22 = 0, ECT = 0, HCI = 0, RHDS1 SHOULD HAVE VV = 1  
:\* ALL OTHER REGISTERS SHOULD BE UNCHANGED

\*\*\*\*\*

013106 000004  
013110 012706 001000  
013114 012737 000014 004514  
013122 004737 034556  
013126 004737 034614  
013132 012700 002210  
013136 012730 177777  
013142 012730 177777  
013146 052730 043010  
013152 012730 001400  
013156 012730 000000  
013162 012730 177777  
013166 012730 000000  
013172 012730 177777  
013176 012730 177777  
013202 012730 000000  
013205 012730 177777  
013212 012730 177776

†ST14: SCOPE  
MOV #STACK, SP ;RESET STACK  
MOV #14, 2†STNM ;SAVE TEST NUMBER  
JSR PC, 2#CLDISK ;SET =:-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER  
JSR PC, 2#CHECK ;CHECK DVA, RDY, MOL, DPR, DRY  
MOV #RHWC, R0 ;ADDR. OF ADDR. OF RHWC IN R0  
MOV #177777, 2(R0)+ ;LOAD 177777 INTO RHWC  
MOV #177777, 2(R0)+ ;LOAD 177777 INTO RHBA  
BIS #43010, 2(R0)+ ;LOAD 43010 INTO RHCS2  
MOV #1400, 2(R0)+ ;LOAD 1400 INTO RHCS1  
MOV #0, 2(R0)+ ;LOAD 0 INTO RHER1  
MOV #177777, 2(R0)+ ;LOAD 177777 INTO RHDST  
MOV #0, 2(R0)+ ;LOAD 0 INTO RHER2  
MOV #177777, 2(R0)+ ;LOAD 177777 INTO RHOF  
MOV #177777, 2(R0)+ ;LOAD 177777 INTO RHCA  
MOV #0, 2(R0)+ ;LOAD 0 INTO RHER3  
MOV #-1, 2(R0)+ ;CLEAR ALL BITS OF RHAS  
MOV #177776, 2(R0)+ ;LOAD 177776 INTO RHMR

4344	013216	053777	004650	167010	BIS	Q#ATTENT,QRHAS	:CLEAR WORKING DRIVE ATA :BIT IN RHAS
4345							
4346	013224	013777	002374	166754	MOV	Q#READIN,QRHCS1	:GET READY FOR READIN :READ IN WITH 20 IN RHCS1
4347							
4348							
4349							
4350							
4351							
4352	013232	004037	034716				:NOW SAVE REGISTERS FOR COMPARISON AFTER READ-IN COMMAND
4353	013236	002210			JSR	RO,Q#SAVER	:SAVE REGISTERS
4354	013240	004522			RHWC		:RHWC IS THE FIRST REGISTER SAVED
4355					SAVERE		:STARTING ADDRESS OF WHERE
4356	013242	000022			18.		:THE REGISTERS ARE SAVED
4357							:NUMBER OF REGISTERS
4358							:SAVED = 18.
4359							
4360	013244	013777	004516	166732	MOV	Q#RP4VEC,QRPVEC	:SET RPO4 VECTOR ADDRESS
4361							:TO 'TIME1' IF P-CLOCK IS PRESENT
4362							:OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4363							: 'TIME' WILL ONLY SAVE
4364							:CURRENT CYLINDER ADDRESS
4365							:AND LOOK AHEAD REGISTERS
4366	013252	013746	002374		MOV	Q#READINI,-(SP)	:GET READY TO MOVE COMMAND
4367	013256	052716	000001		BIS	#GO,(SP)	:GET READY TO SET GO
4368							:WITHOUT INTERRUPT ENABLE
4369	013262	012677	166730		MOV	(SP)+,QRHCS1	:GO WITH
4370							:WITH INTERRUPT DISABLED
4371							
4372							
4373							
4374	013266	104410			WAT		:WAIT FOR RDY BIT TO SET
4375	013270	002216			RHCS1		:WAIT FOR RHCS1 REGISTER
4376	013272	000200			RDY		:WAIT FOR RDY BIT IN RHCS1 REGISTER
4377	013274	000001			1.		:ALLOW 10 MICRO SECONDS
4378	013276	000001			1.		:RDY MUST SET BETWEEN
4379							:00 AND 20 MICRO SECONDS
4380							
4381							
4382	013300	004037	034460				:NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
4383	013304	002230			JSR	RO,Q#FILLRE	:MOV 0 INTO SAVED RHCA
4384	013306	000000			RHCA		:SAVED REGISTER TO CHANGE
4385	013310	004037	034460		0		:DATA
4386	013314	002222			JSR	RO,Q#FILLRE	:MOV 0 INTO SAVED RHDST
4387	013316	000000			RHDST		:SAVED REGISTER TO CHANGE
4388					0		:DATA
4389	013320	004037	035440		JSR	RO,Q#CHREG	:CHANGE BITS IN SAVED REGISTER
4390	013324	002226			RHOF		:CHANGE RHOF REGISTER
4391							
4392	013326	000003			3		:3 BIT BITS TO BE CHANGED
4393	013330	000000			0		:NEW VALUE OF FMT22 IS 0
4394	013332	010000			FMT22		:CHANGE FMT22 BIT
4395	013334	000000			0		:NEW VALUE OF ECI IS 0
4396	013336	004000			ECI		:CHANGE ECI BIT
4397	013340	000000			0		:NEW VALUE OF HCI IS 0
4398	013342	002000			HCI		:CHANGE HCI BIT
4399							

```

4400 013344 004037 035440 JSR RO,3#CHREG ;CHANGE BITS IN SAVED REGISTER
4401 013350 002240 RHDS1 ;CHANGE RHDS1 REGISTER
4402
4403 013352 000001 1 ;1 BIT/BITS TO BE CHANGED
4404 013354 000001 1 ;NEW VALUE OF VV IS 1
4405 013356 000100 VV ;CHANGE VV BIT
4406
4407 ;NOW THAT SAVERE TABLE WITH SAVED REGISTERS HAVE
4408 ;THE EXPECTED VALUE AFTER A READ-IN COMMAND
4409 ;COMPARISONS ARE MADE
4410
4411 013360 004037 035546 JSR RO,3#COMREG ;COMPARE SAVED REGISTERS WITH
4412 ;PRESENT VALUE
4413 013364 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4414 013366 002266 WC ;TEST DATA STARTING FROM 'RAWC'
4415 013370 000022 18. ;18. REGISTERS TO BE COMPARED
4416 013372 013376 5$ ;RETURN TO 5$ ON ERROR
4417 013374 013402 6$ ;RETURN TO 6$ ON NO ERROR
4418
4419 013376 104020 5$: ERROR 20 ;READ-IN COMMAND GAVE AN
4420 013400 000207 RTS 70 ;ERROR
4421
4422 ;GOOD DATA HAS WHAT SHOULD
4423 ;BE IN REGISTER AFTER A
4424 ;READ-IN COMMAND
4425 ;RECEIVED DATA HAS WHAT
4426 ;THE REGISTER ACTUALLY CONTAINS
4427 ;THE FOLLOWING SHOULD
4428 ;BE THE REGISTER CONTENTS
4429 ;RHCA = 0, RHDST = 0
4430 ;RHOF SHOULD HAVE FMT22 = 0.
4431 ;HCI = 0, ECI = 0
4432 ;RHDS1 SHOULD HAVE VV = 1
4433 ;ALL OTHER BITS SHOULD
4434 ;BE UNCHANGED
4435
4436 013402 6$:

```

```

4437
4438
4439
4440
4441
4442
4443
4444
4445 013402 000004
4446 013404 012706 001000
4447 013410 012737 000015 004514
4448
4449 013416 004737 034556
4450
4451
4452
4453 013422 004737 034636
4454
4455 013426 012700 002210
4456
4457 013432 012730 177777
4458
4459 013436 012730 177776
4460
4461 013442 052730 000010
4462 013446 012730 001400
4463
4464 013452 012730 000000
4465
4466 013456 012730 177777
4467
4468 013462 012730 000000
4469
4470 013466 012730 177777
4471
4472 013472 012730 177777
4473
4474 013476 012730 000000
4475
4476
4477
4478 013502 010046
4479 013504 011446
4480 013506 011246
4481
4482 013510 013700 004652
4483 013514 005012
4484 013516 012705 000010
4485 013522 006000
4486 013524 103002
4487 013526 012714 177777
4488
4489 013532 005212
4490 013534 005305
4491 013536 001371
4492 013540 012612

:*****
;*TEST 15 RECALIBRATE COMMAND
;*
;* ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
;* THEN A RECALIBRATE = 6 COMMAND IS GIVEN
;* NO REGISTERS SHOULD CHANGE EXCEPT RHCC = 0
:*****
†ST15: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #15,‡†STNM ;SAVE TEST NUMBER
JSR PC,‡‡CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUMBER
JSR PC,‡‡CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
MOV #RHWC,RO ;ADDR. OF ADDR. OF RHWC IN RO
MOV #177777,‡(RO)+ ;LOAD 177777 INTO RHWC
MOV #177776,‡(RO)+ ;LOAD 177776 INTO RHBA
BIS #010,‡(RO)+ ;LOAD 010 INTO RHCS2
MOV #1400,‡(RO)+ ;LOAD 1400 INTO RHCS1
MOV #0,‡(RO)+ ;LOAD 0 INTO RHER1
MOV #177777,‡(RO)+ ;LOAD 177777 INTO RHDST
MOV #0,‡(RO)+ ;LOAD 0 INTO RHER2
MOV #177777,‡(RO)+ ;LOAD 177777 INTO RHOF
MOV #177777,‡(RO)+ ;LOAD 177777 INTO RHCA
MOV #0,‡(RO)+ ;LOAD 0 INTO RHER3
;NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
MOV RO,-(SP) ;;PUSH RO ON STACK
MOV ‡R4,-(SP) ;;SAVE RHER1 TO REINSTATE LATER
MOV ‡R2,-(SP) ;;SAVE RHCS2 TO BE REINSTATED
;AFTER ALL ATA BITS HAVE BEEN SET
MOV ‡‡TOTALAT,RO ;GET DRIVES PRESENT
CLR ‡R2 ;CLEAR RHCS2 AND CARRY
MOV #8,R5 ;COUNTER
84$: ROR RO ;GET BIT INTO CARRY
;GET BIT INTO CARRY
BCC 85$ ;BRANCH IF NO UNIT ON THIS BIT
MOV ‡-1,‡R4 ;MOVE INTO ERROR REGISTER
;TO SET ATA BIT
85$: INC ‡R2 ;INCREMENT RHCS2 TO NEXT UNIT
DEC R5 ;COUNT
BNE 84$ ;BRANCH IF 8 NOT DONE
MOV (SP)+,‡R2 ;REINSTATE RHCS2

```

```

4493 013542 012614      MOV      (SP)+, @R4      ;REINSTATE RHER1
4494 013544 012600      MOV      (SP)+, R0      ;POP STACK INTO R0
4495 013546 005720      TST      (R0)+         ;GET OVER PHAS IN R0
4496
4497
4498 013550 012730 177776    MOV      #177776, @ (R0)+ ;LOAD 177776 INTO RHMR
4499
4500
4501
4502 013554 013777 002340 166434  MOV      @#RECALI, @RHCS1 ;GET READY FOR RECALI
4503                                     ;RECALIBRATE WITH 6 IN RHCS1
4504
4505                                     ;NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
4506 013562 004037 034716    JSR      R0, @#SAVER    ;SAVE REGISTERS
4507 013566 002210          RHWC      ;RHWC IS THE FIRST REGISTER SAVED
4508 013570 004522          SAVERE    ;STARTING ADDRESS OF WHERE
4509                                     ;THE REGISTERS ARE SAVED
4510 013572 000022          18.      ;NUMBER OF REGISTERS
4511                                     ;SAVED = 18.
4512
4513 013574 013777 004516 166402  MOV      @#RP4VEC, @RPVEC ;SET RP04 VECTOR ADDRESS
4514                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
4515                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
4516                                     ;'TIME' WILL ONLY SAVE
4517                                     ;CURRENT CYLINDER ADDRESS
4518                                     ;AND LOOK AHEAD REGISTERS
4519
4520
4521 013602 013746 002340    MOV      @#RECALI, -(SP) ;GET READY TO MOVE COMMAND
4522 013606 052716 000101    BIS      #GO!IE, (SP)   ;GET READY TO SET GO AND
4523                                     ;ENABLE INTERRUPT
4524 013612 012677 166400    MOV      (SP)+, @RHCS1  ;GO WITH
4525                                     ;6 IN RHCS1 FOR RECALIBRATE
4526                                     ;WITH INTERRUPT ENABLED
4527 013616 011100          MOV      @R1, R0       ;SAVE RHCS1 DURING ABOVE OPERATION
4528 013620 011305          MOV      @R3, R5       ;SAVE RHDS1 DURING ABOVE OPERATION
4529
4530
4531 013622 104410          WAT      ;WAIT FOR DRY BIT TO SET
4532 013624 002240          RHDS1   ;WAIT FOR RHDS1 REGISTER
4533 013626 000200          DRY     ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4534 013630 076377          31999. ;ALLOW 319990 MICRO SECONDS
4535 013632 056701          24001. ;DRY MUST SET BETWEEN
4536                                     ;79980 AND 560000 MICRO SECONDS
4537                                     ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4538                                     ;R0 AND R5 IMMEDIATELY AFTER GO
4539 013634 013746 002340    MOV      @#RECALI, -(SP) ;SAVE COMMAND
4540 013640 052716 004301    BIS      #DVA!GO!IE!RDY, (SP) ;INCLUDE DVA!GO!IE!RDY
4541 013644 005737 004632    TST      @#NUNIT      ;ARE THERE MORE THAN ONE UNIT
4542 013650 001413          BEQ     89$          ;BRANCH IF ONLY ONE UNIT
4543 013652 010037 004660    MOV      R0, @#TMP4    ;GET RHCS1
4544 013656 042737 177677 004660  BIC      #!CIE, @#TMP4 ;KEEP IE BIT
4545 013664 042716 000100          BIC     #IE, (SP)    ;CLEAR IE IN GOOD DATA
4546 013670 053716 004660          BIS     @#TMP4, (SP) ;GET IE AS IS
4547 013674 052716 100000          BIS     #SC, (SP)   ;SET SC IN RHCS1
4548 013700

```

89\$:

*Handwritten mark*





MAINDEC-11-DERPU-B  
DERPUB.P11

MACY11 27(732) 17-SEP-76 13:23 PAGE 99  
RECALIBRATE COMMAND

4605 01404  
4606  
4607  
4608  
4609  
4610  
4611

25:

```

4612 :*****
4613 ;*TEST 16 RECALIBRATE COMMAND
4614 ;* ALL POSSIBLE REGISTERS ARE FILLED WITH 0
4615 ;* THEN A RECALIBRATE =6 COMMAND IS GIVEN
4616 ;* NO REGISTERS SHOULD CHANGE EXCEPT RHCC=0
4617 :*****
4618 :*****
4619 †ST16: SCOPE
4620 014040 000004 MOV #STACK,SP ;RESET STACK
4621 014042 012706 001000 MOV #16,2#1 ;SAVE TEST NUMBER
4622 014046 012737 000016 004514 JSR PC,2#CLC,5K ;SET R1-RHCS1, R2-RHCS2
4623 014054 004737 034556 ;R3-RHDS1, R4-RHER1
4624 ;GIVE RH-11 INITIALIZE
4625 ;SETUP UNIT NUMBER
4626 014060 004737 034636 JSR PC,2#CHECKT ;CHECK DVA, RDY, MOL, DPR, DRY, VV
4627 014064 012700 002210 MOV #RHWC,RO ;ADDR. OF ADDR OF RHWC IN RO
4628
4629
4630 014070 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHWC
4631
4632
4633
4634 014074 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHBA
4635
4636
4637
4638
4639
4640 014100 052730 000000 BIS #0,2(RO)+ ;LOAD 0 INTO RHCS2
4641
4642
4643 014104 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHCS1
4644
4645
4646
4647 014110 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHER1
4648
4649
4650
4651 014114 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHDS1
4652
4653
4654
4655 014120 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHER2
4656
4657
4658
4659 014124 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHOF
4660
4661
4662
4663 014130 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHCA
4664
4665
4666
4667 014134 012730 000000 MOV #0,2(RO)+ ;LOAD 0 INTO RHER3

```

31

```

4668
4669
4670
4671 014140 012730 177777      MOV      #-1,@(RO)+      ;CLEAR ALL BITS OF RHAS
4672
4673
4674 014144 012730 000000      MOV      #0,@(RO)+      ;LOAD 0 INTO RHMR
4675
4676 014150 013777 002340 166040  MOV      @#RECALI,@RHCSI ;GET READY FOR RECALI
4677
4678
4679
4680
4681
4682 014156 004037 034716      JSR      RO,@#SAVER      ;NOW SAVE REGISTERS FOR COMPARISON AFTER RECALIBRATE
4683 014162 002210      RHWC      ;SAVE REGISTERS
4684 014164 004522      S: VERE      ;RHWC IS THE FIRST REGISTER SAVED
4685
4686 014166 000022      18.          ;STARTING ADDRESS OF WHERE
4687
4688
4689 014170 013777 004516 166006  MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
4690
4691
4692
4693
4694
4695
4696
4697 014176 013746 002340      MOV      @#RECALI,-(SP)  ;GET READY TO MOVE COMMAND
4698 014202 052716 000101      BIS      #GO!IE,(SP)    ;GET READY TO SET GO AND
4699
4700 014206 012677 166004      MOV      (SP)+,@RHCSI   ;ENABLE INTERRUPT
4701
4702
4703 014212 011100      MOV      @R1,RO        ;GO WITH
4704 014214 011305      MOV      @R3,R5        ;6 IN RHCSI FOR RECALIBRATE
4705
4706
4707 014216 104410      WAT          ;WITH INTERRUPT ENABLED
4708 014220 002240      RHDS1      ;SAVE RHCSI DURING ABOVE OPERATION
4709 014222 000200      DRY        ;SAVE RHDS1 DURING ABOVE OPERATION
4710 014224 076377      31999.     ;WAIT FOR DRY BIT TO SET
4711 014226 056701      24001.     ;WAIT FOR RHDS1 REGISTER
4712
4713
4714
4715 014230 013746 002340      MOV      @#RECALI,-(SP)  ;WAIT FOR DRY BIT IN RHDS1 REGISTER
4716 014234 052716 004301      BIS      #DVA!GO!IE!RDY,(SP) ;ALLOW 319990 MICRO SECONDS
4717 014240 011637 001124      MOV      (SP),@#SGDDAT  ;DRY MUST SET BETWEEN
4718 014244 022600      CMP      (SP)+,RO      ;79980 AND 560000 MICRO SECONDS
4719
4720 014246 001405      BEQ      88$          ;COMPARE CONTENTS OF RHCSI AND RHDS1 ALREADY SAVED IN
4721 014250 010037 001126      MOV      RO,@#SBDDAT    ;RO AND R5 IMMEDIATELY AFTER GO
4722 014254 010137 004510      MOV      R1,@#REGADR   ;SAVE COMMAND
4723 014260 104021      ERROR     21          ;INCLUDE DVA!GO!IE!RDY
                        ;SAVE FOR PRINTOUT
                        ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
                        ;AND COMMAND SHOULD BE SET
                        ;BRANCH IF GOOD
                        ;BAD DATA
                        ;FAILING REGISTER RHCSI
                        ;DURING ABOVE OPERATION ONLY

```

```

4724                                     ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
4725 014262 012746 030500 88$: MOV #MOL!DPR!VV!PIP, -(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
4726 014266 011637 001124 MOV (SP), @#$GDDAT ;SAVE FOR PRINTOUT
4727 014272 022605 CMP (SP)+, R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV!PIP
4728                                     ;SHOULD BE SET
4729 014274 001405 BEQ 90$ ;BRANCH IF GOOD
4730 014276 010537 001126 MOV R5, @#$BDDAT ;BAD DATA
4731 014302 010337 004510 MOV R3, @#$REGADR ;FAILING REGISTER RHDS1
4732 014306 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
4733                                     ;MOL!DPR!VV!PIP SHOULD BE SET
4734 014310 90$:
4735
4736 ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
4737 014310 004037 034460 JSR RO, @#$FILLRE ;MOV 0 INTO SAVED RHCC
4738 014314 002252 RHCC ;SAVED REGISTER TO CHANGE
4739 014316 000000 0 ;DATA
4740
4741 014320 053737 004650 004546 BIS @#$ATTENT, @#$SAVERE+24 ;SET APPROPRIATE ATA BITS
4742                                     ;FOR WORKING DRIVE IN
4743                                     ;SAVED RHAS LOACTION
4744
4745 014326 004037 035440 JSR RO, @#$CHREG ;CHANGE BITS IN SAVED REGISTER
4746 014332 002240 RHDS1 ;CHANGE RHDS1 REGISTER
4747
4748 014334 000001 1 ;1 BIT/BITS TO BE CHANGED
4749 014336 000001 1 ;NEW VALUE OF ATA IS 1
4750 014340 100000 ATA ;CHANGE ATA BIT
4751
4752 014342 004037 035440 JSR RO, @#$CHREG ;CHANGE BITS IN SAVED REGISTER
4753 014346 002216 RHCS1 ;CHANGE RHCS1 REGISTER
4754
4755 014350 000001 1 ;1 BIT/BITS TO BE CHANGED
4756 014352 000001 1 ;NEW VALUE OF SC IS 1
4757 014354 100000 SC ;CHANGE SC BIT
4758 ;NOW COMPARE REGISTERS AFTER A RECALIBRATE COMMAND
4759
4760 014356 004037 035546 JSR RO, @#$COMREG ;COMPARE SAVED REGISTERS WITH
4761                                     ;PRESENT VALUE
4762 014362 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
4763 014364 002266 WC ;TEST DATA STARTING FROM 'RHWC'
4764 014366 000022 18. ;18. REGISTERS TO BE COMPARED
4765 014370 014374 1$ ;RETURN TO 1$ ON ERROR
4766 014372 014400 2$ ;RETURN TO 2$ ON NO ERROR
4767
4768 014374 104064 1$: ERROR 64 ;RECALIBRATE COMMAND CAUSED
4769 014376 000207 RTS PC ;AN ERROR
4770                                     ;GOOD DATA GIVES WHAT SHOULD BE
4771                                     ;THERE
4772                                     ;RECEIVED DATA GIVES WHAT WAS
4773 014400 2$: ;THERE AFTER A RECALIBRATE
4774
4775 ;*****
4776 ;*TEST 17 UNLOAD COMMAND
4777
4778 ;* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED
4779 ;*

```

```

4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796 014400 000004
4797 014402 012706 001000
4798 014406 012737 000017 004514
4799
4800 014414 004737 034556
4801
4802
4803
4804 014420 005737 004634
4805 014424 001402
4806 014426 000167 001212
4807 014432 005737 000042
4808 014436 001004
4809 014440 005737 001100
4810 014444 001001
4811 014446 000402
4812 014450
4813 014450 000167 001170
4814
4815
4816 014454 017746 165560
4817 014460 042716 167677
4818 014464 022726 010100
4819 014470 001501
4820 014472 104400 014500
4821 014476 000427
4822
4823 014556
4824 014556 104400 014564
4825 014562 000424
4826
4827 014634
4828 014634 000000
4829
4830 014636 004737 034556
4831
4832
4833
4834 014642 004737 034614
4835

```

```

:*      IF THE PROGRAM WORKS UNDER ACT-11 MONITOR
:*      THEN THIS TEST IS NOT PERFORMED
:*
:*      IF NO ACT-11 MONITOR IS PRESENT
:*      THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS
:*      ON SUBSEQUENT PASSES THIS TEST IS NOT DONE
:*
:*      ALL POSSIBLE REGISTERS ARE FILLED WITH ONES
:*      THEN AN UNLOAD COMMAND =2 IS GIVEN
:*      NO REGISTERS SHOULD CHANGE EXCEPT MOL SHOULD=0
:*      THEN THE DRIVE IS POWERED UP BY OPERATOR
:*      AND A PACK ACKNOWLEDGE COMMAND (ALREADY TESTED)
:*      SETS VV-IN RHDS1
:*****
TST17: SCOPE
MOV    #STACK, SP      ;RESET STACK
MOV    #17, @#TSTNM    ;SAVE TEST NUMBER
JSR    PC, @#CLDISK    ;SET R1-RHCS1, R2-RHCS2
                        ;R3-RHDS1, R4-RHER1
                        ;GIVE RH-11 INITIALIZE
                        ;SETUP UNIT NUBER
TST    @#NOPUSH        ;IS THIS 220 START
BEQ    8$              ;BRANCH IF NO
JMP    TST20           ;JUMP TO NEXT TEST
8$:    TST    @#42      ;MONITOR (ACT-11) RETURN ADDRESS
BNE    1$              ;BRANCH IF MONITOR PRESENT
TST    @#$PASS        ;IS THIS FIRST PASS
BNE    1$              ;BRANCH IF NOT FIRST PASS
BR     2$              ;BRANCH TO CONTINUE TEST
1$:    JMP    TST20     ;JUMP TO NEXT TEST
        ;THIS SETTING OF VV IS FOR LOOP ON ERROR ONLY
        ;WHERE UNLOAD TAKES EFFECT AND POWER UP BRINGS VV DOWN
2$:    MOV    @RHDS1, -(SP) ;GET RHDS1
        BIC    #167677, (SP) ;CLEAR EVERYTHING EXCEPT VV AND MOL
        CMP    #VV!MOL, (SP)+ ;ARE VV AND MOL SET
        BEQ    6$              ;BRANCH IF YES
        TYPE  @#+4            ;TYPE ASCIZ STRING
        BR     @#4$          ;GET OVER THE ASCIZ
        ;;.ASCIZ    <15><12>/GET DRIVE HEADS LOADED THEN HIT "CONTINUE"/
64$:   TYPE  @#+4            ;TYPE ASCIZ STRING
        BR     @#5$          ;GET OVER THE ASCIZ
        ;;.ASCIZ    <15><12>/IF ALREADY LOADED THEN HIT "CONTINUE"/
65$:   HALT                    ;WAIT FOR CONTINUE
JSR    PC, @#CLDISK    ;SET R1-RHCS1, R2-RHCS2
                        ;R3-RHDS1, R4-RHER1
                        ;GIVE RH-11 INITIALIZE
                        ;SETUP UNIT NUBER
JSR    PC, @#CHECK     ;CHECK DVA.RDY, MOL, DPR.DRY

```

```

4836
4837
4838
4839 014646 013746 002372
4840 014652 052716 000001
4841
4842 014656 012677 165334
4843
4844
4845
4846
4847
4848 014662 104410
4849 014664 002240
4850 014665 000100
4851 014670 000001
4852 014672 000001
4853
4854 014674
4855 014674 004737 034636
4856
4857
4858 014700 012700 002210
4859
4860
4861
4862 014704 012730 177777
4863
4864
4865
4866 014710 012730 177777
4867
4868
4869
4870 014714 052730 000010
4871
4872
4873 014720 012730 001400
4874
4875
4876
4877 014724 012730 000000
4878
4879
4880
4881 014730 012730 177777
4882
4883
4884
4885 014734 012730 000000
4886
4887
4888
4889 014740 012730 177777
4890
4891

```

```

:SET VV IN RHDS1 WITH PACK ACKNOWLEDGE
MOV @#PKACK, -(SP) ;GET READY TO MOVE COMMAND
BIS #GO, (SP) ;GET READY TO SET GO
;WITHOUT INTERRUPT ENABLE
MOV (SP)+, @RHCS1 ;GO WITH
;22 IN RHCS1 FOR PACK ACKNOWLEDGE
;WITH INTERRUPT DISABLED

WAT ;WAIT FOR VV BIT TO SET
RHDS1 ;WAIT FOR RHDS1 REGISTER
VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
1. ;ALLOW 10 MICRO SECONDS
1. ;VV MUST SET BETWEEN
;00 AND 20 MICRO SECONDS

6$: JSR PC, @#CHECKT ;CHECK DVA, RDY, MOL, DPR, DRY, VV

MOV #RHWC, RO ;ADDR. OF ADDR OFRHWC IN RO
;LOAD ALL POSSIBLE REGISTERS WITH ONES
MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHWC

MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHBA

BIS #10, @ (RO)+ ;LOAD 10 INTO RHCS2

MOV #1400, @ (RO)+ ;LOAD 1400 INTO RHCS1

MOV #0, @ (RO)+ ;LOAD 0 INTO RHER1

MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHDST

MOV #0, @ (RO)+ ;LOAD 0 INTO RHER2

MOV #177777, @ (RO)+ ;LOAD 177777 INTO RHOF

```

```

11444 012730 177777 MOV #177777,2(R0)+ ;LOAD 177777 INTO RHCA
114753 012730 000000 MOV #0,2(R0)+ ;LOAD 0 INTO RHER3

;NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT

MOV R0,-(SP) ;:PUSH R0 ON STACK
MOV R24,-(SP) ;:SAVE RHER1 TO REINSTATE LATER
MOV R22,-(SP) ;:SAVE RHCS2 TO BE REINSTATED
;:AFTER ALL ATA BITS HAVE BEEN SET
MOV #TOTALAT,RC ;:SET DRIVES PRESENT
MOV R2,0 ;:CLEAR RHCS2 AND CARRY
MOV R0,R5 ;:COUNTER
895: R0,R5 ;:SET BIT INTO CARRY
BCC 905 ;:BRANCH IF NO UNIT ON THIS BIT
MOV #1,R5 ;:MOVE INTO ERROR REGISTER
;:TO SET ATA BIT
905: INC R2 ;:INCREMENT RHCS2 TO NEXT UNIT
DEC R5 ;:COUNT
BNE 895 ;:BRANCH IF 8 NOT DONE
MOV (SP)+,R2 ;:REINSTATE RHCS2
MOV (SP)+,R24 ;:REINSTATE RHER1
MOV (SP)+,R0 ;:POP STACK INTO R0
TST (R0)+ ;:SET OVER PHAS IN R0

115022 012730 177776 MOV #177776,2(R0)+ ;LOAD 177776 INTO RHMR
115026 013777 002336 165162 MOV 2#UNLOAD,2RHCS1 ;GET READY FOR UNLOAD
;UNLOAD WITH 2 IN RHCS1

;NOW SAVE REGISTERS FOR COMPARISON AFTER UNLOAD
JSR R0,2#SAVER ;SAVE REGISTERS
RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRESS OF WHERE
;THE REGISTERS ARE SAVED
17. ;NUMBER OF REGISTERS
;SAVED = 17.

115046 013777 004516 165130 MOV 2#RP4VEC,2RPVEC ;SET RP04 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
;TIME WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS

115054 013746 002336 MOV 2#UNLOAD,-(SP) ;GET READY TO MOVE COMMAND

```



```

4948 015060 052716 000001      BIS      #GO, (SP)      ;GET READY TO SET GO
4949 015064 012677 165126      MOV      (SP)+, @RHCS1 ;WITHOUT INTERRUPT ENABLE
4950 015070 011100      MOV      @R1, RC      ;GO WITH
4951 015072 011305      MOV      @R3, R5      ;2 IN RHCS1 FOR UNLOAD
4952 015074 104410      WAT                      ;WITH INTERRUPT DISABLED
4953 015076 002216      RHCS1                    ;WAIT FOR RDY BIT TO SET
4954 015100 000200      RDY                      ;WAIT FOR RHCS1 REGISTER
4955 015102 000001      1.                      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
4956 015104 000001      1.                      ;ALLOW 10 MICRO SECONDS
4957 015106 013746 002336      MOV      @UNLOAD, -(SP) ;RDY MUST SET BETWEEN
4958 015112 052716 004201      BIS      @DVA!GO!RDY, (SP) ;00 AND 20 MICRO SECONDS
4959 015116 005737 004632      TST      @NUNIT        ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
4960 015122 001413      BEQ      94$           ;RD AND R5 IMMEDIATELY AFTER GO
4961 015124 010037 004660      MOV      R0, @TMP4     ;SAVE COMMAND
4962 015130 042737 177677      BIC      @IE, @TMP4    ;INCLUDE DVA!GO!RDY
4963 015136 042716 000100      BIC      @IE, (SP)     ;ARE THERE MORE THAN ONE UNIT
4964 015142 053716 004660      BIS      @TMP4, (SP)   ;BRANCH IF ONLY ONE UNIT
4965 015146 052716 100000      BIS      @SC, (SP)    ;GET RHCS1
4966 015152 011637 001124      MOV      (SP), @SGDDAT ;KEEP IE BIT
4967 015156 022600      CMP      (SP)+, R0     ;CLEAR IE IN GOOD DATA
4968 015160 001405      BEQ      93$           ;GET IE AS IS
4969 015162 010037 001126      MOV      R0, @SBDDAT   ;SET SC IN RHCS1
4970 015166 010137 004510      MOV      R1, @REGADR   ;SAVE FOR PRINTOUT
4971 015172 104021      ERROR    21           ;DURING ABOVE OPERATION ONLY DVA!GO!RDY
4972 015174 012746 020500      MOV      @PIP!DPR!VV, -(SP) ;AND COMMAND SHOULD BE SET
4973 015200 011637 001124      MOV      (SP), @SGDDAT ;BRANCH IF GOOD
4974 015204 022605      CMP      (SP)+, R5     ;BAD DATA
4975 015206 001405      BEQ      95$           ;FAILING REGISTER RHCS1
4976 015210 010537 001126      MOV      R5, @SBDDAT   ;DURING ABOVE OPERATION ONLY
4977 015214 010337 004510      MOV      R3, @REGADR   ;COMMAND AND DVA!GO!RDY SHOULD BE SET
4978 015220 104063      ERROR    63           ;SAVE BITS SET DURING OPERATION IN RHCS1
4979 015222      95$:                  ;SAVE FOR PRINTOUT
4980 015222 004037 035440      JSR      R0, @CHREG    ;DURING ABOVE OPERATION ONLY PIP!DPR!VV
4981 015226 002240      RHCS1                  ;SHOULD BE SET
4982 015230 000002      2                      ;BRANCH IF GOOD
4983 015232 000000      0                      ;BAD DATA
4984 015234 110200      ATA!MOL!DRY           ;FAILING REGISTER RHDS1
4985 015236 000001      1                      ;DURING ABOVE OPERATION ONLY
4986 015236 000001      1                      ;PIP!DPR!VV SHOULD BE SET
4987 015236 000001      1                      ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
4988 015236 000001      1                      ;CHANGE BITS IN SAVED REGISTER
4989 015236 000001      1                      ;CHANGE RHDS1 REGISTER
4990 015236 000001      1                      ;2 BIT/BITS TO BE CHANGED
4991 015236 000001      1                      ;NEW VALUE OF ATA!MOL!DRY IS 0
4992 015236 000001      1                      ;CHANGE A!A!MOL!DRY BIT
4993 015236 000001      1                      ;NEW VALUE OF PIP IS 1

```



```

5004 015240 020000 PIF :CHANGE PIP BIT
5005
5006 015242 004037 035440 JSR RO,2#CHREG :CHANGE BITS IN SAVED REGISTER
5007 015246 002216 RHCS1 :CHANGE RHCS1 REGISTER
5008
5009 015250 000001 I :1 BIT/BITS TO BE CHANGED
5010 015252 000001 I :NEW VALUE OF GO IS 1
5011 015254 000001 GO :CHANGE GO BIT
5012 015256 005737 004632 TST 2#NUNIT :ARE THERE MORE THAN ONE UNIT
5013 015262 001006 BNE 7$ :BRANCH IF MORE THAN ONE UNIT
5014
5015 015264 004037 035440 JSR RO,2#CHREG :CHANGE BITS IN SAVED REGISTER
5016 015270 002216 RHCS1 :CHANGE RHCS1 REGISTER
5017
5018 015272 000001 I :1 BIT/BITS TO BE CHANGED
5019 015274 000000 D :NEW VALUE OF SC IS 0
5020 015276 100000 SC :CHANGE SC BIT
5021 015300 7$:
5022
5023 015300 043737 004650 004546 BIC 2#ATTENT,2#SAVERE+24 :CLEAR APPROPRIATE ATA BITS
5024 :FOR WORKING DRIVE IN SAVED RHAS
5025
5026 :NOW COMPARE REGISTERS AFTER AN UNLOAD COMMAND
5027
5028 015306 004037 035546 JSR RO,2#COMREG :COMPARE SAVED REGISTERS WITH
5029 :PRESENT VALUE
5030 015312 004522 SAVERE :GOOD DATA SAVED IN 'SAVERE'
5031 015314 002266 WC :TEST DATA STARTING FROM 'RHWC'
5032 015316 000021 17. :17. REGISTERS TO BE COMPARED
5033 015320 015324 3$ :RETURN TO 3$ ON ERROR
5034 015322 015330 4$ :RETURN TO 4$ ON NO ERROR
5035
5036 015324 104023 3$: ERROR 23 :UNLOAD COMMAND GAVE
5037 015326 000207 RTS PC :AN ERROR
5038 :GOOD DATA GIVES WHAT SHOULD
5039 :BE THERE
5040 :RECEIVED DATA GIVES WHAT WAS
5041 :THERE AFTER UNLOAD COMMAND
5042
5043 015330 4$:
5044 015330 104400 015336 TYPE +4 :TYPE ASCIZ STRING
5045 015334 000406 BR 96$ :GET OVER THE ASCIZ
5046 :;.ASCIZ <15><12>/ON DRIVE /
5047
5048 015352 013746 004626 96$: MOV 2#UNIT,-(SP) :GET UNIT UNDER TEST
5049 015356 104404 TYPDS
5050 015360 104400 015366 TYPE +4 :TYPE ASCIZ STRING
5051 015364 000446 BR 97$ :GET OVER THE ASCIZ
5052 :;.ASCIZ <15><12>/IF STANDBY BUTTON LIT THEN HIT STANDBY. AFTER HEAD LOAD
5053
5054 015502 104400 015510 97$: TYPE +4 :TYPE ASCIZ STRING
5055 015506 000434 BR 98$ :GET OVER THE ASCIZ
5056 :;.ASCIZ <15><12>/IF STAND-BY NOT LIT THEN-ERROR AFTER UNLOAD COMMAND<15><
5057
5058 015600 005037 041404 98$: CLR 2#PRITEM :CLEAR PREVIOUS ERROR NUMBER
5059 015604 000000 HALT :WAIT FOR CONTINUE
5060 015606 004737 034614 JSR PC,2#CHECK :CHECK DVA.RDY,MOL,DPR,DRY

```

```

5060
5061
5062
5063
5064 :SET VV IN RHDS1
5065 015612 C13746 002372 MOV 2#PKACK,-(SP) ;GET READY TO MOVE COMMAND
5066 015616 052716 000001 SIS #GO,(SP) ;GET READY TO SET GO
5067 ;WITHOUT INTERRUPT ENABLE
5068 015622 012677 164370 MOV (SP)+,2#RHCS1 ;GO WITH
5069 ;22 IN RHCS1 FOR PACK ACKNOWLEDGE
5070 ;WITH INTERRUPT DISABLED
5071
5072 015626 011100 MOV 2#R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
5073 015630 011305 MOV 2#R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
5074
5075 015632 104410 WAT ;WAIT FOR VV BIT TO SET
5076 015634 002240 RHDS1 ;WAIT FOR RHDS1 REGISTER
5077 015636 000100 VV ;WAIT FOR VV BIT IN RHDS1 REGISTER
5078 015640 000001 I. ;ALLOW 10 MICRO SECONDS
5079 015642 000001 I. ;VV MUST SET BETWEEN
5080 ;00 AND 20 MICRO SECONDS

```

```

*****
: *TEST 20 OFFSET AND RETURN TO CENTER LINE COMMAND

```

```

: * THIS TESTS TWO COMMANDS (1)OFFSET (2)RETURN-TO-CENTER-LINE
: * ALL POSSIBLE REGISTERS ARE FILLED WITH ONES (EXCEPT RHOF)
: * AN OFFSET IS GIVEN
: * ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET
: * ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED
: * THEN A RETURN-TO-CENTER-LINE IS GIVEN
: * ALL REGISTERS ARE COMPARED ONLY ATA SHOULD SET
: * AND RHOF SHOULD CLEAR (EXCEPT HCI,ECI,FMT22)
: * ALL OTHER REGISTERS SHOULD REMAIN UNCHANGED
: * THE ABOVE PROCESS IS REPEATED FOR OFFSET REGISTER
: * VALUE 1 TO 377

```

```

*****
: *TEST20: SCOPE

```

```

5098 015644 000004
5099 015646 012767 015706 163232 MOV #15,$LPADR ;SET SCOPE LOOP ADDRESS
5100 015654 012706 001000 MOV #STACK,SP ;RESET STACK
5101 015660 012737 000020 004514 MOV #20,2#TSTNM ;SAVE TEST NUMBER
5102
5103 015666 004737 034556 JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
5104 ;R3-RHDS1, R4-RHER1
5105 ;GIVE RH-11 INITIALIZE
5106 ;SETUP UNIT NUMBER
5107 015672 112737 000001 004520 MOVB #1,2#OFSTVL ;SET OFFSET VALUE TO 1
5108 015700 112737 000034 004521 MOVB #34,2#OFSTVL+1 ;SET HCI,ECI,FMT22
5109
5110
5111 015706 15:
5112
5113 C15706 004737 034556 JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
5114 ;R3-RHDS1, R4-RHER1
5115 ;GIVE RH-11 INITIALIZE

```

```

S116          015712 004737 034636      JSR    PC, @#CHECKT      ; SETUP UNIT NUMBER
S117          015716 012700 002210      MOV    #RHWC, RO         ; CHECK DVA, RDY, MOL, DPR, DRY, VV
S118          015722 012730 177777      MOV    #177777, @ (RO)+ ; ADDR. OF ADDR OF RHWC IN PC
S119          015726 012730 177777      MOV    #177777, @ (RO)+ ; LOAD 177777 INTO RHWC
S120          015732 052730 000010      BIS    #10, @ (RO)+     ; LOAD 10 INTO RHCS2
S121          015736 012730 001400      MOV    #1400, @ (RO)+   ; LOAD 1400 INTO RHCS1
S122          015742 012730 000000      MOV    #0, @ (RO)+     ; LOAD 0 INTO RHER1
S123          015746 012730 177777      MOV    #177777, @ (RO)+ ; LOAD 177777 INTO RHDST
S124          015752 012730 000000      MOV    #0, @ (RO)+     ; LOAD 0 INTO RHER2
S125          015756 013730 004520      MOV    @#OFSTVL, @ (RO)+ ; THE OFFSET REGISTER WILL BE INCREMENTED FROM 0 TO 377
S126          015762 012730 177777      MOV    #177777, @ (RO)+ ; SET OFFSET REGISTER
S127          015766 012730 000000      MOV    #0, @ (RO)+     ; LOAD 177777 INTO RHCA
S128          015772 010046      MOV    RO, -(SP)        ; LOAD 0 INTO RHER3
S129          015774 011446      MOV    @R4, -(SP)       ; NOW SET BITS IN RHAS FOR ALL DRIVES PRESENT
S130          015776 011246      MOV    @R2, -(SP)       ; PUSH RO ON STACK
S131          016000 013700 004652      MOV    @#TOTALAT, RO    ; SAVE RHER1 TO REINSTATE LAT
S132          016004 005012      CLR    @R2              ; SAVE RHCS2 TO BE REINSTATED
S133          016006 012705 000010      MOV    #8, R5           ; AFTER ALL ATA BITS HAVE BEEN SET
S134          015012 006000      ROR    RO               ; GET DRIVES PRESENT
S135          015014 103002      BCC   82$              ; CLEAR RHCS2 AND CARRY
S136          016016 012714 177777      MOV    #-1, @R4        ; COUNTER
S137          016022 005212      INC    @R2              ; GET BIT INTO CARRY
S138          016024 005305      DEC    R5               ; LET BIT INTO CARRY
S139          016026 001371      BNE   82$              ; BRANCH IF NO UNIT ON THIS BIT
S140          016030 012612      MOV    (SP)+, @R2      ; MOVE INTO ERROR REGISTER
S141          016032 012614      MOV    (SP)+, @R4      ; TO SET ATA BIT
S142          016034 012600      MOV    (SP)+, R5       ; INCREMENT RHCS2 TO NEXT UNIT
S143          016036 005720      TST   (RO)+            ; COUNT
S144          016040 012730 177776      MOV    #177776, @ (RO)+ ; BRANCH IF 8 NOT DONE
S145          016044 013777 002366 164.44  MOV    @#OFSETC, @RHCS1 ; REINSTATE RHCS2
S146          ;                                         ; REINSTATE RHER1
S147          ;                                         ; POP STACK INTO RO
S148          ;                                         ; GET OVER PHAS IN RO
S149          016040 012730 177776      MOV    #177776, @ (RO)+ ; LOAD 177776 INTO RHMR
S150          016044 013777 002366 164.44  MOV    @#OFSETC, @RHCS1 ; GET READY FOR OFSETC

```

```

S173
S174
S175 016052 004037 034716 ;NOW SAVE REGISTERS FOR COMPARISON AFTER OFFSET
S176 016056 002210 JSR RO,@#SAVER ;SAVE REGISTERS
S177 016060 004522 RHWC ;RHWC IS THE FIRST REGISTER SAVED
S178 ;STARTING ADDRESS OF WHERE
S179 016062 000022 SAVERE ;THE REGISTERS ARE SAVED
S180 ;NUMBER OF REGISTERS
S181 ;SAVED = 18.
S182 016064 013777 004516 164112 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
S183 ;TO 'TIME1' IF P-CLOCK IS PRESENT
S184 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
S185 ;'TIME' WILL ONLY SAVE
S186 ;CURRENT CYLINDER ADDRESS
S187 ;AND LOOK AHEAD REGISTERS
S188
S189
S190 016072 013746 002366 MOV @#OFSETC,-(SP) ;GET READY TO MOVE COMMAND
S191 016076 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
S192 ;ENABLE INTERRUPT
S193 016102 012677 164110 MOV (SP)+,@RHCS1 ;GO WITH
S194 ;14 IN RHCS1 FOR OFFSET
S195 ;WITH INTERRUPT ENABLED
S196 016106 011100 MOV @R1,RO ;SAVE RHCS1 DURING ABOVE OPERATION
S197 016110 011305 MOV @R3,RS ;SAVE RHDS1 DURING ABOVE OPERATION
S198
S199
S200 016112 104410 WAT ;WAIT FOR DRY BIT TO SET
S201 016114 002240 RHDS1 ;WAIT FOR RHDS1 REGISTER
S202 016116 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
S203 016120 001750 1000. ;ALLOW 10000 MICRO SECONDS
S204 016122 000454 300. ;DRY MUST SET BETWEEN
S205 ;7000 AND 13000 MICRO SECONDS
S206 ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
S207 ;RO AND RS IMMEDIATELY AFTER GO
S208 016124 013746 002366 MOV @#OFSETC,-(SP) ;SAVE COMMAND
S209 016130 052716 004301 BIS #DVA!GO!IE!RD,(SP) ;INCLUDE DVA!GO!IE!RDY
S210 016134 005737 004632 TST @#NUNIT ;ARE THERE MORE THAN ONE UNIT
S211 016140 001413 BEQ 87$ ;BRANCH IF ONLY ONE UNIT
S212 016142 010037 004660 MOV RO,@#TMP4 ;GET RHCS1
S213 016146 042737 177677 004660 BIC #!CIE,@#TMP4 ;KEEP IE BIT
S214 016154 042716 000100 BIC #IE,(SP) ;CLEAR IE IN GOOD DATA
S215 016160 053716 004660 BIS @#TMP4,(SP) ;GET IE AS IS
S216 016164 052716 100000 BIS #SC,(SP) ;SET SC IN RHCS1
S217 016170 87$:
S218 016170 011637 001124 MOV (SP),@#$GDDAT ;SAVE FOR PRINTOUT
S219 016174 022600 CMP (SP)+,RO ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
S220 ;AND COMMAND SHOULD BE SET
S221 016176 001405 BEQ 86$ ;BRANCH IF GOOD
S222 016200 010037 001126 MOV RO,@#$BDDAT ;BAD DATA
S223 016204 010137 004510 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
S224 016210 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
S225 ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
S226 016212 012746 030500 86$: MOV #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
S227 016216 011637 001124 MOV (SP),@#$GDDAT ;SAVE FOR PRINTOUT

```

*Handwritten mark*

*Handwritten mark*

*Handwritten mark*

```

52228 016222 022605          CMP      (SP)+,R5      ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
52229                                     ;SHOULD BE SET
52230 016224 001405          BEQ      88$           ;BRANCH IF GOOD
52231 016226 010537 001126    MOV      R5,2#$BDDAT  ;BAD DATA
52232 016232 010337 004510    MOV      R3,2#$REGADR ;FAILING REGISTER RHDS1
52233 016236 104063          ERROR    63           ;DURING ABOVE OPERATION ONLY
52234                                     ;PIP!MOL!DPR!VV SHOULD BE SET
52235 016240                88$:
52236
52237                                     ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
52238
52239 016240 004037 035440    JSR      RO,2$CHREG   ;CHANGE BITS IN SAVED REGISTER
52240 016244 002216          RHCS1          ;CHANGE RHCS1 REGISTER
52241
52242 016246 000001          1           ;1 BIT/BITS TO BE CHANGED
52243 016250 000001          1           ;NEW VALUE OF SC IS 1
52244 016252 100000          SC           ;CHANGE SC BIT
52245
52246 016254 004037 035440    JSR      RO,2$CHREG   ;CHANGE BITS IN SAVED REGISTER
52247 016260 00224C          RHDS1          ;CHANGE RHDS1 REGISTER
52248
52249 016262 000001          1           ;1 BIT/BITS TO BE CHANGED
52250 016264 000001          1           ;NEW VALUE OF ATA IS 1
52251 016266 100000          ATA          ;CHANGE ATA BIT
52252
52253 016270 053737 004650 004546  BIS      2$ATTENT,2$SAVERE+24 ;SET APPROPRIATE ATA BITS
52254                                     ;FOR WORKING DRIVE IN
52255                                     ;SAVED RHAS LOACTION.
52256
52257                                     ;NOW COMPARE REGISTERS AFTER AN OFFSET COMMAND
52258
52259 016276 004037 035546    JSR      RO,2$COMREG  ;COMPARE SAVED REGISTERS WITH
52260                                     ;PRESENT VALUE
52261 016302 004522          SAVERE        ;GOOD DATA SAVED IN 'SAVERE'
52262 016304 002266          WC           ;TEST DATA STARTING FROM 'RHWC'
52263 016306 000022          18.          ;18. REGISTERS TO BE COMPARED
52264 016310 016314          2$           ;RETURN TO 2$ ON ERROR
52265 016312 016320          3$           ;RETURN TO 3$ ON NO ERROR
52266
52267 016314 104024          2$:          ERROR    24         ;OFFSET COMMAND CAUSED AN ERROR
52268 016316 000207          RTS          PC      ;GOOD DATA IS WHAT SHOULD BE THERE
52269                                     ;RECEIVED DATA GIVES WHAT WAS THERE
52270                                     ;AFTER AN OFFSET COMMAND
52271 016320 013777 004650 163706 3$:      MOV      2$ATTENT,2$RHAS ;CLEAR WORKING DRIVE ATTENTION
52272
52273                                     ;NOW A RETURN TO CENTER LINE COMMAND WILL BE GIVEN
52274
52275 016326 004737 034636    JSR      PC,2$CHECKT  ;CHECK DVA,RDY,MOL,DPR,DRY,VV
52276
52277 016332 013777 002370 163656    MOV      2$RETCL,2$RHCS1 ;GET READY FOR RETCL
52278                                     ;RETURN TO CENTER LINE WITH 16 IN RHCS1
52279
52280
52281                                     ;NOW REGISTERS ARE SAVED FOR COMPARISON AFTER COMMAND
52282 016340 004037 034716    JSR      RO,2$SAVER  ;SAVE REGISTERS
52283 016344 002210          RHWC          ;RHWC IS THE FIRST REGISTER SAVED

```

```

5284 016346 004522          SAVERE          ;STARTING ADDRES OF WHERE
5285 016350 000022          18.            ;THE REGISTERS ARE SAVED
5286 016352 013777 004516 163624  MOV   @#RP4VEC,@RPVEC ;NUMBER OF REGISTERS
5287 016352 013777 004516 163624  ;SAVED = 18.
5288 016352 013777 004516 163624  ;SET RPO4 VECTOR ADDRESS
5289 016352 013777 004516 163624  MOV   @#RP4VEC,@RPVEC ;TO 'TIME1' IF P-CLOCK IS PRESENT
5290 016352 013777 004516 163624  ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5291 016352 013777 004516 163624  ;'TIME' WILL ONLY SAVE
5292 016352 013777 004516 163624  ;CURRENT CYLINDER ADDRESS
5293 016352 013777 004516 163624  ;AND LOOK AHEAD REGISTERS
5294 016352 013777 004516 163624
5295 016352 013777 004516 163624
5296 016352 013777 004516 163624
5297 016360 013746 002370  MOV   @#RETCL,-(SP) ;GET READY TO MOVE COMMAND
5298 016364 052716 000101  BIS   #GO!IE,(SP) ;GET READY TO SET GO AND
5299 016364 052716 000101  ;ENABLE INTERRUPT
5300 016370 012677 163622  MOV   (SP)+,@RHCS1 ;GO WITH
5301 016370 012677 163622  ;16 IN RHCS1 FOR RETURN TO CENTER LINE
5302 016370 012677 163622  ;WITH INTERRUPT ENABLED
5303 016374 011100  MOV   @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
5304 016376 011305  MOV   @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
5305 016376 011305
5306 016400 104410          WAT           ;WAIT FOR DRY BIT TO SET
5307 016400 104410          RHDS1        ;WAIT FOR RHDS1 REGISTER
5308 016402 002240          DRY          ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5309 016404 000200          1000.        ;ALLOW 10000 MICRO SECONDS
5310 016406 001750          1000.        ;DRY MUST SET BETWEEN
5311 016410 001750          ;00 AND 20000 MICRO SECONDS
5312 016410 001750
5313 016410 001750
5314 016410 001750          ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5315 016412 013746 002370  MOV   @#RETCL,-(SP) ;RO AND R5 IMMEDIATELY AFTER GO
5316 016416 052716 004301  BIS   #DVA!GO!IE!RDY,(SP) ;SAVE COMMAND
5317 016422 005737 004632  TST   @#NUNIT ;INCLUDE DVA!GO!IE!RDY
5318 016426 001413 90$          ;ARE THERE MORE THAN ONE UNIT
5319 016430 010037 004660  BEQ   90$ ;BRANCH IF ONLY ONE UNIT
5320 016434 042737 177677  MOV   R0,@#TMP4 ;GET RHCS1
5321 016442 042716 000100  BIC   #!CIE,@#TMP4 ;KEEP IE BIT
5322 016446 053716 004660  BIC   #IE,(SP) ;CLEAR IE IN GOOD DATA
5323 016452 052716 100000  BIS   @#TMP4,(SP) ;GET IE AS IS
5324 016456 90$          BIS   #SC,(SP) ;SET SC IN RHCS1
5325 016456 011637 001124  MOV   (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5326 016462 022600  CMP   (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
5327 016462 022600  ;AND COMMAND SHOULD BE SET
5328 016464 001405  BEQ   89$ ;BRANCH IF GOOD
5329 016466 010037 001126  MOV   R0,@#SBDDAT ;BAD DATA
5330 016472 010137 004510  MOV   R1,@#REGADR ;FAILING REGISTER RHCS1
5331 016475 104021  ERROR 21 ;DURING ABOVE OPERATION ONLY
5332 016475 104021  ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
5333 016500 012746 030500  MOV   #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5334 016504 011637 001124  MOV   (SP),@#SGDDAT ;SAVE FOR PRINTOUT
5335 016510 022605  CMP   (SP)+,R5 ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
5336 016510 022605  ;SHOULD BE SET
5337 016512 001405  BEQ   91$ ;BRANCH IF GOOD
5338 016514 010537 001126  MOV   R5,@#SBDDAT ;BAD DATA
5339 016520 010337 004510  MOV   R3,@#REGADR ;FAILING REGISTER RHDS1

```

29

```

5340 016524 104063          ERROR 63          ;DURING ABOVE OPERATION ONLY
5341                                     ;PIP!MOL!DPR!VV SHOULD BE SET
5342 016526          91$:
5343
5344                                     ;NOW CHANGE SAVED REGISTER TO EXPECTED VALUE
5345
5346 016526 004037 035440    JSR      RO,2#CHREG    ;CHANGE BITS IN SAVED REGISTER
5347 016532 002215          RHCS1      ;CHANGE RHCS1 REGISTER
5348
5349 016534 000001          1          ;1 BIT/BITS TO BE CHANGED
5350 016536 000001          1          ;NEW VALUE OF SC IS 1
5351 016540 100000          SC          ;CHANGE SC BIT
5352
5353 016542 004037 035440    JSR      RO,2#CHREG    ;CHANGE BITS IN SAVED REGISTER
5354 016546 002240          RHDS1      ;CHANGE RHDS1 REGISTER
5355
5356 016550 000001          1          ;1 BIT/BITS TO BE CHANGED
5357 016552 000001          1          ;NEW VALUE OF ATA IS 1
5358 016554 100000          ATA          ;CHANGE ATA BIT
5359
5360 016556 053737 004650 004546  BIS      2#ATTENT,2#SAVERE+24 ;SET APPROPRIATE ATA BITS
5361                                     ;FOR WORKING DRIVE IN
5362                                     ;SAVED RHAS LOACTION
5363 016564 004037 034460    JSR      RO,2#FILLRE    ;MOV BIT15!HCI!ECI!FMT22 INTO SAVED RHOF
5364 016570 002226          RHOF      ;SAVED REGISTER TO CHANGE
5365 016572 116000          BIT15!HCI!ECI!FMT22 ;DATA
5366
5367                                     ;NOW COMPARE REGISTERS AFTER RETURN-TO-CENTER-LINE
5368
5369 016574 004037 035546    JSR      RO,2#COMREG    ;COMPARE SAVED REGISTERS WITH
5370                                     ;PRESENT VALUE
5371 016600 004522          SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
5372 016602 002266          WC          ;TEST DATA STARTING FROM 'RHWC'
5373 016604 000022          18.         ;18. REGISTERS TO BE COMPARED
5374 016606 016612          4$         ;RETURN TO 4$ ON ERROR
5375 016610 016616          5$         ;RETURN TO 5$ ON NO ERROR
5376 016612 104025          4$:      ERROR 25 ;RETURN TO CENTER-LINE
5377 016614 000207          RTS      PC   ;COMMAND CAUSED AN ERROR
5378                                     ;GOOD DATA HAS WHAT SHOULD
5379                                     ;BE THERE
5380                                     ;RECEIVED DATA HAS WHAT WAS
5381                                     ;THERE AFTER COMMAND
5382
5383 016616          5$:
5384
5385 016616 004737 034556    JSR      PC,2#CLDISK    ;SET R1-RHCS1, R2-RHCS2
5386                                     ;R3-RHDS1, R4-RHER1
5387                                     ;GIVE RH-11 INITIALIZE
5388                                     ;SETUP UNIT NUBER
5389 016622 105237 004520    INCB    2#OFSTVL        ;GET NEXT OFFSET VALUE
5390 016626 132737 000100 004520  BITB    #100,2#OFSTVL ;SEE IF UNUSED BIT 6 IS ON
5391 016634 001403          BEQ     7$             ;NO SO DO SOME MORE
5392 016636 062737 000100 004520  ADD     #100,2#OFSTVL ;YES SO BY-PASS IT
5393 016644 105737 004520          TSTB   2#OFSTVL        ;IF ZERO ALL COMBINATIONS ARE
5394                                     ;COMPLETE
5395 016650 001001          BNE     6$             ;BRANCH IF 377 NOT DONE

```





```

5401 :*****
5402 :*TEST 21      OFFSET COMMAND
5403
5404 :*      THIS TEST WILL ONLY GIVE REPEATED OFFSETS
5405
5406 :*****
5407 †ST21: SCOPE
5408 016660 000004      MOV      # 500.,@#TMP1      ;COUNTER
5409 016662 012737 000764 004656
5410 016670      1$:
5411 016670 012706 001000      MOV      #STACK,SP      ;RESET STACK
5412 016674 012737 000021 004514      MOV      #21,@#†STNM      ;SAVE TEST NUMBER
5413
5414      JSR      PC,@#CLDISK      ;SET R1-RHCS1, R2-RHCS2
5415      ;R3-RHDS1, R4-RHER1
5416      ;GIVE RH-11 INITIALIZE
5417 016706 004737 034556      JSR      PC,@#CHECKT      ;SETUP UNIT NUBER
5418      ;CHECK DVA,RDY,MOL,DPR,DRY,VV
5419 016712 004037 034542      JSR      RO,@#OFFSET      ;OFFSET
5420 016716 000260      260      ;260 IN OFFSET REGISTER
5421      ;OFFSET -1200 MICRO INCHES
5422
5423
5424 016720 013746 002366      MOV      @#OFFSETC,-(SP)      ;GET READY TO MOVE COMMAND
5425 016724 052716 000101      BIS      #GO!IE,(SP)      ;GET READY TO SET GO AND
5426      ;ENABLE INTERRUPT
5427 016730 012677 163262      MOV      (SP)+,@#RHCS1      ;GO WITH
5428      ;14 IN RHCS1 FOR OFFSET
5429      ;WITH INTERRUPT ENABLED
5430
5431 016734 104410      WAT      ;WAIT FOR DRY BIT TO SET
5432 016736 002240      RHDS1    ;WAIT FOR RHDS1 REGISTER
5433 016740 000200      DRY      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
5434 016742 000536      350.    ;ALLOW 3500 MICRO SECONDS
5435 016744 000536      350.    ;DRY MUST SET BETWEEN
5436      ;00 AND 7000 MICRO SECONDS
5437 016746 032777 040000 163264      BIT      #ERR,@#RHDS1      ;IS ERR SET?
5438 016754 001417      BEQ      2$      ;NO
5439 016756 104026      ERROR    26      ;REPEATED OFFSETS CAUSED AN ERROR
5440 016760 000004      SCOPE
5441
5442 016762 004737 034556      JSR      PC,@#CLDISK      ;SET R1-RHCS1, R2-RHCS2
5443      ;R3-RHDS1, R4-RHER1
5444      ;GIVE RH-11 INITIALIZE
5445      ;SETUP UNIT NUBER
5446
5447 016766 013746 002340      MOV      @#RECALI,-(SP)      ;GET READY TO MOVE COMMAND
5448 016772 052716 000101      BIS      #GO!IE,(SP)      ;GET READY TO SET GO AND
5449      ;ENABLE INTERRUPT
5450 016776 012677 163214      MOV      (SP)+,@#RHCS1      ;GO WITH
5451      ;6 IN RHCS1 FOR RECALIBRATE
5452      ;WITH INTERRUPT ENABLED
5453
5454 017002 104410      WAT      ;WAIT FOR DRY BIT TO SET
5455 017004 002240      RHDS1    ;WAIT FOR RHDS1 REGISTER
5456 017006 000200      DRY      ;WAIT FOR DRY BIT IN RHDS1 REGISTER

```



5466  
5467  
5468  
5469  
5470  
5471  
5472  
5473  
5474  
5475  
5476  
5477  
5478  
5479  
5480  
5481  
5482  
5483  
5484  
5485  
5486  
5487  
5488  
5489  
5490  
5491  
5492  
5493  
5494  
5495  
5496  
5497  
5498  
5499  
5500  
5501  
5502  
5503  
5504  
5505  
5506  
5507  
5508  
5509  
5510  
5511  
5512  
5513  
5514  
5515  
5516  
5517  
5518  
5519  
5520  
5521

\*\*\*\*\*  
; \*TEST 22 WRITE/READ HEADER AND DATA

; \* WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD  
; \* TRACK 0, SECTOR 0, KEYS=0, NUMBER OF WORDS 256 WORDS  
; \* OF 0  
; \* THEN READ HEADER AND DATA FOR ABOVE.  
; \* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
; \* 1000 0 0, 0, AND 256 OF 0  
; \* THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT  
; \* THE GO BIT, AND ALL THE REGISTERS ARE SAVED  
; \* THEN GO IS GIVEN FOR WRITE HEADER AND DATA  
; \*  
; \* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGED  
; \* THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED  
; \*  
; \* NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED  
; \* WITH ALL ONES, COMMAND IS LOADED INTO REGISTERS EXCEPT  
; \* GO BIT AND ALL REGISTERS ARE SAVED  
; \* GO IS GIVEN FOR THE READ COMMAND  
; \*  
; \* ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE  
; \* THEN THE READ DATA IS COMPARED.

\*\*\*\*\*

017024 000004  
017026 012706 701000  
017032 012737 J00022 004514  
  
017040 004737 034556  
  
  
  
017044 004037 034402  
017050 002400  
017052 000004  
017054 010000  
017056 000000  
017060 000000  
017062 000000  
  
017064 004037 034426  
017070 002410  
017072 000400  
017074 000000

†ST22: SCOPE  
MOV #STACK, SP ;RESET STACK  
MOV #22, @†STNM ;SAVE TEST NUMBER  
  
JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUBER  
  
;FILL WRITE FROM BUFFER WITH HEADER  
  
JSR RO, @#FLHEAD ;SAVE HEADER DATA IN WRFROM  
WRFROM ;LOCATION WHERE SAVED  
4 ;NUMBER OF WORDS SAVED  
10000 ;FIRST DATA WORD  
0 ;SECOND DATA WORD  
0 ;THIRD DATA WORD  
0 ;FOURTH DATA WORD  
  
;FILL WRITE FROM BUFFER WITH DATA  
JSR RO, @#CLAREA ;CLEAR 256. WORDS. FROM WRFROM+10  
WRFROM+10 ;STARTING FROM WRFROM+10  
256. ;256. WORDS  
0 ;FILL WITH 0

; NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA  
; AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS  
; CAN BE MADE TO MAKE SJRE THAT WRITE DID NCT  
; CHANGE WRITE FROM BUFFER

017102	004037	034402	JSR	RD, @FLHEAD	:SAVE HEADER DATA IN REINTO
017104	004037		REINTO		:LOCATION WHERE SAVED
017106	004037		4		:NUMBER OF WORDS SAVED
017108	004037		10000		:FIRST DATA WORD
017110	004037				:SECOND DATA WORD
017112	004037				:THIRD DATA WORD
017114	004037				:FOURTH DATA WORD
017116	004037	034406	JSR	RD, @CLAREA	:CLEAR 256 WORDS FROM REINTO+10
017118	004037		RD, REINTO+10		:STARTING FROM REINTO+10
017120	004037		256		:256 WORDS
017122	004037		0		:FILL WITH 0
:NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED					
017142	002400	036532	JSR	RD, @RUN	:SETUP TO RUN FOR DATA COMMAND
017144	002400		0		:CYLINDER 0
017146	002400		0		:SECTOR 0
017148	002400		0		:TRACK 0
017150	002400		-256, -4		:WORD COUNT (DATA)=256, *
017152	002356		WRFROM		:4 HEADER WORDS
017154	002356				:BUS ADDRESS
017156	002356				:STARTING ADDRESS OF DATA
017158	002356				:BUFFER = WRFROM
017160	002356				:DO NOT INHIBIT BUS ADDRESS INCREMENT
017162	002356				:16 BITS PER WORD FORMAT
017164	002356				:DO NOT INHIBIT ECC CORRECTION
017166	002356				:DO NOT INHIBIT HEADER COMPARE
017168	002356				:GET READY TO DO A WRIFOR
017170	002356				:WRITE HEADER AND DATA WITH 62 IN RHCS1
:NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA					
017152	004037	034716	JSR	RD, @SAVER	:SAVE REGISTERS
017156	002210		RHWC		:RHWC IS THE FIRST REGISTER SAVED
017160	004522		SAVERE		:STARTING ADDRESS OF WHERE
017162	000021		17.		:THE REGISTERS ARE SAVED
017164	000021				:NUMBER OF REGISTERS
017166	000021				:SAVED = 17.
017164	004737	034636	JSR	PC, @CHECKT	:CHECK DVA, RDY, MOL, DPR, DRY, VV
017170	013777	004516	MOV	@RP4VEC, @RPVEC	:SET RP04 VECTOR ADDRESS
017172	004516	163006			:TO 'TIME1' IF P-CLOCK IS PRESENT
017174	004516				:OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
017176	004516				: 'TIME' WILL ONLY SAVE
017178	004516				:CURRENT CYLINDER ADDRESS
017180	004516				:AND LOOK AHEAD REGISTERS
017176	013746	002356	MOV	@WRIFOR, -(SP)	:GET READY TO MOVE COMMAND
017202	052716	000101	BIS	@GO!IE, (SP)	:GET READY TO SET GO AND
017204	052716				:ENABLE INTERRUPT
017206	012677	163004	MOV	(SP)+, @RHCS1	:GO WITH

MA:NDCC-11-DERPUB-8  
DERPUB.F11 T22

MAY11 27(732) 17-SEP-76 13:23 PAGE 119  
WRITE READ HEADER AND DATA

557							:62 IN RHCSI FOR WRITE HEADER AND DATA
558	017212	011100					:WITH INTERRUPT ENABLED
559	017214	011305			MOV JR1,RO		:SAVE RHCSI DURING ABOVE OPERATION
560					MOV JR3,R5		:SAVE RHDS1 DURING ABOVE OPERATION
561							:ONE REVOLUTION=16670 MICRO SEC, ONE SECTOR = 760 MICRO SEC
562							
563	017216	104410			WAT		:WAIT FOR RDY BIT TO SET
564	017220	002216			RHCSI		:WAIT FOR RHCSI REGISTER
565	017222	000200			RDY		:WAIT FOR RDY BIT IN RHCSI REGISTER
566	017224	001614			908.		:ALLOW 9080 MICRO SECONDS
567	017226	001507			839.		:RDY MUST SET BETWEEN
568							:690 AND 17470 MICRO SECONDS
569							:COMPARE CONTENTS OF RHCSI AND RHDS1 ALREADY SAVED IN
570							:RO AND R5 IMMEDIATELY AFTER GO
571	017230	013746	002356		MOV J#WRIFOR,-(SP)		:SAVE COMMAND
572	017234	052716	004101		BIS #IE!GO!DVA,(SP)		:INCLUDE IE!GO!DVA
573	017240	011637	001124		MOV (SP),J#SGDDAT		:SAVE FOR PRINTOUT
574	017244	022600			CMP (SP)+,RO		:DURING ABOVE OPERATION ONLY IE!GO!DVA
575							:AND COMMAND SHOULD BE SET
576	017246	001405			BEQ 645		:BRANCH IF GOOD
577	017250	010037	001126		MOV RO,J#\$BDDAT		:BAD DATA
578	017254	010137	004510		MOV R1,J#REGADR		:FAILING REGISTER RHCSI
579	017260	104021			ERROR 21		:DURING ABOVE OPERATION ONLY
580							:COMMAND AND IE!GO!DVA SHOULD BE SET
581	017262	012746	010500	645:	MOV #MOL!DPR!VV,-(SP)		:SAVE BITS SET DURING OPERATION IN RHDS1
582	017266	011637	001124		MOV (SP),J#SGDDAT		:SAVE FOR PRINTOUT
583	017272	022605			CMP (SP)+,==		:DURING ABOVE OPERATION ONLY MOL!DPR!VV
584							:SHOULD BE SET
585	017274	001405			BEQ 665		:BRANCH IF GOOD
586	017276	010537	001126		MOV R5,J#\$BDDAT		:BAD DATA
587	017302	010337	004510		MOV R3,J#REGADR		:FAILING REGISTER RHDS1
588	017306	104053			ERROR 63		:DURING ABOVE OPERATION ONLY
589							:MOL!DPR!VV SHOULD BE SET
590	017310			665:			
591							:NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
592	017310	004037	034460		JSR RO,J#FILLRE		:MOV 0 INTO SAVED RHWC
593	017314	002210			RHWC		:SAVED REGISTER TO CHANGE
594	017316	000000			0		:DATA
595	017320	004037	034460		JSR RO,J#FILLRE		:MOV WRFROM+(<260.*2>) INTO SAVED RH3A
596	017324	002212			RH3A		:SAVED REGISTER TO CHANGE
597	017326	003410			WRFROM+(<260.*2>)		:DATA
598	017330	004037	034460		JSR RO,J#FILLRE		:MOV 1 INTO SAVED RHDST
599	017334	002222			RHDST		:SAVED REGISTER TO CHANGE
600	017336	000001			1		:DATA
601							:NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
602							:WITH REGISTERS AFTER COMMAND
603	017340	004037	035546		JSR RO,J#COMREG		:COMPARE SAVED REGISTERS WITH
604							:PRESENT VALUE
605	017344	004522			SAVERE		:GOOD DATA SAVED IN 'SAVERE'
606	017346	002266			WC		:TEST DATA STARTING FROM 'RHWC'
607	017350	000021			17.		:17. REGISTERS TO BE COMPARED
608	017352	017356			1\$		:RETURN TO 1\$ ON ERROR
609	017354	017362			2\$		:RETJRN TO 2\$ ON NO ERROR

```

5634 017356 104027 1$: ERROR 27 :WRITE HEADER AND DATA
5635 017360 000207 RTS PC :CAUSED IMPROPER REGISTER
5636 :CHANGE
5637 :GOOD DATA GIVES WHAT SHOULD
5638 :BE THERE
5639 :RECEIVED DATA GIVES WHAT
5640 :WAS THERE AFTER COMMAND
5641
5642 :NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
5643 :NOTHER GOT CHANGED
5644 017362 2$:
5645
5646 017362 004037 036576 JSR RO,2#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5647 017366 003444 REINTO ;GOOD DATA STARTS FROM REINTO
5648 017370 002400 WRFROM ;TEST DATA STARTS FROM WRFROM
5649 017372 000404 260. ;260. WORDS TO BE COMPARED
5650 017374 017400 3$ ;RETURN TO 3$ ON ERROR
5651 017376 017404 4$ ;RETURN TO 4$ ON NO ERROR
5652
5653
5654 017400 104030 3$: ERROR 30 :WRITE HEADER AND DATA
5655 017402 000207 RTS PC ;CHANGED WRITE FROM BUFFER
5656
5657 :NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
5658 :READ INTO BUFFER IS FILLED WITH ONES
5659 017404 4$:
5660
5661 017404 004737 034556 JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
5662 ;R3-RHDS1, R4-RHER1
5663 ;GIVE RH-11 INITIALIZE
5664 ;SETUP UNIT NUBER
5665 017410 004037 034426 JSR RO,2#CLAREA ;CLEAR 260. WORDS, FROM REINTO
5666 017414 003444 REINTO ;STARTING FROM REINTO
5667 017416 000404 260. ;260. WORDS
5668 017420 177777 -1 ;FILL WITH -1
5669
5670
5671 :NOW FILL COMMAND
5672
5673 017422 004037 036532 JSR RO,2#RUN ;SETUP TO RUN FOR DATA COMMAND
5674 017426 000000 0 ;CYLINDER 0
5675 017430 000 0 ;SECTOR 0
5676 017431 000 0 ;TRACK 0
5677 017432 177374 -256.-4 ;WORD COUNT (DATA)=256.+
5678 ;4 HEADER WORDS
5679 017434 003444 REINTO ;BUS ADDRESS
5680 ;STARTING ADDRESS OF DATA
5681 ;BUFFER = REINTO
5682 017436 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
5683 017440 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
5684 ;INHIBIT ECC CORRECTION
5685 ;DO NOT INHIBIT HEADER COMPARE
5686 017442 002362 REFOR ;GET READY TO DO A REFOR
5687 ;READ HEADER AND DATA WITH 72 IN RHCS1
5688
5689

```

```

5690 ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
5691 017444 004037 034716 JSR RO,2#SAVER ;SAVE REGISTERS
5692 017450 002210 RHWC ;RHWC IS THE FIRST REGISTER SAVED
5693 017452 004522 SAVERE ;STARTING ADDRESS OF WHERE
5694 ;THE REGISTERS ARE SAVED
5695 017454 000022 18. ;NUMBER OF REGISTERS
5696 ;SAVED = 18.
5697
5698 017456 004737 034636 JSR PC,2#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
5699
5700
5701 017462 013777 004516 162514 MOV 2#RPO4VEC,2#PVEC ;SET RPO4 VECTOR ADDRESS
5702 ;TO 'TIME1' IF P-CLOCK IS PRESENT
5703 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
5704 ;'TIME' WILL ONLY SAVE
5705 ;CURRENT CYLINDER ADDRESS
5706 ;AND LOOK AHEAD REGISTERS
5707
5708
5709 017470 013746 002362 MOV 2#REFOR,-(SP) ;GET READY TO MOVE COMMAND
5710 017474 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
5711 ;ENABLE INTERRUPT
5712 017500 012677 162512 MOV (SP)+,2#RHCS1 ;GO WITH
5713 ;72 IN RHCS1 FOR READ DATA
5714 ;WITH INTERRUPT ENABLED
5715 017504 011100 MOV 2#R1,RO ;SAVE RHCS1 DURING ABOVE OPERATION
5716 017506 011305 MOV 2#R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
5717
5718
5719 017510 104410 WAT ;WAIT FOR RDY BIT TO SET
5720 017512 002216 RHCS1 ;WAIT FOR RHCS1 REGISTER
5721 017514 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
5722 017516 001614 908. ;ALLOW 9080 MICRO SECONDS
5723 017520 001507 839. ;RDY MUST SET BETWEEN
5724 ;690 AND 17470 MICRO SECONDS
5725 ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
5726 ;RO AND R5 IMMEDIATELY AFTER GO
5727 017522 013746 002362 MOV 2#REFOR,-(SP) ;SAVE COMMAND
5728 017526 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
5729 017532 011637 001124 MOV (SP),2#SGDDAT ;SAVE FOR PRINTOUT
5730 017536 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
5731 ;AND COMMAND SHOULD BE SET
5732 017540 001405 BEQ 67$ ;BRANCH IF GOOD
5733 017542 010037 001126 MOV R0,2#SBDDAT ;BAD DATA
5734 017546 010137 004510 MOV R1,2#REGADR ;FAILING REGISTER RHCS1
5735 017552 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
5736 ;COMMAND AND IE!GO!DVA SHOULD BE SET
5737 017554 012746 010500 67$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5738 017560 011637 001124 MOV (SP),2#SGDDAT ;SAVE FOR PRINTOUT
5739 017564 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5740 ;SHOULD BE SET
5741 017566 001405 BEQ 69$ ;BRANCH IF GOOD
5742 017570 010537 001126 MOV R5,2#SBDDAT ;BAD DATA
5743 017574 010337 004510 MOV R3,2#REGADR ;FAILING REGISTER RHDS1
5744 017600 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
57-5 ;MOL!DPR!VV SHOULD BE SET

```



```

5746 017602          69$:
5747
5748                ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
5749 017602 004037 034460 JSR    RO,2#FILLRE  ;MOV 0 INTO SAVED RHWC
5750 017606 002210        RHWC          ;SAVED REGISTER TO CHANGE
5751 017610 000000        0              ;DATA
5752 017612 004037 034460 JSR    RO,2#FILLRE  ;MOV REINTO+(260.*2) INTO SAVED RHBA
5753 017616 002212        RHBA          ;SAVED REGISTER TO CHANGE
5754 017620 004454        REINTO+(260.*2) ;DATA
5755 017622 004037 034460 JSR    RO,2#FILLRE  ;MOV 1 INTO SAVED RHDST
5756 017626 002222        RHDST         ;SAVED REGISTER TO CHANGE
5757 017630 000001        1              ;DATA
5758
5759                ;COMPARE REGISTER BEFORE READ HEADER AND DATA
5760                ;WITH REGISTERS AFTER COMMAND
5761
5762
5763 017632 004037 035546 JSR    RO,2#COMREG  ;COMPARE SAVED REGISTERS WITH
5764                ;PRESENT VALUE
5765 017636 004522        SAVERE         ;GOOD DATA SAVED IN 'SAVERE'
5766 017640 002266        WC            ;TEST DATA STARTING FROM 'RHWC'
5767 017642 000022        18.          ;18. REGISTERS TO BE COMPARED
5768 017644 017650        5$          ;RETURN TO 5$ ON ERROR
5769 017646 017654        6$          ;RETURN TO 6$ ON NO ERROR
5770
5771 017650 104031 5$:    ERROR 31      ;READ HEADER AND DATA CAUSED
5772 017652 000207        RTS    PC     ;IMPROPER REGISTER CHANGE
5773                ;GOOD DATA GIVES WHAT SHOULD
5774                ;BE THERE
5775                ;RECEIVED DATA GIVES WHAT WAS
5776                ;THERE AFTER COMMAND
5777
5778                ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE
5779                ;THE READ WAS GOOD
5780
5781          017654          6$:
5782 017654 004037 036576 JSR    RO,2#COMPAR  ;COMPARE TWO BLOCKS OF MEMORY
5783 017660 002400        WRFROM        ;GOOD DATA STARTS FROM WRFROM
5784 017662 003444        REINTO       ;TEST DATA STARTS FROM REINTO
5785 017664 000404        260.         ;260. WORDS TO BE COMPARED
5786 017666 017672        7$          ;RETURN TO 7$ ON ERROR
5787 017670 017676        10$         ;RETURN TO 10$ ON NO ERROR
5788
5789
5790 017672 104032 7$:    ERROR 32      ;WRITE HEADER AND DATA
5791 017674 000207        RTS    PC     ;FOLLOWED BY A READ HEADER
5792                ;AND DATA GAVE A READ ERROR
5793                ;ERROR MAY BE IN READ OR WRITE
5794
5795          017676          10$:
5796
5797                ;*****
5798                ;*TEST 23      READ DATA
5799
5800                ;*
5801                ;*      THIS TEST READS DATA WRITTEN BY THE PREVIOUS TEST
                    ;*      THE WRITE FROM BUFFER IS FILLED WITH ALL 0S

```



```

5858 ;AND LOOK AHEAD REGISTERS
5859
5860
5861 017776 013746 002360 MOV 2#READAT, -(SP) ;GET READY TO MOVE COMMAND
5862 020202 052716 000101 BIS #GO!IT, (SP) ;GET READY TO SET GO AND
5863 ;ENABLE INTERRUPT
5864 020006 012672 162204 MOV (SP)+, 2#RHCSI ;GO WITH
5865 ;70 IN RHCSI FOR READ DATA
5866 ;WITH INTERRUPT ENABLED
5867 020012 011100 MOV 2#R1, R0 ;SAVE RHCSI DURING ABOVE OPERATION
5868 020014 011305 MOV 2#R3, R5 ;SAVE RHDS1 DURING ABOVE OPERATION
5869
5870
5871 020016 104410 WAT ;WAIT FOR RDY BIT TO SET
5872 020020 002216 RHCSI ;WAIT FOR RHCSI REGISTER
5873 020022 000200 RDY ;WAIT FOR RDY BIT IN RHCSI REGISTER
5874 020024 001614 908. ;ALLOW 9080 MICRO SECONDS
5875 020026 001507 839. ;RDY MUST SET BETWEEN
5876 ;690 AND 17470 MICRO SECONDS
5877 ;COMPARE CONTENTS OF RHCSI AND RHDS1 ALREADY SAVED IN
5878 ;R0 AND R5 IMMEDIATELY AFTER GO
5879 020030 013746 002360 MOV 2#READAT, -(SP) ;SAVE COMMAND
5880 020034 052716 104101 BIS #IE!GO!DVA, (SP) ;INCLUDE IE!GO!DVA
5881 020040 011637 001124 MOV (SP), 2#$GDDAT ;SAVE FOR PRINTOUT
5882 020044 022600 CMP (SP)+, R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
5883 ;AND COMMAND SHOULD BE SET
5884 020046 001405 BEQ 64$ ;BRANCH IF GOOD
5885 020050 010037 001126 MOV R0, 2#$BDDAT ;BAD DATA
5886 020054 010137 004510 MOV R1, 2#REGADR ;FAILING REGISTER RHCSI
5887 020060 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
5888 ;COMMAND AND IE!GO!DVA SHOULD BE SET
5889 020062 012746 010500 64$: MOV #MOL!DPR!VV, -(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
5890 020066 011637 001124 MOV (SP), 2#$GDDAT ;SAVE FOR PRINTOUT
5891 020072 022605 CMP (SP)+, R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
5892 ;SHOULD BE SET
5893 020074 001405 BEQ 66$ ;BRANCH IF GOOD
5894 020076 010537 001126 MOV R5, 2#$BDDAT ;BAD DATA
5895 020102 010337 004510 MOV R3, 2#REGADR ;FAILING REGISTER RHDS1
5896 02010E 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
5897 ;MOL!DPR!VV SHOULD BE SET
5898 020110 66$:
5899
5900 ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
5901 020110 004037 034460 JSR R0, 2#FILLRE ;MOV 0 INTO SAVED RHWC
5902 020114 002210 RHWC ;SAVED REGISTER TO CHANGE
5903 020116 000000 0 ;DATA
5904 020120 004037 034460 JSR R0, 2#FILLRE ;MOV REINTO+<256.*2> INTO SAVED RHBA
5905 020124 002212 RHBA ;SAVED REGISTER TO CHANGE
5906 020126 004444 REINTO+<256.*2> ;DATA
5907 020130 004037 034460 JSR R0, 2#FILLRE ;MOV 1 INTO SAVED RHDST
5908 020134 002222 RHDST ;SAVED REGISTER TO CHANGE
5909 020136 000001 1 ;DATA
5910
5911 ;NOW COMPARE REGISTERS BEFORE READ DATA WITH
5912 ;AFTER COMMAND
5913

```

```

5914 020140 004037 035546 JSR RD,2#COMREG ;COMPARE SAVED REGISTERS WITH
5915 ;PRESENT VALUE
5916 020144 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
5917 020146 002266 WC ;TEST DATA STARTING FROM 'PHWC'
5918 020150 000022 18. ;18. REGISTERS TO BE COMPARED
5919 020152 020156 1$ ;RETURN TO 1$ ON ERROR
5920 020154 020162 2$ ;RETURN TO 2$ ON NO ERROR
5921
5922 020156 104033 1$: ERROR 33 ;READ DATA CAUSED IMPROPER REGISTER
5923 020160 000207 RTS PC ;CHANGE
5924 ;GOOD DATA GIVES WHAT SHOULD BE THERE
5925 ;RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND
5926 ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE THAT READ
5927 ;WAS GOOD
5928 020162 2$:
5929
5930 020162 004037 036576 JSR RD,2#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
5931 020166 002400 WRFROM ;GOOD DATA STARTS FROM WRFROM
5932 020170 003444 REINTO ;TEST DATA STARTS FROM REINTO
5933 020172 000400 256. ;256. WORDS TO BE COMPARED
5934 020174 020200 3$ ;RETURN TO 3$ ON ERROR
5935 020176 020204 4$ ;RETURN TO 4$ ON NO ERROR
5936
5937
5938 020200 104034 3$: ERROR 34 ;READ DATA COMMAND
5939 020202 000207 RTS PC ;READ INCORRECTLY
5940
5941 020204 4$:
5942
5943 *****
5944 *TEST 24 WRITE/READ DATA
5945
5946 ;* THIS TEST GIVES A WRITE DATA COMMAND FRO CYLINDER 0
5947 ;* TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF ALL ONES
5948 ;* THIS SECTOR IS FORMATED BY PREVIOUS TEST
5949 ;* THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
5950 ;* SAME CYLINDER, TRACK, SECTOR, KEYS
5951 ;* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
5952 ;* 200 ONES AND 56 125252
5953 ;* THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
5954 ;* ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA
5955 ;*
5956 ;*
5957 ;* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
5958 ;* CHANGE. THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
5959 ;*
5960 ;* THEN READ INTO BUFFER IS FILLED WITH 200 OF ZEROS
5961 ;* AND 56 OF 377, WRITE FROM BUFFER IS FILLED WITH 200 ONES
5962 ;* AND 56 OF 377.
5963 ;* THE COMMAND EXCEPT GO IS LOADED. ALL REGISTERS ARE SAVED
5964 ;* GO IS GIVEN
5965 ;*
5966 ;*
5967 ;* ALL FEGISTERS ARE CHECKED
5968 ;* READ DATA IS CHECKED
5969

```

```

5970 :*****
5971 020204 000004          †ST24: SCOPE
5972 020206 012706 001000      MOV      #STACK.SP      ;RESET STACK
5973 020212 012737 002024 004514  MOV      #24,#†STNM     ;SAVE TEST NUMBER
5974
5975 020220 004737 034556      JSR      PC,#CLDISK     ;SET R1-RHCS1, R2-RHCS2
5976                                ;R3-RHDS1, R4-RHER1
5977                                ;GIVE RH-11 INITIALIZE
5978                                ;SETUP UNIT NUBER
5979
5980                                ;NOW FILL WRITE FROM BUFFER -200 OF 1'S AND 56 OF 125252
5981 020224 004037 034426      JSR      RO,#CLAREA     ;CLEAR 200. WORDS, FROM WRFROM
5982 020230 002400                WRFROM     ;STARTING FROM WRFROM
5983 020232 000310                200.      ;200. WORDS
5984 020234 177777                -1        ;FILL WITH -1
5985
5986 020236 004037 034426      JSR      RO,#CLAREA     ;CLEAR 56. WORDS, FROM WRFROM+(200.*2)
5987 020242 003220                WRFROM+(200.*2) ;STARTING FROM WRFROM+(200.*2)
5988 020244 000070                56.      ;56. WORDS
5989 020246 125252                125252   ;FILL WITH 125252
5990
5991
5992                                ;NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
5993                                ;WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
5994                                ;CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
5995 020250 004037 034426      JSR      RO,#CLAREA     ;CLEAR 200. WORDS, FROM REINTO
5996 020254 003444                REINTO    ;STARTING FROM REINTO
5997 020256 000310                200.      ;200. WORDS
5998 020260 177777                -1        ;FILL WITH -1
5999
6000 020262 004037 034426      JSR      RO,#CLAREA     ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
6001 020266 004264                REINTO+(200.*2) ;STARTING FROM REINTO+(200.*2)
6002 020270 000070                56.      ;56. WORDS
6003 020272 125252                125252   ;FILL WITH 125252
6004
6005
6006                                ;NOW WRITE DATA COMMAND WILL BE LOADED
6007
6008 020274 004037 036532      JSR      RO,#RUN        ;SETUP TO RUN FOR DATA COMMAND
6009 020300 000000                0         ;CYLINDER 0
6010 020302 000          .BYTE 0 ;SECTOR 0
6011 020303 000          .BYTE 0 ;TRACK 0
6012 020304 177470                -200.     ;WORD COUNT = 200.
6013 020306 002400                WRFROM    ;BUS ADDRESS
6014                                ;STARTING ADDRESS OF DATA
6015                                ;BUFFER = WRFROM
6016 020310 000000                0         ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6017 020312 010000                FMT22    ;16 BITS PER WORD FORMAT
6018                                ;DO NOT INHIBIT ECC CORRECTION
6019                                ;DO NOT INHIBIT HEADER COMPARE
6020 020314 002354                WRIDAT   ;GET READY TO DO A WRIDAT
6021                                ;WRITE DATA WITH 60 IN RHCS1
6022
6023
6024                                ;NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
6025 020316 004037 034716      JSR      RO,#SAVER     ;SAVE REGISTERS

```

```

6026 020322 002210          RHWC          ;RHWC IS THE FIRST REGISTER SAVED
6027 020324 004522          SAVERE        ;STARTING ADDRESS OF WHERE
6028                                ;THE REGISTERS ARE SAVED
6029 020326 000022          18.           ;NUMBER OF REGISTERS
6030                                ;SAVED = 18.
6031
6032 020330 004737 034636    JSR      PC, @#CHECKT ;CHECK DVA, RDY, MOL, DPR, DRY, VV
6033
6034
6035 020334 013777 004516 161642  MOV      @#RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
6036                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
6037                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6038                                ;'TIME' WILL ONLY SAVE
6039                                ;CURRENT CYLINDER ADDRESS
6040                                ;AND LOOK AHEAD REGISTERS
6041
6042
6043 020342 013746 002354    MOV      @#WRIDAT, -(SP) ;GET READY TO MOVE COMMAND
6044 020346 052716 000101    BIS      #GO!IE, (SP)   ;GET READY TO SET GO AND
6045                                ;ENABLE INTERRUPT
6046 020352 012677 161640    MOV      (SP)+, @RHCSI  ;GO WITH
6047                                ;60 IN RHCSI FOR WRITE DATA
6048                                ;WITH INTERRUPT ENABLED
6049 020356 011100          MOV      @R1, R0        ;SAVE RHCSI DURING ABOVE OPERATION
6050 020360 011305          MOV      @R3, R5        ;SAVE RHDS1 DURING ABOVE OPERATION
6051
6052                                ;ONE REVOLUTION = 16670 MICRO SEC, ONE SECTOR=760 MICRO SEC
6053
6054 020362 104410          WAT          ;WAIT FOR RDY BIT TO SET
6055 020364 002216          RHCSI       ;WAIT FOR RHCSI REGISTER
6056 020366 000200          RDY        ;WAIT FOR RDY BIT IN RHCSI REGISTER
6057 020370 001614          908.       ;ALLOW 9080 MICRO SECONDS
6058 020372 001507          839.       ;RDY MUST SET BETWEEN
6059                                ;690 AND 17470 MICRO SECONDS
6060                                ;COMPARE CONTENTS OF RHCSI AND RHDS1 ALREADY SAVED IN
6061                                ;R0 AND R5 IMMEDIATELY AFTER GO
6062 020374 013746 002354    MOV      @#WRIDAT, -(SP) ;SAVE COMMAND
6063 020400 052716 004101    BIS      #IE!GO!DVA, (SP) ;INCLUDE IE!GO!DVA
6064 020404 011637 001124    MOV      (SP), @#$GDDAT ;SAVE FOR PRINTOUT
6065 020410 022600          CMP      (SP)+, R0      ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6066                                ;AND COMMAND SHOULD BE SET
6067 020412 001405          BEQ      64$          ;BRANCH IF GOOD
6068 020414 010037 001126    MOV      R0, @#$BDDAT   ;BAD DATA
6069 020420 010137 004510    MOV      R1, @#REGADR   ;FAILING REGISTER RHCSI
6070 020424 104021          ERROR     21          ;DURING ABOVE OPERATION ONLY
6071                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
6072 020426 012746 010500    64$: MOV      #MOL!DPR!VV, -(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6073 020432 011637 001124    MOV      (SP), @#$GDDAT ;SAVE FOR PRINTOUT
6074 020436 022605          CMP      (SP)+, R5     ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6075                                ;SHOULD BE SET
6076 020440 001405          BEQ      66$          ;BRANCH IF GOOD
6077 020442 010537 001126    MOV      R5, @#$BDDAT   ;BAD DATA
6078 020446 010337 004510    MOV      R3, @#REGADR   ;FAILING REGISTER RHDS1
6079 020452 104063          ERROR     63          ;DURING ABOVE OPERATION ONLY
6080                                ;MOL!DPR!VV SHOULD BE SET
6081 020454          66$:

```

```

6082
6083
6084 020454 004037 034460      ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
        JSR      RO, @#FILLRE      ;MOV 0 INTO SAVED RHWC
        RHWC      ;SAVED REGISTER TO CHANGE
        0          ;DATA
6085 020460 002210
6086 020462 000000
6087 020464 004037 034460      JSR      RO, @#FILLRE      ;MOV WRFROM+<200.*2> INTO SAVED RHBA
        RHBA      ;SAVED REGISTER TO CHANGE
        WRFROM+<200.*2> ;DATA
6088 020470 002212
6089 020472 003220
6090 020474 004037 034460      JSR      RO, @#FILLRE      ;MOV 1 INTO SAVED RHDST
        RHDST     ;SAVED REGISTER TO CHANGE
        1          ;DATA
6091 020500 002222
6092 020502 000001
6093
6094      ;NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6095      ;AFTER COMMAND
6096
6097 020504 004037 035546      JSR      RO, @#COMREG      ;COMPARE SAVED REGISTERS WITH
        SAVERE     ;PRESENT VALUE
        WC          ;GOOD DATA SAVED IN 'SAVERE'
        18.        ;TEST DATA STARTING FROM 'RHWC'
        1$         ;18. REGISTERS TO BE COMPARED
        2$         ;RETURN TO 1$ ON ERROR
        3$         ;RETURN TO 2$ ON NO ERROR
6100 020512 002266
6101 020514 000022
6102 020516 020522
6103 020520 020526
6104
6105 020522 104035 1$:      ERROR 35      ;WRITE DATA COMMAND CAUSED
        RTS      PC      ;IMPROPER REGISTER CHANGE
6106 020524 000207
6107
6108
6109
6110
6111
6112      ;NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
6113 020526 2$:
6114
6115 020526 004037 036576      JSR      RO, @#COMPAR      ;COMPARE TWO BLOCKS OF MEMORY
        REINTO     ;GOOD DATA STARTS FROM REINTO
        WRFROM     ;TEST DATA STARTS FROM WRFROM
        256.       ;256. WORDS TO BE COMPARED
        3$         ;RETURN TO 3$ ON ERROR
        4$         ;RETURN TO 4$ ON NO ERROR
6116 020532 003444
6117 020534 002400
6118 020536 000400
6119 020540 020544
6120 020542 020550
6121
6122
6123 020544 104036 3$:      ERROR 36      ;WRITE DATA COMMAND CHANGED
        RTS      PC      ;WRITE FROM BUFFER
6124 020546 000207
6125
6126      ;NOW A READ DATA COMMAND WILL BE GIVEN
6127
6128      ;FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF 377
6129 020550 4$:
6130
6131 020550 004737 034556      JSR      PC, @#CLDISK      ;SET R1-RHCS1, R2-RHCS2
        ;R3-RHDS1, R4-RHER1
        ;GIVE RH-11 INITIALIZE
        ;SETUP UNIT NUMBER
6132
6133
6134
6135 020554 004037 034426      JSR      RO, @#CLAREA      ;CLEAR 200. WORDS, FROM REINTO
        REINTO     ;STARTING FROM REINTO
        200.       ;200. WORDS
6136 020560 003444
6137 020562 000310

```

```

6138 020564 000000 0 ;FILL WITH 0
6139
6140 020566 004037 034426 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
6141 020572 004264 REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
6142 020574 000070 56. ;56. WORDS
6143 020576 000377 377 ;FILL WITH 377
6144
6145
6146 ;FILL WRITE FROM BUFFER WITH 200 ONES AND 56 OF 377
6147 020600 004037 034426 JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM
6148 020604 002400 WRFROM ;STARTING FROM WRFROM
6149 020606 000310 200. ;200. WORDS
6150 020610 177777 -1 ;FILL WITH -1
6151
6152 020612 004037 034426 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
6153 020616 003220 WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
6154 020620 000070 56. ;56. WORDS
6155 020622 000377 377 ;FILL WITH 377
6156
6157
6158 ;NOW FILL COMMAND
6159
6160 020624 004037 036532 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6161 020630 000000 0 ;CYLINDER 0
6162 020632 000 .BYTE 0 ;SECTOR 0
6163 020633 000 .BYTE 0 ;TRACK 0
6164 020634 177470 -200. ;WORD COUNT = 200.
6165 020636 003444 REINTO ;BUS ADDRESS
6166 ;STARTING ADDRESS OF DATA
6167 ;BUFFER = REINTO
6168 020640 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6169 020642 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
6170 ;INHIBIT ECC CORRECTION
6171 ;DO NOT INHIBIT HEADER COMPARE
6172 020644 002360 READAT ;GET READY TO DO A READAT
6173 ;READ DATA WITH 70 IN RHCS1
6174
6175
6176 ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6177 020646 004037 034716 JSR RO,@#SAVER ;SAVE REGISTERS
6178 020652 002210 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6179 020654 004522 SAVERE ;STARTING ADDRESS OF WHERE
6180 ;THE REGISTERS ARE SAVED
6181 020656 000022 18. ;NUMBER OF REGISTERS
6182 ;SAVED = 18.
6183
6184 020660 004737 034636 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6185
6186
6187 020664 013777 004516 161312 MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
6188 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6189 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6190 ;'TIME' WILL ONLY SAVE
6191 ;CURRENT CYLINDER ADDRESS
6192 ;AND LOOK AHEAD REGISTERS
6193

```



```

6194
6195 020672 013746 002360      MOV      @#READAT,-(SP) ;GET READY TO MOVE COMMAND
6196 020676 052716 000101      BIS      #GO!IE,(SP) ;GET READY TO SET GO AND
6197                                     ;ENABLE INTERRUPT
6198 020702 012677 161310      MOV      (SP)+,@RHCS1 ;GO WITH
6199                                     ;70 IN RHCS1 FOR READ DATA
6200                                     ;WITH INTERRUPT ENABLED
6201 020706 011100      MOV      @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6202 020710 011305      MO      @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6203
6204
6205 020712 104410      WAT                                     ;WAIT FOR RDY BIT TO SET
6206 020714 002216      RHCS1 ;WAIT FOR RHCS1 REGISTER
6207 020716 000200      RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6208 020720 001614      908. ;ALLOW 9080 MICRO SECONDS
6209 020722 001507      839. ;RDY MUST SET BETWEEN
6210                                     ;690 AND 17470 MICRO SECONDS
6211                                     ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6212                                     ;R0 AND R5 IMMEDIATELY AFTER GO
6213 020724 013746 002360      MOV      @#READAT,-(SP) ;SAVE COMMAND
6214 020730 052716 004101      BIS      #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6215 020734 011637 001124      MOV      (SP),@#$GDDAT ;SAVE FOR PRINTOUT
6216 020740 022600      CMP      (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6217                                     ;AND COMMAND SHOULD BE SET
6218 020742 001405      BEQ      67$ ;BRANCH IF GOOD
6219 020744 010037 001126      MOV      R0,@#$BDDAT ;BAD DATA
6220 020750 010137 004510      MOV      R1,@#REGADR ;FAILING REGISTER RHCS1
6221 020754 104021      ERROR    21 ;DURING ABOVE OPERATION ONLY
6222                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
6223 020756 012746 010500      67$: MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6224 020762 011637 001124      MOV      (SP),@#$GDDAT ;SAVE FOR PRINTOUT
6225 020766 022605      CMP      (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6226                                     ;SHOULD BE SET
6227 020770 001405      BEQ      69$ ;BRANCH IF GOOD
6228 020772 010537 001126      MOV      R5,@#$BDDAT ;BAD DATA
6229 020776 010337 004510      MOV      R3,@#REGADR ;FAILING REGISTER RHDS1
6230 021002 104063      ERROR    63 ;DURING ABOVE OPERATION ONLY
6231                                     ;MOL!DPR!VV SHOULD BE SET
6232 021004      69$:
6233
6234                                     ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
6235 021004 004037 034460      JSR      RO,@#FILLRE ;MOV 0 INTO SAVED RHWC
6236 021010 002210      RHWC ;SAVED REGISTER TO CHANGE
6237 021012 000000      0 ;DATA
6238 021014 004037 034460      JSR      RO,@#FILLRE ;MOV REINTO+(200.*2) INTO SAVED RHBA
6239 021020 002212      RHBA ;SAVED REGISTER TO CHANGE
6240 021022 004264      REINTO+(200.*2) ;DATA
6241 021024 004037 034460      JSR      RO,@#FILLRE ;MOV 1 INTO SAVED RHDST
6242 021030 002222      RHDST ;SAVED REGISTER TO CHANGE
6243 021032 000001      1 ;DATA
6244
6245                                     ;COMPARE REGISTERS BEFORE READ DATA COMMAND
6246                                     ;WITH REGISTERS AFTER COMMAND
6247
6248 021034 004037 035546      JSR      RO,@#COMREG ;COMPARE SAVED REGISTERS WITH
6249                                     ;PRESENT VALUE

```

```

:GOOD DATA SAVED IN 'SAVERE'
:TEST DATA STARTING FROM 'RMWC'
:18. REGISTERS TO BE COMPARED
:RETURN TO 55 ON ERROR
:RETURN TO 65 ON NO ERROR

:READ DATA CAUSED IMPROPER
:REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD BE THERE
:RECEIVED DATA GIVES WHAT WAS THERE
:AFTER COMMAND

:NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

:COMPARE TWO BLOCKS OF MEMORY
:GOOD DATA STARTS FROM 'RFRON'
:WAST DATA STARTS FROM 'REINTO'
:18. WORDS TO BE COMPARED
:RETURN TO 75 ON ERROR
:RETURN TO 105 ON NO ERROR

:INCORRECT DATA AFTER
:READ DATA FOLLOWED BY A
:READ DATA

```

```

SAVERE
NO
10.
55.
65.

55:
75:
105:

65:
75:
105:

RC.2800PAR
RFRON
REINTO
18.
75.
105.

75:
105:

```

```

:GOOD DATA SAVED IN 'SAVERE'
:TEST DATA STARTING FROM 'RMWC'
:18. REGISTERS TO BE COMPARED
:RETURN TO 55 ON ERROR
:RETURN TO 65 ON NO ERROR

:READ DATA CAUSED IMPROPER
:REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD BE THERE
:RECEIVED DATA GIVES WHAT WAS THERE
:AFTER COMMAND

:NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

:COMPARE TWO BLOCKS OF MEMORY
:GOOD DATA STARTS FROM 'RFRON'
:WAST DATA STARTS FROM 'REINTO'
:18. WORDS TO BE COMPARED
:RETURN TO 75 ON ERROR
:RETURN TO 105 ON NO ERROR

:INCORRECT DATA AFTER
:READ DATA FOLLOWED BY A
:READ DATA

```

```

:GOOD DATA SAVED IN 'SAVERE'
:TEST DATA STARTING FROM 'RMWC'
:18. REGISTERS TO BE COMPARED
:RETURN TO 55 ON ERROR
:RETURN TO 65 ON NO ERROR

:READ DATA CAUSED IMPROPER
:REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD BE THERE
:RECEIVED DATA GIVES WHAT WAS THERE
:AFTER COMMAND

:NOW READ INTO BUFFER IS CHECKED FOR GOOD READ

:COMPARE TWO BLOCKS OF MEMORY
:GOOD DATA STARTS FROM 'RFRON'
:WAST DATA STARTS FROM 'REINTO'
:18. WORDS TO BE COMPARED
:RETURN TO 75 ON ERROR
:RETURN TO 105 ON NO ERROR

:INCORRECT DATA AFTER
:READ DATA FOLLOWED BY A
:READ DATA

```

021100  
021101  
021102  
021103  
021104  
021105  
021106  
021107  
021108  
021109  
021110  
021111  
021112  
021113  
021114  
021115  
021116  
021117  
021118  
021119  
021120  
021121  
021122  
021123  
021124  
021125  
021126  
021127  
021128  
021129  
021130  
021131  
021132  
021133  
021134  
021135  
021136  
021137  
021138  
021139  
021140  
021141  
021142  
021143  
021144

000004  
012706 001000  
012737 000025 004514  
004737 034556  
004037 034426  
002400  
000400  
125252  
004037 034426  
003444  
000400  
125252  
004037 036532

```
*****
*TEST 25      WRITE/READ DATA
*
* THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER C
* TRACK 0, SECTOR 0, KEYS 0, 256 WORDS OF 125252
* THIS SECTOR IS FORMATED BY PREVIOUS TEST
* THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
* SAME CYLINDER, TRACK, SECTOR, KEYS
* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
* 256 OF 125252
* THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
* ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA
*
* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
* CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
*
* THEN READ INTO BUFFER IS FILLED WITH 256 OF ZEROS
* WRITE FROM BUFFER IS FILLED WITH
* 256 OF 125252
* THE COMMAND EXCEPT GO IS LOADED. ALL REGISTERS ARE SAVED
* GO IS GIVEN
*
* ALL REGISTERS ARE CHECKED
* READ DATA IS CHECKED
*
```

```
*****
*TEST25:  SCOPE
*          MOV      #STACK.SP      ;RESET STACK
*          MOV      #25,#STNM      ;SAVE TEST NUMBER
*
*          JSR      PC,#CLDISK     ;SET R1-RHCS1, R2-RHCS2
*                                     ;R3-RHDS1, R4-RHER1
*                                     ;GIVE RH-11 INITIALIZE
*                                     ;SETUP UNIT NUBER
*
*          ;NOW FILL WRITE FROM BUFFER - 256 OF 125252
*          JSR      RD,#CLAREA     ;CLEAR 256. WORDS, FROM WRFROM
*          WRFROM   ;STARTING FROM WRFROM
*          256.    ;256. WORDS
*          125252  ;FILL WITH 125252
*
*          ;NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
*          ;WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
*          ;CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
*          JSR      RD,#CLAREA     ;CLEAR 256. WORDS, FROM REINTO
*          REINTO  ;STARTING FROM REINTO
*          256.    ;256. WORDS
*          125252  ;FILL WITH 125252
*
*          ;NOW WRITE DATA COMMAND WILL BE LOADED
*          JSR      RD,#RUN        ;SETUP TO RUN FOR DATA COMMAND
*
```

```

6337 021150 000000          0          ;CYLINDER 0
6338 021152 000          0          ;SECTOR 0
6339 021153 000          0          ;TRACK 0
6340 021154 177400        -256.        ;WORD COUNT = 256.
6341 021156 002400        WRFROM        ;BUS ADDRESS
6342          ;STARTING ADDRESS OF DATA
6343          ;BUFFER = WRFROM
6344 021160 000000          J          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6345 021162 010000        FMT22        ;16 BITS PER WORD FORMAT
6346          ;DO NOT INHIBIT ECC CORRECTION
6347          ;DO NOT INHIBIT HEADER COMPARE
6348 J021164 J02354        WRIDAT        ;GET READY TO DO A WRIDAT
6349          ;WRITE DATA WITH 60 IN RHCS1
6350
6351          ;NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE DATA
6352 JSR          RO,2#SAVER ;SAVE REGISTERS
6353 021166 004037 034716   RHWC        ;RHWC IS THE FIRST REGISTER SAVED
6354 021172 002210        SAVERE        ;STARTING ADDRESS OF WHERE
6355 021174 004522        ;THE REGISTERS ARE SAVED
6356          ;NUMBER OF REGISTERS
6357 021176 000022        18.          ;SAVED = 18.
6358
6359
6360 021200 004737 034636   JSR          PC,2#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6361
6362
6363 J021204 013777 004516 160772 MOV          2#RP4VEC,2#RPEC ;SET RP04 VECTOR ADDRESS
6364          ;TO 'TIME1' IF P-CLOCK IS PRESENT
6365          ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6366          ;'TIME' WILL ONLY SAVE
6367          ;CURRENT CYLINDER ADDRESS
6368          ;AND LOOK AHEAD REGISTERS
6369
6370
6371 021210 013746 002354   MOV          2#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
6372 021216 052716 000101   BIS          #GO!IE,(SP)   ;GET READY TO SET GO AND
6373          ;ENABLE INTERRUPT
6374 021222 012677 160770   MOV          (SP)+,2#RHCS1 ;GO WITH
6375          ;60 IN RHCS1 FOR WRITE DATA
6376          ;WITH INTERRUPT ENABLED
6377 021226 011100        MOV          2#R1,RO       ;SAVE RHCS1 DURING ABOVE OPERATION
6378 021230 011305        MOV          2#R3,R5       ;SAVE RHCS1 DURING ABOVE OPERATION
6379
6380          ;ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
6381
6382 021232 104410        WAT          ;WAIT FOR RDY BIT TO SET
6383 021234 002216        RHCS1        ;WAIT FOR RHCS1 REGISTER
6384 021236 000200        RDY        ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6385 021240 001614        908.        ;ALLOW 9080 MICRO SECONDS
6386 021242 001507        839.        ;RDY MUST SET BETWEEN
6387          ;690 AND 17470 MICRO SECONDS
6388          ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6389          ;RO AND R5 IMMEDIATELY AFTER GO
6390 021244 013746 002354   MOV          2#WRIDAT,-(SP) ;SAVE COMMAND
6391 021250 052716 004101   BIS          #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6392 021254 011637 001124   MOV          (SP),2#SGDAT  ;SAVE FOR PRINTOUT

```

```

6393 021260 022600          CMP      (SP)+,RO      ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6394                                ;AND COMMAND SHOULD BE SET
6395 021262 001405          BEQ      64$          ;BRANCH IF GOOD
6396 021264 010037 001126   MOV      RO,2#SBDDAT  ;BAD DATA
6397 021270 010137 004510   MOV      R1,2#REGADR ;FAILING REGISTER RHCS1
6398 021274 104021          ERROR    21          ;DURING ABOVE OPERATION ONLY
6399                                ;COMMAND AND IE!GO!DVA SHOULD BE SET
6400 021276 012746 010500   64$:  MOV      #MOL!DPR!VV, -(SP) ;SAVE BITS SET DURING OPERATION IN RHCS1
6401 021302 011637 001124   MOV      (SP),2#SGDDAT ;SAVE FOR PRINTOUT
6402 021306 022605          CMP      (SP)+,R5     ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6403                                ;SHOULD BE SET
6404 021310 001405          BEQ      66$          ;BRANCH IF GOOD
6405 021312 010537 001126   MOV      R5,2#SDDAT  ;BAD DATA
6406 021316 010337 004510   MOV      R3,2#REGADR ;FAILING REGISTER RHDS1
6407 021322 104063          ERROR    63          ;DURING ABOVE OPERATION ONLY
6408                                ;MOL!DPR!VV SHOULD BE SET
6409 021324          66$:
6410
6411                                ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6412 021324 004037 034460   JSR      RO,2#FILLRE ;MOV 0 INTO SAVED RHWC
6413                                ;SAVED REGISTER TO CHANGE
6414                                ;DATA
6415 021334 004037 034460   JSR      RO,2#FILLRE ;MOV WRFROM+(256.*2) INTO SAVED RHBA
6416                                ;SAVED REGISTER TO CHANGE
6417                                ;DATA
6418 021344 004037 034460   JSR      RO,2#FILLRE ;MOV 1 INTO SAVED RHDST
6419                                ;SAVED REGISTER TO CHANGE
6420 021352 000001          1          ;DATA
6421
6422                                ;NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6423                                ;AFTER COMMAND
6424
6425 021354 004037 035546   JSR      RO,2#COMREG ;COMPARE SAVED REGISTERS WITH
6426                                ;PRESENT VALUE
6427                                ;GOOD DATA SAVED IN 'SAVERE'
6428                                ;TEST DATA STARTING FROM 'RHWC'
6429                                ;18. REGISTERS TO BE COMPARED
6430                                ;RETURN TO 1$ ON ERROR
6431                                ;RETURN TO 2$ ON NO ERROR
6432 021372 104035          1$:  ERROR    35
6433 021374 000207          RTS      PC
6434                                ;WRITE DATA COMMAND CAUSED
6435                                ;IMPROPER REGISTER CHANGE
6436                                ;GOOD DATA GIVES WHAT SHOULD
6437                                ;BE
6438                                ;RECEIVED DATA GIVES WHAT WAS
6439                                ;THERE AFTER COMMAND
6440
6441                                ;NOW WRITE FROM BUFFER WILL CHECKED FOR NO CHANGE
6442          2$:
6443 021376 004037 036576   JSR      RO,2#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6444 021402 003444          REINTO  ;GOOD DATA STARTS FROM REINTO
6445 021404 002400          WRFROM  ;TEST DATA STARTS FROM WRFROM
6446 021406 000400          256.   ;256. WORDS TO BE COMPARED
6447 021410 021414          3$     ;RETURN TO 3$ ON ERROR
6448 021412 021420          4$     ;RETURN TO 4$ ON NO ERROR

```

```

6449
6450
6451 021414 104036      3$:  ERROR 36      ;WRITE DATA COMMAND CHANGED
6452 021416 000207      RTS      PC      ;WRITE FROM BUFFER
6453
6454      ;NOW A READ DATA COMMAND WILL BE GIVEN
6455
6456      ;FILL READ INTO BUFFER WITH 256 ZEROS
6457 021420      4$:
6458
6459 021420 004737 034556  JSR      PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
6460      ;R3-RHDS1, R4-RHER1
6461      ;GIVE RH-11 INITIALIZE
6462      ;SETUP UNIT NUMBER
6463 021424 004037 034426  JSR      RD,2#CLAREA ;CLEAR 256. WORDS, FROM REINTO
6464 021430 003444      REINTO      ;STARTING FROM REINTO
6465 021432 003400      256.      ;256. WORDS
6466 021434 000000      0          ;FILL WITH 0
6467
6468
6469      ;FILL WRITE FROM BUFFER WITH 256 OF 125252
6470 021436 004037 034426  JSR      RD,2#CLAREA ;CLEAR 256. WORDS, FROM WRFROM
6471 021442 002400      WRFROM      ;STARTING FROM WRFROM
6472 021444 000400      256.      ;256. WORDS
6473 021446 125252      125252     ;FILL WITH 125252
6474
6475
6476      ;NOW FILL COMMAND
6477
6478 021450 004037 036532  JSR      RD,2#RUN    ;SETUP TO RUN FOR DATA COMMAND
6479 021454 000000      0          ;CYLINDER 0
6480 021456 000      .BYTE 0      ;SECTOR 0
6481 021457 000      .BYTE 0      ;TRACK 0
6482 021460 177400      -256.     ;WORD COUNT = 256.
6483 021462 003444      REINTO     ;BUS ADDRESS
6484      ;STARTING ADDRESS OF DATA
6485      ;BUFFER = REINTO
6486 021464 000000      0          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6487 021466 014000      ECI!FMT22 ;16 BITS PER WORD FORMAT
6488      ;INHIBIT ECC CORRECTION
6489      ;DO NOT INHIBIT HEADER COMPARE
6490 021470 002360      READAT    ;GET READY TO DO A READAT
6491      ;READ DATA WITH 70 IN RHCS1
6492
6493
6494      ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6495 021472 004037 034716  JSR      RD,2#SAVER ;SAVE REGISTERS
6496 021476 002210      RHWC      ;RHWC IS THE FIRST REGISTER SAVED
6497 021500 034522      SAVER     ;STARTING ADDRESS OF WHERE
6498      ;THE REGISTERS ARE SAVED
6499 021502 000022      18.      ;NUMBER OF REGISTERS
6500      ;SAVED = 18.
6501
6502 021504 004737 034636  JSR      PC,2#CHECKT ;CHECK DVA.RDY,MOL,DPR.DRY,VV
6503
6504

```

MAINDEC-11-DERPU-B  
DERPUB.F11 T25MACY11 27(732) 17-SEP-76 13:23 PAGE 136  
WRITE/READ DATA

```

6505 021510 013777 004516 160466      MOV      @#RPHVEC,@RPVEC      ;SET RPO4 VECTOR ADDRESS
6506                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
6507                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6508                                     ;'TIME' WILL ONLY SAVE
6509                                     ;CURRENT CYLINDER ADDRESS
6510                                     ;AND LOOK AHEAD REGISTERS
6511
6512
6513 021516 013746 002360      MOV      @#READAT,-(SP)      ;GET READY TO MOVE COMMAND
6514 021522 052716 000101      BIS      #GO!IE,(SP)        ;GET READY TO SET GO AND
6515                                     ;ENABLE INTERRUPT
6516 021526 012677 160464      MOV      (SP)+,@RHCS1        ;GO WITH
6517                                     ;TO IN RHCS1 FOR READ DATA
6518                                     ;WITH INTERRUPT ENABLED
6519 021532 011100      MOV      @R1,R0              ;SAVE RHCS1 DURING ABOVE OPERATION
6520 021534 011305      MOV      @R3,R5              ;SAVE RHDS1 DURING ABOVE OPERATION
6521
6522
6523 021536 104410      WAT                                     ;WAIT FOR RDY BIT TO SET
6524 021540 002216      RHCS1                             ;WAIT FOR RHCS1 REGISTER
6525 021542 000200      RDY                                 ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6526 021544 001614      908.                               ;ALLOW 9080 MICRO SECONDS
6527 021546 001507      839.                               ;RDY MUST SET BETWEEN
6528                                     ;690 AND 17470 MICRO SECONDS
6529                                     ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6530                                     ;R0 AND R5 IMMEDIATELY AFTER GO
6531 021550 013746 002360      MOV      @#READAT,-(SP)      ;SAVE COMMAND
6532 021554 052716 004101      BIS      #IE!GO!DVA,(SP)    ;INCLUDE IE!GO!DVA
6533 021560 011637 001124      MOV      (SP),@#SGDDAT      ;SAVE FOR PRINTOUT
6534 021564 022600      CMP      (SP)+,R0           ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6535                                     ;AND COMMAND SHOULD BE SET
6536 021566 001405      BEQ      67$                ;BRANCH IF GOOD
6537 021570 010037 001126      MOV      R0,@#SBDDAT        ;BAD DATA
6538 021574 010137 004510      MOV      R1,@#REGADR        ;FAILING REGISTER RHCS1
6539 021600 104021      ERROR   21                 ;DURING ABOVE OPERATION ONLY
6540                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
6541 021602 012746 010500      67$: MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6542 021606 011637 001124      MOV      (SP),@#SGDDAT      ;SAVE FOR PRINTOUT
6543 021612 022605      CMP      (SP)+,R5           ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6544                                     ;SHOULD BE SET
6545 021614 001405      BEQ      69$                ;BRANCH IF GOOD
6546 021616 010537 001126      MOV      R5,@#SBDDAT        ;BAD DATA
6547 021622 010337 004510      MOV      R3,@#REGADR        ;FAILING REGISTER RHDS1
6548 021626 104063      ERROR   63                 ;DURING ABOVE OPERATION ONLY
6549                                     ;MOL!DPR!VV SHOULD BE SET
6550 021630      69$:
6551
6552                                     ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
6553 021630 004037 034460      JSR      R0,@#FILLRE        ;MOV 0 INTO SAVED RHWC
6554 021634 002210      RHWC                             ;SAVED REGISTER TO CHANGE
6555 021636 000000      0                                 ;DATA
6556 021640 004037 034460      JSR      R0,@#FILLRE        ;MOV REINT0+(256.*2) INTO SAVED RHBA
6557 021644 002212      RHBA                             ;SAVED REGISTER TO CHANGE
6558 021646 004444      REINT0+(256.*2)              ;DATA
6559 021650 004037 034460      JSR      R0,@#FILLRE        ;MOV 1 INTO SAVED RHDST
6560 021654 002222      RHDST                             ;SAVED REGISTER TO CHANGE

```

```

6551 021656 000001          1          ;DATA
6552
6553          ;COMPARE REGISTERS BEFORE READ DATA COMMAND
6554          ;WITH REGISTERS AFTER COMMAND
6555
6556 021660 004037 035546    JSR      RD,2*COMREG    ;COMPARE SAVED REGISTERS WITH
6557          ;PRESENT VALUE
6558 021664 004522          SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
6559 021666 002266          WC          ;TEST DATA STARTING FROM 'RHWC'
6570 021670 000022          18.       ;18. REGISTERS TO BE COMPARED
6571 021672 021676          5$         ;RETURN TO 5$ ON ERROR
6572 021674 021702          6$         ;RETURN TO 6$ ON NO ERROR
6573
6574 021676 104033          5$:      ERROR   33    ;READ DATA CAUSED IMPROPER
6575 021700 000207          RTS      PC      ;REGISTER CHANGE
6576          ;GOOD DATA GIVES WHAT SHOULD BE THE
6577          ;RECEIVED DATA GIVES WHAT WAS THERE
6578          ;AFTER COMMAND
6579
6580          ;NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
6581 021702          6$:
6582
6583 021702 004037 036576    JSR      RD,2*COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
6584 021706 002400          WRFROM    ;GOOD DATA STARTS FROM WRFROM
6585 021710 003444          REINTO    ;TEST DATA STARTS FROM REINTO
6586 021712 000400          256.     ;256. WORDS TO BE COMPARED
6587 021714 021720          7$         ;RETURN TO 7$ ON ERROR
6588 021716 021724          10$        ;RETURN TO 10$ ON NO ERROR
6589
6590
6591 021720 104034          7$.     ERROR   34    ;INCORRECT DATA AFTER
6592 021722 000207          RTS      PC      ;WRITE DATA FOLLOWED BY A
6593          ;READ DATA
6594 021724          10$:
6595
6596          ;*****
6597          ;*TEST 26      WRITE/READ DATA
6598
6599          ;*
6600          ;*      THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER 0
6601          ;*      TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
6602          ;*      THIS SECTOR IS FORMATED BY PREVIOUS TEST
6603          ;*      THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
6604          ;*      SAME CYLINDER, TRACK, SECTOR, KEYS
6605          ;*      WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
6606          ;*      200 OF 052525 AND 56 OF 377
6607          ;*      THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
6608          ;*      ALL REGISTERS ARE SAVED AND THEN GO IS TO WRITE DATI
6609          ;*
6610          ;*
6611          ;*      THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
6612          ;*      CHANGE, THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE
6613          ;*
6614          ;*      THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS
6615          ;*      AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200
6616          ;*      WORDS OF 52525 AND 56 WORDS OF 0
6616          ;*      THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED

```



```

6617      :*      GU IS GIVEN
6618      :*
6619      :*
6620      :*      ALL REGISTER ARE CHECKED
6621      :*      READ DATA IS CHECKED
6622
6623      :*****
6624      021724 000004      †ST26: SCOPE
6625      021726 012706 001000      MOV      #STACK,SP      ;RESET STACK
6626      021732 012737 000026 004514      MOV      #26,‡#†STNM      ;SAVE TEST NUMBER
6627
6628      021740 004737 034556      JSR      PC,‡#CLDISK      ;SET R1-RHCS1, R2-RHCS2
6629      ;R3-RHDS1, R4-RHER1
6630      ;GIVE RH-11 INITIALIZE
6631      ;SETUP UNIT NUMBER
6632
6633      ;NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377
6634      021744 004037 034426      JSR      RO,‡#CLAREA      ;CLEAR 200. WORDS, FROM WRFROM
6635      WRFROM      ;STARTING FROM WRFROM
6636      200.      ;200. WORDS
6637      021754 052525      52525      ;FILL WITH 52525
6638
6639      021756 004037 034426      JSR      RO,‡#CLAREA      ;CLEAR 56. WORDS, FROM WRFROM+(200.*2)
6640      021762 003220      WRFROM+(200.*2)      ;STARTING FROM WRFROM+(200.*2)
6641      021764 000070      56.      ;56. WORDS
6642      021766 000377      377      ;FILL WITH 377
6643
6644
6645      ;NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
6646      ;WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
6647      ;CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
6648      021770 004037 034426      JSR      RO,‡#CLAREA      ;CLEAR 200. WORDS, FROM REINTO
6649      021774 003444      REINTO      ;STARTING FROM REINTO
6650      021776 000310      200.      ;200. WORDS
6651      022000 052525      52525      ;FILL WITH 52525
6652
6653      022002 004037 034426      JSR      RO,‡#CLAREA      ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
6654      022006 004264      REINTO+(200.*2)      ;STARTING FROM REINTO+(200.*2)
6655      022010 000070      56.      ;56. WORDS
6656      022012 000377      377      ;FILL WITH 377
6657
6658
6659      ;NOW WRITE DATA COMMAND WILL BE LOADED
6660
6661      022014 004037 036532      JSR      RO,‡#RUN      ;SETUP TO RUN FOR DATA COMMAND
6662      022020 000000      0      ;CYLINDER 0
6663      022022 000      .BYTE 0      ;SECTOR 0
6664      022023 000      .BYTE 0      ;TRACK 0
6665      022024 177470      -200.      ;WORD COUNT = 200.
6666      022026 002400      WRFROM      ;BUS ADDRESS
6667      ;STARTING ADDRESS OF DATA
6668      ;BUFFER = WRFROM
6669      022030 000000      0      ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6670      022032 010000      FMT22      ;16 BITS PER WORD FORMAT
6671      ;DO NOT INHIBIT ECC CORRECTION
6672      ;DO NOT INHIBIT HEADER COMPARE

```

```

6673 022034 002354 WRIDAT ;GET READY TO DO A WRIDAT
6674 ;WRITE DATA WITH 60 IN RHCS1
6675
6676
6677 ;NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
6678 022036 004037 034716 JSR RO,@#SAVER ;SAVE REGISTERS
6679 022042 002210 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6680 022044 004522 SAVERE ;STARTING ADDRESS OF WHERE
6681 ;THE REGISTERS ARE SAVED
6682 022046 000022 18. ;NUMBER OF REGISTERS
6683 ;SAVED = 18.
6684
6685 022050 004737 034636 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6686
6687
6688 022054 013777 004516 160122 MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
6689 ;TO 'TIME1' IF P-CLOCK IS PRESENT
6690 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6691 ;'TIME' WILL ONLY SAVE
6692 ;CURRENT CYLINDER ADDRESS
6693 ;AND LOOK AHEAD REGISTERS
6694
6695
6696 022062 013746 002354 MOV @#WRIDAT,-(SP) ;GET READY TO MOVE COMMAND
6697 022066 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
6698 ;ENABLE INTERRUPT
6699 022072 012677 160120 MOV (SP)+,@RHCS1 ;GO WITH
6700 ;60 IN RHCS1 FOR WRITE DATA
6701 ;WITH INTERRUPT ENABLED
6702 022076 011100 MOV @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6703 022100 011305 MOV @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6704
6705 ;ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
6706
6707 022102 104410 WAT ;WAIT FOR RDY BIT TO SET
6708 022104 002216 RHCS1 ;WAIT FOR RHCS1 REGISTER
6709 022106 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6710 022110 001614 908. ;ALLOW 9080 MICRO SECONDS
6711 022112 001507 839. ;RDY MUST SET BETWEEN
6712 ;690 AND 17470 MICRO SECONDS
6713 ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6714 ;R0 AND R5 IMMEDIATELY AFTER GO
6715 022114 013746 002354 MOV @#WRIDAT,-(SP) ;SAVE COMMAND
6716 022120 052716 004101 BIS #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6717 022124 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6718 022130 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6719 ;AND COMMAND SHOULD BE SET
6720 022132 001405 BEQ 64$ ;BRANCH IF GOOD
6721 022134 010037 001126 MOV R0,@#SBDDAT ;BAD DATA
6722 022140 010137 004510 MOV R1,@#REGADR ;FAILING REGISTER RHCS1
6723 022144 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
6724 ;COMMAND AND IE!GO!DVA SHOULD BE SET
6725 022146 012746 010500 64$: MOV #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6726 022152 011637 001124 MOV (SP),@#SGDDAT ;SAVE FOR PRINTOUT
6727 022156 022600 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6728 ;SHOULD BE SET

```



```

6729 022160 001405          BEQ      66$      ;BRANCH IF GOOD
6730 022162 010537 001126  MOV      R5,2#$BDDAT ;BAD DATA
6731 022166 010337 004510  MOV      R3,2#$REGADR ;FAILING REGISTER RHDS!
6732 022172 104053          ERROR    63        ;DURING ABOVE OPERATION ONLY
6733                                     ;MOL!DPR!VV SHOULD BE SET
6734 022174          66$:
6735
6736                                     ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
6737 022174 004037 034460  JSR      RO,2#FILLRE ;MOV 0 INTO SAVED RHWC
6738 022200 002210          RHWC      ;SAVED REGISTER TO CHANGE
6739 022202 000000          0        ;DATA
6740 022204 004037 034460  JSR      RO,2#FILLRE ;MOV WRFROM+(200.*2) INTO SAVED RHBA
6741 022210 002212          RHBA      ;SAVED REGISTER TO CHANGE
6742 022212 003220          WRFROM+(200.*2) ;DATA
6743 022214 004037 034460  JSR      RO,2#FILLRE ;MOV 1 INTO SAVED RHDST
6744 022220 002222          RHDST     ;SAVED REGISTER TO CHANGE
6745 022222 000001          1        ;DATA
6746
6747                                     ;NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
6748                                     ;AFTER COMMAND
6749
6750 022224 004037 035546  JSR      RO,2#COMREG ;COMPARE SAVED REGISTERS WITH
6751                                     ;PRESENT VALUE
6752 022230 004522          SAVERE     ;GOOD DATA SAVED IN 'SAVERE'
6753 022232 002266          WC        ;TEST DATA STARTING FROM 'RHWC'
6754 022234 000022          18.     ;18. REGISTERS TO BE COMPARED
6755 022236 022242          1$      ;RETURN TO 1$ ON ERROR
6756 022240 022246          2$      ;RETURN TO 2$ ON NO ERROR
6757
6758 022242 104035          1$:      ERROR    35      ;WRITE DATA COMMAND CAUSED
6759 022244 000207          RTS      PC      ;IMPROPER REGISTER CHANGE
6760                                     ;GOOD DATA GIVES WHAT SHOULD
6761                                     ;BE
6762                                     ;RECEIVED DATA GIVES WHAT WAS
6763                                     ;THERE AFTER COMMAND
6764
6765                                     ;NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
6766 022246          2$:
6767
6768 022246 004037 036576  JSR      RO,2#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6769 022252 003444          REINTO     ;GOOD DATA STARTS FROM REINTO
6770 022254 002400          WRFROM    ;TEST DATA STARTS FROM WRFROM
6771 022256 000400          256.     ;256. WORDS TO BE COMPARED
6772 022260 022264          3$      ;RETURN TO 3$ ON ERROR
6773 022262 022270          4$      ;RETURN TO 4$ ON NO ERROR
6774
6775
6776 022264 104036          3$:      ERROR    36      ;WRITE DATA COMMAND CHANGED
6777 022266 000207          RTS      PC      ;WRITE FROM BUFFER
6778
6779                                     ;NOW A READ DATA COMMAND WILL BE GIVEN
6780
6781                                     ;FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
6782 022270          4$:
6783
6784 022270 004737 034556  JSR      PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2

```

```

6795 ;R3-RHDS1, R4-RHER1
6796 ;GIVE RH-11 INITIALIZE
6797 ;SETUP UNIT NUBER
6798 ;CLEAR 200. WORDS, FROM REINTO
6799 022274 004037 034426 JSR RO,@#CLAREA ;STARTING FROM REINTO
6799 022300 003444 REINTO ;200. WORDS
6799 022302 000310 200. ;FILL WITH 0
6799 022304 000000 0
6792
6793 022306 004037 034426 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM REINTO+<200.*2>
6794 022312 004264 REINTO+<200.*2> ;STARTING FROM REINTO+<200.*2>
6795 022314 000070 56. ;56. WORDS
6796 022316 000377 377 ;FILL WITH 377
6797
6798
6799 ;FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
6900 022320 004037 034426 JSR RO,@#CLAREA ;CLEAR 200. WORDS, FROM WRFROM
6901 022324 002400 WRFROM ;STARTING FROM WRFROM
6802 022326 000310 200. ;200. WORDS
6803 022330 052525 52525 ;FILL WITH 52525
6804
6805 022332 004037 034426 JSR RO,@#CLAREA ;CLEAR 56. WORDS, FROM WRFROM+<200.*2>
6806 022336 003220 WRFROM+<200.*2> ;STARTING FROM WRFROM+<200.*2>
6807 022340 000070 56. ;56. WORDS
6808 022342 000377 377 ;FILL WITH 377
6809
6810
6811 ;NOW FILL COMMAND
6812
6813 022344 004037 036532 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
6814 022350 000000 0 ;CYLINDER 0
6815 022352 000 .BYTE 0 ;SECTOR 0
6816 022353 000 .BYTE 0 ;TRACK 0
6817 022354 177470 -200. ;WORD COUNT = 200.
6818 022356 003444 REINTO ;BUS ADDRESS
6819 ;STARTING ADDRESS OF DATA
6820 ;BUFFER = REINTO
6821 022360 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
6822 022362 014000 ECI!FMT2 ;16 BITS PER WORD FORMAT
6823 ;INHIBIT ECC CORRECTION
6824 ;DO NOT INHIBIT HEADER COMPARE
6825 022364 002360 READAT ;GET READY TO DO A READAT
6826 ;READ DATA WITH 70 IN RHCS1
6827
6828
6829 ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
6830 022366 004037 034716 JSR RO,@#SAVER ;SAVE REGISTERS
6831 022372 002210 RHWC ;RHWC IS THE FIRST REGISTER SAVED
6832 022374 004522 SAVERE ;STARTING ADDRES OF WHERE
6833 ;THE REGISTERS ARE SAVED
6834 022376 000022 18. ;NUMBER OF REGISTERS
6835 ;SAVED = 18.
6836
6837 022400 004737 034636 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
6838
6839
6840 022404 013777 004516 157572 MOV @#RP4VEC,@#RPVEC ;SET RPO4 VECTOR ADDRESS

```

```

6841                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
6842                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
6843                                     ;'TIME' WILL ONLY SAVE
6844                                     ;CURRENT CYLINDER ADDRESS
6845                                     ;AND LOOK AHEAD REGISTERS
6846
6847
6848 022412 013746 002360      MOV      @#READAT,-(SP) ;GET READY TO MOVE COMMAND
6849 022416 052716 000101      BIS      #GO!IE,(SP) ;GET READY TO SET GO AND
6850                                     ;ENABLE INTERRUPT
6851 022422 012677 157570      MOV      (SP)+,@RHCS1 ;GO WITH
6852                                     ;70 IN RHCS1 FOR READ DATA
6853                                     ;WITH INTERRUPT ENABLED
6854 022426 011100      MOV      @R1,R0 ;SAVE RHCS1 DURING ABOVE OPERATION
6855 022430 011305      MOV      @R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
6856
6857
6858 022432 104410      WAT      ;WAIT FOR RDY BIT TO SET
6859 022434 002216      RHCS1   ;WAIT FOR RHCS1 REGISTER
6860 022436 000200      RDY     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
6861 022440 001614      908.   ;ALLOW 9080 MICRO SECONDS
6862 022442 001507      839.   ;RDY MUST SET BETWEEN
6863                                     ;690 AND 17470 MICRO SECONDS
6864                                     ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
6865                                     ;R0 AND R5 IMMEDIATELY AFTER GO
6866 022444 013746 002360      MOV      @#READAT,-(SP) ;SAVE COMMAND
6867 022450 052716 004101      BIS      #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
6868 022454 011637 001124      MOV      (SP),@#$GDDAT ;SAVE FOR PRINTOUT
6869 022460 022600      CMP      (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!GO!DVA
6870                                     ;AND COMMAND SHOULD BE SET
6871 022462 001405      BEQ      67$ ;BRANCH IF GOOD
6872 022464 010037 001126      MOV      R0,@#$BDDAT ;BAD DATA
6873 022470 010137 004510      MOV      R1,@#REGADR ;FAILING REGISTER RHCS1
6874 022474 104021      ERROR   21 ;DURING ABOVE OPERATION ONLY
6875                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
6876 022476 012746 010500      67$:   MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
6877 022502 011637 001124      MOV      (SP),@#$GDDAT ;SAVE FOR PRINTOUT
6878 022506 022605      CMP      (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
6879                                     ;SHOULD BE SET
6880 022510 001405      BEQ      69$ ;BRANCH IF GOOD
6881 022512 010537 001126      MOV      R5,@#$BDDAT ;BAD DATA
6882 022516 010337 004510      MOV      R3,@#REGADR ;FAILING REGISTER RHDS1
6883 022522 104063      ERROR   63 ;DURING ABOVE OPERATION ONLY
6884                                     ;MOL!DPR!VV SHOULD BE SET
6885 022524      69$:
6886
6887                                     ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
6888 022524 004037 034460      JSR      R0,@#FILLRE ;MOV 0 INTO SAVED RHWC
6889 022530 002210      RHWC   ;SAVED REGISTER TO CHANGE
6890 022532 000000      0      ;DATA
6891 022534 004037 034460      JSR      R0,@#FILLRE ;MOV REINT0+<200.*2> INTO SAVED RHBA
6892 022540 002212      RHBA   ;SAVED REGISTER TO CHANGE
6893 022542 004264      REINT0+<200.*2> ;DATA
6894 022544 004037 034460      JSR      R0,@#FILLRE ;MOV 1 INTO SAVED RHDST
6895 022550 002222      RHDST ;SAVED REGISTER TO CHANGE
6896 022552 000001      1      ;DATA

```

```

6897
6898 ;COMPARE REGISTERS BEFORE READ DATA COMMAND
6899 ;WITH REGISTERS AFTER COMMAND
6900
6901 022554 004037 035546 JSR RD,2#COMREG ;COMPARE SAVED REGISTERS WITH
6902 ;PRESENT VALUE
6903 022560 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
6904 022562 002266 WC ;TEST DATA STARTING FROM 'RHWC'
6905 022564 000022 18. ;18. REGISTERS TO BE COMPARED
6906 022566 022572 5$ ;RETURN TO 5$ ON ERROR
6907 022570 022576 6$ ;RETURN TO 6$ ON NO ERROR
6908
6909 022572 104033 5$: ERROR 33 ;READ DATA CAUSED IMPROPER
6910 022574 000207 RTS PC ;REGISTER CHANGE
6911 ;GOOD DATA GIVES WHAT SHOULD BE THE
6912 ;RECEIVED DATA GIVES WHAT WAS THERE
6913 ;AFTER COMMAND
6914
6915 ;NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
6916 022576 6$:
6917
6918 022576 004037 036576 JSR RD,2#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
6919 022602 002400 WRFROM ;GOOD DATA STARTS FROM WRFROM
6920 022604 003444 REINTO ;TEST DATA STARTS FROM REINTO
6921 022606 000400 256. ;256. WORDS TO BE COMPARED
6922 022610 022614 7$ ;RETURN TO 7$ ON ERROR
6923 022612 022620 10$ ;RETURN TO 10$ ON NO ERROR
6924
6925
6926 022614 104034 7$: ERROR 34 ;INCORRECT DATA AFTER
6927 022616 000207 RTS PC ;WRITE DATA FOLLOWED BY A
6928 ;READ DATA
6929 022620 10$:
6930
6931
6932
6933
6934 ;*****
6935 ;*TEST 27 WRITE/READ DATA USING UNIBUS B
6936
6937 ;*
6938 ;* THIS TEST USES UNIBUS B IF CONNECTED TO THE RH
6939 ;* IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED
6940 ;* THIS TEST GIVES A WRITE DATA COMMAND FOR CYLINDER J
6941 ;* TRACK 0, SECTOR 0, KEYS 0, 200 WORDS OF 052525
6942 ;* THIS SECTOR IS FORMATED BY PREVIOUS TEST
6943 ;* THEN READ DATA COMMAND IS GIVEN FOR 256 WORDS IN
6944 ;* SAME CYLINDER, TRACK, SECTOR, KEYS
6945 ;* THESE COMMANDS USE UNIBUS B FOR DATA
6946 ;* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
6947 ;* 200 OF 052525 AND 56 OF 377
6948 ;* THE WRITE DATA COMMAND IS LOADED EXCEPT GO AND IE
6949 ;* ALL REGISTERS ARE SAVED AND THEN GO IS GIVEN TO WRITE DATA
6950 ;*
6951 ;*
6952 ;* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER
;* CHANGE. THEN WRITE FROM BUFFER IS CHECKED FOR NO CHANGE

```

THEN READ INTO BUFFER IS FILLED WITH 200 IF ZEROS  
AND 56 ALL ONES, WRITE FROM BUFFER IS FILLED WITH 200  
WORDS OF 52525 AND 56 WORDS OF 0  
THE COMMAND EXCEPT GO IS LOADED, ALL REGISTERS ARE SAVED  
GO IS GIVEN

ALL REGISTER ARE CHECKED  
READ DATA IS CHECKED

\*\*\*\*\*  
STEP: SCOPE

000000	000000	156360
000001	000000	000000
000002	000000	000000
000003	000000	000000
000004	000000	000000
000005	000000	000000
000006	000000	000000
000007	000000	000000
000008	000000	000000
000009	000000	000000
000010	000000	000000
000011	000000	000000
000012	000000	000000
000013	000000	000000
000014	000000	000000
000015	000000	000000
000016	000000	000000
000017	000000	000000
000018	000000	000000
000019	000000	000000
000020	000000	000000
000021	000000	000000
000022	000000	000000
000023	000000	000000
000024	000000	000000
000025	000000	000000
000026	000000	000000
000027	000000	000000
000028	000000	000000
000029	000000	000000
000030	000000	000000
000031	000000	000000
000032	000000	000000
000033	000000	000000
000034	000000	000000
000035	000000	000000
000036	000000	000000
000037	000000	000000
000038	000000	000000
000039	000000	000000
000040	000000	000000
000041	000000	000000
000042	000000	000000
000043	000000	000000
000044	000000	000000
000045	000000	000000
000046	000000	000000
000047	000000	000000
000048	000000	000000
000049	000000	000000
000050	000000	000000
000051	000000	000000
000052	000000	000000
000053	000000	000000
000054	000000	000000
000055	000000	000000
000056	000000	000000
000057	000000	000000
000058	000000	000000
000059	000000	000000
000060	000000	000000
000061	000000	000000
000062	000000	000000
000063	000000	000000
000064	000000	000000
000065	000000	000000
000066	000000	000000
000067	000000	000000
000068	000000	000000
000069	000000	000000
000070	000000	000000
000071	000000	000000
000072	000000	000000
000073	000000	000000
000074	000000	000000
000075	000000	000000
000076	000000	000000
000077	000000	000000
000078	000000	000000
000079	000000	000000
000080	000000	000000
000081	000000	000000
000082	000000	000000
000083	000000	000000
000084	000000	000000
000085	000000	000000
000086	000000	000000
000087	000000	000000
000088	000000	000000
000089	000000	000000
000090	000000	000000
000091	000000	000000
000092	000000	000000
000093	000000	000000
000094	000000	000000
000095	000000	000000
000096	000000	000000
000097	000000	000000
000098	000000	000000
000099	000000	000000
000100	000000	000000
000101	000000	000000
000102	000000	000000
000103	000000	000000
000104	000000	000000
000105	000000	000000
000106	000000	000000
000107	000000	000000
000108	000000	000000
000109	000000	000000
000110	000000	000000
000111	000000	000000
000112	000000	000000
000113	000000	000000
000114	000000	000000
000115	000000	000000
000116	000000	000000
000117	000000	000000
000118	000000	000000
000119	000000	000000
000120	000000	000000
000121	000000	000000
000122	000000	000000
000123	000000	000000
000124	000000	000000
000125	000000	000000
000126	000000	000000
000127	000000	000000
000128	000000	000000
000129	000000	000000
000130	000000	000000
000131	000000	000000
000132	000000	000000
000133	000000	000000
000134	000000	000000
000135	000000	000000
000136	000000	000000
000137	000000	000000
000138	000000	000000
000139	000000	000000
000140	000000	000000
000141	000000	000000
000142	000000	000000
000143	000000	000000
000144	000000	000000
000145	000000	000000
000146	000000	000000
000147	000000	000000
000148	000000	000000
000149	000000	000000
000150	000000	000000
000151	000000	000000
000152	000000	000000
000153	000000	000000
000154	000000	000000
000155	000000	000000
000156	000000	000000
000157	000000	000000
000158	000000	000000
000159	000000	000000
000160	000000	000000
000161	000000	000000
000162	000000	000000
000163	000000	000000
000164	000000	000000
000165	000000	000000
000166	000000	000000
000167	000000	000000
000168	000000	000000
000169	000000	000000
000170	000000	000000
000171	000000	000000
000172	000000	000000
000173	000000	000000
000174	000000	000000
000175	000000	000000
000176	000000	000000
000177	000000	000000
000178	000000	000000
000179	000000	000000
000180	000000	000000
000181	000000	000000
000182	000000	000000
000183	000000	000000
000184	000000	000000
000185	000000	000000
000186	000000	000000
000187	000000	000000
000188	000000	000000
000189	000000	000000
000190	000000	000000
000191	000000	000000
000192	000000	000000
000193	000000	000000
000194	000000	000000
000195	000000	000000
000196	000000	000000
000197	000000	000000
000198	000000	000000
000199	000000	000000
000200	000000	000000
000201	000000	000000
000202	000000	000000
000203	000000	000000
000204	000000	000000
000205	000000	000000
000206	000000	000000
000207	000000	000000
000208	000000	000000
000209	000000	000000
000210	000000	000000
000211	000000	000000
000212	000000	000000
000213	000000	000000
000214	000000	000000
000215	000000	000000
000216	000000	000000
000217	000000	000000
000218	000000	000000
000219	000000	000000
000220	000000	000000
000221	000000	000000
000222	000000	000000
000223	000000	000000
000224	000000	000000
000225	000000	000000
000226	000000	000000
000227	000000	000000
000228	000000	000000
000229	000000	000000
000230	000000	000000
000231	000000	000000
000232	000000	000000
000233	000000	000000
000234	000000	000000
000235	000000	000000

```

MOV #1,STIMES           :: DO 1 ITERATION
MOV #STACK_SP          :: RESET STACK
MOV #RT,RT+STIM        :: SAVE TEST NUMBER

JSR  PC,28CLDISK       :SET R1-RMCS1, R2-RMCS2
                          :R3-RMCS1, R4-RMCS1
                          :GIVE RH-1 INITIALIZE
                          :SETUP UNIT NUMBER

IMP TST30 : JUMP TO NEXT TEST FOR RH7C
CLR 28CLBUS           :CLEAR UNIBUS INDICATOR

: NOW FILL WRITE FROM BUFFER-200 OF 52525 AND 56 OF 377
JSR  RO,28CLAREA      :CLEAR 200 WORDS FROM WRFROM
WRFROM                :STARTING FROM WRFROM
200                   :200 WORDS
52525                 :FILL WITH 52525

JSR  RO,28CLAREA      :CLEAR 56 WORDS FROM WRFROM+(200*2)
WRFROM+(200*2)        :STARTING FROM WRFROM+(200*2)
56                    :56 WORDS
377                   :FILL WITH 377

: NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA AS
: WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
: CAN BE MADE TO DETERMINE THAT WRITE DID NOT CHANGE BUFFER
JSR  RO,28CLAREA      :CLEAR 200 WORDS FROM REINTO
REINTO                :STARTING FROM REINTO
200                   :200 WORDS
52525                 :FILL WITH 52525

JSR  RO,28CLAREA      :CLEAR 56 WORDS FROM REINTO+(200*2)
REINTO+(200*2)        :STARTING FROM REINTO+(200*2)
56                    :56 WORDS
377                   :FILL WITH 377

: NOW WRITE DATA COMMAND WILL BE LOADED
JSR  RO,28RUN         :SETUP TO RUN FOR DATA COMMAND
0                      :CYLINDER 0
0                      :SECTOR 0
0                      :TRACK 0

```

.BYTE 0  
.BYTE 0

```

7009 022736 177470 -200. :WORD COUNT = 200.
7010 022740 002400 WRFROM :BUS ADDRESS
7011 :STARTING ADDRESS OF DATA
7012 :BUFFER = WRFROM
7013 022742 000000 C :DO NOT INHIBIT BUS ADDRESS INCREMENT
7014 022744 010000 FMT22 :16 BITS PER WORD FORMAT
7015 :DO NOT INHIBIT ECC CORRECTION
7016 :DO NOT INHIBIT HEADER COMPARE
7017 022746 002354 WRIDAT :GET READY TO DO A WRIDAT
7018 :WRITE DATA WITH 60 IN RHCS1
7019
7020 022750 052777 000000 157240 BIS #PSEL,DRHCS1 :SET PORT B
7021 :THAT IS UNIBUS B
7022
7023 :NOW SAVE REGISTER FOR COMPARISON AFTER WRITE DATA
7024 JSR RO,DRSAVER :SAVE REGISTERS
7025 RHWC :RHWC IS THE FIRST REGISTER SAVED
7026 SAVERE :STARTING ADDRESS OF WHERE
7027 :THE REGISTERS ARE SAVED
7028 022766 000022 18. :NUMBER OF REGISTERS
7029 :SAVED = 18.
7030
7031 022770 004737 034636 JSR PC,DRCHECKT :CHECK DVA,RDY,MOL,DPR,DRY,VV
7032
7033 022774 013777 004516 157202 MOV DRP4VEC,DRPVEC :SET RPO4 VECTOR ADDRESS
7034 :TO 'TIME1' IF P-CLOCK IS PRESENT
7035 :OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7036 :'TIME' WILL ONLY SAVE
7037 :CURRENT CYLINDER ADDRESS
7038 :AND LOOK AHEAD REGISTERS
7039
7040 ;SET PORT SELECT BIT
7041
7042 023002 013746 002354 MOV DRWRIDAT,-(SP) :GET READY TO MOVE COMMAND
7043 023006 052716 000101 BIS #GO!IE!PSEL,(SP) :GET READY TO SET GO AND
7044 :ENABLE INTERRUPT
7045 023012 012677 157200 MOV (SP)+,DRHCS1 :GO WITH
7046 :60 IN RHCS1 FOR WRITE DATA
7047 :WITH INTERRUPT ENABLED
7048 023016 011100 MOV DR1,RO :SAVE RHCS1 DURING ABOVE OPERATION
7049 023020 011305 MOV DR3,R5 :SAVE RHDS1 DURING ABOVE OPERATION
7050
7051 ;ONE REVOLUTION=16670 MICROSEC, ONE SECTOR=760 MICROSEC
7052
7053 023022 104410 WAT :WAIT FOR ROY BIT TO SET
7054 023024 002216 RHCS1 :WAIT FOR RHCS1 REGISTER
7055 023026 000200 RDY :WAIT FOR ROY BIT IN RHCS1 REGISTER
7056 023030 001614 908. :ALLOW 9080 MICRO SECONDS
7057 023032 001507 939. :ROY MUST SET BETWEEN
7058 :690 AND 17470 MICRO SECONDS
7059 :COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7060 :RO AND R5 IMMEDIATELY AFTER GO
7061 023034 013746 002354 MOV DRWRIDAT,-(SP) :SAVE COMMAND
7062 023040 052716 004101 BIS #IE!GO!DVA!PSEL,(SP) :INCLUDE IE!GO!DVA!PSEL

```



```

7065 023044 011637 001124      MOV      (SP),2#SGDDAT      ;SAVE FOR PRINTOUT
7066 023050 022600              CMP      (SP)+,R0          ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
7067                                ;AND COMMAND SHOULD BE SET
7068 023052 001405              BEQ      64$              ;BRANCH IF GOOD
7069 023054 010037 001126      MOV      RC,2#SBDDAT      ;BAD DATA
7070 023060 010137 004510      MOV      R7,2#REGADR      ;FAILING REGISTER RHCS1
7071 023064 104021              ERROR      =              ;DURING ABOVE OPERATION ONLY
7072                                ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
7073 023066 012746 010500 64$:  MOV      #MOL!DPR!VV -(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7074 023072 011637 001124      MOV      (SP),2#SGDDAT      ;SAVE FOR PRINTOUT
7075 023076 022605              CMP      (SP)+,R5          ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7076                                ;SHOULD BE SET
7077 023100 001405              BEQ      66$              ;BRANCH IF GOOD
7078 023102 010537 001126      MOV      R5,2#SBDDAT      ;BAD DATA
7079 023106 010337 004510      MOV      R3,2#REGADR      ;FAILING REGISTER RHDS1
7080 023112 104063              ERROR      63              ;DURING ABOVE OPERATION ONLY
7081                                ;MOL!DPR!VV SHOULD BE SET
7082 023114              66$:
7083
7084
7085
7086                                ;CHECK IF NEM NON EXISTANT MEMORY IS SET
7087                                ;IF SET IT MEANS UNIBUS B IS NOT CONNECTED
7088                                ;SO THIS TEST IS NOT PERFORMED
7089 023114 032777 004000 157072  BIT      #NEM,2#RHCS2      ;TEST NEM
7090 023122 001442              BEQ      11$              ;BRANCH IF UNIBUS B THERE
7091 023124 012737 177777 004642  MOV      #-1,2#UBUSB      ;UNIBUS B NOT THERE
7092 023132 104400 023140      TYPE      +4              ;TYPE ASCIZ STRING
7093 023136 000426              BR      67$              ;GET OVER THE ASCIZ
7094                                ;:.ASCIZ <15><12>/THE RH DOES NOT HAVE UNI BUS B CONNECTED/
7095 023214              67$:
7096 023214 104400 001221      TYPE      ,SCLF
7097 023220 104400 001221      TYPE      ,SCLF
7098 023224 000137 023750      JMP      2#10$            ;JUMP TO NEXT TEST - NO UNIBUS B
7099 023230              11$:
7100 023230 104400 023236      TYPE      +4              ;TYPE ASCIZ STRING
7101 023234 000424              BR      68$              ;GET OVER THE ASCIZ
7102                                ;:.ASCIZ <15><12>/THE RH DOES HAVE UNI BUS B CONNECTED/
7103 023306              68$:
7104 023306 104400 001221      TYPE      ,SCLF
7105 023312 104400 001221      TYPE      ,SCLF
7106
7107                                ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
7108 023316 004037 034460      JSR      R0,2#FILLRE      ;MOV 0 INTO SAVED RHWC
7109 023322 002210              RHWC              ;SAVED REGISTER TO CHANGE
7110 023324 000000              0                ;DATA
7111 023326 004037 034460      JSR      R0,2#FILLRE      ;MOV WRFROM+<200.*2> INTO SAVED RHBA
7112 023332 002212              RHBA              ;SAVED REGISTER TO CHANGE
7113 023334 003220              WRFROM+<200.*2> ;DATA
7114 023336 004037 034460      JSR      R0,2#FILLRE      ;MOV 1 INTO SAVED RHDST
7115 023342 002222              RHDST            ;SAVED REGISTER TO CHANGE
7116 023344 000001              1                ;DATA
7117
7118                                ;NOW COMPARE REGISTERS BEFORE WRITE DATA WITH REGISTERS
7119                                ;AFTER COMMAND
7120

```

MAINDEC-11-DERPU-B  
DERPU8.P11 T27

MACY11 27(732) 17-SEP-76 13:23 PAGE 147  
WRITE/READ DATA USING UNIBUS B

```

7121 023346 004037 035546 JSR RO,2#COMREG ;COMPARE SAVED REGISTERS WITH
7122 ;PRESENT VALUE
7123 023352 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
7124 023354 002266 WC ;TEST DATA STARTING FROM 'PHWC'
7125 023356 000022 18. ;18. REGISTERS TO BE COMPARED
7126 023360 023364 1$ ;RETURN TO 1$ ON ERROR
7127 023362 023370 2$ ;RETURN TO 2$ ON NO ERROR
7128 ;
7129 023364 104074 1$: ERROR 74 ;WHILE USING UNIBUS B
7130 ;WRITE DATA COMMAND CAUSED
7131 023366 000207 RTS PC ;IMPROPER REGISTER CHANGE
7132 ;GOOD DATA GIVES WHAT SHOULD
7133 ;BE
7134 ;RECEIVED DATA GIVES WHAT WAS
7135 ;THERE AFTER COMMAND
7136 ;
7137 ;NOW WRITE FROM BUFFER WILL BE CHECKED FOR NO CHANGE
7138 023370 2$:
7139 ;
7140 023370 004037 036576 JSR RO,2#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
7141 023374 003444 REINTO ;GOOD DATA STARTS FROM REINTO
7142 023376 002400 WRFROM ;TEST DATA STARTS FROM WRFROM
7143 023400 000400 256. ;256. WORDS TO BE COMPARED
7144 023402 023406 3$ ;RETURN TO 3$ ON ERROR
7145 023404 023412 4$ ;RETURN TO 4$ ON NO ERROR
7146 ;
7147 ;
7148 023406 104075 3$: ERROR 75 ;WHILE USING UNIBUS B
7149 ;WRITE DATA COMMAND CHANGED
7150 023410 000207 RTS PC ;WRITE FROM BUFFER
7151 ;
7152 ;NOW A READ DATA COMMAND WILL BE GIVEN
7153 ;
7154 ;FILL READ INTO BUFFER WITH 200 ZEROS AND 56 OF ALL ONES
7155 023412 4$:
7156 ;
7157 023412 004737 034556 JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
7158 ;R3-RHDS1, R4-RHER1
7159 ;GIVE RH-11 INITIALIZE
7160 ;SETUP UNIT NUMBER
7161 023416 004037 034426 JSR RO,2#CLAREA ;CLEAR 200. WORDS, FROM REINTO
7162 023422 003444 REINTO ;STARTING FROM REINTO
7163 023424 000310 200. ;200. WORDS
7164 023426 000000 0 ;FILL WITH 0
7165 ;
7166 023430 004037 034426 JSR RO,2#CLAREA ;CLEAR 56. WORDS, FROM REINTO+(200.*2)
7167 023434 004264 REINTO+(200.*2) ;STARTING FROM REINTO+(200.*2)
7168 023436 000070 56. ;56. WORDS
7169 023440 000377 377 ;FILL WITH 377
7170 ;
7171 ;
7172 ;FILL WRITE FROM BUFFER WITH 200 OF 52525 AND 56 OF 0
7173 023442 004037 034426 JSR RC,2#CLARE ;CLEAR 200. WORDS, FROM WRFROM
7174 023446 002400 WRFROM ;STARTING FROM WRFROM
7175 023450 000310 200 ;200. WORDS
7176 023452 052525 52525 ;FILL WITH 52525

```

7177							
7178	023454	004037	034426	JSR	RO, @CLAREA	:	CLEAR 56. WORDS, FROM WRFOM+(200.*2)
7179	023460	003220			WRFOM+(200.*2)	:	STARTING FROM WRFOM+(200.*2)
7180	023462	000070			56.	:	56. WORDS
7181	023464	000377			377	:	FILL WITH 377
7182							
7183							
7184						:	NOW FILL COMMAND
7185							
7186	023466	004037	036532	JSR	RO, @RUN	:	SETUP TO RUN FOR DATA COMMAND
7187	023472	000000			0	:	CYLINDER 0
7188	023474	000			0	:	SECTOR 0
7189	023475	000			0	:	TRACK 0
7190	023476	177470			-200.	:	WORD COUNT = 200.
7191	023500	003444			REINTO	:	BUS ADDRESS
7192						:	STARTING ADDRESS OF DATA
7193						:	BUFFER = REINTO
7194	023502	000000			0	:	DO NOT INHIBIT BUS ADDRESS INCREMENT
7195	023504	014000			ECI!FMT22	:	16 BITS PER WORD FORMAT
7196						:	INHIBIT ECC CORRECTION
7197						:	DO NOT INHIBIT HEADER COMPARE
7198	023506	002360			READAT	:	GET READY TO DO A READAT
7199						:	READ DATA WITH 70 IN RHCS1
7200							
7201							
7202							
7203	023510	052777	000000	156500	BIS	@PSEL, @RHCS1	: SET PORT B
7204							: THAT IS UNIBUS B
7205							
7206						:	NOW SAVE REGISTERS FOR COMPARISON AFTER READ DATA COMMAND
7207	023516	004037	034716	JSR	RO, @SAVER	:	SAVE REGISTERS
7208	023522	002210			RHWC	:	RHWC IS THE FIRST REGISTER SAVED
7209	023524	004522			SAVERE	:	STARTING ADDRESS OF WHERE
7210						:	THE REGISTERS ARE SAVED
7211	023526	000022			18.	:	NUMBER OF REGISTERS
7212						:	SAVED = 18.
7213							
7214	023530	004737	034636	JSR	PC, @CHECKT	:	CHECK DVA, RDY, MOL, DPR, DRY, VV
7215							
7216							
7217	023534	013777	004516	156442	MOV	@RP4VEC, @RPVEC	: SET RPO4 VECTOR ADDRESS
7218							: TO 'TIME1' IF P-CLOCK IS PRESENT
7219							: OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7220							: 'TIME' WILL ONLY SAVE
7221							: CURRENT CYLINDER ADDRESS
7222							: AND LOOK AHEAD REGISTERS
7223							
7224						:	SET PORT SELECT
7225							
7226	023542	013746	002360	MOV	@READAT, -(SP)	:	GET READY TO MOVE COMMAND
7227	023546	052716	000101	BIS	@GO!IE!PSEL, (SP)	:	GET READY TO SET GO AND
7228						:	ENABLE INTERRUPT
7229	023552	012677	156440	MOV	(SP)+, @RHCS1	:	GO WITH
7230						:	70 IN RHCS1 FOR READ DATA
7231						:	WITH INTERRUPT ENABLED
7232	023556	011100		MOV	JRI, RO	:	SAVE RHCS1 DURING ABOVE OPERATION

```

7233 023560 011305          MOV      R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATICN
7234
7235
7236 023562 104410          WAT                      ;WAIT FOR RDY BIT TO SET
7237 023564 002216          RHCS1                   ;WAIT FOR RHCS1 REGISTER
7238 023566 000200          RDY                     ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7239 023570 001614          908.                   ;ALLOW 9080 MICRO SECONDS
7240 023572 001507          839.                   ;RDY MUST SET BETWEEN
7241                                     ;690 AND 17470 MICRO SECONDS
7242                                     ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7243                                     ;R0 AND R5 IMMEDIATELY AFTER GO
7244 023574 013746 002360          MOV      R0,R#READAT,-(SP) ;SAVE COMMAND
7245 023600 052716 004101          BIS      #IE!GO!DVA!PSEL,(SP) ;INCLUDE IE!GO!DVA!PSEL
7246 023604 011637 001124          MOV      (SP),R#SGDDAT ;SAVE FOR PRINTOUT
7247 023610 022600          CMP      (SP)+,R0       ;DURING ABOVE OPERATION ONLY IE!GO!DVA!PSEL
7248                                     ;AND COMMAND SHOULD BE SET
7249 023612 001405          BEQ      69$            ;BRANCH IF GOOD
7250 023614 010037 001126          MOV      R0,R#SBDAT     ;BAD DATA
7251 023620 010137 004510          MOV      R1,R#REGADR    ;FAILING REGISTER RHCS1
7252 023624 104021          ERROR    21            ;DURING ABOVE OPERATION ONLY
7253                                     ;COMMAND AND IE!GO!DVA!PSEL SHOULD BE SET
7254 023626 012746 010500          69$: MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
7255 023632 011637 001124          MOV      (SP),R#SGDDAT ;SAVE FOR PRINTOUT
7256 023636 022605          CMP      (SP)+,R5       ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7257                                     ;SHOULD BE SET
7258 023640 001405          BEQ      71$            ;BRANCH IF GOOD
7259 023642 010537 001126          MOV      R5,R#SBDAT     ;BAD DATA
7260 023646 010337 004510          MOV      R3,R#REGADR    ;FAILING REGISTER RHDS1
7261 023652 104063          ERROR    63            ;DURING ABOVE OPERATION ONLY
7262                                     ;MOL!DPR!VV SHOULD BE SET
7263 023654          71$:
7264
7265                                     ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
7266 023654 004037 034460          JSR      R0,R#FILLRE    ;MOV 0 INTO SAVED RHWC
7267 023660 002210          RHWC                   ;SAVED REGISTER TO CHANGE
7268 023662 000000          0                       ;DATA
7269 023664 004037 034460          JSR      R0,R#FILLRE    ;MOV REINTO+(200.*2) INTO SAVED RHBA
7270 023670 002212          RHBA                   ;SAVED REGISTER TO CHANGE
7271 023672 004264          REINTO+(200.*2)        ;DATA
7272 023674 004037 034460          JSR      R0,R#FILLRE    ;MOV 1 INTO SAVED RHDST
7273 023700 002222          RHDST                   ;SAVED REGISTER TO CHANGE
7274 023702 000001          1                       ;DATA
7275
7276                                     ;COMPARE REGISTERS BEFORE READ DATA COMMAND
7277                                     ;WITH REGISTERS AFTER COMMAND
7278
7279 023704 004037 035546          JSR      R0,R#COMREG    ;COMPARE SAVED REGISTERS WITH
7280                                     ;PRESENT VALUE
7281 023710 004522          SAVERE                   ;GOOD DATA SAVED IN 'SAVERE'
7282 023712 002266          WC                       ;TEST DATA STARTING FROM 'RHWC'
7283 023714 000022          18.                     ;18. REGISTERS TO BE COMPARED
7284 023716 023722          5$                       ;RETURN TO 5$ ON ERROR
7285 023720 023726          6$                       ;RETURN TO 6$ ON NO ERROR
7286
7287 023722 104072          5$: ERROR 72           ;WHILE USING UNIBUS B
7288                                     ;READ DATA CAUSED IMPROPER

```

```

7289 023724 000207          RTS      PC          ;REGISTER CHANGE
7290                                ;GOOD DATA GIVES WHAT SHOULD BE THE
7291                                ;RECEIVED DATA GIVES WHAT WAS THERE
7292                                ;AFTER COMMAND
7293
7294                                ;NOW READ INTO BUFFER IS CHECKED FOR GOOD READ
7295 023726          6$:
7296
7297 023726 004037 036576      JSR      RD,2#COMPAR    ;COMPARE TWO BLOCKS OF MEMORY
7298 023732 002400          WRFROM    ;GOOD DATA      STARTS FROM WRFROM
7299 023734 003444          REINTO   ;TEST DATA STARTS FROM REINTO
7300 023736 000400          256.      ;256. WORDS TO BE COMPARED
7301 023740 023744          7$       ;RETURN TO 7$ ON ERROR
7302 023742 023750          10$      ;RETURN TO 10$ ON NO ERROR
7303
7304
7305 023744 104073          7$:      ERROR 73      ;WHILE USING UNIBUS B
7306                                ;INCORRECT DATA AFTER
7307 023746 000207          RTS      PC          ;WRITE DATA FOLLOWED BY A
7308                                ;READ DATA
7309 023750          10$:
7310
7311
7312
7313
7314
7315
7316
7317
7318
7319
7320
7321
7322
7323
7324
7325
7326
7327
7328
7329
7330
7331
7332
7333
7334
7335
7336
7337
7338
7339
7340
7341
7342
7343
7344

```

\*\*\*\*\*  
\*TEST 30 IMPLIED SEARCH

```

:* ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEARCH
:* A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN ONE SECTOR
:* OF WORDS
:* WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD
:* TRACK 0, SECTOR 0. KEYS=0, NUMBER OF WORDS
:* 266 (4 HEADER 256 DATA=0 4 HEADER 2 DATA=1
:* THEN READ HEADER AND DATA FOR ABOVE SECTOR 1 ONLY
:* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH
:* 10000 0,0,0 AND 256 OF 0, 10000, 1,0,0, AND 2 OF 1
:* THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT
:* THE GO BIT, AND ALL THE REGISTERS ARE SAVED
:* THEN GO IS GIVEN FOR WRITE HEADER AND DATA

```

```

:* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGE
:* THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED

```

```

:* NOW FOR THE READ COMMAND READ INTO BUFFER IS FILLED
:* WITH 377, WRITE FROM BUFFER IS FILLED WITH 10000,1,0,0,1,1
:* AND 254 OF ZEROS COMMAND IS LOADED INTO REGISTERS EXCEPT
:* GO AND IE THEN ALL REGISTERS ARE SAVED
:* GO IS GIVEN FOR THE READ COMMAND, 256 WORDS

```

```

:* ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
:* THEN THE READ DATA IS COMPARED

```

2

```

7345
7346
7347 023750 000004
7348 023752 012706 001000
7349 023756 012737 000030 004514
7350
7351 023764 004737 034556
7352
7353
7354
7355
7356
7357
7358 023770 004037 034402
7359 023774 002400
7360 023776 000004
7361 024000 010000
7362 024002 000000
7363 024004 000000
7364 024006 000000
7365
7366
7367 024010 004037 034426
7368 024014 002410
7369 024016 000400
7370 024020 000000
7371
7372
7373
7374
7375 024022 004037 034402
7376 024026 003410
7377 024030 000004
7378 024032 010000
7379 024034 000001
7380 024036 000000
7381 024040 000000
7382
7383
7384 024042 004037 034426
7385 024046 003420
7386 024050 000002
7387 024052 000001
7388
7389
7390
7391
7392
7393
7394
7395 024054 004037 034402
7396 024060 003444
7397 024062 000004
7398 024064 010000
7399 024066 000000
7400 024070 000000

```

```

:*****
TST30: SCOPE
MOV #STACK,SP ;RESET STACK
MOV #30,#TSTNM ;SAVE TEST NUMBER
JSR PC,#CLDISK ;SET R1-RHCS1, R2-RHCS2
;R3-RHDS1, R4-RHER1
;GIVE RH-11 INITIALIZE
;SETUP UNIT NUBER
;FILL WRITE FROM BUFFER WITH HEADER
JSR RO,#FLHEAD ;SAVE HEADER DATA IN WRFROM
WRFROM ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
;FILL WRITE FROM BUFFER WITH DATA
JSR RO,#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10
WRFROM+10 ;STARTING FROM WRFROM+10
256. ;256. WORDS
0 ;FILL WITH 0
;FILL WRITE FROM BUFFER WITH NEXT SECTOR HEADER
JSR RO,#FLHEAD ;SAVE HEADER DATA IN WRFROM+(260.*2)
WRFROM+(260.*2) ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
1 ;SECOND DATA WORD
0 ;THIRD DATA WORD
0 ;FOURTH DATA WORD
;FILL WRITE FROM BUFFER WITH NEXT SECTOR DATA
JSR RO,#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+(264.*2)
WRFROM+(264.*2) ;STARTING FROM WRFROM+(264.*2)
2 ;2 WORDS
1 ;FILL WITH 1
;NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA
;AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS
;CAN BE MADE TO MAKE SURE THAT WRITE DID NOT
;CHANGE WRITE FROM BUFFER.
JSR RO,#FLHEAD ;SAVE HEADER DATA IN REINTO
REINTO ;LOCATION WHERE SAVED
4 ;NUMBER OF WORDS SAVED
10000 ;FIRST DATA WORD
0 ;SECOND DATA WORD
0 ;THIRD DATA WORD

```

```

7401 024072 000000 0 ;FOURTH DATA WORD
7402 024074 004037 034426 JSR RO,#CLAREA ;CLEAR 256. WORDS, FROM REINTO+10
7403 024100 003454 REINTO+10 ;STARTING FROM REINTO+10
7404 024102 000400 256. ;256. WORDS
7405 024104 000000 0 ;FILL WITH 0
7406
7407
7408 024106 004037 034402 JSR RO,#FLHEAD ;SAVE HEADER DATA IN REINTO+(260.*2)
7409 024112 004454 REINTO+(260.*2) ;LOCATION WHERE SAVED
7410 024114 000004 4 ;NUMBER OF WORDS SAVED
7411 024116 010000 10000 ;FIRST DATA WORD
7412 024120 000001 1 ;SECOND DATA WORD
7413 024122 000000 0 ;THIRD DATA WORD
7414 024124 000000 0 ;FOURTH DATA WORD
7415 024126 004037 034426 JSR RO,#CLAREA ;CLEAR 2 WORDS, FROM REINTO+(264.*2)
7416 024132 004464 REINTO+(264.*2) ;STARTING FROM REINTO+(264.*2)
7417 024134 000002 2 ;2 WORDS
7418 024136 000001 1 ;FILL WITH 1
7419
7420 ;NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7421
7422 024140 004037 036532 JSR RO,#RUN ;SETUP TO RUN FOR DATA COMMAND
7423 024144 000000 0 ;CYLINDER 0
7424 024146 000 .BYTE 0 ;SECTOR 0
7425 024147 000 .BYTE 0 ;TRACK 0
7426 024150 177366 -262.-4 ;WORD COUNT (DATA)=262.+
7427 ;4 HEADER WORDS
7428 024152 002400 WRFROM ;BUS ADDRESS
7429 ;STARTING ADDRESS OF DATA
7430 ;BUFFER = WRFROM
7431 024154 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7432 024156 010000 FMT22 ;16 BITS PER WORD FORMAT
7433 ;DO NOT INHIBIT ECC CORRECTION
7434 ;DO NOT INHIBIT HEADER COMPARE
7435 024160 002356 WRIFOR ;GET READY TO DO A WRIFOR
7436 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7437
7438
7439
7440 ;NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
7441 024162 004037 034716 JSR RO,#SAVER ;SAVE REGISTERS
7442 024166 002210 RHWC ;RHWC IS THE FIRST REGISTER SAVED
7443 024170 004522 SAVERE ;STARTING ADDRESS OF WHERE
7444 ;THE REGISTERS ARE SAVED
7445 024172 000022 18. ;NUMBER OF REGISTERS
7446 ;SAVED = 18.
7447
7448 024174 004737 034636 JSR PC,#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
7449
7450
7451 024200 013777 004516 155776 MOV #RP4VEC,#RPVEC ;SET RP04 VECTOR ADDRESS
7452 ;TO 'TIME1' IF P-CLOCK IS PRESENT
7453 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7454 ;'TIME' WILL ONLY SAVE
7455 ;CURRENT CYLINDER ADDRESS
7456 ;AND LOOK AHEAD REGISTERS

```

```

7457
7458
7459 024206 013746 002356      MOV    @#WRIFOR, -(SP) ; GET READY TO MOVE COMMAND
7460 024212 052716 000101      BIS    #GO!IE, (SP)   ; GET READY TO SET GO AND
7461                                ; ENABLE INTERRUPT
7462 024216 012677 155774      MOV    (SP)+, @RHCS1 ; GO WITH
7463                                ; 62 IN RHCS1 FOR WRITE HEADER AND DATA
7464                                ; WITH INTERRUPT ENABLED
7465 024222 011100      MOV    @R1, R0        ; SAVE RHCS1 DURING ABOVE OPERATION
7466 024224 011305      MOV    @R3, R5        ; SAVE RHDS1 DURING ABOVE OPERATION
7467                                ; ONE REVOLUTION = 16670 MICRO SEC. ONE SECTOR = 760 MICRO SEC
7468
7469 024226 104410      WAT                                ; WAIT FOR RDY BIT TO SET
7470 024230 002216      RHCS1                            ; WAIT FOR RHCS1 REGISTER
7471 024232 000200      RDY                                ; WAIT FOR RDY BIT IN RHCS1 REGISTER
7472 024234 002001      1025.                            ; ALLOW 10250 MICRO SECONDS
7473 024236 001553      875.                              ; RDY MUST SET BETWEEN
7474                                ; 1500 AND 19000 MICRO SECONDS
7475                                ; COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7476                                ; R0 AND R5 IMMEDIATELY AFTER GO
7477 024240 013746 002356      MOV    @#WRIFOR, -(SP) ; SAVE COMMAND
7478 024244 052716 004101      BIS    #IE!GO!DVA, (SP) ; INCLUDE IE!GO!DVA
7479 024250 011637 001124      MOV    (SP), @#$GDDAT ; SAVE FOR PRINTOUT
7480 024254 022600      CMP    (SP)+, R0       ; DURING ABOVE OPERATION ONLY IE!GO!DVA
7481                                ; AND COMMAND SHOULD BE SET
7482 024256 001405      BEQ    64$             ; BRANCH IF GOOD
7483 024260 010037 001126      MOV    R0, @#$BDDAT   ; BAD DATA
7484 024264 010137 004510      MOV    R1, @#REGADR   ; FAILING REGISTER RHCS1
7485 024270 104021      ERROR  21            ; DURING ABOVE OPERATION ONLY
7486                                ; COMMAND AND IE!GO!DVA SHOULD BE SET
7487 024272 012746 010500      64$: MOV    #MOL!DPR!VV, -(SP) ; SAVE BITS SET DURING OPERATION IN RHDS1
7488 024276 011637 001124      MOV    (SP), @#$GDDAT ; SAVE FOR PRINTOUT
7489 024302 022605      CMP    (SP)+, R5     ; DURING ABOVE OPERATION ONLY MOL!DPR!VV
7490                                ; SHOULD BE SET
7491 024304 001405      BEQ    66$             ; BRANCH IF GOOD
7492 024306 010537 001126      MOV    R5, @#$BDDAT   ; BAD DATA
7493 024312 010337 004510      MOV    R3, @#REGADR   ; FAILING REGISTER RHDS1
7494 024316 104063      ERROR  63            ; DURING ABOVE OPERATION ONLY
7495                                ; MOL!DPR!VV SHOULD BE SET
7496 024320      66$:
7497
7498                                ; NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7499 024320 004037 034460      JSR    R0, @#FILLRE   ; MOV 0 INTO SAVED RHWC
7500                                ; SAVED REGISTER TO CHANGE
7501 024326 000000      0 ; DATA
7502 024330 004037 034460      JSR    R0, @#FILLRE   ; MOV WRFROM+(266.*2) INTO SAVED RHBA
7503                                ; SAVED REGISTER TO CHANGE
7504 024336 003424      WRFROM+(266.*2) ; DATA
7505 024340 004037 034460      JSR    R0, @#FILLRE   ; MOV 2 INTO SAVED RHDST
7506 024344 002222      RHDST ; SAVED REGISTER TO CHANGE
7507 024346 000002      2 ; DATA
7508                                ; NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
7509                                ; WITH REGISTERS AFTER COMMAND
7510
7511 024350 004037 035546      JSR    R0, @#COMREG   ; COMPARE SAVED REGISTERS WITH
7512                                ; PRESENT VALUE

```



```

7513 024354 004522          SAVERE          ;GOOD DATA SAVED IN 'SAVERE'
7514 024356 002266          WC              ;TEST DATA STARTING FROM 'RHWC'
7515 024360 000022          18             ;18. REGISTERS TO BE COMPARED
7516 024362 024366          1$            ;RETURN TO 1$ ON ERROR
7517 024264 024372          2$            ;RETURN TO 2$ ON NO ERROR
7519 024366 104027          1$: ERROR 27    ;WRITE HEADER AND DATA
7520 024370 000207          RTS PC        ;CAUSED IMPROPER REGISTER
7521                                ;CHANGE
7522                                ;GOOD DATA GIVES WHAT SHOULD
7523                                ;BE THERE
7524                                ;RECEIVED DATA GIVES WHAT
7525                                ;WAS THERE AFTER COMMANT
7526
7527                                ;NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
7528                                ;NOTHING GOT CHANGED
7529 024372          2$:
7530
7531 024372 004037 036576    JSR    RO,2#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
7532 024376 003444          REINTO        ;GOOD DATA      STARTS FROM REINTO
7533 024400 002400          WRFROM        ;TEST DATA STARTS FROM WRFROM
7534 024402 000412          266.         ;266. WORDS TO BE COMPARED
7535 024404 024410          3$           ;RETURN TO 3$ ON ERROR
7536 024406 024414          4$           ;RETURN TO 4$ ON NO ERROR
7537
7538
7539 024410 104030          3$: ERROR 30    ;WRITE HEADER AND DATA
7540 024412 000207          RTS PC        ;CHANGED WRITE FROM BUFFER
7541
7542                                ;NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
7543                                ;FOR SECTOR 1, 256 WORDS
7544
7545                                ;READ INTO BUFFER IS FILLED WITH ONES
7546 024414          4$:
7547
7548 024414 004737 034556    JSR    PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
7549                                ;R3-RHDS1, R4-RHER1
7550                                ;GIVE RH-11 INITIALIZE
7551                                ;SETUP UNIT NUBER
7552 024420 004037 034426    JSR    RO,2#CLAREA ;CLEAR 260. WORDS, FROM REINTO
7553 024424 003444          REINTO        ;STARTING FROM REINTO
7554 024426 000404          260.         ;260. WORDS
7555 024430 000377          377         ;FILL WITH 377
7556
7557                                ;WRITE FROM BUFFER IS FILLED WITH EXPECTED DATA
7558
7559 024432 004037 034402    JSR    RO,2#FLHEAD ;SAVE HEADER DATA IN WRFROM
7560 024436 002400          WRFROM        ;LOCATION WHERE SAVED
7561 024440 000006          6             ;NUMBER OF WORDS SAVED
7562 024442 010000          10000        ;FIRST DATA WORD
7563 024444 000001          1             ;SECOND DATA WORD
7564 024446 000000          0             ;THIRD DATA WORD
7565 024450 000000          0             ;FOURTH DATA WORD
7566 024452 000001          1             ;FIFTH DATA WORD
7567 024454 000001          1             ;SIXTH DATA WORD
7568 024456 004037 034426    JSR    RO,2#CLAREA ;CLEAR 254 WORDS, FROM WRFROM+(6.*2)

```

```

7569 024462 002414      WRFROM+(6.*2)      ;STARTING FROM WRFROM+(6.*2)
7570 024464 000254      254                ;254 WORDS
7571 024466 000000      0                  ;FILL WITH 0
7572
7573                    ;NOW FILL COMMAND
7574
7575 024470 004037 036532 JSR      RD,@#RUN    ;SETUP TO RUN FOR DATA COMMAND
7576 024474 000000      0                  ;CYLINDER 0
7577 024476      001      .BYTE      -              ;SECTOR 1
7578 024477      000      .BYTE      -              ;TRACK 0
7579 024500 177374      -256.-4           ;WORD COUNT (DATA)=256.+
7580
7581 024502 003444      REINTO            ;4 HEADER WORDS
7582
7583                    ;BUS ADDRESS
7584 024504 000000      0                  ;STARTING ADDRESS OF DATA
7585 024506 014000      ECI!FMT22        ;BUFFER = REINTO
7586
7587                    ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7588 024510 002362      REFOR            ;16 BITS PER WORD FORMAT
7589
7590                    ;INHIBIT ECC CORRECTION
7591
7592                    ;DO NOT INHIBIT HEADER COMPARE
7593
7594                    ;GET READY TO DO A REFOR
7595
7596                    ;READ HEADER AND DATA WITH 72 IN RHCS1
7597
7598
7599
7600
7601
7602
7603 024512 004037 034716 ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
7604 024516 002210      JSR      RD,@#SAVER ;SAVE REGISTERS
7605 024520 004522      RHWG          ;RHWG IS THE FIRST REGISTER SAVED
7606
7607                    ;STARTING ADDRESS OF WHERE
7608
7609                    ;THE REGISTERS ARE SAVED
7610
7611                    ;NUMBER OF REGISTERS
7612
7613                    ;SAVED = 18.
7614
7615
7616
7617
7618
7619
7620
7621 024524 004737 034636 JSR      PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631 024530 013777 004516 155446 MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
7632
7633                    ;TO 'TIME1' IF P-CLOCK IS PRESENT
7634
7635                    ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7636
7637                    ;'TIME' WILL ONLY SAVE
7638
7639                    ;CURRENT CYLINDER ADDRESS
7640
7641                    ;AND LOOK AHEAD REGISTERS
7642
7643
7644
7645
7646
7647
7648
7649
7650
7651 024536 013746 002362 MOV      @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
7652 024542 052716 000101 BIS      #GO!IE,(SP) ;GET READY TO SET GO AND
7653
7654                    ;ENABLE INTERRUPT
7655
7656
7657
7658
7659
7660
7661
7662
7663
7664 024546 012677 155444 MOV      (SP)+,@RHCS1 ;GO WITH
7665
7666                    ;72 IN RHCS1 FOR READ DATA
7667
7668                    ;WITH INTERRUPT ENABLED
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701
7702
7703
7704
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7730
7731
7732
7733
7734
7735
7736
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7890
7891
7892
7893
7894
7895
7896
7897
7898
7899
7900
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

```



77200  
77201  
77202  
77203  
77204  
77205  
77206  
77207  
77208  
77209  
77210  
77211  
77212  
77213  
77214  
77215  
77216  
77217  
77218  
77219  
77220  
77221  
77222  
77223  
77224  
77225  
77226  
77227  
77228  
77229  
77230  
77231  
77232  
77233  
77234  
77235  
77236

024722  
024723  
024724  
024725  
024726  
024727  
024728  
024729  
024730  
024731  
024732  
024733  
024734  
024735  
024736  
024737  
024738  
024739  
024740  
024741  
024742  
024743  
024744

026576

:NOW READ INTO BUFFER WILL BE CHECKED TO SEE  
:THAT READ WAS GOOD

65:

JSR RD,2#COMPAR  
WAFROM  
REINTO  
R50.  
ICS

:COMPARE TWO BLOCKS OF MEMORY  
:GOOD DATA STARTS FROM WAFROM  
:TEST DATA STARTS FROM REINTO  
:260. WORDS TO BE COMPARED  
:RETURN TO 75 ON ERROR  
:RETURN TO ICS ON NO ERROR

75:

ERROR 32  
RTS PC

:WRITE HEADER AND DATA  
:FOLLOWED BY A READ HEADER  
:AND DATA GAVE A READ ERROR  
:ERROR MAY BE IN READ OR WRITE

ICS:

\*\*\*\*\*  
:TEST 31 IMPLIED SEEK

:\* ONLY NEW ADDITION IN THIS TEST IS AN IMPLIED SEEK FROM CYLINDER 0  
:\* TO CYLINDER 1. A WRITE HEADER AND DATA IS GIVEN FOR MORE THAN  
:\* ONE SECTOR OF WORDS

:\* WRITE HEADER AND DATA CYLINDER 0, FORMAT 16 BITS PER WORD  
:\* TRACK 18, SECTOR 21, KEYS=0, NUMBER OF WORDS 266 WORDS  
:\*(4 HEADER, 256 DATA=1125, 4 HEADER 2 DATA = 2000  
:\* THEN READ HEADER AND DATA FOR ABOVE, TRACK 0, SECTOR 0, CYL=1  
:\* WRITE FROM BUFFER AND READ INTO BUFFER ARE FILLED WITH  
:\* 10000 0 0 0 AND 256 OF 1125, 10001 0 0 0 AND 2 OF 2000  
:\* THE WRITE COMMAND IS THEN LOADED INTO THE REGISTERS EXCEPT  
:\* THE GC BIT, AND ALL THE REGISTERS ARE SAVED  
:\* THEN GO IS GIVEN FOR WRITE HEADER AND DATA

:\* THEN ALL REGISTERS ARE COMPARED TO CHECK FOR IMPROPER CHANGE  
:\* THEN WRITE FROM BUFFER IS CHECKED TO SEE THAT NOTHING CHANGED

:\* NOW FOR READ COMMAND READ INTO BUFFER IS FILLED  
:\* WITH ALL ONES, WRITE FROM BUFFER IS LOADED WITH  
:\* 10001 0 0 0 AND 2 OF 2000, 254 OF ZEROS COMMAND IS LOADED  
:\* INTO REGISTERS EXCEPT GO AND IE ALL REGISTERS ARE SAVED  
:\* GO IS GIVEN FOR THE READ COMMAND

:\* ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE  
:\* THEN THE READ DATA IS COMPARED.

\*\*\*\*\*

†ST31: SCOPE  
MOV #STACK,SP :RESET STACK  
MOV #31,†STNM :SAVE TEST NUMBER  
JSR PC,2#CLDISK :SET R1-RHCS1, R2-RHCS2

024744 000004  
024746 012706 001000  
024752 012737 000031 004514  
024760 004737 034556

;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUBER

:FILL WRITE FROM BUFFER WITH HEADER

7743 024764 004037 034402  
7744 024770 002400  
7745 024772 000004  
7746 024774 010000  
7747 024776 011025  
7748 025000 000000  
7749 025002 000000

JSR RO,3#FLHEAD ;SAVE HEADER DATA IN WRFROM  
WRFROM ;LOCATION WHERE SAVED  
4 ;NUMBER OF WORDS SAVED  
10000 ;FIRST DATA WORD  
<18.\*40>!<21.> ;SECOND DATA WORD  
0 ;THIRD DATA WORD  
0 ;FOURTH DATA WORD

:FILL WRITE FROM BUFFER WITH DATA  
JSR RO,3#CLAREA ;CLEAR 256. WORDS, FROM WRFROM+10  
WRFROM+10 ;STARTING FROM WRFROM+10  
256. ;256. WORDS  
<18.\*40>!21. ;FILL WITH <18.\*40>!21.

7751 025004 004037 034426  
7752 025010 002410  
7753 025012 000400  
7754 025014 001125

:FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER

7759 025016 004037 034402  
7760 025022 003410  
7761 025024 000004  
7762 025026 010001  
7763 025030 000000  
7764 025032 000000  
7765 025034 000000

JSR RO,3#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.\*2>  
WRFROM+<260.\*2> ;LOCATION WHERE SAVED  
4 ;NUMBER OF WORDS SAVED  
10001 ;FIRST DATA WORD  
0 ;SECOND DATA WORD  
0 ;THIRD DATA WORD  
0 ;FOURTH DATA WORD

:FILL WRITE FROM BUFFER WITH NEXT TRACK DATA  
JSR RO,3#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<264.\*2>  
WRFROM+<264.\*2> ;STARTING FROM WRFROM+<264.\*2>  
2 ;2 WORDS  
2000 ;FILL WITH 2000

7769 025036 004037 034426  
7770 025042 003420  
7771 025044 000002  
7772 025046 002000

:NOW READ INTO BUFFER WILL BE FILLED WITH SAME DATA  
:AS WRITE FROM BUFFER SO THAT AFTER A WRITE COMPARISONS  
:CAN BE MADE TO MAKE SURE THAT WRITE DID NOT  
:CHANGE WRITE FROM BUFFER.

7773  
7774  
7775  
7776  
7777  
7778  
7779  
7780 025050 004037 034402  
7781 025054 003444  
7782 025056 000004  
7783 025060 010000  
7784 025062 011025  
7785 025064 000000  
7786 025066 000000  
7787 025070 004037 034426  
7788 025074 003454  
7789 025076 000400  
7790 025100 001125

JSR RO,3#FLHEAD ;SAVE HEADER DATA IN REINTO  
REINTO ;LOCATION WHERE SAVED  
4 ;NUMBER OF WORDS SAVED  
10000 ;FIRST DATA WORD  
<18.\*40>!<21.> ;SECOND DATA WORD  
0 ;THIRD DATA WORD  
0 ;FOURTH DATA WORD  
JSR RO,3#CLAREA ;CLEAR 256. WORDS, FROM REINTO+10  
REINTO+10 ;STARTING FROM REINTO+10  
256. ;256. WORDS  
<18.\*40>!21. ;FILL WITH <18.\*40>!21.

7791  
7792

```

7793 025102 004037 034402 JSR RO, @#FLHEAD ;SAVE HEADER DATA IN REINTO+(<260.*2)
7794 025106 004454 REINTO+(<260.*2) ;LOCATION WHERE SAVED
7795 025110 000004 4 ;NUMBER OF WORDS SAVED
7796 025112 010001 10001 ;FIRST DATA WORD
7797 025114 000000 0 ;SECOND DATA WORD
7798 025116 000000 0 ;THIRD DATA WORD
7799 025120 000000 0 ;FOURTH DATA WORD
7800 025122 004037 034426 JSR RO, @#CLAREA ;CLEAR 2 WORDS FROM REINTO+(<264.*2)
7801 025126 004464 REINTO+(<264.*2) ;STARTING FROM REINTO+(<264.*2)
7802 025130 000002 2 ;2 WORDS
7803 025132 002000 2000 ;FILL WITH 2000
7804
7805 ;NOW THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
7806
7807 025134 004037 036532 JSR RO, @#RUN ;SETUP TO RUN FOR DATA COMMAND
7808 025140 000000 0 ;CYLINDER 0
7809 025142 025 .BYTE 21. ;SECTOR 21.
7810 025143 022 .BYTE 18. ;TRACK 18.
7811 025144 177266 -262.-4 ;WORD COUNT (DATA)=262.+
7812 ;4 HEADER WORDS
7813 025146 002400 WRFROM ;BUS ADDRESS
7814 ;STARTING ADDRESS OF DATA
7815 ;BUFFER = WRFROM
7816 025150 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7817 025152 010000 FMT22 ;16 BITS PER WORD FORMAT
7818 ;DO NOT INHIBIT ECC CORRECTION
7819 ;DO NOT INHIBIT HEADER COMPARE
7820 025154 002356 WRIFOR ;GET READY TO DO A WRIFOR
7821 ;WRITE HEADER AND DATA WITH 62 IN RHCS1
7822
7823
7824 ;NOW SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
7825 025156 004037 034716 JSR RO, @#SAVER ;SAVE REGISTERS
7826 025162 002210 RHWC ;RHWC IS THE FIRST REGISTER SAVED
7827 025164 004522 SAVERE ;STARTING ADDRESS OF WHERE
7828 ;THE REGISTERS ARE SAVED
7829 025166 000022 18. ;NUMBER OF REGISTERS
7830 ;SAVED = 18.
7831
7832 025170 004737 034636 JSR PC, @#CHECKT ;CHECK DVA, RDY, MOL, DPR, DRY, VV
7833
7834
7835 025174 013777 004516 155002 MOV @#RP4VEC, @#RPVEC ;SET RP04 VECTOR ADDRESS
7836 ;TO 'TIME1' IF P-CLOCK IS PRESENT
7837 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7838 ;'TIME' WILL ONLY SAVE
7839 ;CURRENT CYLINDER ADDRESS
7840 ;AND LOOK AHEAD REGISTERS
7841
7842
7843 025202 013746 002356 MOV @#WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
7844 025206 052716 000101 BIS @#GO!IE, (SP) ;GET READY TO SET GO AND
7845 ;ENABLE INTERRUPT
7846 025212 012677 155000 MOV (SP)+, @#RHCS1 ;GO WITH
7847 ;62 IN RHCS1 FOR WRITE HEADER AND DATA
7848 ;WITH INTERRUPT ENABLED

```

```

7849 025216 011100      MOV      R1,R3      ;SAVE RHCS1 DURING ABOVE OPERATION
7850 025220 011305      MOV      R3,R5      ;SAVE RHDS1 DURING ABOVE OPERATION
7851                                     ;ONE REVOLUTION = 16670 MICRO1 SEC. ONE SECTOR = 760 MICRO1 SEC
7852
7853 025222 104410      WAT                                     ;WAIT FOR RDY BIT TO SET
7854 025224 002216      RHCS1      ;WAIT FOR RHCS1 REGISTER
7855 025226 000200      RDY      ;WAIT FOR RDY BIT IN RHCS1 REGISTER
7856 025230 003237      1695.      ;ALLOW 16950 MICRO SECONDS
7857 025232 001515      645.      ;RDY MUST SET BETWEEN
7858                                     ;8500 AND 25400 MICRO SECONDS
7859                                     ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
7860                                     ;R0 AND R5 IMMEDIATELY AFTER GO
7861 025234 013746 002356      MOV      R0,R5      ;SAVE COMMAND
7862 025240 052716 004101      BIS      IE,GO!DVA,(SP) ;INCLUDE IE!GO!DVA
7863 025244 011637 001124      MOV      (SP),R0      ;SAVE FOR PRINTOUT
7864 025250 022600      CMP      (SP)+,R0      ;DURING ABOVE OPERATION ONLY IE!GO!DVA
7865                                     ;AND COMMAND SHOULD BE SET
7866 025252 001405      BEQ      645      ;BRANCH IF GOOD
7867 025254 010037 001126      MOV      R0,R5      ;BAD DATA
7868 025260 010137 004510      MOV      R1,R0      ;FAILING REGISTER RHCS1
7869 025264 104021      ERROR    21      ;DURING ABOVE OPERATION ONLY
7870                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
7871 025266 012746 010500      645:      MOV      MOL,DPR!VV -(SP) ;SAVE BITS SET DURING OPERATION IN RHCS1
7872 025272 011637 001124      MOV      (SP),R0      ;SAVE FOR PRINTOUT
7873 025276 022605      CMP      (SP)+,R5      ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
7874                                     ;SHOULD BE SET
7875 025300 001405      BEQ      665      ;BRANCH IF GOOD
7876 025302 010537 001126      MOV      R5,R0      ;BAD DATA
7877 025306 010337 004510      MOV      R3,R0      ;FAILING REGISTER RHDS1
7878 025312 104063      ERRCR    63      ;DURING ABOVE OPERATION ONLY
7879                                     ;MOL!DPR!VV SHOULD BE SET
7880 025314      665:
7881
7882                                     ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
7883 025314 004037 034460      JSR      R0,R#FILLRE ;MOV 0 INTO SAVED RHWC
7884 025320 002210      RHWC      ;SAVED REGISTER TO CHANGE
7885 025322 000000      0      ;DATA
7886 025324 004037 034460      JSR      R0,R#FILLRE ;MOV WRFROM+(266.*2) INTO SAVED RHBA
7887 025330 002212      RHBA      ;SAVED REGISTER TO CHANGE
7888 025332 003424      WRFROM+(266.*2) ;DATA
7889 025334 004037 034460      JSR      R0,R#FILLRE ;MOV 1 INTO SAVED RHCA
7890 025340 002230      RHCA      ;SAVED REGISTER TO CHANGE
7891 025342 000001      1      ;DATA
7892 025344 004037 034460      JSR      R0,R#FILLRE ;MOV 1 INTO SAVED RHCC
7893 025350 002252      RHCC      ;SAVED REGISTER TO CHANGE
7894 025352 000001      1      ;DATA
7895 025354 004037 034460      JSR      R0,R#FILLRE ;MOV 1 INTO SAVED RHDST
7896 025360 002222      RHDST      ;SAVED REGISTER TO CHANGE
7897 025362 000001      1      ;DATA
7898
7899                                     ;NOW COMARE REGISTERS BEFORE WRITE HEADER AND DATA
7900                                     ;WITH REGISTERS AFTER COMMAND
7901
7902 025364 004037 035546      JSR      R0,R#COMREG ;COMPARE SAVED REGISTERS WITH
7903                                     ;PRESENT VALUE
7904 025370 004522      SAVERE      ;GOOD DATA SAVED IN 'SAVERE'

```

```

7905 025372 002266          WC          ;TEST DATA STARTING FROM 'RHWC'
7906 025374 000022          18.         ;18. REGISTERS TO BE COMPARED
7907 025376 025402          1$          ;RETURN TO 1$ ON ERROR
7908 025400 025406          2$          ;RETURN TO 2$ ON NO ERRCR
7909
7910 025402 104027          1$:         ;WRITE HEADER AND DATA
7911 025404 000207          RTS         ;CAUSED IMPROPER REGISTER
7912                                     CHANGE
7913                                     ;GOOD DATA GIVES WHAT SHOULD
7914                                     ;BE THERE
7915                                     ;RECEIVED DATA GIVES WHAT
7916                                     ;WAS THERE AFTER COMMANT
7917
7918                                     ;NOW WRITE FROM BUFFER WILL BE CHECKED TO SEE THAT
7919                                     ;NOTHING GOT CHANGED
7920 025406          2$:
7921
7922 025406 004037 036576     JSR      RO,2*COMPAR ;COMPARE TWO BLOCKS OF MEMORY
7923 025412 003444          REINTO      ;GOOD DATA STARTS FROM REINTO
7924 025414 002400          WRFROM      ;TEST DATA STARTS FROM WRFROM
7925 025416 000412          266.         ;266. WORDS TO BE COMPARED
7926 025420 025424          3$          ;RETURN TO 3$ ON ERROR
7927 025422 025430          4$          ;RETURN TO 4$ ON NO ERROR
7928
7929
7930 025424 104030          3$:         ;WRITE HEADER AND DATA
7931 025426 000207          RTS         ;CHANGED WRITE FROM BUFFER
7932
7933                                     ;NOW A READ HEADER AND DATA COMMAND WILL BE GIVEN
7934                                     ;READ INTO BUFFER IS FILLED WITH ONES
7935 025430          4$:
7936
7937 025430 004737 034556     JSR      PC,2*CLDISK ;SET R1-RHCS1, R2-RHCS2
7938                                     ;R3-RHDS1, R4-RHER1
7939                                     ;GIVE RH-11 INITIALIZE
7940                                     ;SETUP UNIT NUBER
7941 025434 004037 034426     JSR      RC,2*CLAREA ;CLEAR 260. WORDS, FROM REINTO
7942 025440 003444          REINTO      ;STARTING FROM REINTO
7943 025442 000404          260.         ;260. WORDS
7944 025444 177777          -1          ;FILL WITH -1
7945
7946                                     ;WRITE FROM BUFFER IS FILLED WITH 10001,0,0,0,2000,2000. AND 254 OF 0
7947
7948 025446 004037 034402     JSR      RO,2*FLHEAD ;SAVE HEADER DATA IN WRFROM
7949 025452 002400          WRFROM      ;LOCATION WHERE SAVED
7950 025454 000006          6          ;NUMBER OF WORDS SAVED
7951 025456 010001          10001       ;FIRST DATA WORD
7952 025460 000000          0          ;SECOND DATA WORD
7953 025462 000000          0          ;THIRD DATA WORD
7954 025464 000000          0          ;FOURTH DATA WORD
7955 025466 002000          2000        ;FIFTH DATA WORD
7956 025470 002000          2000        ;SIXTH DATA WORD
7957 025472 004037 034426     JSR      RO,2*CLAREA ;CLEAR 254. WORDS, FROM WRFROM+(6*2)
7958 025476 002414          WRFROM+(6*2) ;STARTING FROM WRFROM+(6*2)
7959 025500 000376          254.         ;254. WORDS
7960 025502 000000          0          ;FILL WITH 0

```



```

7961
7962
7963                ;NOW FILL COMMAND
7964
7965 025504 004037 036532 JSR      RD, @#RUN      ;SETUP TO RUN FOR DATA COMMAND
7966 025510 000001          1                ;CYLINDER 1
7967 025512          000          0                ;SECTOR 0
7968 025513          000          0                ;TRACK 0
7969 025514 177374          -256.-4      ;WORD COUNT (DATA)=256.+
7970
7971 025516 003444 REINTO          ;4 HEADER WORDS
7972
7973                ;BUS ADDRESS
7974 025520 000000          0                ;STARTING ADDRESS OF DATA
7975 025522 014000          0                ;BUFFER = REINTO
7976                ;DO NOT INHIBIT BUS ADDRESS INCREMENT
7977                ;16 BITS PER WORD FORMAT
7978 025524 002362 REFOR          ;INHIBIT ECC CORRECTION
7979                ;DO NOT INHIBIT HEADER COMPARE
7980                ;GET READY TO DO A REFOR
7981                ;READ HEADER AND DATA WITH 72 IN RHCS1
7982
7983                ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
7984 025526 004037 034716 JSR      RO, @#SAVER  ;SAVE REGISTERS
7985 025532 002210          RHWC          ;RHWC IS THE FIRST REGISTER SAVED
7986 025534 004522          SAVERE        ;STARTING ADDRESS OF WHERE
7987 025536 000022          18.          ;THE REGISTERS ARE SAVED
7988                ;NUMBER OF REGISTERS
7989                ;SAVED = 18.
7990 025540 004737 034636 JSR      PC, @#CHECKT ;CHECK DVA, RDY, MOL, DPR, DRY, VV
7991
7992
7993 025544 013777 004516 154432 MOV     @#RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
7994                ;TO 'TIME1' IF P-CLOCK IS PRESENT
7995                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
7996                ;'TIME' WILL ONLY SAVE
7997                ;CURRENT CYLINDER ADDRESS
7998                ;AND LOOK AHEAD REGISTERS
7999
8000
8001 025552 013746 002362 MOV     @#REFOR, -(SP) ;GET READY TO MOVE COMMAND
8002 025556 052716 000101 BIS     #GO!IE, (SP)  ;GET READY TO SET GO AND
8003                ;ENABLE INTERRUPT
8004 025562 012677 154430 MOV     (SP)+, @RHCS1 ;GO WITH
8005                ;72 IN RHCS1 FOR READ DATA
8006                ;WITH INTERRUPT ENABLED
8007 025566 011100          MOV     @R1, RO    ;SAVE RHCS1 DURING ABOVE OPERATION
8008 025570 011305          MOV     @R3, R5    ;SAVE RHDS1 DURING ABOVE OPERATION
8009
8010
8011 025572 104410          WAT                ;WAIT FOR RDY BIT TO SET
8012 025574 002216          RHCS1          ;WAIT FOR RHCS1 REGISTER
8013 025576 000200          RDY          ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8014 025600 001614          908.          ;ALLOW 9080 MICRO SECONDS
8015 025602 001507          839.          ;RDY MUST SET BETWEEN
8016                ;690 AND 17470 MICRO SECONDS

```

```

8017      :COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8018      :RO AND R5 IMMEDIATELY AFTER GO
8019 025604 013746 002362  MOV      2*REFOR,-(SP) ;SAVE COMMAND
8020 025610 052716 004101  BIS      #IE!GO!DVA,(SP) ;INCLUDE IE!GO!DVA
8021 025614 011637 001124  MOV      (SP),2*$GDDAT ;SAVE FOR PRINTOUT
8022 025620 022600      CMP      (SP)+,RO ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8023      ;AND COMMAND SHOULD BE SET
8024 025622 001405      BEQ      67$ ;BRANCH IF GOOD
8025 025624 010037 001126  MOV      RO,2*$BDDAT ;BAD DATA
8026 025630 010137 004510  MOV      R1,2*$REGADR ;FAILING REGISTER RHCS1
8027 025634 104021      ERROR   -21 ;DURING ABOVE OPERATION ONLY
8028      ;COMMAND AND IE!GO!DVA SHOULD BE SET
8029 025636 012746 010500 67$: MOV      #MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8030 025642 011637 001124  MOV      (SP),2*$GDDAT ;SAVE FOR PRINTOUT
8031 025646 022605      CMP      (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8032      ;SHOULD BE SET
8033 025650 001405      BEQ      69$ ;BRANCH IF GOOD
8034 025652 010537 001126  MOV      R5,2*$BDDAT ;BAD DATA
8035 025656 010337 004510  MOV      R3,2*$REGADR ;FAILING REGISTER RHDS1
8036 025662 104063      ERROR   63 ;DURING ABOVE OPERATION ONLY
8037      ;MOL!DPR!VV SHOULD BE SET
8038 025664      69$:
8039
8040      :CHANGE SAVED REGISTERS TO EXPECTED VALUES
8041 025664 004037 034460  JSR      RO,2*$FILLRE ;MOV 0 INTO SAVED RHWC
8042 025670 002210      RHWC      ;SAVED REGISTER TO CHANGE
8043 025672 000000      0 ;DATA
8044 025674 004037 034460  JSR      RO,2*$FILLRE ;MOV REINTO+(260.*2) INTO SAVED RHBA
8045 025700 002212      RHBA      ;SAVED REGISTER TO CHANGE
8046 025702 004454      REINTO+(260.*2) ;DATA
8047 025704 004037 034460  JSR      RO,2*$FILLRE ;MOV 1 INTO SAVED RHDST
8048 025710 002222      RHDST     ;SAVED REGISTER TO CHANGE
8049 025712 000001      1 ;DATA
8050
8051      :COMPARE REGISTERS BEFORE READ HEADER AND DATA
8052      :WITH REGISTERS AFTER COMMAND
8053
8054 025714 004037 035546  JSR      RO,2*$COMREG ;COMPARE SAVED REGISTERS WITH
8055      ;PRESENT VALUE
8056 025720 004522      SAVERE    ;GOOD DATA SAVED IN 'SAVERE'
8057 025722 002266      WC ;TEST DATA STARTING FROM 'RHWC'
8058 025724 000022      18. ;18. REGISTERS TO BE COMPARED
8059 025726 025732      5$ ;RETURN TO 5$ ON ERROR
8060 025730 025736      6$ ;RETURN TO 6$ ON NO ERROR
8061
8062
8063 025732 104031 5$: ERROR   31 ;READ HEADER AND DATA CAUSED
8064 025734 000207      RTS     PC ;IMPROPER REGISTER CHANGE
8065      ;GOOD DATA GIVES WHAT SHOULD
8066      ;BE THERE
8067      ;RECEIVED DATA GIVES WHAT WAS
8068      ;THERE AFTER COMMAND
8069
8070      ;NOW READ INTO BUFFER WILL BE CHECKED TO SEE
8071      ;THAT READ WAS GOOD
8072 025736      6$:

```

```

8073
8074 025736 004037 036576 JSR RO, @#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
8075 025742 002400 WRFROM ;GOOD DATA STARTS FROM WRFROM
8076 025744 003444 REINTO ;TEST DATA STARTS FROM REINTO
8077 025746 000404 260. ;260. WORDS TO BE COMPARED
8078 025750 025754 7$ ;RETURN TO 7$ ON ERROR
8079 025752 025760 10$ ;RETURN TO 10$ ON NO ERROR

```

```

8083 025754 104032 7$: ERROR 32 ;WRITE HEADER AND DATA
8084 025756 000207 RTS PC ;FOLLOWED BY A READ HEADER
;AND DATA GAVE A READ ERROR
;ERROR MAY BE IN READ OR WRITE
8086
8087 025760 10$:

```

```

8090 *****
8091 ;*TEST 32 SEEK TEST
8092

```

```

8093 ;* THIS TEST GETS THE HEADS OUT TO CYLINDER 10 NOT BY ONE
8094 ;* SEEK BUT BY TEN IMPLIED SEEKS ONE CYLINDER AT A TIME
8095 ;*
8096 ;* THIS TEST STARTS WITH A (ALREADY TESTED) RECALIBRATE
8097 ;* THAT IS CYLINDER ZERO. THEN ON CYLINDER 0 SECTOR
8098 ;* #21 TRACK #18 IT WRITES 266 WORDS THERE BY GETTING
8099 ;* THE HEAD TO CYLINDER 1 THEN IT WRITES 266 WORDS ON
8100 ;* CYLINDER 1 SECTOR #21 TRACK #18 THERE BY GETTING
8101 ;* THE HEADS TO CYLINDER 2
8102 ;* THIS IS REPEATED 10 TIMES GETTING THE
8103 ;* HEADS TO CYLINDER 10
8104 ;* THEN A SEEK COMMAND IS GIVEN TO CYLINDER 0
8105 ;* AND DATA ALREADY WRITTEN IS CHECKED

```

```

8106 *****
8107 ;*TEST32: SCOPE

```

```

8108 025760 000004 MOV #STACK, SP ;RESET STACK
8109 025762 012706 001000 MOV #32, @#TSTNM ;SAVE TEST NUMBER
8110 025766 012737 000032 C04514

```

```

8112 025774 004737 034556 JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
8113 ;R3-RHDS1, R4-RHER1
8114 ;GIVE RH-11 INITIALIZE
8115 ;SETUP UNIT NUMBER
8116 ;THE FOLLOWING MOVES ARE TO INITIALIZE TEST FROM
8117 ;CYLINDER 0
8118 ;THESE LOCATIONS ARE CHANGED DURING TEST TO ENABLE
8119 ;GOING TO NEXT CYLINDER

```

```

8120 026000 012737 010000 026156 MOV #10000, @#ST1+10
8121 026006 012737 001125 026176 MOV #<<18.*40>!21.>, @#ST2+10
8122 026014 012737 010001 026210 MOV #10001, @#ST3+10
8123 026022 012737 002000 026230 MOV #2000, @#ST4+10
8124 026030 012737 000000 026236 MOV #0, @#ST5+4
8125 026036 012737 000001 026440 MOV #1, @#ST6+6
8126 026044 012737 000001 026450 MOV #1, @#ST6+16
8127 026052 012737 010001 026532 MOV #10001, @#ST9+10
8128 026060 012737 002000 026552 MOV #2000, @#ST10+10

```

```

8129 026066 012737 005001 026572      MOV      #1, @#ST11+4
8130                                     ;THIS IS TO GET THE HEADS TO CYLINDER 0
8131 026074 013777 004516 154102      MOV      @#RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS
8132                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
8133                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8134                                     ;'TIME' WILL ONLY SAVE
8135                                     ;CURRENT CYLINDER ADDRESS
8136                                     ;AND LOOK AHEAD REGISTERS
8137
8138 026102 013746 002340      MOV      @#RECALI, -(SP) ;GET READY TO MOVE COMMAND
8139 026106 052716 000101      BIS      #GO!IE, (SP) ;GET READY TO SET GO AND
8140                                     ;ENABLE INTERRUPT
8141 026112 012677 154100      MOV      (SP)+, @RHCS1 ;GO WITH
8142                                     ;6 IN RHCS1 FOR RECALIBRATE
8143                                     ;WITH INTERRUPT ENABLED
8144 026116 011100      MOV      @R1, RO ;SAVE RHCS1 DURING ABOVE OPERATION
8145 026120 011305      MOV      @R3, RS ;SAVE RHDS1 DURING ABOVE OPERATION
8146
8147 026122 104410      WAT                                     ;WAIT FOR DRY BIT TO SET
8148 026124 002240      RHDS1 ;WAIT FOR RHDS1 REGISTER
8149 026126 000200      DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
8150 026130 076377      31999. ;ALLOW 319990 MICRO SECONDS
8151 026132 056701      24001. ;DRY MUST SET BETWEEN
8152                                     ;79980 AND 560000 MICRO SECONDS
8153 026134 012737 000012 001176      MOV      #10., @#STMP1 ;TEN COUNT TO GET TO CYLINDER 10
8154
8155 026142 004737 034556      JSR      PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
8156                                     ;R3-RHDS1, R4-RHER1
8157                                     ;GIVE FH-11 INITIALIZE
8158                                     ;SETUP UNIT NUBER
8159
8160                                     ;FILL WRITE FROM BUFFER WITH HEADER
8161 026146      ST1:
8162
8163 026146 004037 034402      JSR      RO, @#FLHEAD ;SAVE HEADER DATA IN WRFROM
8164 026152 002400      WRFROM ;LOCATION WHERE SAVED
8165 026154 000004      4 ;NUMBER OF WORDS SAVED
8166 026156 010000      10000 ;FIRST DATA WORD
8167 026160 011025      <18.*400>!21. ;SECOND DATA WORD
8168 026162 000000      0 ;THIRD DATA WORD
8169 026164 000000      0 ;FOURTH DATA WORD
8170
8171                                     ;FILL WRITE FROM BUFFER WITH DATA
8172 026166      ST2:
8173 026166 004037 034426      JSR      RO, @#CLAREA ;CLEAR 256. WORDS. FROM WRFROM+!0
8174 026172 002410      WRFROM+10 ;STARTING FROM WRFROM+10
8175 026174 000400      256. ;256. WORDS
8176 026176 001125      <0.*2000>!<18.*40>!21. ;FILL WITH <0.*2000>!<18.*40>!21.
8177
8178
8179                                     ;FILL WRITE FROM BUFFER WITH NEXT TRACK HEADER
8180 026200      ST3:
8181
8182 026200 004037 034402      JSR      RO, @#FLHEAD ;SAVE HEADER DATA IN WRFROM+<260.*2>
8183 026204 003410      WRFROM+<260.*2> ;LOCATION WHERE SAVED
8184 026206 000004      4 ;NUMBER OF WORDS SAVED

```

8195	026210	010001		10001	;FIRST DATA WORD
8196	026212	000000		0	;SECOND DATA WORD
8197	026214	000000		0	;THIRD DATA WORD
8198	026216	000000		0	;FOURTH DATA WORD
8199					
8190					:FILL WRITE FROM BUFFER WITH NEXT TRACK DATA
8191	026220		ST4:		
8192	026220	004037	034426	JSR RO,@#CLAREA	;CLEAR 2 WORDS FROM WRFROM+<264.*2>
8193	026224	003420		WRFROM+<264.*2>	;STARTING FROM WRFROM+<264.*2>
8194	026226	000002		2	;2 WORDS
8195	026230	002000		1.*2000	;FILL WITH 1.*2000
8196					
8197					:THE WRITE HEADER AND DATA COMMAND WILL BE FILLED
8198	026232		ST5:		
8199					
8200	026232	004037	036532	JSR RO,@#RUN	;SETUP TO RUN FOR DATA COMMAND
8201	026236	000000		0	;CYLINDER 0
8202	026240	025	.BYTE	21.	;SECTOR 21.
8203	026241	022	.BYTE	18.	;TRACK 18.
8204	026242	177366		-262.-4	:WORD COUNT (DATA)=262.+
8205					;4 HEADER WORDS
8206	026244	002400		WRFROM	;BUS ADDRESS
8207					;STARTING ADDRESS OF DATA
8208					;BUFFER = WRFROM
8209	026246	000000		0	;DO NOT INHIBIT BUS ADDRESS INCREMENT
8210	026250	010000		FMT22	;16 BITS PER WORD FORMAT
8211					;DO NOT INHIBIT ECC CORRECTION
8212					;DO NOT INHIBIT HEADER COMPARE
8213	026252	002356		WRIFOR	;GET READY TO DO A WRIFOR
8214					:WRITE HEADER AND DATA WITH 62 IN RHCS1
8215					
8216					
8217					:SAVE REGISTERS FOR COMPARISON AFTER WRITE HEADER AND DATA
8218	026254	004037	034716	JSR RO,@#SAVER	;SAVE REGISTERS
8219	026260	002210		RHWC	;RHWC IS THE FIRST REGISTER SAVED
8220	026262	004522		SAVERE	;STARTING ADDRESS OF WHERE
8221					;THE REGISTERS ARE SAVED
8222	026264	000022		18.	;NUMBER OF REGISTERS
8223					;SAVED = 18.
8224					
8225	026266	004737	034636	JSR PC,@#CHECKT	;CHECK DVA,RDY,MOL,DPR,DRY,VV
8226					
8227					
8228	026272	013777	004516 153704	MOV @#RP4VEC,@RPVEC	;SET RP04 VECTOR ADDRESS
8229					;TO 'TIME1' IF P-CLOCK IS PRESENT
8230					;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8231					; 'TIME' WILL ONLY SAVE
8232					;CURRENT CYLINDER ADDRESS
8233					;AND LOOK AHEAD REGISTERS
8234					
8235					
8236	026300	013746	002356	MOV @#WRIFOR,-(SP)	;GET READY TO MOVE COMMAND
8237	026304	052716	000101	BIS #GO!IE,(SP)	;GET READY TO SET GO AND
8238					;ENABLE INTERRUPT
8239	026310	012677	153702	MOV (SP)+,@RHCS1	;GO WITH
8240					;62 IN RHCS1 FOR WRITE HEADER AND DATA

```

8241                                     ;WITH INTERRUPT ENABLED
8242 026314 011100                       MOV   R1,R0                       ;SAVE RHCS1 DURING ABOVE OPERATION
8243 026316 011305                       MOV   R3,R5                       ;SAVE RHDS1 DURING ABOVE OPERATION
8244
8245                                     ;ONE REVOLUTION = 16670 MICRO SECONDS, ONE SECTOR = 760 MICRO SEC.
8246                                     ;MAX TIME ALLOWED = ONE REVOLUTION + SEEK + 2 SECTORS
8247                                     ;MIN TIME ALLOWED = 2 SECTORS + SEEK
8248
8249
8250 026320 104410                       WAT                               ;WAIT FOR RDY BIT TO SET
8251 026322 002216                       RHCS1                             ;WAIT FOR RHCS1 REGISTER
8252 026324 000200                       RDY                               ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8253 026326 003237                       1695.                             ;ALLOW 16950 MICRO SECONDS
8254 026330 001515                       845.                               ;RDY MUST SET BETWEEN
8255                                     ;8500 AND 25400 MICRO SECONDS
8256                                     ;COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8257                                     ;R0 AND R5 IMMEDIATELY AFTER GO
8258 026332 013745 002356                 MOV   R0,R5                       ;SAVE COMMAND
8259 026336 052716 004101                 BIS   R0,R5                       ;INCLUDE IE!GO!DVA
8260 026342 011637 001124                 MOV   R0,R5                       ;SAVE FOR PRINTOUT
8261 026346 022600                       CMP   R0,R5                       ;DURING ABOVE OPERATION ONLY IE!GO!DVA
8262                                     ;AND COMMAND SHOULD BE SET
8263 026350 001405                       BEQ   R0,R5                       ;BRANCH IF GOOD
8264 026352 010037 001126                 MOV   R0,R5                       ;BAD DATA
8265 026356 010137 004510                 MOV   R1,R5                       ;FAILING REGISTER RHCS1
8266 026362 104021                       ERROR 21                          ;DURING ABOVE OPERATION ONLY
8267                                     ;COMMAND AND IE!GO!DVA SHOULD BE SET
9268 026364 012746 010500 64$:          MOV   R0,R5                       ;SAVE BITS SET DURING OPERATION IN RHDS1
8269 026370 011637 001124                 MOV   R0,R5                       ;SAVE FOR PRINTOUT
8270 026374 022605                       CMP   R0,R5                       ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
8271                                     ;SHOULD BE SET
8272 026376 001405                       BEQ   R0,R5                       ;BRANCH IF GOOD
8273 026400 010537 001126                 MOV   R5,R5                       ;BAD DATA
8274 026404 010337 004510                 MOV   R3,R5                       ;FAILING REGISTER RHDS1
8275 026410 104063                       ERROR 63                          ;DURING ABOVE OPERATION ONLY
8276                                     ;MOL!DPR!VV SHOULD BE SET
8277 026412                               66$:
8278
8279                                     ;NOW CHANGES SAVED REGISTERS TO EXPECTED VALUES
8280 026412 004037 034460                 JSR   R0,R5                       ;MOV 0 INTO SAVED RHWC
8281 026416 002210                       RHWC                             ;SAVED REGISTER TO CHANGE
8282 026420 000000                       0                                ;DATA
8283 026422 004037 034460                 JSR   R0,R5                       ;MOV WRFROM+<266.*2> INTO SAVED RHBA
8284 026426 002212                       RHBA                             ;SAVED REGISTER TO CHANGE
8285 026430 003424                       WRFROM+<266.*2>                 ;DATA
8286
8287 026432 004037 034460  ST6:          JSR   R0,R5                       ;MOV 1 INTO SAVED RHCC
8288 026436 002252                       RHCC                             ;SAVED REGISTER TO CHANGE
8289 026440 000001                       1                                ;DATA
8290 026442 004037 034460                 JSR   R0,R5                       ;MOV 1 INTO SAVED RHCA
8291 026446 002230                       RHCA                             ;SAVED REGISTER TO CHANGE
8292 026450 000001                       1                                ;DATA
8293 026452 004037 034460                 JSR   R0,R5                       ;MOV 1 INTO SAVED RHDST
8294 026456 002222                       RHDST                           ;SAVED REGISTER TO CHANGE
8295 026460 000001                       1                                ;DATA
8296

```

```

8297                                     ;COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
8298                                     ;WITH REGISTERS AFTER COMMAND
8299
8300 026462 004037 035546      JSR      RD,2#COMREG      ;COMPARE SAVED REGISTERS WITH
8301                                     ;PRESENT VALUE
8302 026466 004522      SAVERE      ;GOOD DATA SAVED IN 'SAVERE'
8303 026470 002266      WC          ;TEST DATA STARTING FROM 'RHWC'
8304 026472 000022      18.        ;18. REGISTERS TO BE COMPARED
8305 026474 026500      ST7       ;RETURN TO ST7 ON ERROR
8306 026476 026504      ST8       ;RETURN TO ST8 ON NO ERROR
8307 026500 104027      ST9:     ERROR 27 ;WRITE HEADER AND DATA CAUSED
8308 026500 000207      RTS        PC ;IMPROPER REGISTER CHANGE
8309                                     ;GOOD DATA GIVES WHAT SHOULD BE
8310                                     ;THERE
8311                                     ;RECEIVED DATA GIVES WHAT WAS BE
8312                                     ;THERE AFTER COMMAND
8313
8314                                     ;SETUP TO READ HEADER AND DATA FOR NEXT TRACK
8315 ST8:     ;FILL READ INTO BUFFER WITH ALL ONES
8316 026504
8317
8318 026504 004737 034556      JSR      PC,2#CLDISK    ;SET R1-RHCS1, R2-RHCS2
8319                                     ;R3-RHDS1, R4-RHER1
8320                                     ;GIVE RH-11 INITIALIZE
8321                                     ;SETUP UNIT NUBER
8322 026510 004037 034426      JSR      RD,2#CLAREA    ;CLEAR 260. WORDS, FROM REINTO
8323 026514 003444      REINTO     ;STARTING FROM REINTO
8324 026516 000404      260.      ;260. WORDS
8325 026520 177777      -1        ;FILL WITH -1
8326
8327
8328                                     ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
8329 ST9:     ;
8330 026522
8331 026522 004037 034402      JSR      RD,2#FLHEAD    ;SAVE HEADER DATA IN WRFROM
8332 026526 002400      WRFROM     ;LOCATION WHERE SAVED
8333 026530 000004      4          ;NUMBER OF WORDS SAVED
8334 026532 010001      10001     ;FIRST DATA WORD
8335 026534 000000      0         ;SECOND DATA WORD
8336 026536 000000      0         ;THIRD DATA WORD
8337 026540 000000      0         ;FOURTH DATA WORD
8338 026542
8339 026542 004037 034426      ST10:  JSR      RD,2#CLAREA    ;CLEAR 2 WORDS, FROM WRFROM+(4*2)
8340 026546 002410      WRFROM+(4*2) ;STARTING FROM WRFROM+(4*2)
8341 026550 000002      2         ;2 WORDS
8342 026552 002000      1*2000    ;FILL WITH 1*2000
8343
8344 026554 004037 034426      JSR      RD,2#CLAREA    ;CLEAR 254. WORDS, FROM WRFROM+(6*2)
8345 026560 002414      WRFROM+(6*2) ;STARTING FROM WRFROM+(6*2)
8346 026562 000376      254.     ;254. WORDS
8347 026564 000000      0         ;FILL WITH 0
8348
8349
8350 ST11:   ;FILL COMMAND INTO REGISTERS
8351 026566
8352

```

8353	026566	004037	036532	JSR	RO, @#RUN	: SETUP TO RUN FOR DATA COMMAND
8354	026572	000001				: CYLINDER 1
8355	026574	000		.BYTE	0	: SECTOR 0
8356	026575	000		.BYTE	0	: TRACK 0
8357	026576	177374			-256.-4	: WORD COUNT (DATA)=256.+
8359	026600	003444		REINTO		: 4 HEADER WORDS
8360						: BUS ADDRESS
8361						: STARTING ADDRESS OF DATA
8362	026602	000000		0		: BUFFER = REINTO
8363	026604	014000		ECI!FMT22		: DO NOT INHIBIT BUS ADDRESS INCREMENT
8364						: 16 BITS PER WORD FORMAT
8365						: INHIBIT ECC CORRECTION
8366	026606	002362		REFOR		: DO NOT INHIBIT HEADER COMPARE
8367						: GET READY TO DO A REFOR
8368						: READ HEADER AND DATA WITH 72 IN RHCS1
8369						
8370						: SAVE REGISTERS FOR COMPARISON AFTER READ HEADER
8371						: AND DATA
8372	026610	004037	034716	JSR	RO, @#SAVER	: SAVE REGISTERS
8373	026614	002210		RHWC		: RHWC IS THE FIRST REGISTER SAVED
8374	026616	004522		SAVERE		: STARTING ADDRESS OF WHERE
8375						: THE REGISTERS ARE SAVED
8376	026620	000020		16.		: NUMBER OF REGISTERS
8377						: SAVED = 16.
8378						
8379	026622	004737	034636	JSR	PC, @#CHECKT	: CHECK DVA, RDY, MOL, DPR, DRY, VV
8380						
8381						
8382	026626	013777	004516 153350	MOV	@#RP4VEC, @RPVEC	: SET RPO4 VECTOR ADDRESS
8383						: TO 'TIME1' IF P-CLOCK IS PRESENT
8384						: OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8385						: 'TIME' WILL ONLY SAVE
8386						: CURRENT CYLINDER ADDRESS
8387						: AND LOOK AHEAD REGISTERS
8388						
8389						
8390	026634	013746	002362	MOV	@#REFOR, -(SP)	: GET READY TO MOVE COMMAND
8391	026640	052716	000101	BIS	#GO!IE, (SP)	: GET READY TO SET GO AND
8392						: ENABLE INTERRUPT
8393	026644	012677	153346	MOV	(SP)+, @RHCS1	: GO WITH
8394						: 72 IN RHCS1 FOR READ DATA
8395						: WITH INTERRUPT ENABLED
8396	026650	011100		MOV	@R1, RO	: SAVE RHCS1 DURING ABOVE OPERATION
8397	026652	011305		MOV	@R3, RS	: SAVE RHDS1 DURING ABOVE OPERATION
8398						
8399						
8400	026654	104410		WAT		: WAIT FOR RDY BIT TO SET
8401	026656	002216		RHCS1		: WAIT FOR RHCS1 REGISTER
8402	026660	000200		RDY		: WAIT FOR RDY BIT IN RHCS1 REGISTER
8403	026662	001614		908.		: ALLOW 9080 MICRO SECONDS
8404	026664	001507		839.		: RDY MUST SET BETWEEN
8405						: 690 AND 17470 MICRO SECONDS
8406						: COMPARE CONTENTS OF RHCS1 AND RHDS1 ALREADY SAVED IN
8407						: RO AND RS IMMEDIATELY AFTER GO
8408	026666	013746	002362	MOV	@#REFOR, -(SP)	: SAVE COMMAND

2



```

026767 004101 BIS IE!GO!DVA (SP) :INCLUDE IE!GO!DVA
026768 004102 MOV (SP),J#SGDDAT :SAVE FOR PRINTOUT
026769 004103 CMP (SP)+,R0 :DURING ABOVE OPERATION ONLY IE!GO!DVA
:AND COMMAND SHOULD BE SET
026770 001405 BEQ 645 :BRANCH IF GOOD
026771 004104 MOV R0,J#SBDDAT :BAD DATA
026772 004105 MOV R1,J#REGADR :FAILING REGISTER RHCS1
026773 004106 ERROR 21 :DURING ABOVE OPERATION ONLY
:COMMAND AND IE!GO!DVA SHOULD BE SET
026774 004510 6-5: MOV #MOL!DPR!VV (SP) :SAVE BITS SET DURING OPERATION IN RHCS1
MOV (SP),J#SGDDAT :SAVE FOR PRINTOUT
CMP (SP)+,R5 :DURING ABOVE OPERATION ONLY MOL!DPR!VV
:SHOULD BE SET
026775 001405 BEQ 645 :BRANCH IF GOOD
026776 004104 MOV R5,J#SBDDAT :BAD DATA
026777 004105 MOV R6,J#REGADR :FAILING REGISTER RHCS1
026778 004106 ERROR 63 :DURING ABOVE OPERATION ONLY
:MOL!DPR!VV SHOULD BE SET

026779 004037 034460 :CHANGE SAVED REGISTERS TO EXPECTED VALUES
JSR R0,J#FILLRE :MOV 0 INTO SAVED RHWC
RHWC :SAVED REGISTER TO CHANGE
C :DATA
026780 004037 034460 JSR R0,J#FILLRE :MOV REINTC+(260.*2) INTO SAVED RHBA
RHBA :SAVED REGISTER TO CHANGE
REINTC+(260.*2) :DATA
026781 004037 034460 JSR R0,J#FILLRE :MOV 1 INTO SAVED RHDST
RHDST :SAVED REGISTER TO CHANGE
1 :DATA

:COMPARE REGISTERS BEFORE READ HEADER AND DATA WITH
:REGISTERS AFTER COMMAND
026776 004037 035546 JSR R0,J#CCMREG :COMPARE SAVED REGISTERS WITH
:PRESENT VALUE
SAVERE :GOOD DATA SAVED IN 'SAVERE'
WC :TEST DATA STARTING FROM 'RHWC'
IB :IB. REGISTERS TO BE COMPARED
ST12 :RETURN TO ST12 ON ERROR
ST13 :RETURN TO ST13 ON NO ERROR
026776 004037 035546 ST12: ERROR 31 :READ HEADER AND DATA CAUSED
RTS PC :IMPROPER REGISTER CHANGE
:GOOD DATA GIVES WHAT SHOULD
:BE THERE
:RECEIVED DATA GIVES WHAT
:WAS THERE AFTER COMMAND
026776 004037 035546 ST13: ;READ INTO BUFFER IS CHECKED FOR PROPER READ

026776 004037 036576 JSR R0,J#COMPAR :COMPARE TWO BLOCKS OF MEMORY
WRFROM :GOOD DATA STARTS FROM WRFROM
REINTC :TEST DATA STARTS FROM REINTC
260. :260. WORDS TO BE COMPARED
ST14 :RETURN TO ST14 ON ERROR

```

```

0465 027034 027042
0466 027036 104032
0467 027040 000207
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520

```

```

ST15 : RETURN TO ST15 ON NO ERROR
ST14: ERROR 32 : WRITE HEADER AND DATA
      RTS PC : WITH AN IMPLIED SEEK
           : FOLLOWED BY A READ
           : HEADER AND DATA ON THE
           : NEXT TRACK GAVE A
           : READ ERROR
           : ERROR MAY BE READ OR WRITE

```

```

: THE HEADS HAVE ADVANCED ONE CYLINDER BY AN IMPLIED
: SEEK
: CHANGES WILL BE MADE TO ENABLE GOING TO THE NEXT
: CYLINDER AND THEN THE ABOVE WILL BE REPEATED
: TILL CYLINDER 10 IS REACHED
ST15: INC 2*ST1+10
      ADD #(<1.*2000>),2*ST2+10
      INC 2*ST3+10
      ADD #(<1.*2000>),2*ST4+10
      INC 2*ST5+4
      INC 2*ST6+6
      INC 2*ST6-16
      INC 2*ST9+10
      ADD #(<1.*2000>),2*ST10+10
      INC 2*ST11+4
      DEC 2*STMP1 : COUNT FOR TEN TIMES
      BNE ST16 : BRANCH IF 10 NOT DONE
      BR ST17 : 10 COMPLETED SO CONTINUE
ST16: JMP ST1 : JUMP AS 10 NOT DONE

```

```

: THE HEADS ARE NOW AT CYLINDER 10
: ALL REGISTERS WILL BE SAVED AND A SEEK WILL BE GIVEN
: TO CYLINDER 0

```

```

: FILL REGISTERS FOR A SEEK COMMAND
ST17: JSR PC,2*CLDISK : SET R1-RHCS1, R2-RHCS2
           : R3-RHDS1, R4-RHER1
           : GIVE RH-11 INITIALIZE
           : SETUP UNIT NUBER
      JSR R0,2*SEEKCY : SEEK FOR
      0 : CYLINDER 0
      : SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
      JSR R0,2*SAVER : SAVE REGISTERS
      RHWC : RHWC IS THE FIRST REGISTER SAVED
      SAVERE : STARTING ADDRES OF WHERE
           : THE REGISTERS ARE SAVED
           : NUMBER OF REGISTERS
           : SAVED = 18.
      18.
      MOV 2*RP4VEC,2*RPVEC : SET RPO4 VECTOR ADDRESS
           : TO 'TIME1' IF P-CLOCK IS PRESENT

```



;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT  
; 'TIME' WILL ONLY SAVE  
; CURRENT CYLINDER ADDRESS  
; AND LOOK AHEAD REGISTERS

```

027166 013746 002364 MOV 2#SEECOM, -(SP) ; GET READY TO MOVE COMMAND
027172 052716 000101 BIS 2#GO!IE, (SP) ; GET READY TO SET GO AND
; ENABLE INTERRUPT
027176 012677 153014 MOV (SP)+, 2#RHCSI ; GO WITH
; 4 IN RHCSI FOR SEEK
; WITH INTERRUPT ENABLED
027200 011100 MOV 2#R1, R0 ; SAVE RHCSI DURING ABOVE OPERATION
027204 011305 MOV 2#R3, R5 ; SAVE RHCSI DURING ABOVE OPERATION
; SEEK FOR ONE CYLINDER=7MILI SEC., FOR TEN=70 MILI SEC

027206 104410 WAT ; WAIT FOR DRY BIT TO SET
027210 002240 RHDS1 ; WAIT FOR RHDS1 REGISTER
027212 000200 DRY ; WAIT FOR DRY BIT IN RHDS1 REGISTER
027214 015530 7000. ; ALLOW 70000 MICRO SECONDS
027216 000043 35. ; DRY MUST SET BETWEEN
; 69650 AND 70350 MICRO SECONDS
; COMPARE CONTENTS OF RHCSI AND RHDS1 ALREADY SAVED IN
; R0 AND R5 IMMEDIATELY AFTER GO
027220 013746 002364 MOV 2#SEECOM, -(SP) ; SAVE COMMAND
027224 052716 004301 BIS 2#DVA!GO!IE!RDY, (SP) ; INCLUDE DVA!GO!IE!RDY
027230 011637 001124 MOV (SP), 2#SGDDAT ; SAVE FOR PRINTOUT
027234 022600 CMP (SP)+, R0 ; DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
; AND COMMAND SHOULD BE SET
027236 001405 BEQ 64$ ; BRANCH IF GOOD
027240 010037 001126 MOV R0, 2#SBDDAT ; BAD DATA
027244 010137 004510 MOV R1, 2#REGADR ; FAILING REGISTER RHCSI
027250 104021 ERROR 21 ; DURING ABOVE OPERATION ONLY
; COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
027252 012746 030500 64$: MOV 2#PIP!MOL!DPR!VV, -(SP) ; SAVE BITS SET DURING OPERATION IN RHDS1
027256 011637 001124 MOV (SP), 2#SGDDAT ; SAVE FOR PRINTOUT
027262 022605 CMP (SP)+, R5 ; DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
; SHOULD BE SET
027264 001405 BEQ 66$ ; BRANCH IF GOOD
027266 010537 001126 MOV R5, 2#SBDDAT ; BAD DATA
027272 010337 004510 MOV R3, 2#REGADR ; FAILING REGISTER RHDS1
027276 104063 ERROR 63 ; DURING ABOVE OPERATION ONLY
; PIP!MOL!DPR!VV SHOULD BE SET

027300 66$: ; NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
; JSR R0, 2#FILLRE ; MOV 0 INTO SAVED RHCC
; RHCC ; SAVED REGISTER TO CHANGE
; 0 ; DATA
027310 053737 004650 004546 BIS 2#ATTENT, 2#SAVERE+24 ; SET APPROPRIATE ATA BITS
; FOR WORKING DRIVE IN
; SAVED RHAS LOACTION

027316 004037 035440 JSR R0, 2#CHREG ; CHANGE BITS IN SAVED REGISTER
027322 002240 RHDS1 ; CHANGE RHDS1 REGISTER

```

```

8577 027324 000001      1      ;1 BIT/BITS TO BE CHANGED
8578 027326 000001      1      ;NEW VALUE OF ATA IS 1
8579 027330 100000      ATA     ;CHANGE ATA BIT
8580
8581 027332 004037 035440 JSR     RD,2*CHREG ;CHANGE BITS IN SAVED REGISTER
8582 027336 002216      RHCS1  ;CHANGE RHCS1 REGISTER
8583
8584 027340 000001      1      ;1 BIT/BITS TO BE CHANGED
8585 027342 000001      1      ;NEW VALUE OF SC IS 1
8586 027344 100000      SC      ;CHANGE SC BIT
8587
8588      ;COMPARE REGISTERS AFTER A SEEK COMMAND
8589
8590
8591 027346 004037 035546 JSR     RD,2*COMREG ;COMPARE SAVED REGISTERS WITH
8592      ;PRESENT VALUE
8593 027352 004522      SAVERE ;GOOD DATA SAVED IN 'SAVERE'
8594 027354 002266      WC      ;TEST DATA STARTING FROM 'RHWC'
8595 027356 000022      18.   ;18. REGISTERS TO BE COMPARED
8596 027360 027364      ST18  ;RETURN TO ST18 ON ERROR
8597 027362 027370      ST19  ;RETURN TO ST19 ON NO ERROR
8598
8599 027364 104037      ST18:  ERROR 37   ;SEEK COMMAND CAUSED AN
8600 027366 000207      RTS     PC    ;ERROR
8601      ;GOOD DATA GIVES WHAT SHOULD
8602      ;BE THERE
8603      ;RECEIVED DATA GIVES WHAT WAS
8604      ;THERE AFTER A SEEK COMMAND
8605
8606      ;AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
8607      ;NO ERROR BITS
8608      ;A READ HEADER AND DATA WILL BE DONE ON CYLINDER 0
8609      ;SECTOR 21 TRACK 19, EXPECTED DATA IS 1125
8610      ;FOR 10 WORDS
8611      ;CLEAR READ INTO BUFFER WITH ALL ONES
8612 027370      ST19:  JSR     RD,2*CLAREA ;CLEAR 260. WORDS, FROM REINTO
8613 027370 004037 034426 REINTO ;STARTING FROM REINTC
8614 027374 003444      260.  ;260. WORDS
8615 027376 000404      -1     ;FILL WITH -1
8616 027400 177777
8617
8618      ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
8619
8620 027402 004037 034402 JSR     RD,2*FLHEAD ;SAVE HEADER DATA IN WRFROM
8621 027406 002400      WRFROM ;LOCATION WHERE SAVED
8622 027410 000004      4      ;NUMBER OF WORDS SAVED
8623 027412 010000      10000 ;FIRST DATA WORD
8624 027414 011025      <18.*400>!<21.> ;SECOND DATA WORD
8625 027416 000000      0      ;THIRD DATA WORD
8626 027420 000000      0      ;FOURTH DATA WORD
8627 027422 004037 034426 JSR     RD,2*CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<4.*2>
8628 027426 002410      WRFROM+<4.*2> ;STARTING FROM WRFROM+<4.*2>
8629 027430 000012      10.   ;10. WORDS
8630 027432 001125      <18.*40>!<21.> ;FILL WITH <18.*40>!<21.>
8631
8632 027434 004037 034426 JSR     RD,2*CLAREA ;CLEAR 246. WORDS, FROM WRFROM+<14.*2>

```

```

8633 027440 002434 WRFROM+(14.*2) ;STARTING FROM WRFROM+(14.*2)
8634 027442 000366 246. ;246. WORDS
8635 027444 177777 -1 ;FILL WITH -1
8636
8637
8638 ;FILL READ HEADER AND DATA COMMAND FOR 10 WORDS
8639
8640 027446 004037 036532 JSR RD,2*RUN ;SETUP TO RUN FOR DATA COMMAND
8641 027452 000000 C ;CYLINDER 0
8642 027454 025 .BYTE 21. ;SECTOR 21.
8643 027455 022 .BYTE 18. ;TRACK 18.
8644 027456 177752 -10.-4 ;WORD COUNT (DATA)=10.+
8645 ;4 HEADER WORDS
8646 027460 003444 REINTO ;BUS ADDRESS
8647 ;STARTING ADDRESS OF DATA
8648 ;BUFFER = REINTO
8649 027462 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8650 027464 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
8651 ;INHIBIT ECC CORRECTION
8652 027466 002362 REFOR ;DO NOT INHIBIT HEADER COMPARE
8653 ;GET READY TO DO A REFOR
8654 ;READ HEADER AND DATA WITH 72 IN RHCS1
8655
8656
8657 027470 004737 034636 JSR PC,2*CHECKT ;CHECK DVA.RDY,MOL,DPR,DRY,VV
8658
8659
8660 027474 013777 004516 152502 MOV 2*RP4VEC,2*RPVEC ;SET RPO4 VECTOR ADDRESS
8661 ;TO 'TIME1' IF P-CLOCK IS PRESENT
8662 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8663 ;'TIME' WILL ONLY SAVE
8664 ;CURRENT CYLINDER ADDRESS
8665 ;AND LOOK AHEAD REGISTERS
8666
8667
8668 027502 013746 002362 MOV 2*REFOR,-(SP) ;GET READY TO MOVE COMMAND
8669 027506 052716 000101 BIS *GO!IE,(SP) ;GET READY TO SET GO AND
8670 ;ENABLE INTERRUPT
8671 027512 012677 152500 MOV (SP)+,2*RHCS1 ;GO WITH
8672 ;72 IN RHCS1 FOR READ DATA
8673 ;WITH INTERRUPT ENABLED
8674
8675
8676 027516 104410 WAT ;WAIT FOR RDY BIT TO SET
8677 027520 002216 RHCS1 ;WAIT FOR RHCS1 REGISTER
8678 027522 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8679 027524 001614 908. ;ALLOW 9080 MICRO SECONDS
8680 027526 00:507 839. ;RDY MUST SET BETWEEN
8681 ;690 AND 17470 MICRO SECONDS
8682
8683 ;CHECK READ WORDS
8684
8685 027530 004037 036576 JSR RD,2*COMPAR ;COMPARE TWO BLOCKS OF MEMORY
8686 027534 002400 WRFROM ;GOOD DATA STARTS FROM WRFROM
8687 027536 003444 REINTO ;TEST DATA STARTS FROM REINTO
8688 027540 000404 260. ;260. WORDS TO BE COMPARED

```

8689	027542	027546
8690	027544	027552
8691		
8692		
8693		
8694	027546	104032
8695	027550	000207
8696		
8697		
8698	027552	
8699		
8700		

ST26  
ST27

:RETURN TO ST26 ON ERROR  
:RETURN TO ST27 ON NO ERROR

ST26: ERROR 32  
RTS PC

:READ HEADER AND DATA  
:FOLLOWING A SEEK TO CYLINDER 0  
:FROM CYLINDER 10 GAVE AN  
:ERROR

ST27:

```

8701
8702
8703
8704
8705
8706
8707
8708
8709
8710
8711
8712
8713 027552 000004
8714 027554 012706 001000
8715 027560 012737 000033 004514
8716
8717 027566 004737 034556
8718
8719
8720
8721
8722 027572 013777 004516 152404
8723
8724
8725
8726
8727
8728
8729 027600 013746 002340
8730 027604 052716 000101
8731
8732 027610 012677 152402
8733
8734
8735 027614 011100
8736 027616 011305
8737
8738 027620 104410
8739 027622 002240
8740 027624 000200
8741 027626 076377
8742 027630 056701
8743
8744
8745
8746 027632 004037 034526
8747 027636 000012
8748
8749
8750 027640 004037 034716
8751 027644 002210
8752 027646 004522
8753
8754 027650 000022
8755
8756

```

```

:*****
:*TEST 33      SEEK TEST
:*
:* THIS TEST DEPENDS ON HEADER AND DATA WRITTEN BY THE
:* PREVIOUS TEST.  AT THIS POINT THE HEADS ARE ON
:* CYLINDER 0
:*
:* ALL REGISTERS ARE SAVED
:* A SEEK IS GIVEN TO CYLINDER 9
:* ALL REGISTERS ARE CHECKED FOR IMPROPER CHANGE
:* DATA WRITTEN ON CYLINDER 9 IS CHECKED
:*****
†ST33: SCOPE
      MOV      #STACK,SP      ;RESET STACK
      MOV      #33,‡†STNM     ;SAVE TEST NUMBER
      JSR      PC,‡#CLDISK    ;SET R1-RHCS1, R2-RHCS2
                                   ;R3-RHDS1, R4-RHER1
                                   ;GIVE RH-11 INITIALIZE
                                   ;SETUP UNIT NUBER
      ;THIS GETS HEADS TO CYLINDER 0
      MOV      ‡#RP4VEC,‡RPVEC ;SET RPO4 VECTOR ADDRESS
                                   ;TO 'TIME1' IF P-CLOCK IS PRESENT
                                   ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
                                   ;'TIME' WILL ONLY SAVE
                                   ;CURRENT CYLINDER ADDRESS
                                   ;AND LOOK AHEAD REGISTERS
      MOV      ‡#RECALI, -(SP) ;GET READY TO MOVE COMMAND
      BIS      #GO!IE, (SP)    ;GET READY TO SET GO AND
                                   ;ENABLE INTERRUPT
      MOV      (SP)+,‡RHCS1    ;GO WITH
                                   ;6 IN RHCS1 FOR RECALIBRATE
                                   ;WITH INTERRUPT ENABLED
      MOV      ‡R1,R0          ;SAVE RHCS1 DURING ABOVE OPERATION
      MOV      ‡R3,R5          ;SAVE RHDS1 DURING ABOVE OPERATION
      WAT                      ;WAIT FOR DRY BIT TO SET
      RHDS1                    ;WAIT FOR RHDS1 REGISTER
      DRY                      ;WAIT FOR DRY BIT IN RHDS1 REGISTER
      31999.                   ;ALLOW 319990 MICRO SECONDS
      24001.                   ;DRY MUST SET BETWEEN
                                   ;79980 AND 560000 MICRO SECONDS
      ;FILL REGISTERS FOR A SEEK COMMAND
      JSR      R0,‡#SEEKCY     ;SEEK FOR
      10.                      ;CYLINDER 10.
      ;SAVE REGISTERS FOR COMPARISON AFTER SEEK COMMAND
      JSR      R0,‡#SAVER     ;SAVE REGISTERS
      RHWC                    ;RHWC IS THE FIRST REGISTER SAVED
      SAVERE                   ;STARTING ADDRES OF WHERE
                                   ;THE REGISTERS ARE SAVED
      18.                      ;NUMBER OF REGISTERS
                                   ;SAVED = 18.

```

```

8757 027652 013777 004516 152324      MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
8758                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
8759                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8760                                     ;'TIME' WILL ONLY SAVE
8761                                     ;CURRENT CYLINDER ADDRESS
8762                                     ;AND LOOK AHEAD REGISTERS
8763
8764
8765 027660 013746 002364      MOV      @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
8766 027664 052716 000101      BIS      #GO!IE,(SP) ;GET READY TO SET GO AND
8767                                     ;ENABLE INTERRUPT
8768 027670 012677 152322      MOV      (SP)+,@RHCSI ;GO WITH
8769                                     ;4 IN RHCSI FOR SEEK
8770                                     ;WITH INTERRUPT ENABLED
8771 027674 011100      MOV      @R1,R0 ;SAVE RHCSI DURING ABOVE OPERATION
8772 027676 011305      MOV      @R3,R5 ;SAVE RHDSI DURING ABOVE OPERATION
8773
8774                                     ;SEEK FOR ONE CYLINDER=7 MILI SEC., FOR TEN=70 MILI SEC
8775
8776 027700 104410      WAT                                     ;WAIT FOR DRY BIT TO SET
8777 027702 002240      RHDS1 ;WAIT FOR RHDS1 REGISTER
8778 027704 000200      DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
8779 027706 015530      7000. ;ALLOW 70000 MICRO SECONDS
8780 027710 000043      35. ;DRY MUST SET BETWEEN
8781                                     ;69650 AND 70350 MICRO SECONDS
8782                                     ;COMPARE CONTENTS OF RHCSI AND RHDS1 ALREADY SAVED IN
8783                                     ;R0 AND R5 IMMEDIATELY AFTER GO
8784 027712 013746 002364      MOV      @#SEECOM,-(SP) ;SAVE COMMAND
8785 027716 052716 004301      BIS      #DVA!GO!IE!RDY,(SP) ;INCLUDE DVA!GO!IE!RDY
8786 027722 011637 001124      MOV      (SP),@#SGDDAT ;SAVE FOR PRINTOUT
8787 027726 022600      CMP      (SP)+,R0 ;DURING ABOVE OPERATION ONLY DVA!GO!IE!RDY
8788                                     ;AND COMMAND SHOULD BE SET
8789 027730 001405      BEQ      67$ ;BRANCH IF GOOD
8790 027732 010037 001126      MOV      R0,@#SBDDAT ;BAD DATA
8791 027736 010137 004510      MOV      R1,@#REGADR ;FAILING REGISTER RHCSI
8792 027742 104021      ERROR   21 ;DURING ABOVE OPERATION ONLY
8793                                     ;COMMAND AND DVA!GO!IE!RDY SHOULD BE SET
8794 027744 012746 030500      67$: MOV      #PIP!MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
8795 027750 011637 001124      MOV      (SP),@#SGDDAT ;SAVE FOR PRINTOUT
8796 027754 022605      CMP      (SP)+,R5 ;DURING ABOVE OPERATION ONLY PIP!MOL!DPR!VV
8797                                     ;SHOULD BE SET
8798 027756 001405      BEQ      69$ ;BRANCH IF GOOD
8799 027760 010537 001126      MOV      R5,@#SBDDAT ;BAD DATA
8800 027764 010337 004510      MOV      R3,@#REGADR ;FAILING REGISTER RHDSI
8801 027770 104063      ERROR   63 ;DURING ABOVE OPERATION ONLY
8802                                     ;PIP!MOL!DPR!VV SHOULD BE SET
8803 027772      69$:
8804
8805                                     ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUE
8806 027772 004037 034460      JSR      R0,@#FILLRE ;MOV 10. INTO SAVED RHCC
8807 027776 002252      RHCC ;SAVED REGISTER TO CHANGE
8808 030000 000012      10. ;DATA
8809
8810 030002 053737 004650 004546      BIS      @#ATTENT,@SAVERE+24 ;SET APPROPRIATE ATA BITS
8811                                     ;FOR WORKING DRIVE IN
8812                                     ;SAVED RHAS LOACTION

```





```

8813
8814 030010 004037 035440 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
8815 030014 002240 RHDS1 ;CHANGE RHDS1 REGISTER
8816
8817 030016 000001 1 ;1 BIT/BITS TO BE CHANGED
8818 030020 000001 1 ;NEW VALUE OF ATA IS 1
8819 030022 100000 ATA ;CHANGE ATA BIT
8820
8821 030024 004037 035440 JSR RO,2#CHREG ;CHANGE BITS IN SAVED REGISTER
8822 030030 002216 RHCS1 ;CHANGE RHCS1 REGISTER
8823
8824 030032 000001 1 ;1 BIT/BITS TO BE CHANGED
8825 030034 000001 1 ;NEW VALUE OF SC IS 1
8826 030036 100000 SC ;CHANGE SC BIT
8827
8828 ;COMPARE REGISTERS AFTER A SEEK COMMAND
8829
8830 030040 004037 035546 JSR RO,2#COMREG ;COMPARE SAVED REGISTERS WITH
8831 ;PRESENT VALUE
8832 030044 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
8833 030046 002266 WC ;TEST DATA STARTING FROM 'RHWC'
8834 030050 000022 18. ;18. REGISTERS TO BE COMPARED
8835 030052 030056 1$ ;RETURN TO 1$ ON ERROR
8836 030054 030062 2$ ;RETURN TO 2$ ON NO ERROR
8837
8838 030056 104037 1$: ERROR 37 ;SEEK COMMAND CAUSED
8839 030060 000207 RTS PC ;ERROR
8840 ;GOOD DATA GIVES WHAT SHOULD
8841 ;BE THERE
8842 ;RECEIVED DATA GIVES WHAT WAS
8843 ;THERE AFTER A SEEK COMMAND
8844
8845
8846 ;AT THIS POINT THE CURRENT CYLINDER IS GOOD AND THERE ARE
8847 ;NO ERROR BITS
8848 ;A READ HEADER AND DATA WILL BE DONE ON CYLINDER 9
8849 ;SECTOR 21 TRACK 18. EXPECTED DATA IS 23125
8850 ;FOR 20 WORDS
8851 ;CLEAR READ INTO BUFFER WITH ALL ONES
8852 030062 2$:
8853
8854 030062 004737 034556 JSR PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
8855 ;R3-RHDS1, R4-RHER1
8856 ;GIVE RH-11 INITIALIZE
8857 ;SETUP UNIT NUMBER
8858 030066 004037 034426 JSR RO,2#CLAREA ;CLEAR 260. WORDS. FROM REINTO
8859 030072 003444 REINTO ;STARTING FROM REINTO
8860 030074 000404 260. ;260. WORDS
8861 030076 177777 -1 ;FILL WITH -1
8862
8863 ;FILL WRITE FROM BUFFER WITH EXPECTED DATA
8864
8865 030100 004037 034402 JSR RO,2#FLHEAD ;SAVE HEADER DATA IN WRFROM
8866 030104 002400 WRFROM ;LOCATION WHERE SAVED
8867 030106 000004 4 ;NUMBER OF WORDS SAVED
8868 030110 010011 10011 ;FIRST DATA WORD

```



```

8869 030112 011025 <18.*400>!<21.> ;SECOND DATA WORD
8870 030114 000000 0 ;THIRD DATA WORD
8871 030116 000000 0 ;FOURTH DATA WORD
8872 030120 004037 034426 JSR R0,@#CLAREA ;CLEAR 20. WORDS, FROM WRFROM+<4.*2>
8873 030124 002410 WRFROM+<4.*2> ;STARTING FROM WRFROM+<4.*2>
8874 030126 000024 20. ;20. WORDS
8875 030130 023125 <9.*2000>!<18.*40>!<21.> ;FILL WITH <9.*2000>!<18.*40>!<21.>
8876
8877 030132 004037 034426 JSR R0,@#CLAREA ;CLEAR 250. WORDS, FROM WRFROM+<24.*2>
8878 030136 002460 WRFROM+<24.*2> ;STARTING FROM WRFROM+<24.*2>
8879 030140 000372 250. ;250. WORDS
8880 030142 177777 -1 ;FILL WITH -1
8881
8882
8883 ;FILL READ HEADER AND DATA COMMAND FOR 10 WORDS
8884
8885 030144 004037 036532 JSR R0,@#RUN ;SETUP TO RUN FOR DATA COMMAND
8886 030150 000011 9. ;CYLINDER 9.
8887 030152 025 .BYTE 21. ;SECTOR 21.
8888 030153 022 .BYTE 18. ;TRACK 18.
8889 030154 177750 -20.-4 ;WORD COUNT (DATA)=20.+
8890 ;4 HEADER WORDS
8891 030156 003444 REINTO ;BUS ADDRESS
8892 ;STARTING ADDRESS OF DATA
8893 ;BUFFER = REINTO
8894 030160 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
8895 030162 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
8896 ;INHIBIT ECC CORRECTION
8897 ;DO NOT INHIBIT HEADER COMPARE
8898 030164 002362 REFOR ;GET READY TO DO A REFOR
8899 ;READ HEADER AND DATA WITH 72 IN RHCS1
8900
8901
8902 030166 004737 034636 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
8903
8904
8905 030172 013777 004516 152004 MOV @#RP4VEC,@RPVEC ;SET RP04 VECTOR ADDRESS
8906 ;TO 'TIME1' IF P-CLOCK IS PRESENT
8907 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8908 ;'TIME' WILL ONLY SAVE
8909 ;CURRENT CYLINDER ADDRESS
8910 ;AND LOOK AHEAD REGISTERS
8911
8912
8913 030200 013746 002362 MOV @#REFOR,-(SP) ;GET READY TO MOVE COMMAND
8914 030204 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
8915 ;ENABLE INTERRUPT
8916 030210 012677 152002 MOV (SP)+,@RHCS1 ;GO WITH
8917 ;72 IN RHCS1 FOR READ DATA
8918 ;WITH INTERRUPT ENABLED
8919
8920
8921 030214 104410 WAT ;WAIT FOR RDY BIT TO SET
8922 030216 002216 RHCS1 ;WAIT FOR RHCS1 REGISTER
8923 030220 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
8924 030222 001614 908. ;ALLOW 9080 MICRO SECONDS

```

8925 030224 001507

839.

:RDY MUST SET BETWEEN  
:690 AND 17470 MICRO SECONDS

8926  
8927

:CHECK READ WORDS

8928  
8929

8930 030226 004037 036575

JSR RD.#COMPAR

:COMPARE TWO BLOCKS OF MEMCRY  
:GOOD DATA STARTS FROM WRFROM  
:TEST DATA STARTS FROM REINTO  
:260. WORDS TO BE COMPARED  
:RETURN TO 3\$ ON ERROR  
:RETURN TO 4\$ ON NO ERROR

8931 030232 002400

WRFROM

8932 030234 003444

REINTO

8933 030236 000404

260.

8934 030240 030244

3\$

8935 030242 030250

4\$

8936

8937

8938 030244 104032

3\$: ERROR 32

:READ HEADER AND DATA  
:FOLLOWING A SEEK TO CYLINDER 9  
:FROM CYLINDER 0 GAVE AN  
:ERROR

8939 030246 000207

RTS PC

8940

8941

8942 030250

4\$:

8943

8944

8945

8946

8947

8948

```

8949 :*****
8950 :*TEST 34 WRITE CHECK HEADER AND DATA
8951
8952 :* WRITE CHECK HEADER AND DATA CYLINDER 5 FORMAT 16 BITS PER WORD
8953 :* TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 266
8954 :* CONSISTING OF
8955 :* 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)
8956 :* 10 WORDS OF 177777
8957 :* 10 WORDS OF 0
8958 :* 10 WORDS OF 052525
8959 :* 10 WORDS OF 125252
8960 :* 16 WORDS OF LEFT ROTATING ZERO (EG 177776,177775)
8961 :* 16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)
8962 :* 174 WORDS OF 377
8963 :* 4 WORDS OF HEADER
8964 :* 2 WORDS OF 12345
8965
8966
8967 :* FIRST THE ABOVE DATA IS WRITTEN BY A WRITE HEADER AND
8968 :* DATA COMMAND
8969 :* CHECK FOR NO ERRORS
8970 :* THEN THE ABOVE DATA IS READ BY A READ HEADER AND DATA
8971 :* CHECK FOR NO ERRORS
8972 :* THEN THE ABOVE WRITE CHECK HEADER AND DATA IS GIVEN
8973 :*****
8974 030250 000004          ST34: SCOPE
8975 030252 012706 001000  MOV      #STACK,SP      ;RESET STACK
8976 030256 012737 000034 004514  MOV      #34,#TSTNM     ;SAVE TEST NUMBER
8977
8978 030264 004737 034556  JSR      PC,#C'DISK     ;SET R1-RHCS1, R2-RHCS2
8979                                     ;R3-RHDS1, R4-RHER1
8980                                     ;GIVE RH-11 INITIALIZE
8981                                     ;SETUP UNIT NUMBER
8982                                     ;GET HEADS TO CYLINDER 5
8983
8984 030270 004037 034526  JSR      RO,#SEEKCY     ;SEEK FOR
8985 030274 000005 5          ;CYLINDER 5
8986 030276 013777 004516 151700  MOV      @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
8987                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
8988                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
8989                                     ;'TIME' WILL ONLY SAVE
8990                                     ;CURRENT CYLINDER ADDRESS
8991                                     ;AND LOOK AHEAD REGISTERS
8992
8993 030304 013746 002364  MOV      @#SEECOM,-(SP) ;GET READY TO MOVE COMMAND
8994 030310 052716 000101  BIS      #GO!IE,(SP)   ;GET READY TO SET GO AND
8995                                     ;ENABLE INTERRUPT
8996 030314 012677 151676  MOV      (SP)+,@RHCS1  ;GO WITH
8997                                     ;4 IN RHCS1 FOR SEEK
8998                                     ;WITH INTERRUPT ENABLED
8999 030320 011100  MOV      @R1,R0        ;SAVE RHCS1 DURING ABOVE OPERATION
9000 030322 011305  MOV      @R3,R5        ;SAVE RHDS1 DURING ABOVE OPERATION
9001
9002
9003 030324 104410  WAT                                     ;WAIT FOR DRY BIT TO SET
9004 030326 002240  RHDS1                       ;WAIT FOR RHDS1 REGISTER

```

```

9005 030330 000200          DRY          ;WAIT FOR DRY BIT IN RHDS1 REGISTER
9006 030332 015530          7000.        ;ALLOW 7000 MICRO SECONDS
9007 030334 000043          35.          ;DRY MUST SET BETWEEN
9008                                ;59650 AND 70350 MICRO SECONDS
9009
9010 030336 004737 034556  JSR    PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
9011                                ;R3-RHDS1, R4-RHER1
9012                                ;GIVE RH-11 INITIALIZE
9013                                ;SETUP UNIT NUMBER
9014                                ;FILL WRITE FROM BUFFER WITH HEADER
9015
9016 030342 004037 034402  JSR    RO, @#FLHEAD ;SAVE HEADER DATA IN WRFROM
9017 030346 002400          WRFROM        ;LOCATION WHERE SAVED
9018 030350 000004          4            ;NUMBER OF WORDS SAVED
9019 030352 010005          10005         ;FIRST DATA WORD
9020 030354 003404          <7*400>!4    ;SECOND DATA WORD
9021 030356 000000          0            ;THIRD DATA WORD
9022 030360 000000          0            ;FOURTH DATA WORD
9023
9024                                ;10 WORDS OF OF THE FOLLOWING DATA
9025                                ; 12344,17777,0,52525,125252
9026 030362 004037 034426  JSR    RO, @#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<4*2>
9027 030366 002410          WRFROM+<4*2> ;STARTING FROM WRFROM+<4*2>
9028 030370 000012          10.          ;10. WORDS
9029 030372 012344          <5*2000>!<7*40>!4 ;FILL WITH <5*2000>!<7*40>!4
9030
9031 030374 004037 034426  JSR    RO, @#CLAREA ;CL AR 10. WORDS, FROM WRFROM+<14.*2>
9032 030400 002434          WRFROM+<14.*2> ;STARTING FROM WRFROM+<14.*2>
9033 030402 000012          10.          ;10. WORDS
9034 030404 177777          -1          ;FILL WITH -1
9035
9036 030406 004037 034426  JSR    RO, @#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<24.*2>
9037 030412 002460          WRFROM+<24.*2> ;STARTING FROM WRFROM+<24.*2>
9038 030414 000012          10.          ;10. WORDS
9039 030416 000000          0            ;FILL WITH 0
9040
9041 030420 004037 034426  JSR    RO, @#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<34.*2>
9042 030424 002504          WRFROM+<34.*2> ;STARTING FROM WRFROM+<34.*2>
9043 030426 000012          10.          ;10. WORDS
9044 030430 052525          52525         ;FILL WITH 52525
9045
9046 030432 004037 034426  JSR    RO, @#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<44.*2>
9047 030436 002530          WRFROM+<44.*2> ;STARTING FROM WRFROM+<44.*2>
9048 030440 000012          10.          ;10. WORDS
9049 030442 125252          125252        ;FILL WITH 125252
9050
9051                                ;FILL LEFT ROTATING ZEROS FROM WRFROM+<54.*2>
9052                                MOV    R1, -(SP) ;PUSH R1 ON STACK
9053 030444 010146          177776        MOV    #177776, R0 ;DATA
9054 030446 012700          000020        MOV    #16, R5 ;COUNT
9055 030452 012705          002554        MOV    #WRFROM+<54.*2>, R1 ;WHERE DATA GOES
9056 030456 012701          000261
9057 030462 000261
9058 030464 010021          15: MOV    RO, (R1)+ ;STORE DATA
9059 030466 006100          RCL    RO ;GET ZERO ONE BIT LEFT
9060 030470 005305          DEC    R5 ;COUNT 16

```

```

9:00:00 030472 001374          BNE      15          :BRANCH IF 15 NOT DONE
9:00:01          :FILL LEFT ROTATING ONE INTO WRFROM+65.*2
9:00:02          CLC
9:00:03          MOV      #1,RO
9:00:04          MOV      RO,(R1)+
9:00:05          RSL      RO
9:00:06          BCC      25
9:00:07          MOV      (SP)+,R1          ;;POP STACK INTO R1
9:00:08
9:00:09          :FILL REST OF DATA
9:00:10          JSR      RO,2#CLAREA          :CLEAR 174. WORDS, FROM WRFROM+95.*2
9:00:11          WRFROM+(95.*2)          :STARTING FROM WRFROM+ 95.*2
9:00:12          :174. WORDS
9:00:13          :FILL WITH 377
9:00:14
9:00:15          JSR      RO,2#FLHEAD          :SAVE HEADER DATA IN WRFROM+250.*2
9:00:16          WRFROM+(250.*2)          :LOCATION WHERE SAVED
9:00:17          4          :NUMBER OF WORDS SAVED
9:00:18          10005          :FIRST DATA WORD
9:00:19          (<7*400>!5          :SECOND DATA WORD
9:00:20          0          :THIRD DATA WORD
9:00:21          0          :FOURTH DATA WORD
9:00:22          JSR      RO,2#CLAREA          :CLEAR 2 WORDS, FROM WRFROM+(264.*2)
9:00:23          WRFROM+(264.*2)          :STARTING FROM WRFROM+(264.*2)
9:00:24          2          :2 WORDS
9:00:25          (<5*2000>!(<7*40>!5          ;FILL WITH <5*2000>!(<7*40>!5
9:00:26
9:00:27          :READ INTO BUFFER WILL BE CLEARED
9:00:28          JSR      RO,2#CLAREA          :CLEAR 266. WORDS, FROM REINTO
9:00:29          REINTO          :STARTING FROM REINTO
9:00:30          266          :266. WORDS
9:00:31          177400          :FILL WITH 177400
9:00:32
9:00:33          :THE WRITE HEADER AND DATA COMMAND WILL BE LOADED
9:00:34
9:00:35          JSR      RO,2#RUN          :SETUP TO RUN FOR DATA COMMAND
9:00:36          5          :CYLINDER 5
9:00:37          4          :SECTOR 4
9:00:38          7          :TRACK 7
9:00:39          -262.-4          :WORD COUNT (DATA)=262.+
9:00:40          WRFROM          :4 HEADER WORDS
9:00:41          :BUS ADDRESS
9:00:42          :STARTING ADDRESS OF DATA
9:00:43          :BUFFER = WRFROM
9:00:44          0          :DO NOT INHIBIT BUS ADDRESS INCREMENT
9:00:45          FMT22          :16 BITS PER WORD FORMAT
9:00:46          :DO NOT INHIBIT ECC CORRECTION
9:00:47          :DO NOT INHIBIT HEADER COMPARE
9:00:48          WRIFOR          :SET READY TO DO A WRIFOR
9:00:49          :WRITE HEADER AND DATA WITH 62 IN RHCS:

```

000001

25:

034426

034422

034426

034426

036532

.BYTE  
.BYTE

030604 000000  
030606 010000

030610 002356

```

030512 004037 034716 ;NOW SAVE REGISTER FOR COMPARISON AFTER WRITE HEADER AND DATA
030516 002210 JSR RO,0#SAVER ;SAVE REGISTERS
030520 004522 RHWC ;RHWC IS THE FIRST REGISTER SAVED
SAVERE ;STARTING ADDRES OF WHERE
;THE REGISTERS ARE SAVED
;NUMBER OF REGISTERS
;SAVED = 18.
030622 000022 18.
030624 004737 034636 JSR PC,0#CHECKT ;CHECK DVA, RDY, MOL, DPR, DPY, VV
030630 013777 004516 151346 MOV 0#RP4VEC,0#RPEC ;SET RPO4 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
;'TIME' WILL ONLY SAVE
;CURRENT CYLINDER ADDRESS
;AND LOOK AHEAD REGISTERS
030636 013746 002356 MOV 0#WRIFOR, -(SP) ;GET READY TO MOVE COMMAND
030642 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
;ENABLE INTERRUPT
030646 012677 151344 MOV (SP)+,0#RHCS1 ;GO WITH
;E2 IN RHCS1 FOR WRITE HEADER AND DATA
;WITH INTERRUPT ENABLED
030652 104410 WAT ;WAIT FOR RDY BIT TO SET
030654 002216 RHCS1 ;WAIT FOR RHCS1 REGISTER
030656 000200 RDY ;WAIT FOR RDY BIT IN RHCS1 REGISTER
030660 001732 986. ;ALLOW 9860 MICRO SECONDS
030662 001502 834. ;RDY MUST SET BETWEEN
;1520 AND 19200 MICRO SECONDS
030664 004037 034460 ;NOW CHANGE SAVED REGISTERS TO EXPECTED VALUES
030670 002210 JSR RO,0#FILLRE ;MOV 0 INTO SAVED RHWC
RHWC ;SAVED REGISTER TO CHANGE
0 ;DATA
030672 000000 0
030674 004037 034460 JSR RO,0#FILLRE ;MOV WRFROM+(266.*2) INTO SAVED RHBA
030700 002212 RHBA ;SAVED REGISTER TO CHANGE
030702 003424 WRFROM+(266.*2) ;DATA
030704 004037 034460 JSR RO,0#FILLRE ;MOV 3406 INTO SAVED RHDST
030710 002222 RHDST ;SAVED REGISTER TO CHANGE
030712 003406 3406 ;DATA
;NOW COMPARE REGISTERS BEFORE WRITE HEADER AND DATA
;WITH REGISTERS AFTER COMMAND
030714 004037 035546 JSR RO,0#COMREG ;COMPARE SAVED REGISTERS WITH
;PRESENT VALUE
030720 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
030722 002266 WC ;TEST DATA STARTING FROM 'RHWC'
030724 000022 18. ;18. REGISTERS TO BE COMPARED
030726 030732 3$ ;RETURN TO 3$ ON ERROR

```

```

9173 030730 030736          4$          ;RETURN TO 4$ ON NO ERROR
9174
9175 030732 104027          3$:      ERROR 27          ;WRITE HEADER AND DATA
9176 030734 000207          RTS      PC          ;CAUSED IMPROPER REGISTER
9177
9178
9179
9180
9181
9182
9183
9184 030736          4$:      ;NOW FILL COMMAND FOR READ
9185
9186 030736 004737 034556    JSR      PC,@CLDISK    ;SET R1-RHCS1, R2-RHCS2
9187
9188
9189
9190
9191 030742 004037 036532    JSR      RO,@RUN      ;SETUP TO RUN FOR DATA COMMAND
9192 030746 000005          S          ;CYLINDER 5
9193 030750          .BYTE 4          ;SECTOR 4
9194 030751          .BYTE 7          ;TRACK 7
9195 030752 177366          -262.-4    ;WORD COUNT (DATA)=262.+
9196
9197 030754 003444          REINTO    ;4 HEADER WORDS
9198
9199
9200 030756 000000          0          ;BUS ADDRESS
9201 030760 014000          ECI!FMT22 ;STARTING ADDRESS OF DATA
9202
9203
9204 030762 002362          REFOR     ;BUFFER = REINTO
9205
9206
9207
9208
9209 030764 004037 034716    JSR      RO,@SAVER    ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9210 030770 002210          RHWC      ;16 BITS PER WORD FORMAT
9211 030772 004522          SAVER     ;INHIBIT ECC CORRECTION
9212
9213
9214 030774 000022          18.      ;DO NOT INHIBIT HEADER COMPARE
9215
9216
9217 030776 004737 034636    JSR      PC,@CHECKT   ;GET READY TO DO A REFOR
9218
9219
9220 031002 013777 004516 151174 MOV      @RPHVEC,@RPVEC ;READ HEADER AND DATA WITH 72 IN RHCS1
9221
9222
9223
9224
9225
9226
9227 031010 013746 002362    MOV      @REFOR,-(SP) ;NOW SAVE REGISTERS FOR COMPARISON AFTER READ HEADER AND DATA
9228 031014 052716 000101    BIS      @GO!IE,(SP) ;SAVE REGISTERS

```



```

9230 031020 012577 151172      MOV      (SP)+, @RHCSI      ;ENABLE INTERRUPT
9231                                     ;GO WITH
9232                                     ;72 IN RHCSI FOR READ DATA
9233                                     ;WITH INTERRUPT ENABLED
9234
9235 031024 104410      WAT                                     ;WAIT FOR RDY BIT TO SET
9236 031026 002216      RHCSI                                ;WAIT FOR RHCSI REGISTER
9237 031030 000200      RDY                                  ;WAIT FOR RDY BIT IN RHCSI REGISTER
9238 031032 001732      986.                                ;ALLOW 9860 MICRO SECONDS
9239 031034 001502      834.                                ;RDY MUST SET BETWEEN
9240                                     ;1520 AND 18200 MICRO SECONDS
9241
9242                                     ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
9243 031036 004037 034460      JSR      RO, @#FILLRE          ;MOV 0 INTO SAVED RHWC
9244 031042 002210      RHWC                                ;SAVED REGISTER TO CHANGE
9245 031044 000000      0                                   ;DATA
9246 031046 004037 034460      JSR      RO, @#FILLRE          ;MOV REINTO+<266.*2> INTO SAVED RHBA
9247 031052 002212      RHBA                                ;SAVED REGISTER TO CHANGE
9248 031054 004470      REINTO+<266.*2>                 ;DATA
9249 031056 004037 034460      JSR      RO, @#FILLRE          ;MOV 3406 INTO SAVED RHDST
9250 031062 002222      RHDST                              ;SAVED REGISTER TO CHANGE
9251 031064 003406      3406                              ;DATA
9252
9253                                     ;COMPARE REGISTERS BEFORE READ HEADER AND DATA
9254                                     ;WITH REGISTERS AFTER COMMAND
9255
9256 031066 004037 035546      JSR      RO, @#COMREG          ;COMPARE SAVED REGISTERS WITH
9257                                     ;PRESENT VALUE
9258 031072 004522      SAVERE                             ;GOOD DATA SAVED IN 'SAVERE'
9259 031074 002266      WC                                 ;TEST DATA STARTING FROM 'RHWC'
9260 031076 000022      18.                                ;18. REGISTERS TO BE COMPARED
9261 031100 031104      5$                                  ;RETURN TO 5$ ON ERROR
9262 031102 031110      6$                                  ;RETURN TO 6$ ON NO ERROR
9263
9264
9265 031104 104031      5$:  ERROR  31                      ;READ HEADER AND DATA CAUSED
9266 031106 000207      RTS    PC                          ;IMPROPER REGISTER CHANGE
9267                                     ;GOOD DATA GIVES WHAT SHOULD
9268                                     ;BE THERE RECEIVED DATA GIVES WHAT WAS
9269                                     ;RECEIVED DATA GIVES WHAT WAS
9270                                     ;THERE AFTER COMMAND
9271
9272                                     ;NOW READ INTO BUFFER WILL BE . LOCKED TO SEE
9273                                     ;THAT READ WAS GOOD
9274 031110      6$:
9275
9276 031110 004037 036576      JSR      RO, @#COMPAR          ;COMPARE TWO BLOCKS OF MEMORY
9277 031114 002400      WRFROM                             ;GOOD DATA STARTS FROM WRFROM
9278 031116 003444      REINTO                             ;TEST DATA STARTS FROM REINTO
9279 031120 000412      266.                               ;266. WORDS TO BE COMPARED
9280 031122 031126      7$                                  ;RETURN TO 7$ ON ERROR
9281 031124 031132      10$                               ;RETURN TO 10$ ON NO ERROR
9282
9283
9284

```

```

9285 031126 104032      7S:  ERROR 32      ;WRITE HEADER AND DATA
9286 031130 000207      RTS      PC      ;FOLLOWED BY A READ HEADER
9287                                ;AND DATA GAVE A READ ERROR
9288                                ;ERROR MAY BE IN READ OR WRITE
9289
9290                                ;A WRITE, READ HAS BEEN SUCCESSFULLY COMPLETED
9291                                ;NOW A WRITE CHECK HEADER AND DATA WILL BE GIVEN
9292                                ;FILL THE WRITE CHECK HEADER AND DATA
9293 031132      10S:
9294
9295 031132 004737 034556  JSR      PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2
9296                                ;R3-RHDS1, R4-RHER1
9297                                ;GIVE RH-11 INITIALIZE
9298                                ;SETUP UNIT NUBER
9299
9300 031136 004037 036532  JSR      RD, @#RUN    ;SETUP TO RUN FOR DATA COMMAND
9301 031142 000005      5      ;CYLINDER 5
9302 031144      004      4      ;SECTOR 4
9303 031145      007      7      ;TRACK 7
9304 031146 177366      -262.-4 ;WORD COUNT (DATA)=262. +
9305                                ;4 HEADER WORDS
9306 031150 002400  WRFROM ;BUS ADDRESS
9307                                ;STARTING ADDRESS OF DATA
9308                                ;BUFFER = WRFROM
9309 031152 000000      0      ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9310 031154 014000  ECI!FMT22 ;16 BITS PER WORD FORMAT
9311                                ;INHIBIT ECC CORRECTION
9312                                ;DO NOT INHIBIT HEADER COMPARE
9313 031156 002352  WRCHDT ;GET READY TO DO A WRCHDT
9314                                ;WRITE CHECK HEADER AND DATA WITH 52 IN RHCS1
9315
9316
9317                                ;SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
9318 031160      ST25:
9319 031160 004037 034716  JSR      RD, @#SAVER ;SAVE REGISTERS
9320 031164 002210  RHWC    ;RHWC IS THE FIRST REGISTER SAVED
9321 031166 004522  SAVERE ;STARTING ADDRESS OF WHERE
9322                                ;THE REGISTERS ARE SAVED
9323 031170 000022      18.    ;NUMBER OF REGISTERS
9324                                ;SAVED = 18.
9325
9326 031172 004737 034636  JSR      PC, @#CHECKT ;CHECK DVA, RDY, MOL, DPR, DRY, VV
9327
9328
9329 031176 013777 004516 151000 MOV      @#RP4VEC, @RPVEC ;SET RP04 VECTOR ADDRESS
9330                                ;TO 'TIME1' IF P-CLOCK IS PRESENT
9331                                ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9332                                ;'TIME' WILL ONLY SAVE
9333                                ;CURRENT CYLINDER ADDRESS
9334                                ;AND LOOK AHEAD REGISTERS.
9335
9336
9337 031204 013746 002352  MOV      @#WRCHDT, -(SP) ;GET READY TO MOVE COMMAND
9338 031210 052716 000101  BIS      #GO!IE, (SP)  ;GET READY TO SET GO AND
9339                                ;ENABLE INTERRUPT
9340 031214 012677 150776  MOV      (SP)+, @RHCS1 ;GO WITH

```

```

9341                                     ;52 IN RHCSI FOR WRITE CHECK HEADER AND DATA
9342                                     ;WITH INTERRUPT ENABLED
9343 031220 011100 MOV R1,R0 ;SAVE RHCSI DURING ABOVE OPERATION
9344 031222 011305 MOV R3,R5 ;SAVE RHDS1 DURING ABOVE OPERATION
9345
9346
9347 031224 104410 WAT ;WAIT FOR RDY BIT TO SET
9348 031226 002216 RHCS1 ;WAIT FOR RHCSI REGISTER
9349 031230 000200 RDY ;WAIT FOR RDY BIT IN RHCSI REGISTER
9350 031232 001732 986. ;ALLOW 9860 MICRO SECONDS
9351 031234 001502 834. ;RDY MUST SET BETWEEN
9352                                     ;1520 AND 18200 MICRO SECONDS
9353                                     ;COMPARE CONTENTS OF RHCSI AND RHDS1 ALREADY SAVED IN
9354                                     ;R0 AND R5 IMMEDIATELY AFTER GO
9355 031236 013746 002352 MOV R2,WRCHDT,-(SP) ;SAVE COMMAND
9356 031242 052716 004101 BIS IE!DVA!GO,(SP) ;INCLUDE IE!DVA!GO
9357 031246 011637 001124 MOV (SP),R2,$GDDAT ;SAVE FOR PRINTOUT
9358 031252 022600 CMP (SP)+,R0 ;DURING ABOVE OPERATION ONLY IE!DVA!GO
9359                                     ;AND COMMAND SHOULD BE SET
9360 031254 001405 BEQ 64$ ;BRANCH IF GOOD
9361 031256 010037 001126 MOV R0,$BDDAT ;BAD DATA
9362 031262 010137 004510 MOV R1,$REGADR ;FAILING REGISTER RHCSI
9363 031266 104021 ERROR 21 ;DURING ABOVE OPERATION ONLY
9364                                     ;COMMAND AND IE!DVA!GO SHOULD BE SET
9365 031270 012746 010503 64$: MOV $MOL!DPR!VV,-(SP) ;SAVE BITS SET DURING OPERATION IN RHDS1
9366 031274 011637 001124 MOV (SP),R2,$GDDAT ;SAVE FOR PRINTOUT
9367 031300 022605 CMP (SP)+,R5 ;DURING ABOVE OPERATION ONLY MOL!DPR!VV
9368                                     ;SHOULD BE SET
9369 031302 001405 BEQ 66$ ;BRANCH IF GOOD
9370 031304 010537 001126 MOV R5,$BDDAT ;BAD DATA
9371 031310 010337 004510 MOV R3,$REGADR ;FAILING REGISTER RHDS1
9372 031314 104063 ERROR 63 ;DURING ABOVE OPERATION ONLY
9373                                     ;MOL!DPR!VV SHOULD BE SET
9374 031316 66$:
9375
9376                                     ;CHANGE SAVED REGISTERS TO EXPECTED VALUES
9377 031316 004037 034460 JSR R0,$FILLRE ;MOV 0 INTO SAVED RHWC
9378 031322 002210 RHWC ;SAVED REGISTER TO CHANGE
9379 031324 000000 0 ;DATA
9380 031326 004037 034460 JSR R0,$FILLRE ;MOV WRFROM+(266.*2) INTO SAVED RHBA
9381 031332 002212 RHBA ;SAVED REGISTER TO CHANGE
9382 031334 003424 WRFROM+(266.*2) ;DATA
9383 031336 004037 034460 JSR R0,$FILLRE ;MOV 3406 INTO SAVED RHDST
9384 031342 002222 RHDST ;SAVED REGISTER TO CHANGE
9385 031344 003406 3406 ;DATA
9386
9387                                     ;COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA
9388                                     ;WITH REGISTERS AFTER COMMAND
9389
9390 031346 004037 035546 JSR R0,$COMREG ;COMPARE SAVED REGISTERS WITH
9391                                     ;PRESENT VALUE
9392 031352 004522 SAVERE ;GOOD DATA SAVED IN 'SAVERE'
9393 031354 002266 WC ;TEST DATA STARTING FROM 'RHWC'
9394 031356 000022 18. ;18. REGISTERS TO BE COMPARED
9395 031360 031364 8$ ;RETURN TO 8$ ON ERROR
9396 031362 031370 9$ ;RETURN TO 9$ ON NO ERROR

```

9397  
9398 031364 104040  
9399 031356 000207

9\$: ERROR 40  
RTS PC

:WRITE CHECK CAUSED  
:AN IMPROPER REGISTER  
:CHANGE  
:GOOD DATA GIVES WHAT  
:SHOULD BE THERE  
:RECEIVED DATA GIVES WHAT  
:WAS THERE AFTER COMMAND

9400  
9401  
9402  
9403  
9404  
9405  
9406  
9407 031370

9\$:

\*\*\*\*\*  
:TEST 35 WRITE CHECK DATA  
: THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST  
: WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS  
: TRACK 7, SECTOR 4, KEYS 0. NUMBER OF WORDS 258  
: CONSISTING OF  
: 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)  
: 10 WORDS OF 177777  
: 10 WORDS OF 0  
: 10 WORDS OF 052525  
: 10 WORD OF 125252  
: 16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)  
: 16 WORDS OF LEFT ROTATING ONE (EG 1.2.4.10)  
: 174 WORDS OF 377  
: 2 WORDS OF 12345

9408  
9409  
9410  
9411  
9412  
9413  
9414  
9415  
9416  
9417  
9418  
9419  
9420  
9421  
9422  
9423  
9424  
9425  
9426  
9427  
9428

FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER  
THEN THE ABOVE WRITE CHECK DATA IS GIVEN

9429 031370 000004  
9430 031372 012706 001000  
9431 031376 012737 000035 004514

\*\*\*\*\*  
†ST35: SCOPE

MOV #STACK, SP ;RESET STACK  
MOV #35, @†STNM ;SAVE TEST NUMBER

9432 031404 004737 034556  
9433  
9434  
9435  
9436  
9437  
9438  
9439

JSR PC, @#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUMBER

;GET HEADS TO CYLINDER 5

9440 031410 004037 034526  
9441 031414 000005  
9442 031416 013777 004516 150560

JSR R0, @#SEEKCY ;SEEK FOR  
5 ;CYLINDER 5  
MOV @#RP4VEC, @RPVEC ;SET RPO4 VECTOR ADDRESS  
;TO 'TIME1' IF P-CLOCK IS PRESENT  
;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT  
;'TIME' WILL ONLY SAVE  
;CURRENT CYLINDER ADDRESS  
;AND LOOK AHEAD REGISTERS

9443  
9444  
9445  
9446  
9447  
9448  
9449 031424 013746 002364  
9450 031430 052716 000101  
9451  
9452 031434 012677 150556

MOV @#SEECOM, -(SP) ;GET READY TO MOVE COMMAND  
BIS #GO!IE, (SP) ;GET READY TO SET GO AND  
;ENABLE INTERRUPT  
MOV (SP)+, @RHCS1 ;GO WITH

```

9453                                     ;4 IN RHCS1 FOR SEEK
9454                                     ;WITH INTERRUPT ENABLED
9455
9456 031440 104410      WAT                ;WAIT FOR DRY BIT TO SET
9457 031442 002240      RHDS1             ;WAIT FOR RHDS1 REGISTER
9458 031444 000200      DRY                ;WAIT FOR DRY BIT IN RHDS1 REGISTER
9459 031446 004704      2500.             ;ALLOW 25000 MICRO SECONDS
9460 031450 004704      2500.             ;DRY MUST SET BETWEEN
9461                                     ;00 AND 50000 MICRO SECONDS
9462
9463 031452 004737 034556 JSR      PC,2#CLDISK ;SET R1-RHCS1, R2-RHCS2
9464                                     ;R3-RHDS1, R4-RHER1
9465                                     ;GIVE RH-11 INITIALIZE
9466                                     ;SETUP UNIT NUBER
9467
9468 031456 004037 034426 JSR      RO,2#CLAREA ;10 WORDS OF EACH 12344,17777 0 52525 125252
9469 031462 002400      WRFROM           ;CLEAR 10. WORDS, FROM WRFROM
9470 031464 000012      10.              ;STARTING FROM WRFROM
9471 031466 012344      <5*2000>!<7*40>!4 ;10. WORDS
9472                                     ;FILL WITH \5*2000>!<7*40>!4
9473 031470 004037 034426 JSR      RO,2#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<10.*2>
9474 031474 002424      WRFROM+<10.*2> ;STARTING FROM WRFROM+<10.*2>
9475 031476 000012      10.              ;10. WORDS
9476 031500 177777      -1                ;FILL WITH -1
9477
9478 031502 004037 034426 JSR      RO,2#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<20.*2>
9479 031506 002450      WRFROM+<20.*2> ;STARTING FROM WRFROM+<20.*2>
9480 031510 000012      10.              ;10. WORDS
9481 031512 000000      0                ;FILL WITH 0
9482
9483 031514 004037 034426 JSR      RO,2#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<30.*2>
9484 031520 002474      WRFROM+<30.*2> ;STARTING FROM WRFROM+<30.*2>
9485 031522 000012      10.              ;10. WORDS
9486 031524 052525      52525            ;FILL WITH 52525
9487
9488 031526 004037 034426 JSR      RO,2#CLAREA ;CLEAR 10. WORDS, FROM WRFROM+<40.*2>
9489 031532 002520      WRFROM+<40.*2> ;STARTING FROM WRFROM+<40.*2>
9490 031534 000012      10.              ;10. WORDS
9491 031536 125252      125252           ;FILL WITH 125252
9492
9493
9494                                     ;FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
9495 031540 010146      MOV      R1, -(SP) ;PUSH R1 ON STACK
9496 031542 012700 177776 MOV      #177776, R0 ;DATA
9497 031546 012705 000020 MOV      #16, R5 ;COUNT
9498 031552 012701 002544 MOV      #WRFROM+<50.*2>, R1 ;WHERE DATA GOES
9499 031556 000261      SEC
9500 031560 010021 1$: MOV      RO, (R1)+ ;STORE DATA
9501 031562 006100      ROL      RO ;GET ZERO ONE BIT LEFT
9502 031564 005305      DEC      R5 ;COUNT 16
9503 031566 001374      BNE      1$ ;BRANCH IF 16 NOT DONE
9504
9505                                     ;FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
9506 031570 000241      CLC
9507 031572 012700 000001 MOV      #1, R0
9508 031576 010021 2$: MOV      RO, (R1)+

```

2 1/2

```

9509 031600 006300 ASL RO
9510 031602 103375 BCC 2$
9511 031604 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
9512 ;FILL REST OF DATA
9513 031606 004037 034426 JSR RO,@#CLAREA ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
9514 031612 002644 WRFROM+<82.*2> ;STARTING FROM WRFROM+<82.*2>
9515 031614 000256 174. ;174. WORDS
9516 031616 000377 377 ;FILL WITH 377
9517
9518 031620 004037 034426 JSR RO,@#CLAREA ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
9519 031624 003400 WRFROM+<256.*2> ;STARTING FROM WRFROM+<256.*2>
9520 031626 000002 2 ;2 WORDS
9521 031630 012345 <5*2000>!<7*40>!5 ;FILL WITH <5*2000>!<7*40>!5
9522
9523
9524 ;FILL THE WRITE CHECK HEADER AND DATA
9525
9526 031632 004037 036532 JSR RO,@#RUN ;SETUP TO RUN FOR DATA COMMAND
9527 031536 000005 5 ;CYLINDER 5
9528 031640 004 .BYTE 4 ;SECTOR 4
9529 031641 007 .BYTE 7 ;TRACK 7
9530 031642 177400 -256. ;WORD COUNT = 256.
9531 031644 002400 WRFROM ;BUS ADDRESS
9532 ;STARTING ADDRESS OF DATA
9533 ;BUFFER = WRFROM
9534 031646 000000 0 ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9535 031650 014000 ECI!FMT22 ;16 BITS PER WORD FORMAT
9536 ;INHIBIT ECC CORRECTION
9537 ;DO NOT INHIBIT HEADER COMPARE
9538 031652 002350 WRCHK ;GET READY TO DO A WRCHK
9539 ;WRITE CHECK DATA WITH 50 IN RHCS1
9540
9541
9542 ;SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
9543 031654 004037 034716 JSR RO,@#SAVER ;SAVE REGISTERS
9544 031660 002210 RH.1C ;RHW1 IS THE FIRST REGISTER SAVED
9545 031662 004522 SAVERE ;STARTING ADDRESS OF WHERE
9546 ;THE REGISTERS ARE SAVED
9547 031664 000022 18. ;NUMBER OF REGISTERS
9548 ;SAVED = 18.
9549
9550 031666 004737 034636 JSR PC,@#CHECKT ;CHECK DVA,RDY,MOL,DPR,DRY,VV
9551
9552
9553 031672 013777 004516 150304 MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
9554 ;TO 'TIME1' IF P-CLOCK IS PRESENT
9555 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9556 ;'TIME' WILL ONLY SAVE
9557 ;CURRENT CYLINDER ADDRESS
9558 ;AND LOOK AHEAD REGISTERS
9559
9560
9561 031700 013746 002350 MOV @#WRCHK,-(SP) ;GET READY TO MOVE COMMAND
9562 031704 052716 000101 BIS #GO!IE,(SP) ;GET READY TO SET GO AND
9563 ;ENABLE INTERRUPT
9564 031710 012677 150302 MOV (SP)+,@RHCS1 ;GO WITH

```



9621  
9622 032060 104040  
9623 032062 000207  
9624  
9625  
9626  
9627  
9628  
9629  
9630 032064  
9631  
9632  
9633  
9634  
9635  
9636  
9637  
9638  
9639  
9640  
9641  
9642  
9643  
9644  
9645  
9646  
9647  
9648  
9649  
9650  
9651  
9652  
9653  
9654  
9655  
9656 032064 000004  
9657 032066 012706 001000  
9658 032072 012737 000036 004514  
9659  
9660 032100 004737 034556  
9661  
9662  
9663  
9664  
9665  
9666 032104 000167 000474  
9667 032110 005737 004642  
9668 032114 001402  
9669 032116 000137 032604  
9670  
9671  
9672 032122  
9673  
9674 032122 004037 034526  
9675 032126 000005  
9676 032130 013777 004516 150046

8\$: ERROR 40 ;WRITE CHECK CAUSED  
RTS PC ;AN IMPROPER REGISTER  
;CHANGE  
;GOOD DATA GIVES WHAT  
;SHOULD BE THERE  
;RECEIVED DATA GIVES WHAT  
;WAS THERE AFTER COMMANDS

9\$:

\*\*\*\*\*  
\*TEST 36 WRITE CHECK DATA USING UNIBUS B  
\* THIS TEST USES UNIBUS B IF CONNECTED TO THE RH  
\* IF UNIBUS B IS NOT CONNECTED THEN THIS TEST IS NOT PERFORMED  
\* THE DATA FOR THIS TEST IS WRITTEN ON DISK BY PREVIOUS TEST  
\* WRITE CHECK DATA CYLINDER 5, FORMAT 16 BITS PER WORDS  
\* TRACK 7, SECTOR 4, KEYS 0, NUMBER OF WORDS 258  
\* CONSISTING OF  
\* 10 WORDS OF 12344 (6 BITS FOR CYL, 5 FOR TRACK, 5 FOR SECTOR)  
\* 10 WORDS OF 177777  
\* 10 WORDS OF 0  
\* 10 WORDS OF 052525  
\* 10 WORD OF 125252  
\* 16 WORDS OF LEFT ROTATING ZERO (EG177776,177775)  
\* 16 WORDS OF LEFT ROTATING ONE (EG 1,2,4,10)  
\* 174 WORDS OF 377  
\* 2 WORDS OF 12345

\* FIRST THE ABOVE DATA IS FILLED INTO WRITE FROM BUFFER  
\* THEN THE ABOVE WRITE CHECK DATA IS GIVEN

\*\*\*\*\*

TST36: SCOPE  
MOV #STACK,SP ;RESET STACK  
MOV #36,#TSTNM ;SAVE TEST NUMBER  
JSR PC,#CLDISK ;SET R1-RHCS1, R2-RHCS2  
;R3-RHDS1, R4-RHER1  
;GIVE RH-11 INITIALIZE  
;SETUP UNIT NUBER

JMP TST37 ;JUMP TO NEXT TEST FOR RH70  
TST @#UBUSB ;IS UNIBUS B THERE  
BEQ 11\$ ;UNIBUS B THERE SO CONTINUE  
JMP @#9\$ ;NO UNIBUS B JUMP OUT

11\$: ;GET HEADS TO CYLINDER 5

JSR R0,#SEEKCY ;SEEK FOR  
5 ;CYLINDER 5  
MOV @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS



```

9677 ;TO 'TIME1' IF P-CLOCK IS PRESENT
9678 ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
9679 ;'TIME' WILL ONLY SAVE
9680 ;CURRENT CYLINDER ADDRESS
9681 ;AND LOOK AHEAD REGISTERS
9682
9683 032136 013746 002364 MOV @#SEECOM, -(SP) ;GET READY TO MOVE COMMAND
9684 032142 052716 000101 BIS #GO!IE, (SP) ;GET READY TO SET GO AND
9685 ;ENABLE INTERRUPT
9686 032146 012677 150044 MOV (SP)+, @RHCS1 ;GO WITH
9687 ;4 IN RHCS1 FOR SEEK
9688 ;WITH INTERRUPT ENABLED
9689
9690 032152 104410 WAT ;WAIT FOR DRY BIT TO SET
9691 032154 002240 RHDS1 ;WAIT FOR RHDS1 REGISTER
9692 032156 000200 DRY ;WAIT FOR DRY BIT IN RHDS1 REGISTER
9693 032160 004704 2500. ;ALLOW 2500 MICRO SECONDS
9694 032162 004704 2500. ;DRY MUST SET BETWEEN
9695 ;00 AND 50000 MICRO SECONDS
9696
9697 032164 004737 034556 JSR PC, @CLDISK ;SET R1-RHCS1, R2-RHCS2
9698 ;R3-RHDS1, R4-RHER1
9699 ;GIVE RH-11 INITIALIZE
9700 ;SETUP UNIT NUMBER
9701 ;10 WORDS OF EACH 12344, 17777, 0, 52525, 125252
9702 032170 004037 034426 JSR RO, @CLAREA ;CLEAR 10. WORDS FROM WRFROM
9703 032174 002400 WRFROM ;STARTING FROM WRFROM
9704 032176 000012 10. ;10. WORDS
9705 032200 012344 <5*2000>!<7*40>!4 ;FILL WITH <5*2000>!<7*40>!4
9706
9707 032202 004037 034426 JSR RO, @CLAREA ;CLEAR 10. WORDS FROM WRFROM+<10.*2>
9708 032206 002424 WRFROM+<10.*2> ;STARTING FROM WRFROM+<10.*2>
9709 032210 000012 10. ;10. WORDS
9710 032212 177777 -1 ;FILL WITH -1
9711
9712 032214 004037 034426 JSR RO, @CLAREA ;CLEAR 10. WORDS FROM WRFROM+<20.*2>
9713 032220 002450 WRFROM+<20.*2> ;STARTING FROM WRFROM+<20.*2>
9714 032222 000012 10. ;10. WORDS
9715 032224 000000 0 ;FILL WITH 0
9716
9717 032226 004037 034426 JSR RO, @CLAREA ;CLEAR 10. WORDS FROM WRFROM+<30.*2>
9718 032232 002474 WRFROM+<30.*2> ;STARTING FROM WRFROM+<30.*2>
9719 032234 000012 10. ;10. WORDS
9720 032236 052525 52525 ;FILL WITH 52525
9721
9722 032240 004037 034426 JSR RO, @CLAREA ;CLEAR 10. WORDS FROM WRFROM+<40.*2>
9723 032244 002520 WRFROM+<40.*2> ;STARTING FROM WRFROM+<40.*2>
9724 032246 000012 10. ;10. WORDS
9725 032250 125252 125252 ;FILL WITH 125252
9726
9727
9728 ;FILL LEFT ROTATING ZEROS FROM WRFROM+<50.*2>
9729 032252 010146 MOV R1, -(SP) ;PUSH R1 ON STACK
9730 032254 012700 177776 MOV #177776, R0 ;DATA
9731 032260 012705 000020 MOV #16, R5 ;COUNT
9732 032264 012701 002544 MOV #WRFROM+<50.*2>, R1 ;WHERE DATA GOES

```

```

9733 032270 000261          SEC
9734 032272 010021          1$: MOV      RO,(R1)+      ;STORE DATA
9735 032274 006100          ROL      RO              ;GET ZERO ONE BIT LEFT
9736 032276 005305          DEC      R5              ;COUNT 16
9737 032200 001374          BNE     1$              ;BRANCH IF 16 NOT DONE
9738
9739                          ;FILL LEFT ROTATING ONE INTO WRFROM+<65.*2>
9740 032302 000241          CLC
9741 032304 012700 000001    MOV      #1,RO
9742 032310 010021          2$: MOV      RO,(R1)+
9743 032312 006300          ASL      RO
9744 032314 103375          BCC     2$
9745 032316 012601          MOV     (SP)+,R1        ;:POP STACK INTO R1
9746                          ;FILL REST OF DATA
9747 032320 004037 034426    JSR     RO,@#CLAREA    ;CLEAR 174. WORDS, FROM WRFROM+<82.*2>
9748 032324 002644          WRFROM+<82.*2>        ;STARTING FROM WRFROM+<82.*2>
9749 032326 000256          174.
9750 032330 000377          377.                  ;:174. WORDS
9751                          ;FILL WITH 377
9752 032332 004037 034426    JSR     RO,@#CLAREA    ;CLEAR 2 WORDS, FROM WRFROM+<256.*2>
9753 032336 003400          WRFROM+<256.*2>      ;STARTING FROM WRFROM+<256.*2>
9754 032340 000002          2
9755 032342 012345          <5*2000>!<7*40>!5    ;2 WORDS
9756                          ;FILL WITH <5*2000>!<7*40>!5
9757
9758                          ;FILL THE WRITE CHECK HEADER AND DATA
9759
9760 032344 004037 036532    JSR     RO,@#RUN        ;SETUP TO RUN FOR DATA COMMAND
9761 032350 000005          5
9762 032352 004          .BYTE 4
9763 032353 007          .BYTE 7
9764 032354 177400          -256.
9765 032356 002400          WRFROM
9766                          ;CYLINDER 5
9767                          ;SECTOR 4
9768                          ;TRACK 7
9769 032360 000000          0
9770 032362 014000          ECI!FMT22
9771                          ;WORD COUNT = 256.
9772                          ;BUS ADDRESS
9773                          ;STARTING ADDRESS OF DATA
9774                          ;BUFFER = WRFROM
9775                          ;DO NOT INHIBIT BUS ADDRESS INCREMENT
9776                          ;16 BITS PER WORD FORMAT
9777 032364 002350          WRCHK
9778                          ;INHIBIT ECC CORRECTION
9779                          ;DO NOT INHIBIT HEADER COMPARE
9780                          ;GET READY TO DO A WRCHK
9781                          ;WRITE CHECK DATA WITH 50 IN RHCS!
9782
9783 032366 052777 000000 147622  BIS     #PSEL,@RHCS1    ;SET PORT B
9784                          ;THAT IS UNIBUS B
9785
9786                          ;SAVE REGISTERS FOR COMPARISON AFTER WRITE CHECK
9787 032374 004037 034716    JSR     RO,@#SAVER    ;SAVE REGISTERS
9788 032400 002210          RHWC
9789 032402 004522          SAVERE
9790                          ;RHWC IS THE FIRST REGISTER SAVED
9791                          ;STARTING ADDRESS OF WHERE
9792                          ;THE REGISTERS ARE SAVED
9793 032404 000022          18.
9794                          ;NUMBER OF REGISTERS
9795                          ;SAVED = 19.
9796
9797 032406 004737 03463E    JSR     PC,@#CHECKT   ;CHECK DVA.RDY.MOL.DPR.DRY.VV
9798

```

032412 013777 004516 147564  
032420 013746 002350  
032424 052716 004101  
032430 012677 147562  
032434 011100  
032438 011305  
032440 104410  
032442 002216  
032444 000200  
032446 001732  
032450 001502  
032452 013746 002350  
032456 052716 004101  
032462 011637 001124  
032466 022600  
032470 001405  
032472 010337 001126  
032476 010137 004510  
032502 104021  
032504 012746 010500  
032510 011637 001124  
032514 022605  
032516 001405  
032520 010537 001126  
032524 010337 004510  
032530 104063  
032532  
032532 004037 034460  
032536 002210  
032540 000000  
032542 004037 034460  
032546 002212

```
MOV     2#RPP4VEC,3#RPPVE : SET RPP4 VECTOR ADDRESS  
: TO 'TIME1' IF P-CLOCK IS PRESENT  
: OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT  
: 'TIME' WILL ONLY SAVE  
: CURRENT CYLINDER ADDRESS  
: AND LOOK AHEAD REGISTERS  
  
: SET PORT SELECT  
  
MOV     2#RWCHCK-(SP) : GET READY TO MOVE COMMAND  
BIS     2#GO!IE!PSEL,(SP) : GET READY TO SET GO AND  
: ENABLE INTERRUPT  
MOV     (SP)+,2#RHCSI : GO WITH  
: 50 IN RHCSI FOR WRITE CHECK DATA  
: WITH INTERRUPT ENABLED  
MOV     2#R1,R0 : SAVE RHCSI DURING ABOVE OPERATION  
MOV     2#R3,R5 : SAVE RHCSI DURING ABOVE OPERATION  
  
WAT     : WAIT FOR RDY BIT TO SET  
RHCSI : WAIT FOR RHCSI REGISTER  
RDY : WAIT FOR RDY BIT IN RHCSI REGISTER  
986. : ALLOW 9860 MICRO SECONDS  
634. : RDY MUST SET BETWEEN  
: 1520 AND 18200 MICRO SECONDS  
  
: COMPARE CONTENTS OF RHCSI AND RHDSI ALREADY SAVED IN  
: R0 AND R5 IMMEDIATELY AFTER GO  
MOV     2#RWCHCK-(SP) : SAVE COMMAND  
BIS     2#IE!DVA!PSEL!GO,(SP) : INCLUDE IE!DVA!PSEL!GO  
MOV     (SP),2#SGDDAT : SAVE FOR PRINTOUT  
CMP     (SP)+,R0 : DURING ABOVE OPERATION ONLY IE!DVA!PSEL!GO  
: AND COMMAND SHOULD BE SET  
BEG     675 : BRANCH IF GOOD  
MOV     R0,2#SBDDAT : BAD DATA  
MOV     R1,2#REGADR : FAILING REGISTER RHCSI!  
ERROR  21 : DURING ABOVE OPERATION ONLY  
: COMMAND AND IE!DVA!PSEL!GO SHOULD BE SET  
MOV     2#MOL!DPR!VV-(SP) : SAVE BITS SET DURING OPERATION IN RHDSI  
MOV     (SP),2#SGDDAT : SAVE FOR PRINTOUT  
CMP     (SP)+,R5 : DURING ABOVE OPERATION ONLY MOL!DPR!VV  
: SHOULD BE SET  
BEG     698 : BRANCH IF GOOD  
MOV     R5,2#SBDDAT : BAD DATA  
MOV     R3,2#REGADR : FAILING REGISTER RHDSI!  
ERROR  63 : DURING ABOVE OPERATION ONLY  
: MOL!DPR!VV SHOULD BE SET  
  
: CHANGE SAVED REGISTERS TO EXPECTED VALUES  
JSR     R0,2#FILLRE : MOV 0 INTO SAVED RHWC  
RHWC : SAVED REGISTER TO CHANGE  
0 : DATA  
JSR     R0,2#FILLRE : MOV RIFROM+(256.*2) INTO SAVED RHBA  
RHBA : SAVED REGISTER TO CHANGE
```

032412 013777 004516 147564  
032420 013746 002350  
032424 052716 004101  
032430 012677 147562  
032434 011100  
032438 011305  
032440 104410  
032442 002216  
032444 000200  
032446 001732  
032450 001502  
032452 013746 002350  
032456 052716 004101  
032462 011637 001124  
032466 022600  
032470 001405  
032472 010337 001126  
032476 010137 004510  
032502 104021  
032504 012746 010500  
032510 011637 001124  
032514 022605  
032516 001405  
032520 010537 001126  
032524 010337 004510  
032530 104063  
032532  
032532 004037 034460  
032536 002210  
032540 000000  
032542 004037 034460  
032546 002212

032550 003400  
032552 004037 034460  
032556 002222  
032560 003435

WHFROM+ (256.\*2) :DATA  
JSR RO.3#FILLRE :MOV 3405 INTO SAVED RHDST  
RHDST :SAVED REGISTER TO CHANGE  
3405 :DATA

:COMPARE REGISTERS BEFORE WRITE CHECK HEADER AND DATA  
:WITH REGISTER AFTER COMMAND

032562 004037 035546

JSR RO.3#COMREG :COMPARE SAVED REGISTERS WITH  
:PRESENT VALUE

032566 004522  
032570 002266  
032574 003022  
032578 002600  
032582 002604

SAVERE :GOOD DATA SAVED IN 'SAVERE'  
WC :TEST DATA STARTING FROM 'RHW' C  
18. :18. REGISTERS TO BE COMPARED  
99 :RETURN TO 99 ON ERROR  
99 :RETURN TO 99 ON NO ERROR

032600 104076

99: ERROR 76 :WHILE USING UNIBUS B

032602 000207

RTS PC :WRITE CHECK CAUSED  
:AN IMPROPER REGISTER  
:CHANGE  
:GOOD DATA GIVES WHAT  
:SHOULD BE THERE  
:RECEIVED DATA GIVES WHAT  
:WAS THERE AFTER COMMANDS

032604

99:

\*\*\*\*\*  
\*TEST 37 WRITE PROTECT BUTTON  
\* IF STARTING ADDRESS 220 IS USED THIS TEST WILL NOT BE PERFORMED  
\*  
\* IF THE PROGRAM WORKS UNDER ACT-11 MONITOR  
\* THEN THIS TEST IS NOT PERFORMED  
\*  
\* IF NO ACT-11 MONITOR IS PRESENT  
\* THEN THIS TEST IS PERFORMED ONLY ON THE FIRST PASS  
\* ON SUBSEQUENT PASSES THIS TEST IS NOT DONE  
\*  
\* WRITE FROM BUFFER IS FILLED WITH ALL ONES AND  
\* SECTOR 0, TRACK 0, CYLINDER 0 IS FILLED WITH  
\* ALL ONES  
\* ALL REGISTERS ARE SAVED THEN WRITE LOCK BUTTON IS  
\* PRESSED AND ALL REGISTERS ARE CHECKED.  
\* WRITE FROM BUFFER IS FILLED WITH 377 AND A WRITE IS  
\* ATTEMPTED TO SECTOR 0, TRACK 0, CYLINDER 0 70. WORDS  
\* ALL REGISTERS ARE CHECKED  
\* THE SAME SECTOR IS READ AND DATA COMPARED TO SEE  
\* THAT NOTHING GOT DESTROYED (READ DATA SHOULD BE ALL  
\* ONES AND NOT 377)  
\* THEN WRITE LOCK BUTTON IS PRESSED TO UNLOCK  
\* WRITE LOCKS AND ALL REGISTERS ARE COMPARED  
\*

032550  
032552  
032556  
032560  
032562  
032566  
032570  
032574  
032578  
032582  
032600  
032602  
032604

```

9901
9902
9903 032604 000004
9904 032606 012706 001000
9905 032612 012737 000037 004514
9906
9907 032620 004737 034556
9908
9909
9910
9911 032624 005737 004634
9912 032630 001402
9913 032632 000167 001170
9914 032636 005737 000042
9915 032642 001004
9916 032644 005737 001100
9917 032650 001001
9918 032652 000402
9919 032654
9920 032654 000167 001146
9921
9922
9923
9924 032660
9925 032660 004037 034426
9926 032664 002400
9927 032666 000400
9928 032670 177777
9929
9930
9931
9932
9933 032672 004037 036532
9934 032676 000000
9935 032700 000
9936 032701 000
9937 032702 177400
9938 032704 002400
9939
9940
9941 032706 000000
9942 032710 010000
9943
9944
9945 032712 002354
9946
9947
9948
9949 032714 004737 034636
9950
9951
9952 032720 013777 004516 147256
9953
9954
9955
9956

```

```

*****
TST37: SCOPE
MOV #STACK.SP :RESET STACK
MOV #37.2#TSTNM :SAVE TEST NUMBER
JSR PC.2#CLDISK :SET R1-RHCS1, R2-RHCS2
:R3-RHDS1, R4-RHER1
:GIVE RH-11 INITIALIZE
:SETUP UNIT NUBER
TST 2#NOPUSH :IS THIS 220 START
BEQ 9$ :BRANCH IF NO
JMP TST40 :JUMP TO NEXT TEST
9$: TST 2#42 :MONITOR (ACT-11) RETURN ADDRESS
BNE 1$ :BRANCH IF MONITOR PRESENT
TST 2#SPASS :IS THIS FIRST PASS
BNE 1$ :BRANCH IF NOT FIRST PASS
BR 2$ :BRANCH TO CONTINUE TEST
1$: JMP TST40 :JUMP TO NEXT TEST
:FILL SECTOR 0, TRACK 0, CYL 0 WITH ONES
:
: FILL WRITE FROM BUFFER
2$: JSR RC.2#CLAREA :CLEAR 256. WORDS FROM WRFROM
WRFROM :STARTING FROM WRFROM
256. :256. WORDS
-1 :FILL WITH -1
:
:FILL WRITE DATA COMMAND
JSR RC.2#RUN :SETUP TO RUN FOR DATA COMMAND
0 :CYLINDER 0
0 :SECTOR 0
0 :TRACK 0
-256. :WORD COUNT = 256
WRFROM :BUS ADDRESS
:STARTING ADDR OF DATA
:BUFFER = WRF
0 :DO NOT INHI BUS ADDRESS INCREMENT
FMT22 :16 BITS PER WORD FORMAT
:DO NOT INHI ECC CORRECTION
:DO NOT INHI T HEADER COMPARE
WRIDAT :GET READ DO A WRIDAT
:WRITE DATA WITH N RHCS1
JSR PC.2#CHECKT :CHECK -RDY.MOL.DPR.DRY.VV
MOV 2#RP4VEC.3#RVEC :SET R... VECTOR ADDRESS
:TO 'TIME1' IF P-CLOCK IS PRESENT
:OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
:'TIME' WILL ONLY SAVE
:CURRENT CYLINDER ADDRESS

```

```

99957                                     :AND LOOK AHEAD REGISTERS
99958
99959
99960 032726 013746 002354      MOV    @#WRIDAT,-(SP)  ;GET READY TO MOVE COMMAND
99961 032732 052716 000101      BIS    @GO!IE,(SP)   ;GET READY TO SET GO AND
99962                                     ;ENABLE INTERRUPT
99963 032736 012677 147254      MOV    (SP)+,@RHCSI  ;GO WITH
99964                                     ;GO IN RHCSI FOR WRITE DATA
99965                                     ;WITH INTERRUPT ENABLED
99966                                     ;TIME IS NOT CRITICAL
99967
99968 032742 104410      WAT                                     ;WAIT FOR RDY BIT TO SET
99969 032744 002216      RHCSI                                ;WAIT FOR RHCSI REGISTER
99970 032746 000200      RDY                                  ;WAIT FOR RDY BIT IN RHCSI REGISTER
99971 032750 004704      2500.                               ;ALLOW 25000 MICRO SECONDS
99972 032752 004704      2500.                               ;RDY MUST SET BETWEEN
99973                                     ;00 AND 5000 MICRO SECONDS
99974
99975                                     ;SAVE REGISTERS FOR COMPARISON AFTER WRITE PROTECT
99976                                     ;BUTTON HAS BEEN HIT
99977 032754 004037 034716      JSR    RD,@#SAVER   ;SAVE REGISTERS
99978 032760 002210      RHWC                                ;RHWC IS THE FIRST REGISTER SAVED
99979 032762 004522      SAVERE                             ;STARTING ADDRES OF WHERE
99980                                     ;THE REGISTERS ARE SAVED
99981 032764 000022      18.                                ;NUMBER OF REGISTERS
99982                                     ;SAVED = 18.
99983
99984 032766 104400 032774      TYPE                                ;;TYPE ASCIZ STRING
99985 032772 000407      BR    67$                          ;;GET OVER THE ASCIZ
99986                                     ;;.ASCIZ <15><12>/ON DRIVE /
99987
99988 033012 013746 004626      MOV    @#UNIT,-(SP) ;GET UNIT UNDER TEST
99989 033016 104404      TYPDS
99990 033020 104400 033026      TYPE                                ;;TYPE ASCIZ STRING
99991 033024 000442      BR    68$                          ;;GET OVER THE ASCIZ
99992                                     ;;.ASCIZ <15><12>/PUSH WRITE PROTECT BUTTON TO LOCK OUT WRITES THEN HIT CON
99993
99994 033132 000000      68$: HALT
99995                                     ;THE ONLY REGISTER THAT SHOULD CHANGE IS RHDSI - BIT #11
99996                                     ;--WRL
99997
99998 033134 004037 035440      JSR    RD,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
99999 033140 002240      RHDSI                                ;CHANGE RHDSI REGISTER
10000
10001 033142 000001      1                                     ;1 BIT/BITS TO BE CHANGED
10002 033144 000001      1                                     ;NEW VALUE OF WRL IS 1
10003 033146 004000      WRL                                  ;CHANGE WRL BIT
10004
10005                                     ;COMPARE ALL REGISTERS BEFORE WRITE WAS LOCKED
10006                                     ;OUT WITH REGISTER VALUES AFTER WRITE WAS LOCKED OUT
10007
10008 033150 004037 035546      JSR    RD,@#COMREG  ;COMPARE SAVED REGISTERS WITH
10009                                     ;PRESENT VALUE
10010 033154 004522      SAVERE                             ;GOOD DATA SAVED IN 'SAVERE'
10011 033156 002266      WC                                  ;TEST OF "A" STARTING FROM 'RHWC'
10012 033160 000022      18.                                ;18. REGISTERS TO BE COMPARED

```



```

10013 033162 033166          3$          :RETURN TO 3$ ON ERROR
10014 033164 033172          4$          :RETURN TO 4$ ON NO ERROR
10015
10016 033166 104041          3$:      ERROR  41          :LOCKING OUT WRITE BY
10017 033170 000207          RTS      PC              :WRITE LOCK BUTTON CAUSED
10018
10019
10020
10021
10022
10023
10024
10025
10026
10027
10028 033172 013737 033172 001110 4$:      MOV      2*4$,2*$LPERR ;NOW A WRITE WILL BE ATTEMPTED WITH WRITE LOCKED
10029
10030
10031 03320C 004737 034556          JSR      PC,2*CLDISK    ;OUT BY BUTTON
10032
10033
10034
10035 033204 000737 034426          JSR      RD,2*CLAREA   ;FILL WRITE FROM BUFFER WITH 377
10036 033210 000000          WRFROM
10037 033212 000400          256.
10038 033214 000377          377
10039
10040
10041
10042
10043 033216 004037 036532          JSR      RD,2*RUN      ;TRY A ONE WORD WRITE
10044 033222 000000          0
10045 033224 000000          .BYTE   0
10046 033225 000000          .BYTE   0
10047 033226 177777          -1
10048 033230 002400          WRFROM
10049
10050
10051 033232 000000          0
10052 033234 010000          FMT22
10053
10054
10055 033236 002354          WRIDAT
10056
10057
10058
10059 033240 004037 034716          JSR      RD,2*SAVER   ;WRITE DATA WITH 60 IN RHCS1
10060 033244 002210          RHWC
10061 033246 004522          SAVERE
10062
10063 033250 000022          18.
10064
10065
10066 033252 013777 004516 146724      MOV      2*RP4VEC,2*RPVEC ;SAVE REGISTERS
10067
10068

```

```

:SET R1-RHCS1, R2-RHCS2
:R3-RHDS1, R4-RHER1
:GIVE RH-11 INITIALIZE
:SETUP UNIT NUBER
:CLEAR 256. WORDS FROM WRFROM
:STARTING FROM WRFROM
:256. WORDS
:FILL WITH 377
;SCOPE LOOP STARTS FROM HERE
;SETUP TO RUN FOR DATA COMMAND
;CYLINDER 0
;SECTOR 0
;TRACK 0
;WORD COUNT = 1.
;BUS ADDRESS
;STARTING ADDRESS OF DATA
;BUFFER = WRFROM
;DO NOT INHIBIT BUS ADDRESS INCREMENT
;15 BITS PER WORD FORMAT
;DO NOT INHIBIT ECC CORRECTION
;DO NOT INHIBIT HEADER COMPARE
;GET READY TO DO A WRIDAT
;SAVE REGISTERS
;RHWC IS THE FIRST REGISTER SAVED
;STARTING ADDRES OF WHERE
;THE REGISTERS ARE SAVED
;NUMBER OF REGISTERS
;SAVED = 18.
;SET RP04 VECTOR ADDRESS
;TO 'TIME1' IF P-CLOCK IS PRESENT
;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT

```

```

10069                                ; 'TIME' WILL ONLY SAVE
10070                                ; CURRENT CYLINDER ADDRESS
10071                                ; AND LOOK AHEAD REGISTERS
10072
10073
10074 033260 013746 002354  MOV   @#WRIDAT, -(SP) ; GET READY TO MOVE COMMAND
10075 033264 052716 000101  BIS   #GO! IE, (SP)  ; GET READY TO SET GO AND
10076                                ; ENABLE INTERRUPT
10077 033270 012677 146722  MOV   (SP)+, @RHCS1 ; GO WITH
10078                                ; 60 IN RHCS1 FOR WRITE DATA
10079                                ; WITH INTERRUPT ENABLED
10080                                ; TIME IS NOT CRITICAL
10081
10082 033274 104410  WAT   ; WAIT FOR RDY BIT TO SET
10083 033276 002216  RHCS1 ; WAIT FOR RHCS1 REGISTER
10084 033300 000200  RDY   ; WAIT FOR RDY BIT IN RHCS1 REGISTER
10085 033302 004704  2500. ; ALLOW 25000 MICRO SECONDS
10086 033304 004704  2500. ; RDY MUST SET BETWEEN
10087                                ; 00 AND 5000 MICRO SECONDS
10088
10089                                ; CHANGE SAVED REGISTERS TO EXPECTED VALUE
10090 033306 017737 146676 004522  MOV   @RHWC, @#SAVERE ; RHWC IS UNPREDICTABLE
10091 033314 017737 146672 004524  MOV   @RHBA, @#SAVERE+2 ; RHBA IS UNPREDICTABLE
10092 033322 017746 146666  MOV   @PHCS2, -(SP) ; GET RHCS2
10093 033326 042716 177477  BIC   #!C<IR!OR>, (SP) ; KEEP IR AND OR
10094 033332 042737 000300 004526  BIC   #IR!OR, @#SAVERE+4 ; CLEAR SAVED IR OR
10095 033340 052637 004526  BIS   (SP)+, @#SAVERE+4 ; SET OR IR AS REQUIRED
10096
10097 033344 004037 035440  JSR   RD, @#CHREG ; CHANGE BITS IN SAVED REGISTER
10098 033350 002216  RHCS1 ; CHANGE RHCS1 REGISTER
10099
10100 033352 000002  2 ; 2 BIT/BITS TO BE CHANGED
10101 033354 000001  1 ; NEW VALUE OF SC IS 1
10102 033356 100000  SC ; CHANGE SC BIT
10103 033360 000001  1 ; NEW VALUE OF TRE IS 1
10104 033362 040000  TRE ; CHANGE TRE BIT
10105 033364 004037 034460  JSR   RD, @#FILLRE ; MOV 1 INTO SAVED RHDST
10106 033370 002222  RHDST ; SAVED REGISTER TO CHANGE
10107 033372 000001  1 ; DATA
10108
10109 033374 004037 035440  JSR   RD, @#CHREG ; CHANGE BITS IN SAVED REGISTER
10110 033400 002220  RHER1 ; CHANGE RHER1 REGISTER
10111
10112 033402 000001  1 ; 1 BIT/BITS TO BE CHANGED
10113 033404 000001  1 ; NEW VALUE OF WLE IS 1
10114 033406 004000  WLE ; CHANGE WLE BIT
10115
10116 033410 004037 035440  JSR   RD, @#CHREG ; CHANGE BITS IN SAVED REGISTER
10117 033414 002240  RHDS1 ; CHANGE RHDS1 REGISTER
10118
10119 033416 000002  2 ; 2 BIT/BITS TO BE CHANGED
10120 033420 000001  1 ; NEW VALUE OF ATA IS 1
10121 033422 100000  ATA ; CHANGE ATA BIT
10122 033424 000001  1 ; NEW VALUE OF ERR IS 1
10123 033426 040000  ERR ; CHANGE ERR BIT
10124

```





MAINDEC-11-DERPU-B  
DERPUB.P11 T37

MACY11 27(732) 17-SEP-76 13:23 PAGE 203  
WRITE PROTECT BUTTON

```

10181                                     ;BUFFER = REINTO
10182 033524 000000 0                                     ;DO NOT INHIBIT BUS ADDRESS INCREMENT
10183 033526 014000 ECI:FMT22                               ;16 BITS PER WORD FORMAT
10184                                     ;INHIBIT ECC CORRECTION
10185                                     ;DO NOT INHIBIT HEADER COMPARE
10185 033530 002360 READAT                               ;GET READY TO DO A READAT
10187                                     ;READ DATA WITH 70 IN RHCSI
10188
10189
10190 033532 013777 004516 146444 MOV  @#RP4VEC,@RPVEC ;SET RPO4 VECTOR ADDRESS
10191                                     ;TO 'TIME1' IF P-CLOCK IS PRESENT
10192                                     ;OR TO 'TIME2' IF P-CLOCK IS NOT PRESENT
10193                                     ;'TIME' WILL ONLY SAVE
10194                                     ;CURRENT CYLINDER ADDRESS
10195                                     ;AND LOOK AHEAD REGISTERS
10196
10197
10198 033540 013746 002360 MOV  @#READAT,-(SP) ;GET READY TO MOVE COMMAND
10199 033544 052716 000101 BIS   @GO!IE,(SP) ;GET READY TO SET GO AND
10200                                     ;ENABLE INTERRUPT
10201 033550 012677 146442 MOV  (SP)+,@RHCSI ;GO WITH
10202                                     ;70 IN RHCSI FOR READ DATA
10203                                     ;WITH INTERRUPT ENABLED
10204
10205
10206 033554 104410 WAT                                     ;WAIT FOR RDY BIT TO SET
10207 033556 002216 RHCSI                               ;WAIT FOR RHCSI REGISTER
10208 033560 090200 RDY                               ;WAIT FOR RDY BIT IN RHCSI REGISTER
10209 033562 001614 908. ;ALLOW 9080 MICRO SECONDS
10210 033564 001507 839. ;RDY MUST SET BETWEEN
10211                                     ;690 AND 17470 MICRO SECONDS
10212
10213                                     ;COMPARE READ DATA
10214
10215 033566 004037 036576 JSR   RD,@#COMPAR ;COMPARE TWO BLOCKS OF MEMORY
10216 033572 002400 WRFROM ;GOOD DATA STARTS FROM WRFROM
10217 033574 033444 REINTO ;TEST DATA STARTS FROM REINTO
10218 033576 000400 256. ;256. WORDS TO BE COMPARED
10219 033600 033604 7$. ;RETURN TO 7$ ON ERROR
10220 033602 033606 8$. ;RETURN TO 8$ ON NO ERROR
10221
10222
10223 033604 104043 7$: ERROR 43 ;WRITING WITH WRITE
10224                                     ;LOCKED CHANGED DISK
10225                                     ;GOOD DATA GIVES WHAT WAS
10226                                     ;ON DISK BEFORE WRITE WITH
10227                                     ;WRITE LOCK WAS ATTEMPTED
10228
10229                                     ;RECEIVED DATA GIVES WHAT
10230                                     ;WAS READ BACK AFTER WRITE
10231                                     ;WITH WRITE LOCKED WAS ATTEMPTED
10232
10233                                     ;SAVE REGISTERS FOR COMPARISON AFTER WRITE LOCK HAS BEEN
10234                                     ;UNLOCKED
10235 033606 004037 034716 8$: JSR   RD,@#SAVER ;SAVE REGISTERS
10236 033606

```

```

10237 033612 002210          RHWC          ;RHWC IS THE FIRST REGISTER SAVED
10238 033614 004522          SAVERE       ;STARTING ADDRESS OF WHERE
10239                                ;THE REGISTERS ARE SAVED
10240 033616 000022          18.          ;NUMBER OF REGISTERS
10241                                ;SAVED = 18.
10242
10243 033620          ST20:
10244 033620 104400 033626    TYPE          ;:TYPE ASCIZ STRING
10245 033624 000407          BR          64$      ;:GET OVER THE ASCIZ
10246                                ;:ASCIZ <15><12>/ON DRIVE /
10247 033644          64$:
10248 033644 013746 004626    MOV          @#UNIT,-(SP) ;GET UNIT UNDER TEST
10249 033650 104404          TYPDS
10250 033652 104400 033660    TYPE          ;:TYPE ASCIZ STRING
10251 033655 000440          BR          65$      ;:GET OVER THE ASCIZ
10252                                ;:ASCIZ <15><12>/PUSH WRITE PROTCT BUTTON TO UNLOCK WRITES THEN HIT CONTIN
10253 033760          65$:
10254 033760 000000          HALT
10255                                ;THE ONLY BIT THAT SHOULD CHANGE IS WRL-BIT #11 IN RHDS1
10256
10257 033762 004037 035440    JSR          RO,@#CHREG   ;CHANGE BITS IN SAVED REGISTER
10258 033766 002240          RHDS1         ;CHANGE RHDS1 REGISTER
10259
10260 033770 000001          1
10261 033772 000000          0
10262 033774 004000          WRL          ;1 BIT/BITS TO BE CHANGED
10263                                ;NEW VALUE OF WRL IS 0
10264                                ;CHANGE WRL BIT
10265                                ;COMPARE ALL REGISTERS BEFORE WRITE LOCK WAS UNLOCKED
10266                                ;WITH REGISTERS AFTER WRITE WAS UNLOCKED
10267 033776 004037 035546    JSR          RO,@#COMREG   ;COMPARE SAVED REGISTERS WITH
10268                                ;PRESENT VALUE
10269 034002 004522          SAVERE       ;GOOD DATA SAVED IN 'SAVERE'
10270 034004 002266          WC          ;TEST DATA STARTING FROM 'RHWC'
10271 034006 000022          18.          ;18. REGISTERS TO BE COMPARED
10272 034010 034014          9$          ;RETURN TO 9$ ON ERROR
10273 034012 034020          10$         ;RETURN TO 10$ ON NO ERROR
10274
10275 034014 104044          9$:          ERROR      44
10276 034016 000207          RTS        PC
10277                                ;UNLOCKING WRITES BY WRITE
10278                                ;LOCK BUTTON CAUSED AN ERROR
10279                                ;GOOD DATA GIVES WHAT SHOULD
10280                                ;BE THERE
10281                                ;RECEIVED DATA GIVES WHAT WAS
10282                                ;THERE AFTER WRITES WERE
10283                                ;UNLOCKED
10284                                ;ON THIS ERROR NO LOOPING IS RECOMMENDED
10285                                ;JUST A HALT ON ERROR WILL DO THE SAME
10286                                ;THING AS ONLY THE REGISTERS ARE READ
10287 034020 012737 177777 041404 10$: MOV          #-1,@#PRITEM ;CLEAR PREVIOUS ITEM NUMBER
10288
10289
10290
10291
10292
;*****
;*****

```

# K16

MAINDEC-11-DERPU-B  
DERPUB.P11 T40

MACY11 27(732) 17-SEP-76 13:23 PAGE 205  
END OF DRIVE

```

10293          : *TEST 40          END OF DRIVE
10294
10295          : *          THIS IS THE END OF TEST FOR ONE DRIVE
10296          : *          IF THERE ARE MORE DRIVES THEN THE PROGRAM
10297          : *          JUMPS TO TEST 5 FOR NEXT DRIVE TEST
10298          : *          END PASS IS REACHED ONLY AFTER ALL DRIVES ARE COMPLETE
10299
10300          : *****
10301 034026 000004          †ST40: SCOPE
10302 034030 012767 000001 145152          MOV      #1,$TIMES          ;;DO 1 ITERATION
10303 034036 012767 000000 143732          MOV      #0,$PS          ;;REINSTATE PS TO 0
10304 034044 104400 034052          TYPE    +4          ;;TYPE ASCIZ STRING
10305 034050 000425          BR      64$          ;;GET OVER THE ASCIZ
10306          :;.ASCIZ      <15><12>/TOTAL ERRORS ON THIS PASS ON UNIT NO. /
10307 034124          64$:
10308 034124 013746 004626          MOV      @#UNIT,-(SP)          ;GET READY TO TYPE UNIT NUMBER
10309 034130 104404          TYPDS
10310 034132 104400 034140          TYPE    +4          ;;TYPE ASCIZ STRING
10311 034136 000402          BR      65$          ;;GET OVER THE ASCIZ
10312          :;.ASCIZ      /= /
10313 034144          65$:
10314 034144 013746 001112          MOV      @#$ERTTL,-(SP)          ;GET READY TO TYPE NUMBER OF ERRORS
10315 034150 104404          TYPDS
10316 034152 005037 001112          CLR      @#$ERTTL          ;CLEAR TOTAL NUMBER OF ERRORS
10317 034156 005737 004635          TST     @#$SELECT          ;STARTING FROM 200 ?
10318 034162 001415          BEQ     3$          ;BRANCH IF YES
10319 034164 005067 144712          CLR     $TSTNM          ;CLEAR TEST NUMBER
10320 034170 005237 001100          INC     @#$PASS          ;INCREASE PASS COUNT
10321 034174 104400 034362          TYPE    $ENDMG          ;TYPE END PASS #
10322 034200 013746 001100          MOV     @#$PASS,-(SP)
10323 034204 104404          TYPDS
10324 034206 104400 034377          TYPE    $ENULL
10325 034212 000137 007572          JMP     @#TST4          ;JUMP TEST 4
10326 034216 012737 177777 041404 3$:          MOV     #-1,@#PRITEM          ;CLEAR PREVIOUS ITEM NUMBER
10327 034224 005337 004630          DEC     @#NOUNITS          ;NO. OF UNITS PRESENT DECREMENT
10328 034230 001413          BEQ     $EOP          ;BRANCH IF ALL DRIVES COMPLETE
10329 034232 013700 004626          MOV     @#UNIT,R0          ;UNIT UNDER TEST
10330 034236 012701 004606          MOV     #UNITS,R1          ;TABLE
10331 034242 022100          1$:          CMP     (R1)+,R0          ;IS THIS UNIT JUST TESTED
10332 034244 001401          BEQ     2$          ;BRANCH IF YES
10333 034246 000775          BR      1$          ;BRANCH IF NO
10334 034250 011137 004626          2$:          MOV     (R1),@#UNIT          ;THIS IS NEXT UNIT
10335 034254 000137 007572          JMP     @#TST4          ;GO FOR NEXT TESTS.

```

```

10336
10337
10338
10339
10340
10341
10342
10343
10344
10345 034260
10346 034260 000004
10347 034262 005067 144614
10348 034266 005067 144716
10349 034272 005267 144602
10350 034276 042767 100000 144574
10351 034304 005327
10352 034306 000001
10353 034310 003022
10354 034312 012737
10355 034314 000001
10356 034316 034306
10357 034320 104400 034362
10358 034324 016746 144550
10359 034330 104404
10360 034332 104400 034377
10361 034336
10362
10363 034336 013700 000042
10364 034342 001405
10365 034344 000005
10366 034346 004710
10367 034350 000240
10368 034352 000240
10369 034354 000240
10370 034356
10371 034356 000137 006170
10372 034362 005015 047105 020104
10373 034370 040520 051523 021440
10374 034376 000
10375 034377 377 377 000
10376
10377
10378
10379
10380
10391
10382
10383
10384
10385
10386
10387
10388
10389
10390
10391

```

```

;*****
.SBTTL END OF PASS ROUTINE

;*INCREMENT THE PASS NUMBER ($PASS)
;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO TST1

$EOP:
SCOPE
CLR $TSTNM ;;ZERO THE TEST NUMBER
CLR $TIMES ;;ZERO THE NUMBER OF ITERATIONS
INC $PASS ;;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;;YES
MOV (PC)+,2(PC)+ ;;RESTORE COUNTER
$ENDCT: .WORD 1
SCOPE
TYPE $ENDMG ;;TYPE "END PASS #"
MOV $PASS,-(SP) ;;SAVE $PASS FOR TYPEOUT
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE , $ENULL ;;TYPE A NULL CHARACTER

$GET42:
MOV 2#42,RO ;;GET MONITOR ADDRESS
BEQ $DOAGN ;;BRANCH IF NO MONITOR
RESET ;;CLEAR THE WORLD
$ENDAD: JSR PC,(RO) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11
$DOAGN:
JMP 2#TST1 ;;RETURN
$ENDMG: .ASCIZ <15><12>/END PASS #/
$ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING

.SBTTL SUBROUTINES

;THIS FILLS MEMORY WITH GIVEN DATA
;USED CHIEFLY FOR HEADER INFORMATION
;CALL IS
; JSR RO,2#FLHEAD ;FILL HEADER
; LOC ;LOCATION WHERE SAVED

```

```

10392      :          XN          ;NUMBER OF WORDS
10393      :          XD1         ;DATA REPEATED XN TIMES
10394      :          XD2         ;DATA REPEATED XN TIMES
10395      :
10396      :
10397      :
10398      :
10399      034402      FLHEAD:
10400      034402      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
10401      034404      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
10402      034406      012001      MOV      (R0)+,R1     ;R1 HAS ADDRESS OF WHERE TO SAVE
10403      034410      012002      MOV      (R0)+,R2     ;R2 HAS NUMBER OF WORDS
10404
10405      ;NOW FILL DATA
10406
10407      034412      012021      1$:      MOV      (R0)+,(R1)+  ;SAVE DATA
10408      034414      005302      DEC      R2          ;DECREMENT COUNT
10409      034416      001375      BNE     1$          ;BRANCH IF INCOMPLETE
10410      034420      012602      MOV      (SP)+,R2   ;POP STACK INTO R2
10411      034422      012601      MOV      (SP)+,R1   ;POP STACK INTO R1
10412      034424      000200      RTS      R0
10413
10414
10415
10416
10417      ;THIS CLEARS ANY BLOCK OF MEMORY.
10418      ;FILLING IT WITH ANY DATA
10419      ;CALL IS
10420      ;          JSR      R0,#CLAREA
10421      ;          F          ;FROM
10422      ;          N          ;NUMBER OF WORDS
10423      ;          D          ;DATA TO BE FILLED
10424
10425      ;R1 WILL HAVE STARTING ADDRESS OF BLOCK TO BE FILLED
10426      ;R2 WILL HAVE NUMBER OF WORDS
10427      ;R3 WILL HAVE DATA
10428      034426      CLAREA:
10429      034426      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
10430      034430      010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
10431      034432      010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
10432      034434      012001      MOV      (R0)+,R1     ;FROM
10433      034436      012002      MOV      (R0)+,R2     ;NUMBER
10434      034440      012003      MOV      (R0)+,R3     ;DATA
10435      034442      010321      1$:      MOV      R3,(R1)+     ;MOVE DATA
10436      034444      005302      DEC      R2          ;COUNT
10437      034446      001375      BNE     1$          ;BRANCH IF NOT COMPLETE
10438      034450      012603      MOV      (SP)+,R3   ;POP STACK INTO R3
10439      034452      012602      MOV      (SP)+,R2   ;POP STACK INTO R2
10440      034454      012601      MOV      (SP)+,R1   ;POP STACK INTO R1
10441      034456      000200      RTS      R0          ;RETURN TO MAIN PROGRAM
10442
10443
10444
10445
10446
10447

```

```

9265 035544 000200          RTS      RO          ;RETURN TO MAIN PROGRAM
9266
9267
9268
9269
9270
9271
9272          ;THIS IS A SUBROUTINE TO COMPARE REGISTERS
9273          ;GOOD DATA IS ALREADY SAVED IN 'SAVERE'
9274          ;TEST DATA IS IN THE REGISTERS
9275          ;CALL IS
9276          JSR      RO, @COMREG
9277          SAVERE          ;GOOD DATA
9278          RHCSI          ;ADDRESS OF ADDRESS TEST DATA
9279          N.              ;RETURN FOR ERROR
9280          RG              ;RETURN FOR GOOD COMPARISON
9281          ;ON RETURN WITH ERROR '$GDDAT' HAS GOOD DATA, '$BDDAT' HAS BAD DATA
9282          ;'REGADR' HAS REGISTER ADDRESS
9283
9284          COMREG:
9285          MOV      R1, -(SP)          ;: PUSH R1 ON STACK
9286          MOV      R2, -(SP)          ;: PUSH R2 ON STACK
9287          MOV      R3, -(SP)          ;: PUSH R3 ON STACK
9288          MOV      R4, -(SP)          ;: PUSH R4 ON STACK
9289          MOV      R5, -(SP)          ;: PUSH R5 ON STACK
9290          MOV      (RO)+, R1          ;: R1 HAS ADDRESS OF GOOD DATA
9291          MOV      (RO)+, R2          ;: R2 HAS ADDRESS OF ADDRESS OF TEST DATA
9292          MOV      (RO)+, R3          ;: R3 HAS NUMBER OF WORDS
9293          MOV      (RO)+, R4          ;: R4 HAS RETURN FOR ERROR
9294          MOV      (RO), RO          ;: RO HAS RETURN ON NO ERROR
9295          ;NOW SAVE REGISTERS
9296          JSR      PC, @PUTREG          ;: SAVE REGISTERS
9297          MOV      @SAVERE+25, @R5+1 ;: MAKE UPPER BYTE OF R HAS SAME
9298          MOV      # -2, R5          ;: PRESET R5 TO -2
9299          ;NOW COMPARES WILL MADE
9300          ADD      #2, R5          ;: INCREMENT TO INDEX
9301          CMP      (R1)+, (R2)+          ;: COMPARE REGISTER CONTENTS
9302          BEQ      2$          ;: BRANCH IF GOOD
9303          MOV      -(R1), @SGDDAT          ;: SAVE GOOD DATA
9304          MOV      -(R2), @SBDDAT          ;: SAVE BAD DATA
9305          MOV      RHWC(R5), @REGADR          ;: SAVE ADDRESS OF FAILING REGISTER
9306          JSR      PC, @R4          ;: RETURN TO MAIN PROGRAM
9307          ;TO PRINT ERROR
9308          CMP      (R1)+, (R2)+          ;: UNDO -(R1) AND -(R2) FOR ERRORS
9309          MOV      @SWR, -(SP)          ;: GET SWITCH SETTING
9310          BIC      #1C600, (SP)          ;: KEEP ONLY SWITCH 7 AND 8
9311          CMP      #SW07, (SP)+          ;: IS 7 SET AND 8 DOWN
9312          BEQ      3$          ;: BRANCH OUT IF YES
9313          DEC      R3          ;: ARE ALL COMPARES DONE
9314          BNE      1$          ;: BRANCH IF NOT COMPLETE
9315
9316          1$:
9317          MOV      (SP)+, R5          ;: POP STACK INTO R5
9318          MOV      (SP)+, R4          ;: POP STACK INTO R4
9319          MOV      (SP)+, R3          ;: POP STACK INTO R3
9320          MOV      (SP)+, R2          ;: POP STACK INTO R2

```

9321 035674 012601  
9322 035676 000200  
9323 035700 000000

MOV (SP)+,R1  
RTS R0  
4S: .WORD 0

::POP STACK INTO R1  
:RETURN TO MAIN PROGRAM  
:TEMP STORAGE



9324  
9325  
9326  
9327  
9328  
9329  
9330  
9331  
9332  
9333  
9334  
9335  
9336  
9337  
9338  
9339  
9340  
9341  
9342  
9343  
9344  
9345  
9346  
9347  
9348  
9349  
9350  
9351  
9352  
9353  
9354  
9355  
9356  
9357  
9358  
9359  
9360  
9361  
9362  
9363  
9364  
9365  
9366  
9367  
9368  
9369  
9370  
9371  
9372  
9373  
9374  
9375  
9376  
9377  
9378  
9379

035702 000000  
035704  
035704 005067 142066  
035710 012737 177777 041404  
035716 104400 035724  
035722 000421  
035766  
035766 013746 004514  
035772 104401  
035774 104400 036002  
036000 000414  
036032  
036032 013746 001110  
036036 104401  
036040 104400 001221  
036044 104400 036052  
036050 000430  
036132  
036132 104400 036140  
036136 000430  
036220  
036220 104400 036226  
036224 000422  
036272  
036272 104407  
036274 062716 000002  
036300 012637 001106  
036304 104400 036312  
036310 000417

; HERE IS A DETAILED EXPLANATION OF HOW THE LOOP ON ERROR WORKS.  
; ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE  
; PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.

; WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT  
; THE PROGRAM GOES BACK TO CAN BE CHANGED.  
; THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -  
; 1. IT MUST BE WITHIN THE TEST UNDER CONSIDERATION  
; 2. LOOP ON ERROR SWITCH MUST BE SET  
; 3. THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION  
; IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION  
; THE PROGRAM WILL REVERT TO NORMAL OPERATION. HOWEVER, IF LOOP ON  
; TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED  
; THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT  
; COMES TO THE END OF THE TEST UNDER CONSIDERATION.

; AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN  
; NORMAL OPERATION WILL CONTINUE.

TESTAD: 0 ; FIRST ADDRESS OF TEST  
OPERSEL:  
CLR PS ; MAKE PROCESSOR STATUS ZERO  
MOV #1, @PRITEM ; CLEAR PREVIOUS ITEM NUMBER  
TYPE +4 ; TYPE ASCIZ STRING  
BR 64\$ ; GET OVER THE ASCIZ  
;;.ASCIZ <15><12>/THE PROGRAM WAS IN TEST NUMBER /  
64\$:  
MOV @TSTNM, -(SP) ; GET READY TO TYPE TEST  
TYPC ; NUMBER  
TYPE +4 ; TYPE ASCIZ STRING  
BR 65\$ ; GET OVER THE ASCIZ  
;;.ASCIZ <15><12>/THE LOOP BACK PC WAS /  
65\$:  
MOV @SLPERR, -(SP) ; GET READY TO TYPE LOOP BACK PC  
TYPC ; NUMBER  
TYPE , SCRLF ; TYPE ASCIZ STRING  
TYPE +4 ; TYPE ASCIZ STRING  
BR 66\$ ; GET OVER THE ASCIZ  
;;.ASCIZ <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/  
66\$:  
TYPE +4 ; TYPE ASCIZ STRING  
BR 67\$ ; GET OVER THE ASCIZ  
;;.ASCIZ <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/  
67\$:  
TYPE +4 ; TYPE ASCIZ STRING  
BR 68\$ ; GET OVER THE ASCIZ  
;;.ASCIZ <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>  
68\$:  
RDOCT  
ADD #2, (SP) ; GET LPADR  
MOV (SP)+, @SLPADR  
TYPE +4 ; TYPE ASCIZ STRING  
BR 69\$ ; GET OVER THE ASCIZ

```

9380          ::.ASCIZ      <15><12>/TYPE THE PC WHERE YOU WANT/
9381 036350    695:      TYPE          ;;TYPE ASCIZ STRING
9382 036350 104400 036356    BR          705      ;;GET OVER THE ASCIZ
9383 036354 000440          ;;.ASCIZ      <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETUR
9384
9385 036456    705:      RDOCT
9386 036456 104407          MOV      (SP)+,2*SLPERR ;GET LPERR
9387 036460 012637 001110    MOV      2*SLPADR,-(SP)
9388 036464 013746 001106    RTI
9389 036470 000002
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9400
9401
9402
9403
9404 036472          PUTREG:  MOV      RO,-(SP)      ;;PUSH RO ON STACK
9405 036472 010046          MOV      R1,-(SP)      ;;PUSH R1 ON STACK
9406 036474 010146          MOV      R2,-(SP)      ;;PUSH R2 ON STACK
9407 036476 010246          MOV      @RWC,RO      ;;STARTING ADDRESS OF REG
9408 036500 012700 002210    MOV      @WC,R1      ;;STARTING ADDRESS OF WERE SAVED
9409 036504 012701 002266    MOV      @RHC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
9410 036510 012702 000022    105:  MOV      2(RO)+,(R1)+ ;SAVE HARDWARE REG.
9411 036514 013021          DEC      R2
9412 036516 005302          BNE     105
9413 036520 001375          MOV      (SP)+,R2      ;;POP STACK INTO R2
9414 036522 012602          MOV      (SP)+,R1      ;;POP STACK INTO R1
9415 036524 012601          MOV      (SP)+,RO      ;;POP STACK INTO RO
9416 036526 012600          RTS
9417 036530 000207          PC
  
```

```

;THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
;IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
;THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
;AND NOT THE REGISTERS THEMSELVES. THIS WILL MAKE
;ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
  
```

```

9417
9418
9419
9420
9421
9422
9423
9424
9425
9426
9427
9428
9429
9430 036532 012077 143472
9431 036536 012077 143450
9432 036542 012077 143442
9433 036546 012077 143440
9434 036552 013746 004626
9435 036556 052016
9436 036560 012677 143430
9437 036564 012077 143436
9438
9439 036570 013077 143422
9440 036574 000200
9441
9442
9443
9444
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461 036576
9462 036576 010146
9463 036600 010246
9464 036602 010346
9465 036604 010446
9466 036606 010546
9467 036610 012001
9468 036612 012002
9469 036614 012003
9470 036616 012005
9471 036620 011000
9472 036622 010304
  
```

```

: THIS IS A DATA COMMAND SETUP SUBROUTINE
: THE CALL IS
: JSR      RD, @#RUN
:
: BYTE    C      : CYLINDER
: BYTE    S      : SECTOR
:         T      : TRACK
:         -W     : WORD COUNT
:         B      : BUS ADDRESS
:         BAI    : BUS ADDRESS INHIBIT
:         FMT22!ECI!HCI : FMT22=1 =16 BIT WORDS
:                        : ECI = ECC CORRECTION INHIBIT
:                        : HCI = HEADER COMPARE INHIBIT
:
: RUN:     COM
:         MOV    (RO)+, @RHCA : CYLINDER
:         MOV    (RO)+, @RHDS : DESIRED SECTOR/TRACK
:         MOV    (RO)+, @RHWC : WORD COUNT
:         MOV    (RO)+, @RHBA : BUS ADDRESS
:         MOV    @#UNIT, -(SP) : GET UNIT NO
:         BIS    (RO)+, (SP)   : SET BUS ADDRESS INHIBIT
:         MOV    (SP)+, @RHCS2 : UNIT NO AND BAI TO RHCS2
:         MOV    (RO)+, @RHOF : FORMAT, ECC INHIBIT, HEADER
:                        : COMPARE, IF THERE
:         MOV    @#(RO)+, @RHCS1 : COMMAND IN RHCS1
:         RTS    RO           : RETURN TO MAIN PROGRAM
  
```

```

: THIS IS A SUBROUTINE TO COMPARE TWO BLOCKS IN MEMORY
: R1 HAS GOOD DATA BUFFER ADDRESS
: R2 HAS TEST DATA BUFFER ADDRESS
: R5 HAS ADDRESS OF RETURN ON ERROR
: R3 HAS NUMBER OF WORDS TO BE COMPARED
: R4 HAS ONE MORE THAN NUMBER OF WORDS TO BE COMPARED
: CALL IS
: JSR      RO, @#CCMPAR
:
: G        : ADDRESS OF GOOD DATA
: T        : ADDRESS OF TEST DATA
: N        : NUMBER OF WORDS TO BE COMPARED
: RE       : RETURN ON ERROR
: RG       : RETURN ON NO ERROR
  
```

```

COMPAR:
: 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      (RO)+, R1 : ADDRESS OF GOOD DATA BUFFER
: MOV      (RO)+, R2 : ADDRESS OF TEST DATA BUFFER
: MOV      (RO)+, R3 : NO OF WORDS TO BE COMPARED
: MOV      (RO)+, R5 : RETURN ON ERROR
: MOV      (RO), RO  : RETURN ON NO ERROR
: MOV      R3, R4   : NO OF WORDS TO BE COMPARED
  
```

```

9473 036624 005204      INC      R4
9474 036626 010437 004512 15:      MOV      R4, @#ERWORD ; FOR ERROR WORD NO
9475 036632 022122      CMP      (R1)+, (R2)+ ; COMPARE GOOD WITH TEST DATA
9476 036634 001417      BEQ      25 ; BRANCH IF GOOD
9477
9478 036636 014137 001124      MOV      -(R1), @#SGDOAT ; GOOD DATA
9479 036642 014237 001126      MOV      -(R2), @#SBDOAT ; BAD DATA
9480 036646 160337 004512      SUB      R3, @#ERWORD ; ERROR WORD NO.
9481 036652 004715      JSR      PC, @#5 : RETURN TO PRINT ERROR
9482 036654 022122      CMP      (R1)+, (R2)+ ; UNDO -(R1) AND -(R2) FOR ERRORS
9483 036656 013746 001136      MOV      @#SWR, -(SP) ; GET SWITCH SETTING
9484 036662 042716 177177      BIC      #1C600, (SP) ; KEEP ONLY SWITCH 7 AND 8
9485 036666 022726 000200      CMP      @#SW07, (SP)+ ; IS 7 SET AND 8 RESET
9486 036672 001402      BEQ      35 ; BRANCH OUT IF YES
9487 036674 005303 25:      DEC      R3 ; COUNT
9488 036676 001353      BNE      15 ; BRANCH IF ALL NOT DEVICE
9489
9490 036700 012605 35:      MOV      (SP)+, R5 ; POP STACK INTO R5
9491 036702 012604      MOV      (SP)+, R4 ; POP STACK INTO R4
9492 036704 012603      MOV      (SP)+, R3 ; POP STACK INTO R3
9493 036706 012602      MOV      (SP)+, R2 ; POP STACK INTO R2
9494 036710 012601      MOV      (SP)+, R1 ; POP STACK INTO R1
9495 036712 000200      RTS      R0 ; RETURN TO MAIN PROGRAM
9496 ;* THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
9497 ;* ADDRESS FROM 176700 TO ANY TYPED VALUE
9498
9499 036714 BASECH:
9500 036714 104400 036722 TYPE      +4 ; TYPE ASCIZ STRING
9501 036720 000424 BR      645 ; GET OVER THE ASCIZ
9502 ;;.ASCIZ <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS /
9503 036772 645:
9504 036772 013746 002216 MOV      @#RHCS1, -(SP) ; GET READY TO TYPE OLD BASE
9505 036776 104401 TYPOC
9506 037000 104400 037006 TYPE      +4 ; TYPE ASCIZ STRING
9507 037004 000425 BR      655 ; GET OVER THE ASCIZ
9508 ;;.ASCIZ <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR'/
9509 037060 655:
9510 037060 104407 RDOCT
9511 037062 012700 002206 MOV      @#RHDB, R0 ; GET STARTING ADDRESS OF REGISTERS
9512 037066 012701 000024 MOV      #20, R1 ; NUMBER OF REGISTERS
9513 037072 042710 177700 15:      BIC      #1C77, (R0) ; CLEAR OLD BASE
9514 037076 051620 BIS      (SP), (R0)+ ; SET NEW BASE
9515 037100 005301 DEC      R1 ; COUNT
9516 037102 001373 BNE      15 ; BRANCH IF 20 NOT DONE
9517 037104 104400 037112 TYPE      +4 ; TYPE ASCIZ STRING
9518 037110 000417 BR      665 ; GET OVER THE ASCIZ
9519 ;;.ASCIZ <15><12>/PRESENT VECTOR ADDRESS IS /
9520 037150 665:
9521 037150 013746 002204 MOV      @#RPVEC, -(SP) ; GET READY TO TYPE OLD VECTOR ADDRESS
9522 037154 104401 TYPOC
9523 037156 104400 037164 TYPE      +4 ; TYPE ASCIZ STRING
9524 037162 000437 BR      675 ; GET OVER THE ASCIZ
9525 ;;.ASCIZ <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR
9526 037262 675:
9527 037262 104407 RDOCT
9528 037264 012637 002204 MOV      (SP)+, @#RPVEC ; SETUP VECTOR ADDRESS

```

```

9529 037270 104400 037276      TYPE      +4      ;;TYPE ASCIZ STRING
9530 037274 000421      BR      68$      ;;GET OVER THE ASCIZ
9531      ;;.ASCIZ      <15><12>/RESTART PROGRAM FROM 200 OR 210/
9532 037340      68$:
9533 037340 104400 037346      TYPE      +4      ;;TYPE ASCIZ STRING
9534 037344 000414      BR      69$      ;;GET OVER THE ASCIZ
9535      ;;.ASCIZ      <15><12>/NEW BASE WILL REMAIN/
9536 037376      69$:
9537 037376 013746 002216      MOV      @RHC1,-(SP)
9538 037402 104401      TYPOC
9539 037404 104400 037412      TYPE      +4      ;;TYPE ASCIZ STRING
9540 037410 000415      BR      70$      ;;GET OVER THE ASCIZ
9541      ;;.ASCIZ      <15><12>/NEW VECTOR WILL REMAIN /
9542 037444      70$:
9543 037444 013746 002204      MOV      @RPVEC,-(SP)
9544 037450 104401      TYPOC
9545 037452 104400 037460      TYPE      +4      ;;TYPE ASCIZ STRING
9546 037456 000416      BR      71$      ;;GET OVER THE ASCIZ
9547      ;;.ASCIZ      <15><12>/UNTIL PROGRAM IS RELOADED/
9548 037514      71$:
9549 037514 000000      HALT
9550

```

9551	037516			RPVECT:			
9552	037516	104400	037524		TYPE	+4	:::TYPE ASCIZ STRING
9553	037522	000411			BR	64\$	:::GET OVER THE ASCIZ
9554					:::ASCIZ		/TRAPED FROM PC = /
9555	037546			54\$:			
9556	037546	104401			TYPOC		:TYPE FROM PC
9557	037550	012777	037516	142426	MOV	*RPVECT,@RPVEC	:RESTORE TRAP RPO4 VECTOR
9558	037556	000000			HALT		:CHANGE TO CONTINUE
9559							
9560							
9561							
9562							

```

9563 ;*****
9564 ;*****
9565 ;*****
9566 .SBTTL SCOPE HANDLER ROUTINE
9567
9568 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9569 ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
9570 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
9571 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9572 ;*SW14=1 LOOP ON TEST
9573 ;*SW11=1 INHIBIT ITERATIONS
9574 ;*SW09=1 LOOP ON ERROR
9575 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
9576 ;*CALL
9577 ;* SCOPE ;;SCOPE=IOT
9578
9579 037560 $SCOPE:
9580 037560 032777 040000 141350 1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
9581 037566 001111 BNE $OVER ;;YES IF SW14=1
9582 ;*****START OF CODE FOR THE XOR TESTER*****
9583 037570 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
9584 ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
9585 037572 013746 000004 MOV $ERRVEC,-($P) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
9586 037576 012737 037616 000004 MOV $S,$ERRVEC ;;SET FOR TIMEOUT
9587 037604 005737 177060 TST $#177060 ;;TIME OUT ON XOR?
9588 037610 012637 000004 MOV ($P)+,$ERRVEC ;;RESTORE THE ERROR VECTOR
9589 037614 000463 BR $SVLAD ;;GO TO THE NEXT TEST
9590 037616 022626 5$: CMP ($P)+,$(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
9591 037620 012637 000004 MOV ($P)+,$ERRVEC ;;RESUME THE ERROR VECTOR
9592 037624 000423 BR 7$ ;;LOOP ON THE PRESENT TEST
9593 037626 6$;;*****END OF CODE FOR THE XOR TESTER*****
9594 037626 032777 000400 141302 BIT #BIT08,$SWR ;;LOOP ON SPEC. TEST?
9595 037634 001404 BEQ 2$ ;;BR IF NO
9596 037636 127767 141274 141236 CMPB $SWR,$STNM ;;ON THE RIGHT TEST? SWR<7:0>
9597 037644 001462 BEQ $OVER ;;BR IF YES
9598 037646 105767 141231 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
9599 037652 001421 BEQ 3$ ;;BR IF NO
9600 037654 126767 141235 141221 CMPB $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
9601 037662 101015 BHI 3$ ;;BR IF NO
9602 037664 032777 001000 141244 BIT #BIT09,$SWR ;;LOOP ON ERROR?
9603 037672 001404 BEQ 4$ ;;BR IF NO
9604 037674 016767 141210 141204 7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
9605 037702 000443 BR $OVER
9606 037704 105067 141173 4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
9607 037710 005067 141274 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
9608 037714 000415 BR 1$ ;;ESCAPE TO THE NEXT TEST
9609 037716 032777 004000 141212 3$: BIT #BIT11,$SWR ;;INHIBIT ITERATIONS?
9610 037724 001011 BNE 1$ ;;BR IF YES
9611 037726 005767 141146 TST $PASS ;;IF FIRST PASS OF PROGRAM
9612 037732 001406 BEQ 1$ ;; INHIBIT ITERATIONS
9613 037734 005267 141144 INC $ICNT ;;INCREMENT ITERATION COUNT
9614 037740 026767 141244 141136 CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
9615 037746 002021 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
9616 037750 012767 000001 141126 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
9617 037756 016767 000044 141224 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
9618 037764 105267 141112 $SVLAD: INCB $STNM ;;COUNT TEST NUMBERS

```

K01

MAINDEC-11-DERPU-B MACY11 27(732) 30-DEC-76 10:30 PAGE 187  
DERPUB.P11 SCOPE HANDLER ROUTINE

9619	037770	011667	141112		MOV	(SP), \$LPADR	:: SAVE SCOPE LOOP ADDRESS
9620	037774	011667	141110		MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS
9621	040000	005067	141206		CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS
9622	040004	112767	000001	141103	MOVB	#1, \$ERMAX	:: ONLY ALLOW ONE(I) ERROR ON NEXT TEST
9623	040012	016777	141064	141120	SOVER: MOV	\$TSTNM, @DISPLAY	:: DISPLAY TEST NUMBER
9624	040020	016716	141062		MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS
9625	040024	000002			RTI		:: FIXES PS
9626	040026	000004			SMXCNT: 4		:: MAX. NUMBER OF ITERATIONS



```

9627 ;*****
9628
9629 .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
9630
9631 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
9632 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
9633 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
9634 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
9635 ;*REPLACED WITH SPACES.
9636 ;*CALL:
9637 ;*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
9638 ;*      TYPDS                    ;;GO TO THE ROUTINE
9639
9640 STYPDS:
9641 040030 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
9642 040032 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
9643 040034 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
9644 040036 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
9645 040040 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
9646 040042 012746 02020C      MOV      #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
9647 040046 016605 000020      MOV      20(SP),R5    ;;GET THE INPUT NUMBER
9648 040052 100004      BPL      1$           ;;BR IF INPUT IS POS.
9649 040054 005405      NEG      R5           ;;MAKE THE BINARY NUMBER POS.
9650 040056 112766 000055 000001      MOVVB   #'-,1(SP)    ;;MAKE THE ASCII NUMBER NEG.
9651 040064 005000      CLR      R0           ;;ZERO THE CONSTANTS INDEX
9652 040066 012703 040244      MOV      #S0BLK,R3    ;;SETUP THE OUTPUT POINTER
9653 040072 112723 000040      MOVVB   #' ,(R3)+    ;;SET THE FIRST CHARACTER TO A BLANK
9654 040076 005002      CLR      R2           ;;CLEAR THE BCD NUMBER
9655 040100 016001 040234      MOV      $DTBL(R0),R1 ;;GET THE CONSTANT
9656 040104 160105      SJB     R1,R5         ;;FORM THIS BCD DIGIT
9657 040106 002402      BLT     4$           ;;BR IF DONE
9658 040110 005202      INC     R2           ;;INCREASE THE BCD DIGIT BY 1
9659 040112 000774      BR      3$
9660 040114 060105      4$:      ADD     R1,R5         ;;ADD BACK THE CONSTANT
9661 040116 005702      TST     R2           ;;CHECK IF BCD DIGIT=0
9662 040120 001002      BNE     5$           ;;FALL THROUGH IF 0
9663 040122 105716      TSTB   (SP)         ;;STILL DOING LEADING 0'S?
9664 040124 100407      BMI     7$           ;;BR IF YES
9665 040126 106316      5$:      ASLB   (SP)         ;;MSD?
9666 040130 103003      BCC     6$           ;;BR IF NO
9667 040132 116663 000001 177777      MOVVB   1(SP),-1(R3)  ;;YES--SET THE SIGN
9668 040140 052702 000060      6$:      BIS     #'0,R2       ;;MAKE THE BCD DIGIT ASCII
9669 040144 052702 000040      7$:      BIS     #' ,R2       ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
9670 040150 110223      MOVVB   R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
9671 040152 005720      TST     (R0)+       ;;JUST INCREMENTING
9672 040154 020027 000010      CMP     R0,#10      ;;CHECK THE TABLE INDEX
9673 040160 002746      BLT     2$           ;;GO DO THE NEXT DIGIT
9674 040162 003002      BGT     8$           ;;GO TO EXIT
9675 040164 010502      MOV     R5,R2       ;;GET THE LSD
9676 040166 000764      BR      6$           ;;GO CHANGE TO ASCII
9677 040170 105726      8$:      TSTB   (SP)+       ;;WAS THE LSD THE FIRST NON-ZERO?
9678 040172 100003      BPL     9$           ;;BR IF NO
9679 040174 116663 177777 177776      9$:      MOVVB   -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
9680 040202 105013      CLRB   (R3)         ;;SET THE TERMINATOR
9681 040204 012605      MOV     (SP)+,R5    ;;POP STACK INTO R5
9682 040206 012603      MOV     (SP)+,R3    ;;POP STACK INTO R3

```

```

9683 040210 012602          MOV      (SP)+,R2          ;; POP STACK INTO R2
9684 040212 012601          MOV      (SP)+,R1          ;; POP STACK INTO R1
9685 040214 012600          MOV      (SP)+,R0          ;; POP STACK INTO R0
9686 040216 104400 040244  TYPE      $DBLK          ;; NOW TYPE THE NUMBER
9687 040222 016666 000002 000004  MOV      2(SP),4(SP)      ;; ADJUST THE STACK
9688 040230 012616          MOV      (SP)+,(SP)
9689 040232 000002          RTI                          ;; RETURN TO USER
9690 040234 023420          $DTBL: 10000.
9691 040236 001750          1000.
9692 040240 000144          100.
9693 040242 000012          10.
9694 040244 000004          $DBLK: .BLKW 4
9695                                     ;*****
9696                                     .SBTTL  TYPE ROUTINE
9697
9698                                     ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
9699                                     ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
9700                                     ;*NOTE1:          $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
9701                                     ;*NOTE2:          $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
9702                                     ;*NOTE3:          $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
9703                                     ;*
9704                                     ;*CALL:
9705                                     ;*1) USING A TRAP INSTRUCTION
9706                                     ;*      TYPE      ,MESADR          ;; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
9707                                     ;*OR
9708                                     ;*      TYPE
9709                                     ;*      MESADR
9710                                     ;*
9711                                     ;*
9712
9713 040254 105767 140675          $TYPE:  TSTB      $TPFLG          ;; IS THERE A TERMINAL?
9714 040260 100002          BPL      1$                ;; BR IF YES
9715 040262 000000          HALT                    ;; HALT HERE IF NO TERMINAL
9716 040264 000407          SR                          ;; LEAVE
9717 040266 010046          1$:  MOV      RO,-(SP)      ;; SAVE RO
9718 040270 017600 000002          MOV      2(SP),RO        ;; GET ADDRESS OF ASCIZ STRING
9719 040274 112046          2$:  MOV      (RO)+,-(SP)  ;; PUSH CHARACTER TO BE TYPED ONTO STACK
9720 040276 001005          BNE      4$                ;; BR IF IT ISN'T THE TERMINATOR
9721 040300 005726          TST      (SP)+            ;; IF TERMINATOR POP IT OFF THE STACK
9722 040302 012600          60$: MOV      (SP)+,RO        ;; RESTORE RO
9723 040304 062716 000002          3$:  ADD      #2,(SP)        ;; ADJUST RETURN PC
9724 040310 000002          RTI                          ;; RETURN
9725 040312 122716 000011          4$:  CMP      #THT,(SP)     ;; BRANCH IF <HT>
9726 040316 001426          BEQ      8$                ;; BRANCH IF NOT <CRLF>
9727 040320 122716 000200          CMP      #TCRLF,(SP)
9728 040324 001004          BNE      5$                ;; POP <CR><LF> EQUIV
9729 040326 005726          TST      (SP)+            ;; TYPE A CR AND LF
9730 040330 104400          TYPE
9731 040332 001221          $CRLF
9732 040334 000757          BR      2$                ;; GET NEXT CHARACTER
9733 040336 004767 000056          5$:  JSR      PC,$TYPEC      ;; GO TYPE THIS CHARACTER
9734 040342 126726 140606          6$:  CMP      $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
9735 040346 001352          BNE      2$                ;; IF NO GO GET NEXT CHAR.
9736 040350 016746 140576          MOV      $NULL,-(SP)     ;; GET # OF FILLER CHARS. NEEDED
9737                                     ;; AND THE NULL CHAR.
9738 040354 105366 000001          7$:  DECB      1(SP)        ;; DOES A NULL NEED TO BE TYPED?

```

```

9739 040360 002770          BLT      6$          ;; BR IF NO--GO POP THE NULL OFF OF STACK
9740 040362 004767- 000032  JSR      PC,$TYPEC  ;; GO TYPE A NULL
9741 040366 105367 000072  DECB    $CHARCNT    ;; DO NOT COUNT AS A COUNT
9742 040372 000770          BR       7$          ;; LOOP

```

; HORIZONTAL TAB PROCESSOR

```

9746 040374 112716 000040  8$:     MOVB    #40,(SP)  ;; REPLACE TAB WITH SPACE
9747 040400 004767 000014  9$:     JSR      PC,$TYPEC  ;; TYPE A SPACE
9748 040404 132767 000007 000052  BITB    #7,$CHARCNT    ;; BRANCH IF NOT AT
9749 040412 001372          BNE     9$            ;; TAB STOP
9750 040414 005726          TST     (SP)+         ;; POP SPACE OFF STACK
9751 040416 000726          BR       2$            ;; GET NEXT CHARACTER
9752 040420 105777 140522  $TYPEC: TSTB    @STPS     ;; WAIT UNTIL PRINTER IS READY
9753 040424 100375          BPL     $TYPEC
9754 040426 116677 000002 140514  MOVB    2(SP),@STPB    ;; LOAD CHAR TO BE TYPED INTO DATA REG.
9755 040434 122766 000015 000002  CMPB    #15,2(SP)     ;; BRANCH IF
9756 040442 001003          BNE     1$            ;; NOT <CR>
9757 040444 105067 000014          CLRB   $CHARCNT
9758 040450 000406          BR      $TYPEX
9759 040452 122766 000012 000002  1$:     CMPB    #12,2(SP)  ;; BRANCH IF
9760 040460 002002          BGE     $TYPEX       ;; <LF>
9761 040462 105227          INCB   (PC)+         ;; INC SPACE
9762 040464 000000  $CHARCNT: .WORD 0    ;; COUNT
9763 040466 000207  $TYPEX:  RTS      PC
9764          ;; EQUATES
9765          THT=11
9766          TCRLF=200

```

;\*\*\*\*\*

.SBTTL TTY INPUT ROUTINE

```

9772 040470 000000  $TKCNT: .WORD 0    ;; NUMBER OF ITEMS IN QUEUE
9773 040472 000000  $TKQIN: .WORD 0    ;; INPUT POINTER
9774 040474 000000  $TKQOUT: .WORD 0   ;; OUTPUT POINTER
9775 040476 000011  $TKQSRT: .BLKB 9.  ;; TTY KEYBOARD QUEUE
9776          $TKQEND=.
9777          .EVEN

```

```

; *TK INITIALIZE ROUTINE
; *THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
; *SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT

```

```

; *CALL
; * JSR      PC,$TKINT
; * RETURN

```

```

9787 040510 005067 177754  $TKINT: CLR     $TKCNT    ;; CLEAR COUNT OF ITEMS IN QUEUE
9788 040514 012767 040476 177750  MOV     # $TKQSRT,$TKQIN  ;; MOVE THE STARTING ADDRESS OF THE
9789 040522 016767 177744 177744  MOV     $TKQIN,$TKQOUT    ;; QUEUE INTO THE INPUT & OUTPUT POINTERS.
9790 040530 012737 040560 000060  MOV     # $TKQSRT,@ $TKVEC  ;; INITIALIZE THE KEYBOARD VECTOR
9791 040536 012737 000200 000062  MOV     #200,@ $TKVEC+2    ;; "BR" LEVEL 4
9792 040544 005777 140374          TST     @ $TKB           ;; CLEAR DONE FLAG
9793 040550 012777 000100 140364  MOV     #BIT06,@ $TKS     ;; ENABLE INTERRUPT
9794 040556 000207          RTS      PC           ;; RETURN TO CALLER

```

```

9795
9796 ;*TK SERVICE ROUTINE
9797 ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
9798
9799 040560 117746 140360 $TKSRV: MOVB 2$TKB,-(SP) ;; PICKUP THE CHARACTER
9800 040564 042716 177600 BIC #1C17,(SP) ;; STRIP THE JUNK
9801 040570 021627 000003 CMP (SP),#3 ;; IS IT A CONTROL C?
9802 040574 001006 BNE 1$ ;; BRANCH IF NO
9803 040576 104400 041075 TYPE $CNTLC ;; TYPE A CONTROL-C (1C)
9804 040602 004767 177702 JSR PC,$TKINT ;; INIT THE KEYBOARD
9805 040606 000167 175072 JMP OPERSEL ;; CONTROL C RESTART
9806 040612 022767 000011 177650 1$: CMP #9,$TKCNT ;; IS THE QUEUE FULL?
9807 040620 001004 BNE 2$ ;; BRANCH IF NO
9808 040622 104400 001214 TYPE $BELL ;; RING THE TTY BELL
9809 040626 005726 TST (SP)+ ;; CLEAN CHARACTER OFF OF STACK
9810 040630 000002 RTI ;; RETURN
9811 040632 005267 177632 2$: INC $TKCNT ;; COUNT THIS CHARACTER
9812 040636 112677 177630 MOVB (SP)+,2$TKQIN ;; AND PUT IT IN QUEUE
9813 040642 005267 177624 INC $TKQIN ;; UPDATE THE POINTER
9814 040646 026727 177620 040507 CMP $TKQIN,$$TKQEND ;; GO OFF THE END?
9815 040654 001003 BNE 3$ ;; BRANCH IF NO
9816 040656 012767 040476 177606 MOV $$TKQSR, $TKQIN ;; RESET THE POINTER
9817 040664 000002 3$: RTI ;; RETURN
9818 *****
9819 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
9820 ;*CALL:
9821 ;* RDOCHR ;; INPUT A SINGLE CHARACTER FROM THE TTY
9822 ;* RETURN HERE ;; CHARACTER IS ON THE STACK
9823
9824
9825 040666 011646 $RDOCHR: MOV (SP),-(SP) ;; PUSH DOWN THE PC AND
9826 040670 016666 000004 000002 MOV 4(SP),2(SP) ;; THE PS
9827 040676 005066 000004 CLR 4(SP) ;; GET READY FOR A CHARACTER
9828 040702 005046 CLR -(SP) ;; SETUP TO LOAD A NEW
9829 040704 012746 040712 MOV #1$,-(SP) ;; PS WORD THAT
9830 040710 000002 RTI ;; WILL ALLOW INTERRUPTS
9831 040712 005767 177552 1$: TST $TKCNT ;; WAIT ON A CHARACTER
9832 040716 001775 BEQ 1$
9833 040720 005367 177544 DEC $TKCNT ;; DECREMENT THE COUNTER
9834 040724 117766 177544 000004 MOVB 2$TKQOUT,4(SP) ;; GET ONE CHARACTER
9835 040732 005267 177536 INC $TKQOUT ;; UPDATE THE POINTER
9836 040736 026727 177532 040507 CMP $TKQOUT,$$TKQEND ;; DID IT GO OFF OF THE END?
9837 040744 001003 BNE 2$ ;; BRANCH IF NO
9838 040746 012767 040476 177520 MOV $$TKQSR,$TKQOUT ;; RESET THE POINTER
9839 040754 000002 2$: RTI ;; RETURN
9840 *****
9841 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
9842 ;*CALL:
9843 ;* RDLIN ;; INPUT A STRING FROM THE TTY
9844 ;* RETURN HERE ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
9845 ;* ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
9846
9847 040756 010346 $RDLIN: MOV R3, -(SP) ;; SAVE R3
9848 040760 012703 041064 1$: MOV $$TTYIN,R3 ;; GET ADDRESS
9849 040764 022703 041075 2$: CMP $$TTYIN+9.,R3 ;; BUFFER FULL?
9850 040770 101405 BLOS 4$ ;; BR IF YES

```

```

9851 040772 104405          RDCHR          ; GO READ ONE CHARACTER FROM THE TTY
9852 040774 112613          MOVW          (SP)+,(R3) ; GET CHARACTER
9853 040776 122713 000177    CMPB          #'177,(R3) ; IS IT A RUBOUT
9854 041002 001003          BNE          3$        ; SKIP IF NOT
9855 041004 104400 001220    4$: TYPE      $QUES    ; TYPE A '?'
9856 041010 000763          BR           1$        ; CLEAR THE BUFFER AND LOOP
9857 041012 111367 000044    3$: MOVW      (R3),9$   ; ECHO THE CHARACTER
9858 041016 104400 041062    TYPE          9$
9859 041022 122723 000015    CMPB          #'15,(R3)+ ; CHECK FOR RETURN
9860 041026 001356          BNE          2$        ; LOOP IF NOT RETURN
9861 041030 105063 177777    CLR          -1(R3)    ; CLEAR RETURN (THE 15)
9862 041034 104400 001222    TYPE          $LF      ; TYPE A LINE FEED
9863 041040 012603          MOV          (SP)+,R3  ; RESTORE R3
9864 041042 011646          MOV          (SP),-(SP) ; ADJUST THE STACK AND PUT ADDRESS OF THE
9865 041044 016666 000004 000002    MOV          4(SP),2(SP) ; FIRST ASCII CHARACTER ON IT
9866 041052 012766 041064 000004    MOV          $TTYIN,4(SP)
9867 041060 000002          RTI
9868 041062 000          9$: .BYTE      0        ; RETURN
9869 041063 000          .BYTE      0        ; STORAGE FOR ASCII CHAR. TO TYPE
9870 041064 000011          .BLKB      9         ; TERMINATOR
9871 041075 136 006503 000012  STTYIN: .BLKB 9         ; RESERVE 9 BYTES FOR TTY INPUT
SCNTLC: .ASCIZ /?C/<15><12> ; CONTROL "C"
; FROM THE TTY

;*****
.SBTL  READ AN OCTAL NUMBER FROM THE TTY

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

9891 041102 011646          SRDOCT: MOV      (SP),-(SP) ; PROVIDE SPACE FOR THE
9892 041104 016666 000004 000002    MOV          4(SP),2(SP) ; INPUT NUMBER
9893 041112 010046          MOV          R0,-(SP)  ; PUSH R0 ON STACK
9894 041114 010146          MOV          R1,-(SP)  ; PUSH R1 ON STACK
9895 041116 010246          MOV          R2,-(SP)  ; PUSH R2 ON STACK
9896 041120 104406          1$: RDLIN      ; READ AN ASCII LINE
9897 041122 012600          MOV          (SP)+,R0  ; GET ADDRESS OF 1ST CHARACTER
9898 041124 010067 000100    MOV          R0,5$     ; AND SAVE IT
9899 041130 005001          CLR          R1        ; CLEAR DATA WORD
9900 041132 005002          CLR          R2
9901 041134 112046          2$: MOVW      (R0)+,-(SP) ; PICKUP THIS CHARACTER
9902 041136 001420          BEQ          3$        ; IF ZERO GET OUT
9903 041140 122716 000060    CMPB          #'0,(SP) ; MAKE SURE THIS CHARACTER
9904 041144 003026          BGT          4$        ; IS AN OCTAL DIGIT
9905 041146 122716 000067    CMPB          #'7,(SP)
9906 041152 002423          BLT          4$

```

9907	041154	006301		ASL	R1	::#2
9908	041156	006102		ROL	R2	
9909	041160	006301		ASL	R1	::#4
9910	041162	006102		ROL	R2	
9911	041164	006301		ASL	R1	::#8
9912	041166	006102		ROL	R2	
9913	041170	042716	177770	BIC	#1C7,(SP)	::STRIP THE ASCII JUNK
9914	041174	062601		ADD	(SP)+,R1	::ADD IN THIS DIGIT
9915	041176	000756		BR	2\$	::LOOP
9916	041200	005726	3\$:	TST	(SP)+	::CLEAN TERMINATOR FROM STACK
9917	041202	010166	000012	MOV	R1,12(SP)	::SAVE THE RESULT
9918	041206	010267	000026	MOV	R2,\$SHIOCT	
9919	041212	012602		MOV	(SP)+,R2	::POP STACK INTO R2
9920	041214	012601		MOV	(SP)+,R1	::POP STACK INTO R1
9921	041216	012600		MOV	(SP)+,R0	::POP STACK INTO R0
9922	041220	000002		RTI		::RETURN
9923	041222	005726	4\$:	TST	(SP)+	::CLEAN PARTIAL FROM STACK
9924	041224	105010		CLRB	(R0)	::SET A TERMINATOR
9925	041226	104400		TYPE		::TYPE UP THRU THE BAD CHAR.
9926	041230	000000	5\$:	.WORD	0	
9927	041232	104400	001220	TYPE	\$QUES	::"? "CR" & "LF"
9928	041236	000730		BR	1\$	::TRY AGAIN
9929	041240	000000		\$SHIOCT: .WORD	0	::HIGH ORDER BITS GO HERE

```

9930 ;*****
9931
9932 .SBTTL ERROR HANDLER ROUTINE
9933
9934 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
9935 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9936 ;*AND GO TO $ERRTYP ON ERROR
9937 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9938 ;*SW15=1 HALT ON ERROR
9939 ;*SW13=1 INHIBIT ERROR TYPEOUTS
9940 ;*SW10=1 BELL ON ERROR
9941 ;*SW09=1 LOOP ON ERROR
9942 ;*CALL
9943 ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
9944
9945 041242 $ERROR:
9946
9947
9948 041242 REGSAV:
9949 041242 012737 177777 004644 MOV #-1,2,$ERFLGS ;SET ERROR FLAG
9950 041250 REGSA1:
9951
9952
9953 041250 105267 137627 7$: INCB $ERFLG ;;SET THE ERROR FLAG
9954 041254 001775 BEQ 7$ ;;DON'T LET THE FLAG GO TO ZERO
9955 041256 016777 137620 137654 MOV $STNM,2,$DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
9956 041264 032777 002000 137644 BIT #BIT10,2,$SWR ;;BELL ON ERROR?
9957 041272 001402 BEQ 1$ ;;NO - SKIP
9958 041274 104400 001214 TYPE $BELL ;;RING BELL
9959 041300 005267 137606 1$: INC $ERTTL ;;COUNT THE NUMBER OF ERRORS
9960 041304 011667 137606 MOV (SP), $ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
9961 041310 162767 000002 137600 SUB #2, $ERRPC
9962 041316 117767 137574 137570 MOVB 2,$ERRPC, $ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
9963 041324 032777 020000 137604 BIT #BIT13,2,$SWR ;;SKIP TYPEOUT IF SET
9964 041332 001004 BNE 20$ ;;SKIP TYPEOUTS
9965 041334 004767 000046 JSR PC, $ERRTYP ;;GO TO USER ERROR ROUTINE
9966 041340 104400 001221 TYPE , $CRLF
9967 041344
9968 041344 005777 137566 20$: TST 2,$SWR ;;HALT ON ERROR
9969 041350 100001 BPL 3$ ;;SKIP IF CONTINUE
9970 041352 000000 HALT ;;HALT ON ERROR!
9971 041354 032777 001000 137554 3$: BIT #BIT09,2,$SWR ;;LOOP ON ERROR SWITCH SET?
9972 041362 001402 BEQ 4$ ;;BR IF NO
9973 041364 016716 137520 MOV $LPERR, (SP) ;;FUDGE RETURN FOR LOOPING
9974 041370 005767 137616 4$: TST $ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS
9975 041374 001402 BEQ 5$ ;;BR IF NONE
9976 041376 016716 137610 MOV $ESCAPE, (SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
9977 041402 5$:
9978 041402 000002 RTI ;;RETURN
9979
9980 ;*****
9981 .SBTTL ERROR MESSAGE TYPEOUT ROUTINE
9982 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9983 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
9984 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
9985 ;*IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
;*WITH ONLY MINOR CHANGES

```

```

9986      ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
9987      ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
9988      ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
9989      ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
9990      ;*AND NOT THE ERROR MESSAGE AND HEADER.
9991      PRITEM: 0          ;PREVIOUS ITEM NO. LOCATION
9992      SERRTYP: MOV      2#SWR, -(SP) ;GET SWITCH SETTING
9993      BIC              2#C500, (SP) ;KEEP ONLY SWITCH 8 AND 6
9994      CMP              2#SW06, (SP)+ ;IS 6 SET AND 8 RESET
9995      BNE              1S          ;IF NOT BRANCH
9996      BR               2S          ;BRANCH IF SW 6 IS SET AND 8 RESET
9997      1S: JMP          2#TYPERR    ;JUMP IF SW 8 IS SET
9998      ;OR IF SW 8 IS RESET AND SW 6 IS RESET
9999      2S:
10000
10001      041404 000000 001136      TYPE      ,.+4          ;;TYPE ASCIZ STRING
10002      041406 013746 177277      BR        64S          ;;GET OVER THE ASCIZ
10003      ;;.ASCIZ      <15><12>/RHWC = /
10004      041412 042716 000100      64S:
10005      041416 022726 000100      MOV      2#WC, -(SP)    ;GET READY TO TYPE RHWC CONTENTS
10006      041422 001001 042346      TYPOC
10007
10008
10009      041462 104400 041470      TYPE      ,.+4          ;;TYPE ASCIZ STRING
10010      041466 000406 00226E      BR        65S          ;;GET OVER THE ASCIZ
10011      ;;.ASCIZ      <15><12>/RHBA = /
10012      041504 013746 002270      65S:
10013      041504 013746 104401      MOV      2#BA, -(SP)   ;GET READY TO TYPE RHBA CONTENTS
10014      041510 104401 041520      TYPOC
10015
10016
10017      041512 104400 041520      TYPE      ,.+4          ;;TYPE ASCIZ STRING
10018      041516 000406 002272      BR        66S          ;;GET OVER THE ASCIZ
10019      ;;.ASCIZ      <15><12>/RHCS2 = /
10020      041534 013746 002272      66S:
10021      041534 013746 104401      MOV      2#CS2, -(SP) ;GET READY TO TYPE RHCS2 CONTENTS
10022      041540 104401 041550      TYPOC
10023
10024
10025      041542 104400 041550      TYPE      ,.+4          ;;TYPE ASCIZ STRING
10026      041546 000406 002274      BR        67S          ;;GET OVER THE ASCIZ
10027      ;;.ASCIZ      <15><12>/RHCS1 = /
10028      041564 013746 002274      67S:
10029      041564 013746 104401      MOV      2#CS1, -(SP) ;GET READY TO TYPE RHCS1 CONTENTS
10030      041570 104401 041600      TYPOC
10031
10032
10033      041572 104400 041600      TYPE      ,.+4          ;;TYPE ASCIZ STRING
10034      041576 000406 002316      BR        68S          ;;GET OVER THE ASCIZ
10035      ;;.ASCIZ      <15><12>/RHDS1 = /
10036      041614 013746 002316      68S:
10037      041614 013746 104401      MOV      2#DS1, -(SP) ;GET READY TO TYPE RHDS1 CONTENTS
10038      041620 104401 041630      TYPOC
10039
10040
10041      041622 104400 041630      TYPE      ,.+4          ;;TYPE ASCIZ STRING
    
```



10042	041626	000406		BR	69\$		::GET OVER THE ASCIZ
10043				::.ASCIZ		<15><12>/RHER1 = /	
10044	041644		69\$:				
10045	041644	013746	002276	MOV	2#ER1,-(SP)		;GET READY TO TYPE RHER1 CONTENTS
10046	041650	104401		TYPOC			
10047							
10048							
10049	041652	104400	041660	TYPE	..+4		::TYPE ASCIZ STRING
10050	041656	000406		BR	70\$		::GET OVER THE ASCIZ
10051				::.ASCIZ		<15><12>/RHER2 = /	
10052	041674		70\$:				
10053	041674	013746	002302	MOV	2#ER2,-(SP)		;GET READY TO TYPE RHER2 CONTENTS
10054	041700	104401		TYPOC			
10055							
10056							
10057	041702	104400	041710	TYPE	..+4		::TYPE ASCIZ STRING
10058	041706	000406		BR	71\$		::GET OVER THE ASCIZ
10059				::.ASCIZ		<15><12>/RHER3 = /	
10060	041724		71\$:				
10061	041724	013746	002310	MOV	2#ER3,-(SP)		;GET READY TO TYPE RHER3 CONTENTS
10062	041730	104401		TYPOC			
10063							
10064							
10065	041732	104400	041740	TYPE	..+4		::TYPE ASCIZ STRING
10066	041736	000406		BR	72\$		::GET OVER THE ASCIZ
10067				::.ASCIZ		<15><12>/RHDST = /	
10068	041754		72\$:				
10069	041754	013746	002300	MOV	2#DST,-(SP)		;GET READY TO TYPE RHDST CONTENTS
10070	041760	104401		TYPOC			
10071							
10072							
10073	041762	104400	041770	TYPE	..+4		::TYPE ASCIZ STRING
10074	041766	000406		BR	73\$		::GET OVER THE ASCIZ
10075				::.ASCIZ		<15><12>/RHCA = /	
10076	042004		73\$:				
10077	042004	013746	002306	MOV	2#CA,-(SP)		;GET READY TO TYPE RHCA CONTENTS
10078	042010	104401		TYPOC			
10079							
10080							
10081	042012	104400	042020	TYPE	..+4		::TYPE ASCIZ STRING
10082	042016	000406		BR	74\$		::GET OVER THE ASCIZ
10083				::.ASCIZ		<15><12>/RHAS = /	
10084	042034		74\$:				
10085	042034	013746	002312	MOV	2#AS,-(SP)		;GET READY TO TYPE RHAS CONTENTS
10086	042040	104401		TYPOC			
10087							
10088							
10089	042042	104400	042050	TYPE	..+4		::TYPE ASCIZ STRING
10090	042046	000406		BR	75\$		::GET OVER THE ASCIZ
10091				::.ASCIZ		<15><12>/RHOF = /	
10092	042064		75\$:				
10093	042064	013746	002304	MOV	2#OF,-(SP)		;GET READY TO TYPE RHOF CONTENTS
10094	042070	104401		TYPOC			
10095							
10096							
10097	042072	104400	042100	TYPE	..+4		::TYPE ASCIZ STRING

10098	042076	000406		BR	76\$	;;GET OVER THE ASCIZ
10099				;;.ASCIZ	<15><12>/RHMR = /	
10100	042114		76\$:			
10101	042114	013746	002314	MOV	2#MR,-(SP)	;GET READY TO TYPE RHMR CONTENTS
10102	042120	104401		TYPOC		
10103						
10104						
10105	042122	104400	042130	TYPE	+4	;;TYPE ASCIZ STRING
10106	042126	000406		BR	77\$	;;GET OVER THE ASCIZ
10107				;;.ASCIZ	<15><12>/RHLA = /	
10108	042144		77\$:			
10109	042144	013746	002332	MOV	2#LA,-(SP)	;GET READY TO TYPE RHLA CONTENTS
10110	042150	104401		TYPOC		
10111						
10112						
10113	042152	104400	042160	TYPE	+4	;;TYPE ASCIZ STRING
10114	042156	000406		BR	78\$	;;GET OVER THE ASCIZ
10115				;;.ASCIZ	<15><12>/RHCC = /	
10116	042174		78\$:			
10117	042174	013746	002330	MOV	2#CC,-(SP)	;GET READY TO TYPE RHCC CONTENTS
10118	042200	104401		TYPOC		
10119						
10120						
10121	042202	104400	042210	TYPE	+4	;;TYPE ASCIZ STRING
10122	042206	000406		BR	79\$	;;GET OVER THE ASCIZ
10123				;;.ASCIZ	<15><12>/RHEC1 = /	
10124	042224		79\$:			
10125	042224	013746	002324	MOV	2#EC1,-(SP)	;GET READY TO TYPE RHEC1 CONTENTS
10126	042230	104401		TYPOC		
10127						
10128						
10129	042232	104400	042240	TYPE	+4	;;TYPE ASCIZ STRING
10130	042236	000406		BR	80\$	;;GET OVER THE ASCIZ
10131				;;.ASCIZ	<15><12>/RHEC2 = /	
10132	042254		80\$:			
10133	042254	013746	002326	MOV	2#EC2,-(SP)	;GET READY TO TYPE RHEC2 CONTENTS
10134	042260	104401		TYPOC		
10135						
10136						
10137	042262	104400	042270	TYPE	+4	;;TYPE ASCIZ STRING
10138	042266	000406		BR	81\$	;;GET OVER THE ASCIZ
10139				;;.ASCIZ	<15><12>/RHDT = /	
10140	042304		81\$:			
10141	042304	013746	002320	MOV	2#DT,-(SP)	;GET READY TO TYPE RHDT CONTENTS
10142	042310	104401		TYPOC		
10143						
10144						
10145	042312	104400	042320	TYPE	+4	;;TYPE ASCIZ STRING
10146	042316	000406		BR	82\$	;;GET OVER THE ASCIZ
10147				;;.ASCIZ	<15><12>/RHSN = /	
10148	042334		82\$:			
10149	042334	013746	002322	MOV	2#SN,-(SP)	;GET READY TO TYPE RHSN CONTENTS
10150	042340	104401		TYPOC		
10151						
10152	042342	005037	041404	CLR	2#PRITEM	;CLEAR PREVIOUS ERROR ITEM
10153	042346			TYPERR:		

```

10154 042346 104400 001221      TYPE      $CRLF      ;"CARRIAGE RETURN" & "LINE FEED"
10155 042352 010046      MOV      RO,-(SP)   ;SAVE RO
10156 042354 005000      CLR      RO        ;PICKUP THE ITEM INDEX
10157 042356 153700 001114      B1SB     @#$ITEMB,RO
10158 042362 001004      BNE      1$        ;IF ITEM NUMBER IS ZERO, JUST
10159                                ;TYPE THE PC OF THE ERROR
10160 042364 016746 136526      MOV      $ERRPC,-(SP) ;SAVE $ERRPC FOR TYPEOUT
10161                                ;ERROR ADDRESS
10162 042370 104401      TYP0C     ;GO TYPE--OCTAL ASCII(ALL DIGITS)
10163 042372 000454      BR      10$       ;GET OUT
10164 042374 005300      1$:      DEC      RO        ;ADJUST THE INDEX SO THAT IT WILL
10165 042376 006300      ASL      RO        ;WORK FOR THE ERROR TABLE
10166 042400 006300      ASL      RO
10167 042402 006300      ASL      RO
10168 042404 062700 001224      ADD      $ERRTB,RO  ;FORM TABLE POINTER
10169 042410 020037 041404      CMP      RO,@$PRITEM ;WAS PREVIOUS ERROR SAME
10170 042414 001002      BNE      13$      ;BRANCH IF NOT
10171 042416 022020      CMP      (RO)+,(RO)+ ;POP RO OVER EM AND DH
10172 042420 000420      BR      5$
10173 042422 010037 041404      13$:     MOV      RO,@$PRITEM ;SAVE NEW ERROR ITEM
10174 042426 012067 000004      MOV      (RO)+,2$  ;PICKUP "ERROR MESSAGE" POINTER
10175 042432 001404      BEQ      3$        ;SKIP TYPEOUT IF NO POINTER
10176 042434 104400      TYPE     ;TYPE THE "ERROR MESSAGE"
10177 042436 000000      2$:      .WORD    0        ;"ERROR MESSAGE" POINTER GOES HERE
10178 042440 104400 001221      TYPE     $CRLF     ;"CARRIAGE RETURN" & "LINE FEED"
10179 042444 012067 000004      3$:      MOV      (RO)+,4$  ;PICKUP "DATA HEADER" POINTER
10180 042450 001404      BEQ      5$        ;SKIP TYPEOUT IF 0
10181 042452 104400      TYPE     ;TYPE THE "DATA HEADER"
10182 042454 000000      4$:      .WORD    0        ;"DATA HEADER" POINTER GOES HERE
10183 042456 104400 001221      TYPE     $CRLF     ;"CARRIAGE RETURN" & "LINE FEED"
10184 042462 010146      5$:      MOV      R1,-(SP)  ;SAVE R1
10185 042464 012001      MOV      (RO)+,R1  ;PICKUP "DATA TABLE" POINTER
10186 042466 001415      BEQ      9$        ;BR IF NO DATA TO BE TYPED
10187 042470 012000      MOV      (RO)+,RO  ;PICKUP "DATA FORMAT" POINTER
10188 042472 105720      6$:      TSTB     (RO)+     ;"OCTAL" OR "DECIMAL"
10189 042474 001003      BNE      7$        ;BR IF DECIMAL
10190 042476 013146      MOV      @$(R1)+,-(SP) ;SAVE @$(R1)+ FOR TYPEOUT
10191 042500 104401      TYP0C     ;GO TYPE--OCTAL ASCII(ALL DIGITS)
10192 042502 000402      BR      8$
10193 042504      7$:      BR      8$
10194 042504 013146      MOV      @$(R1)+,-(SP) ;SAVE @$(R1)+ FOR TYPEOUT
10195 042506 104404      TYPDS     ;GO TYPE--DECIMAL ASCII WITH SIGN
10196 042510 005711      8$:      TST      (R1)     ;IS THERE ANOTHER NUMBER?
10197 042512 001403      BEQ      9$        ;BR IF NO
10198 042514 104400 042530      TYPE     ,11$     ;TYPE TWO(2) SPACES
10199 042520 000764      BR      6$        ;LOOP
10200
10201 042522 012601      9$:      MOV      (SP)+,R1  ;RESTORE R1
10202 042524 012600      10$:     MOV      (SP)+,RO  ;"CARRIAGE RETURN" & "LINE FEED"
10203 042526 000207      RTS      PC        ;RETURN
10204 042530 020040 000      11$:     .ASCIZ  / /      ;TWO(2) SPACES
10205      .EVEN
10206      ;*****
10207      ;*****
10208      ;*****
10209

```

```

10210 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
10211
10212 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
10213 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
10214 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
10215 ;*CALL:
10216 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
10217 ;*      TYPOS    ;;CALL FOR TYPEOUT
10218 ;*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
10219 ;*      .BYTE   M              ;;M=1 OR 0
10220 ;*                                     ;;1=TYPE LEADING ZEROS
10221 ;*                                     ;;0=SUPPRESS LEADING ZEROS
10222 ;*
10223 ;*$STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
10224 ;*$TYPOS OR $TYPOC
10225 ;*CALL:
10226 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
10227 ;*      TYPON    ;;CALL FOR TYPEOUT
10228 ;*
10229 ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
10230 ;*CALL:
10231 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
10232 ;*      TYPOC    ;;CALL FOR TYPEOUT
10233
10234 042534 017646 000000          $TYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
10235 042540 116667 000001 000211  MOVB     1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
10236 042546 112667 000207          MOVB     (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
10237 042552 062716 000002          ADD      #2,(SP)        ;;ADJUST RETURN ADDRESS
10238 042556 000406          BR       $TYPON
10239 042560 112767 000001 000171  $TYPOC: MOVB     #1,$OFILL    ;;SET THE ZERO FILL SWITCH
10240 042566 112767 000006 000165  MOVB     #6,$OMODE+1    ;;SET FOR SIX(6) DIGITS
10241 042574 112767 000005 000154  $TYPON: MOVB     #5,$OCNT    ;;SET THE ITERATION COUNT
10242 042602 010346          MOV      R3,-(SP)      ;;SAVE R3
10243 042604 010446          MOV      R4,-(SP)      ;;SAVE R4
10244 042606 010546          MOV      R5,-(SP)      ;;SAVE R5
10245 042610 116704 000145          MOVB     $OMODE+1,R4   ;;GET THE NUMBER OF DIGITS TO TYPE
10246 042614 005404          NEG      R4
10247 042616 062704 000006          ADD      #6,R4        ;;SUBTRACT IT FOR MAX. ALLOWED
10248 042622 110467 000132          MOVB     R4,$OMODE    ;;SAVE IT FOR USE
10249 042626 116704 000125          MOVB     $OFILL,R4    ;;GET THE ZERO FILL SWITCH
10250 042632 016605 000012          MOV      12(SP),R5    ;;PICKUP THE INPUT NUMBER
10251 042636 005003          CLR      R3           ;;CLEAR THE OUTPUT WORD
10252 042640 006105          1$:     ROL      R5        ;;ROTATE MSB INTO "C"
10253 042642 000404          BR       3$          ;;GO DO MSB
10254 042644 006105          2$:     ROL      R5        ;;FORM THIS DIGIT
10255 042646 006105          ROL      R5
10256 042650 006105          ROL      R5
10257 042652 010503          MOV      R5,R3
10258 042654 006103          3$:     ROL      R3        ;;GET LSB OF THIS DIGIT
10259 042656 105367 000076          DECB     $OMODE       ;;TYPE THIS DIGIT?
10260 042662 100016          BPL      7$          ;;BR IF NO
10261 042664 042703 177770          BIC     #177770,R3    ;;GET RID OF JUNK
10262 042670 001002          BNE     4$          ;;TEST FOR 0
10263 042672 005704          TST     R4           ;;SUPPRESS THIS 0?
10264 042674 001403          BEQ     5$          ;;BR IF YES
10265 042676 005204          4$:     INC      R4        ;;DON'T SUPPRESS ANYMORE 0'S

```

10266	042700	052703	000060		BIS	#'0,R3	:: MAKE THIS DIGIT ASCII
10267	042704	052703	000040	5\$:	BIS	#',R3	:: MAKE ASCII IF NOT ALREADY
10268	042710	110367	000040		MOVB	R3,B\$	:: SAVE FOR TYPING
10269	042714	104400	042754		TYPE	B\$	:: GO TYPE THIS DIGIT
10270	042720	105367	000032	7\$:	DECB	\$OCNT	:: COUNT BY 1
10271	042724	003347			BGT	2\$	:: BR IF MORE TO DO
10272	042726	002402			BLT	6\$	:: BR IF DONE
10273	042730	005204			INC	R4	:: INSURE LAST DIGIT ISN'T A BLANK
10274	042732	000744			BR	2\$	:: GO DO THE LAST DIGIT
10275	042734	012605		6\$:	MOV	(SP)+,R5	:: RESTORE R5
10276	042736	012604			MOV	(SP)+,R4	:: RESTORE R4
10277	042740	012603			MOV	(SP)+,R3	:: RESTORE R3
10278	042742	016666	000002 000004		MOV	2(SP),4(SP)	:: SET THE STACK FOR RETURNING
10279	042750	012616			MOV	(SP)+,(SP)	
10280	042752	000002			RTI		:: RETURN
10281	042754	000		8\$:	.BYTE	0	:: STORAGE FOR ASCII DIGIT
10282	042755	000			.BYTE	0	:: TERMINATOR FOR TYPE ROUTINE
10283	042756	000		\$OCNT:	.BYTE	0	:: OCTAL DIGIT COUNTER
10284	042757	000		\$DFILL:	.BYTE	0	:: ZERO FILL SWITCH
10285	042760	000000		\$OMODE:	.WORD	0	:: NUMBER OF DIGITS TO TYPE

```

10286 ;*****
10287
10288 .SBTTL TRAP DECODER
10289
10290 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10291 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10292 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
10293 ;*GO TO THAT ROUTINE.
10294

```

```

10295 042762 010046 STRAP: MOV RO, -(SP) ;: SAVE RO
10296 042764 016600 000002 MOV 2(SP), RO ;: GET TRAP ADDRESS
10297 042770 005740 TST -(RO) ;: BACKUP BY 2
10298 042772 111000 MOVB (RO), RO ;: GET RIGHT BYTE OF TRAP
10299 042774 006300 ASL RO ;: POSITION FOR INDEXING
10300 042776 016000 043004 MOV $TRPAD(RO), RO ;: INDEX TO TABLE
10301 043002 000200 RTS RO ;: GO TO ROUTINE
10302

```

```

10303 .SBTTL TRAP TABLE
10304
10305 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10306 ;*BY THE "TRAP" INSTRUCTION.
10307

```

```

10308 ; ROUTINE
10309 ; -----
10310
10311 $TRPAD:
10312 043004 040254 $TYPE ;: CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
10313 043006 042560 $TYPOC ;: CALL=TYPOC TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
10314 043010 042534 $TYPOS ;: CALL=TYPOS TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
10315 043012 042574 $TYPON ;: CALL=TYPON TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
10316 043014 040030 $TYPDS ;: CALL=TYPDS TRAP+4(104404) TYPE DECIMAL NUMBER (WITH SIGN)
10317 043016 040666 $RDCHR ;: CALL=RDCHR TRAP+5(104405) TTY TYPEIN CHARACTER ROUTINE
10318 043020 040756 $RDLIN ;: CALL=RDLIN TRAP+6(104406) TTY TYPEIN STRING ROUTINE
10319 043022 041102 $RDOCT ;: CALL=RDOCT TRAP+7(104407) READ AN OCTAL NUMBER FROM TTY
10320 043024 035250 WAIT.T ;: CALL=WAT TRAP+10(104410) DONT ADD ABOVE THIS TRAP
10321
10322
10323

```

```

10324 ;*****
10325
10326 .SBTTL POWER DOWN AND UP ROUTINES
10327
10328 :POWER DOWN ROUTINE
10329 043026 012737 043154 000024 $PWRDN: MOV $SILLUP, @#PWRVEC ;;SET FOR FAST UP
10330 043034 012737 000340 000026 MOV #340, @#PWRVEC+2 ;;PRIO:7
10331 043042 010046 MOV R0, -(SP) ;;PUSH R0 ON STACK
10332 043044 010146 MOV R1, -(SP) ;;PUSH R1 ON STACK
10333 043046 010246 MOV R2, -(SP) ;;PUSH R2 ON STACK
10334 043050 010346 MOV R3, -(SP) ;;PUSH R3 ON STACK
10335 043052 010446 MOV R4, -(SP) ;;PUSH R4 ON STACK
10336 043054 010546 MOV R5, -(SP) ;;PUSH R5 ON STACK
10337 043056 010667 000076 MOV SP, $SAVR6 ;;SAVE SP
10338 043062 012737 043074 000024 MOV $PWRUP, @#PWRVEC ;;SET UP VECTOR
10339 043070 000000 HALT
10340 043072 000776 BR .-2 ;;HANG UP
10341
10342 :POWER UP ROUTINE
10343 043074 016706 00006C $PWRUP: MOV $SAVR6, SP ;;GET SP
10344 043100 005067 000054 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
10345 043104 005267 000050 1$: INC $SAVR6 ;;WAIT FOR THE INC
10346 043110 001375 BNE 1$ ;;OF WORD
10347 043112 012605 MOV (SP)+, R5 ;;POP STACK INTO R5
10348 043114 012604 MOV (SP)+, R4 ;;POP STACK INTO R4
10349 043116 012603 MOV (SP)+, R3 ;;POP STACK INTO R3
10350 043120 012602 MOV (SP)+, R2 ;;POP STACK INTO R2
10351 043122 012601 MOV (SP)+, R1 ;;POP STACK INTO R1
10352 043124 012600 MOV (SP)+, R0 ;;POP STACK INTO R0
10353 043126 012737 043026 000024 MOV $PWRDN, @#PWRVEC ;;SET UP THE POWER DOWN VECTOR
10354 043134 012737 000340 000026 MOV #340, @#PWRVEC+2 ;;PRIO:7
10355 043142 104400 TYPE REPORT THE POWER FAILURE
10356 043144 043162 $PWRMG: .WORD $POWER POWER FAIL MESSAGE POINTER
10357 043146 012716 MOV (PC)+, (SP) ;;RESTART AT BEGIN
10358 043150 004712 $PWRAD: .WORD BEGIN ;;RESTART ADDRESS
10359 043152 000002 RTI
10360 043154 000000 $SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
10361 043156 000776 BR .-2 ;;BEFORE THE POWER DOWN WAS COMPLETE
10362 043160 000000 $SAVR6: 0 ;;PUT THE SP HERE
10363 043162 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
10364 043170 000122
10365 .EVEN
  
```

```

10366 ;*****
10367 ;
10368 ;ERROR AND MESSAGE TABLE CONDIMENTS
10369 ;
10370 ;*****
10371 ;
10372 ;
10373 ;
10374 ;
10375 043172 050122 032060 042040 EM1: .ASCIZ /RPO4 DID NOT INTERRUPT/
10376 043200 042111 047040 052117
10377 043206 044440 052116 051105
10378 043214 052522 052120 000
10379 043221 111 052116 051105 EM2: .ASCIZ /INTERRUPT ENABLE BIT DOWN BUT EXPECTED BIT DID NOT SET/
10380 043226 052522 052120 042440
10381 043234 040516 046102 020105
10382 043242 044502 020124 047504
10383 043250 047127 041040 052125
10384 043256 042440 050130 041505
10385 043264 042524 020104 044502
10386 043272 020124 044504 020104
10387 043300 047516 020124 042523
10388 043306 000124
10389 043310 050122 032060 042040 EM3: .ASCIZ /RPO4 DID NOT INTERRUPT WHEN EXPECTED BIT DID SET/
10390 043316 042111 047040 052117
10391 043324 044440 052116 051105
10392 043332 052522 052120 053440
10393 043340 042510 020116 054105
10394 043346 042520 052103 042105
10395 043354 041040 052111 042040
10396 043362 042111 051440 052105
10397 043370 000
10398 043371 105 050130 041505 EM4: .ASCIZ /EXPECTED BIT DID SET BUT TIME IS IN ERROR TIME IN 10 MICROSEC. DECIMAL/
10399 043376 042524 020104 044502
10400 043404 020124 044504 020104
10401 043412 042523 020124 052502
10402 043420 020124 044524 042515
10403 043426 044440 020123 047111
10404 043434 042440 051122 051117
10405 043442 052040 046511 020105
10406 043450 047111 030440 020060
10407 043456 044515 051103 051517
10408 043464 041505 020056 042504
10409 043472 044503 040515 000114
10410 043500 044122 051501 042040 EM5: .ASCIZ /RHAS DOES NOT CLEAR BY MOVING IN ALL ONES/
10411 043506 042517 020123 047516
10412 043514 020124 046103 040505
10413 043522 020122 054502 046440
10414 043530 053117 047111 020107
10415 043536 047111 040440 046114
10416 043544 047440 042516 000123
10417 043552 047514 042101 047111 EM6: .ASCIZ /LOADING RHER1 FOR ALL UNITS DID NOT SET ANY RHAS BITS/
10418 043560 020107 044122 051105
10419 043566 020061 047506 020122
10420 043574 046101 020114 047125
10421 043602 052111 020123 044504

```



10422	043610	020104	047516	020124
10423	043616	042523	020124	047101
10424	043622	020131	044122	051501
10425	043627	041040	052111	030123
10426	043634	050123	041505	043111
10427	043640	042511	020104	042522
10428	043646	044507	052123	051105
10429	043654	047040	047117	042440
10430	043662	044530	052123	047101
10431	043670	026124	051440	020117
10432	043676	041101	051117	020124
10433	043704	041120	043517	040522
10434	043712	000115		
10435	043720	052123	050117	042520
10436	043722	020104	051104	053111
10437	043730	020105	040510	020123
10438	043736	047515	020114	044502
10439	043744	020124	047111	051040
10440	043752	042110	030523	051440
10441	043760	052105	000	
10442	043771	127	052111	020110
10443	043776	050123	047111	046104
10444	044004	020105	047520	042527
10445	044012	042522	020104	047504
10446	044020	047127	051040	041510
10447	044026	031123	051440	047510
10448	044034	046125	020104	047117
10449	044042	054514	044040	053101
10450	044050	020105	047125	052111
10451	044056	047040	035117	040440
10452	044064	042116	044440	020122
10453	044072	042523	000124	
10454	044076	043101	042524	020122
10455	044104	050123	047111	046104
10456	044112	020105	047520	042527
10457	044120	042522	020104	050125
10458	044126	020054	047516	050040
10459	044134	041501	020113	041501
10460	044142	047113	020056	044122
10461	044150	051504	020061	044123
10462	044156	052517	042114	044040
10463	044164	053101	020105	047515
10464	044172	036514	026061	053040
10465	044200	036526	000060	
10466	044204	044527	044124	051440
10467	044212	044520	042116	042514
10468	044220	050040	053517	051105
10469	044228	042105	020054	047516
10470	044234	044440	042116	040511
10471	044242	044514	042532	020054
10472	044250	044122	051503	020061
10473	044256	044123	052517	042114
10474	044264	044040	053101	020105
10475	044272	047507	030075	020054
10476	044300	053104	036501	026061
10477	044306	051040	054504	030475

EM7: .ASCIZ /SPECIFIED REGISTER NON EXISTANT, SO ABORT PROGRAM/

EM10: .ASCIZ /STOPPED DRIVE HAS MOL BIT IN RHDS1 SET/

EM11: .ASCIZ /WITH SPINDLE POWERED DOWN RHCS2 SHOULD ONLY HAVE UNIT NO: AND IR SET/

EM12: .ASCIZ /AFTER SPINDLE POWERED UP, NO PACK ACKN. RHDS1 SHOULD HAVE MOL=1, VV=0/

EM13: .ASCIZ /WITH SPINDLE POWERED, NO INTIALIZE, RHCS1 SHOULD HAVE GO=0, DVA=1. RDY=

10478	044314	020054	042511	030075	
10479	044322	000			
10480	044323	101	052106	051105	EM14: .ASCIZ /AFTER SPINDLE POWERED UP RHCC SHOULD BE=0/
10481	044330	051440	044520	042116	
10482	044336	042514	050040	053517	
10483	044344	051105	042105	052440	
10484	044352	020120	044122	041503	
10485	044360	051440	047510	046125	
10486	044366	020104	042502	030075	
10487	044374	000			
10488	044375	120	041501	020113	EM15: .ASCII /PACK ACKNOWLEDGE COMMAND CAUSED AN ERROR/<15><12>
10489	044402	041501	047113	053517	
10490	044410	042514	043504	020105	
10491	044416	047503	046515	047101	
10492	044424	020104	040503	051525	
10493	044432	042105	040440	020116	
10494	044440	051105	047522	006522	
10495	044446	012			
10496	044447	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
10497	044454	040504	040524	044440	
10498	044462	020123	042502	047506	
10499	044470	042522	041440	046517	
10500	044476	040515	042116	020054	
10501	044504	042103	040103	040504	
10502	044512	040524	044440	020123	
10503	044520	043101	042524	020122	
10504	044526	047503	046515	047101	
10505	044534	000104			
10506	044536	047516	047455	020120	EM16: .ASCII /NO-OP COMMAND CAUSED AN ERROR/<15><12>
10507	044544	047503	046515	047101	
10508	044552	020104	040503	051525	
10509	044560	042105	040440	020116	
10510	044566	051105	047522	006522	
10511	044574	012			
10512	044575	107	047517	020104	.ASCIZ /GOOD DATA IS BEFORE COMMAND, REC DATA IS AFTER COMMAND/
10513	044582	040504	040524	044440	
10514	044610	020123	042502	047506	
10515	044616	042522	041440	046517	
10516	044624	040515	042116	020054	
10517	044632	042522	020103	040504	
10518	044640	040524	044440	020123	
10519	044646	043101	042524	020122	
10520	044654	047503	046515	047101	
10521	044662	000104			
10522	044664	051104	053111	020105	EM17: .ASCII /DRIVE CLEAR COMMAND CAUSED AN ERROR/<15><12>
10523	044672	046103	040505	020122	
10524	044700	047503	046515	047101	
10525	044706	020104	040503	051525	
10526	044714	042105	040440	020116	
10527	044722	051105	047522	006522	
10528	044730	012			
10529	044731	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES AFTER COMMAND/
10530	044736	040504	040524	043440	
10531	044744	053111	051505	051440	
10532	044752	047510	046125	020104	
10533	044760	042502	020054	042522	

10534	044766	020103	040504	040524	
10535	044774	043440	053111	051505	
10536	045002	040440	052106	051105	
10537	045010	041440	046517	040515	
10538	045016	042116	000		
10539	045021	122	040505	026504	EM20: .ASCII /READ-IN COMMAND CAUSED AN ERROR/<15><12>
10540	045026	047111	041440	046517	
10541	045034	040515	042116	041440	
10542	045042	052501	042523	020104	
10543	045050	047101	042440	051122	
10544	045056	051117	005015		
10545	045062	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONTENTS AFTER COMMAND/
10546	045070	052101	020101	044507	
10547	045076	042526	020123	044123	
10548	045104	052517	042114	041040	
10549	045112	026105	051040	041505	
10550	045120	042040	052101	020101	
10551	045126	044507	042526	020123	
10552	045134	042522	027107	041440	
10553	045142	047117	042524	052116	
10554	045150	020123	043101	042524	
10555	045156	020122	047503	046515	
10556	045164	047101	000104		
10557	045170	044122	051503	020061	EM21: .ASCIZ /RHCSI CONTENTS DURING COMMAND WAS IN ERROR/
10558	045176	047503	052116	047105	
10559	045204	051524	042040	051125	
10560	045212	047111	020107	047503	
10561	045220	046515	047101	020104	
10562	045226	040527	020123	047111	
10563	045234	042440	051122	051117	
10564	045242	000			
10565	045243	122	042110	030523	EM22: .ASCIZ /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/
10566	045250	041440	047117	042524	
10567	045256	052116	020123	052504	
10568	045264	044522	043516	041440	
10569	045272	046517	040515	042116	
10570	045300	053440	051501	044440	
10571	045306	020116	051105	047522	
10572	045314	050122			
10573	045316	047125	047514	042101	EM23: .ASCII /UNLOAD COMMAND CAUSED AN ERROR/<15><12>
10574	045324	041440	046517	040515	
10575	045332	042116	041440	052501	
10576	045340	042523	020104	047101	
10577	045346	042440	051122	051117	
10578	045354	005015			
10579	045356	047507	042117	042040	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REGISTER CONT. AFTER COMMAND/
10580	045364	052101	020101	044507	
10581	045372	042526	020123	044123	
10582	045400	052517	042114	041040	
10583	045406	026105	051040	041505	
10584	045414	042040	052101	020101	
10585	045422	044507	042526	020123	
10586	045430	042522	044507	052123	
10587	045436	051105	041440	047117	
10588	045444	027124	040440	052106	
10589	045452	051105	041440	046517	

10590	045460	040515	042116	000	
10591	045465	117	043106	042523	EM24: .ASCII /OFFSET COMMAND CAUSED AN ERROR/<15><12>
10592	045472	020124	047503	046515	
10593	045500	047101	020104	040503	
10594	045506	051525	042105	040440	
10595	045514	020116	051105	047522	
10596	045522	006522	012		
10597	045525	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
10598	045532	040504	040524	043440	
10599	045540	053111	051505	051440	
10600	045546	047510	046125	020104	
10601	045554	042502	020054	042522	
10602	045562	020103	040504	040524	
10603	045570	043440	053111	051505	
10604	045576	051040	043505	020056	
10605	045604	047503	052116	020056	
10606	045612	043101	042524	020122	
10607	045620	047503	046515	047101	
10608	045626	000104			
10609	045630	042522	052524	047122	EM25: .ASCII /RETURN TO CENTER LINE COMMAND CAUSED AN ERROR/<15><12>
10610	045636	052040	020117	042503	
10611	045644	052116	051105	046040	
10612	045652	047111	020105	047503	
10613	045660	046515	047101	020104	
10614	045666	040503	051525	042105	
10615	045674	040440	020116	051105	
10616	045702	047522	006522	012	
10617	045707	107	047517	020104	.ASCIZ /GOOD DATA GIVES SHOULD BE, REC DATA GIVES REG. CONT. AFTER COMMAND/
10618	045714	040504	040524	043440	
10619	045722	053111	051505	051440	
10620	045730	047510	046125	020104	
10621	045736	042502	020054	042522	
10622	045744	020103	040504	040524	
10623	045752	043440	053111	051505	
10624	045760	051040	043505	020056	
10625	045766	047503	052116	020056	
10626	045774	043101	042524	020122	
10627	046002	047503	046515	047101	
10628	046010	000104			
10629	046012	030065	020060	043117	EM26: .ASCIZ /500 OFFSET COMMANDS ONE AFTER THE OTHER CAUSED AN ERROR/
10630	046020	051506	052105	041440	
10631	046026	046517	040515	042116	
10632	046034	020123	047117	020105	
10633	046042	043101	042524	020122	
10634	046050	044124	020105	052117	
10635	046056	042510	020122	040503	
10636	046064	051525	042105	040440	
10637	046072	020116	051105	047522	
10638	046100	000122			
10639	046102	051127	052111	020105	EM27: .ASCII /WRITE HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
10640	046110	042510	042101	051105	
10641	046116	040440	042116	042040	
10642	046124	052101	020101	040503	
10643	046132	051525	042105	044440	
10644	046140	050115	047522	042520	
10645	046146	020122	042522	044507	

10646	046154	052123	051105	041440	
10647	046162	040510	043516	006505	
10648	046170	012			
10649	046171	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10650	046176	040504	040524	043440	
10651	046204	053111	051505	053440	
10652	046212	040510	020124	044123	
10653	046220	052517	042114	041040	
10654	046226	020105	044124	051105	
10655	046234	006505	012		
10656	046237	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
10657	046244	042526	020104	040504	
10658	046252	040524	043440	053111	
10659	046260	051505	053440	040510	
10660	046266	020124	040527	020123	
10661	046274	044124	051105	020105	
10662	046302	043101	042524	020122	
10663	046310	047503	046515	047101	
10664	046316	000104			
10665	046320	051127	052111	020105	EM30: .ASCIZ /WRITE HEADER AND DATA CHANGED WRITE FROM BUFFER/
10666	046326	042510	042101	051105	
10667	046334	040440	042116	042040	
10668	046342	052101	020101	044103	
10669	046350	047101	042507	020104	
10670	046356	051127	052111	020105	
10671	046364	051106	046517	041040	
10672	046372	043125	042506	000122	
10673	046400	042522	042101	044040	EM31: .ASCII /READ HEADER AND DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
10674	046406	040505	042504	020122	
10675	046414	047101	020104	040504	
10676	046422	040524	041440	052501	
10677	046430	042523	020104	046511	
10678	046436	051120	050117	051105	
10679	046444	051040	043505	051511	
10680	046452	042524	020122	044103	
10681	046460	047101	042507	005015	
10682	046466	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10683	046474	052101	020101	044507	
10684	046502	042526	020123	044127	
10685	046510	052101	051440	047510	
10686	046516	046125	020104	042502	
10687	046524	052040	042510	042522	
10688	046532	005015			
10689	046534	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
10690	046542	042105	042040	052101	
10691	046550	020101	044507	042526	
10692	046556	020123	044127	052101	
10693	046564	053440	051501	052040	
10694	046572	042510	042522	040440	
10695	046600	052106	051105	041440	
10696	046606	046517	040515	042116	
10697	046614	000			
10698	046615	127	044522	042524	EM32: .ASCIZ /WRITE HEADER DATA FOLLOWED BY READ HEADER AND DATA CAUSED DATA ERROR/
10699	046622	044040	040505	042504	
10700	046630	020122	040504	040524	
10701	046636	043040	046117	047514	

10702	046644	042527	020104	054502	
10703	046652	051040	040505	020104	
10704	046660	042510	042101	051105	
10705	046666	040440	042116	042040	
10706	046674	052101	020101	040503	
10707	046702	051525	042105	042040	
10708	046710	052101	020101	051105	
10709	046716	047522	000122		
10710	046722	042522	042101	042040	EM33: .ASCII /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
10711	046730	052101	020101	040503	
10712	046736	051525	042105	044440	
10713	046744	050115	047522	042520	
10714	046752	020122	042522	044507	
10715	046760	052123	051105	041440	
10716	046766	040510	043516	006505	
10717	046774	012			
10718	046775	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10719	047002	040504	040524	043440	
10720	047010	053111	051505	053440	
10721	047016	040510	020124	044123	
10722	047024	052517	042114	041040	
10723	047032	020105	044124	051105	
10724	047040	006505	012		
10725	047043	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
10726	047050	042526	020104	040504	
10727	047056	040524	043440	053111	
10728	047064	051505	053440	040510	
10729	047072	020124	040527	020123	
10730	047100	044124	051105	020105	
10731	047106	043101	042524	020122	
10732	047114	047503	046515	047101	
10733	047122	000104			
10734	047124	042522	042101	042040	EM34: .ASCIZ /READ DATA INCORRECT/
10735	047132	052101	020101	047111	
10736	047140	047503	051122	041505	
10737	047146	000124			
10738	047150	051127	052111	020105	EM35: .ASCII /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
10739	047156	040504	040524	041440	
10740	047164	046517	040515	042116	
10741	047172	041440	052501	042523	
10742	047200	020104	046511	051120	
10743	047206	050117	051105	051040	
10744	047214	043505	051511	042524	
10745	047222	020122	044103	047101	
10746	047230	042107	005015		
10747	047234	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10748	047242	052101	020101	044507	
10749	047250	042526	020123	044127	
10750	047256	052101	051440	047510	
10751	047264	046125	020104	042502	
10752	047272	052040	042510	042522	
10753	047300	005015			
10754	047302	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
10755	047310	042105	042040	052101	
10756	047316	020101	044507	042526	
10757	047324	020123	042522	044507	

10758	047332	052123	051105	041440	
10759	047340	047117	042524	052116	
10760	047346	020123	043101	042524	
10761	047354	020122	047503	046515	
10762	047362	047101	000104		
10763	047366	051127	052111	020105	EM36: .ASCIZ /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
10764	047374	040504	040524	041440	
10765	047402	046517	040515	042116	
10766	047410	041440	040510	043516	
10767	047416	042105	053440	044522	
10768	047424	042524	043040	047522	
10769	047432	020115	052502	043106	
10770	047440	051105	000		
10771	047443	123	042505	020113	EM37: .ASCII /SEEK COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
10772	047450	047503	046515	047101	
10773	047456	020104	040503	051525	
10774	047464	042105	044440	050115	
10775	047472	047522	042520	020122	
10776	047500	042522	044507	052123	
10777	047506	051105	041440	040510	
10778	047514	043516	006505	012	
10779	047521	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10780	047526	040504	040524	043440	
10781	047534	053111	051505	053440	
10782	047542	040510	020124	044123	
10783	047550	052517	042114	041040	
10784	047556	020105	044124	051105	
10785	047564	006505	012		
10786	047567	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER SEEK COMMAND/
10787	047574	042526	020104	040504	
10788	047602	040524	043440	053111	
10789	047610	051505	051040	043505	
10790	047616	051511	042524	020122	
10791	047624	047503	052116	047105	
10792	047632	051524	040440	052106	
10793	047640	051105	051440	042505	
10794	047646	020113	047503	046515	
10795	047654	047101	000104		
10796	047660	051127	052111	020105	EM40: .ASCII /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
10797	047666	044103	041505	020113	
10798	047674	040503	051525	042105	
10799	047702	044440	050115	047522	
10800	047710	042520	020122	042522	
10801	047716	044507	052123	051105	
10802	047724	041440	040510	043516	
10803	047732	006505	012		
10804	047735	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10805	047742	040504	040524	043440	
10806	047750	053111	051505	053440	
10807	047756	040510	020124	044123	
10808	047764	052517	042114	041040	
10809	047772	020105	044124	051105	
10810	050000	006505	012		
10811	050003	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
10812	050010	042526	020104	040504	
10813	050016	040524	043440	053111	

MAINDEC-11-DERPU-B MACY11 27(732) 30-DEC-76 10:30 PAGE 211  
 DERPUB.P11 POWER DOWN AND UP ROUTINES

10814	050024	051505	051040	043505	
10815	050032	051511	042524	020122	
10816	050040	047503	052116	047105	
10817	050046	051524	040440	052106	
10818	050054	051105	041440	046517	
10819	050062	040515	042116	000	
10820	050067	114	041517	044513	EM41: .ASCII /LOCKING OUT WRITE BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/
10821	050074	043516	047440	052125	
10822	050102	053440	044522	042524	
10823	050110	041040	020131	051127	
10824	050116	052111	020105	047514	
10825	050124	045503	041040	052125	
10826	050132	047524	020116	040503	
10827	050140	051525	042105	044440	
10828	050146	050115	047522	042520	
10829	050154	020122	042522	044507	
10830	050162	052123	051105	041440	
10831	050170	040510	043516	006505	
10832	050176	012			
10833	050177	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10834	050204	040504	040524	043440	
10835	050212	053111	051505	053440	
10836	050220	040510	020124	044123	
10837	050226	052517	042114	041040	
10838	050234	020105	044124	051105	
10839	050242	006505	012		
10840	050245	122	041505	044505	.ASCIIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITES WERE LOCKED OUT/
10841	050252	042526	020104	040504	
10842	050260	040524	043440	053111	
10843	050266	051505	051040	043505	
10844	050274	051511	042524	020122	
10845	050302	047503	052116	047105	
10846	050310	051524	040440	052106	
10847	050316	051105	053440	044522	
10848	050324	042524	020123	042527	
10849	050332	042522	046040	041517	
10850	050340	042513	020104	052517	
10851	050346	000124			
10852	050350	052101	042524	050115	EM42: .ASCII /ATTEMPTING TO WRITE WITH WRITES LOCKED OUT CAUSED IMPROPER REGISTER CHA
10853	050356	044524	043516	052040	
10854	050364	020117	051127	052111	
10855	050372	020105	044527	044124	
10856	050400	053440	044522	042524	
10857	050406	020123	047514	045503	
10858	050414	042105	047440	052125	
10859	050422	041440	052501	042523	
10860	050430	020104	046511	051120	
10861	050436	050117	051105	051040	
10862	050444	043505	051511	042524	
10863	050452	020122	044103	047101	
10864	050460	042507	005015		
10865	050464	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
10866	050472	052101	020101	044507	
10867	050500	042526	020123	044127	
10868	050506	052101	051440	047510	
10869	050514	046125	020104	042502	



10870	050522	052040	042510	042522	
10871	050530	005015			
10872	050532	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE/
10873	050540	042105	042040	052101	
10874	050546	020101	044507	042526	
10875	050554	020123	042522	044507	
10876	050562	052123	051105	041440	
10877	050570	047117	042524	052116	
10878	050576	020123	043101	042524	
10879	050604	020122	052101	042524	
10880	050612	050115	042524	020104	
10881	050620	051127	052111	000105	
10882	050626	051127	052111	047111	EM43: .ASCII /WRITING WITH WRITES LOCKED OUT CHANGED DISK DATA/<15><12>
10883	050634	020107	044527	044124	
10884	050642	053440	044522	042524	
10885	050650	020123	047514	045503	
10886	050656	042105	047440	052125	
10887	050664	041440	040510	043516	
10888	050672	042105	042040	051511	
10889	050700	020113	040504	040524	
10890	050706	005015			
10891	050710	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT WAS ON DISK BEFORE WRITE WITH WRITE LOCKED OUT/<15>
10892	050716	052101	020101	044507	
10893	050724	042526	020123	044127	
10894	050732	052101	053440	051501	
10895	050740	047440	020116	044504	
10896	050746	045523	041040	043105	
10897	050754	051117	020105	051127	
10898	050762	052111	020105	044527	
10899	050770	044124	053440	044522	
10900	050776	042524	046040	041517	
10901	051004	042513	020104	052517	
10902	051012	006524	012		
10903	051015	127	051501	040440	.ASCII /WAS ATTEMPTED/<15><12>
10904	051022	052124	046505	052120	
10905	051030	042105	005015		
10906	051034	042522	042503	053111	.ASCII /RECEIVED DATA GIVES WHAT WAS READ BACK AFTER WRITE/<15><12>
10907	051042	042105	042040	052101	
10908	051050	020101	044507	042526	
10909	051056	020123	044127	052101	
10910	051064	053440	051501	051040	
10911	051072	040505	020104	040502	
10912	051100	045503	040440	052106	
10913	051106	051105	053440	044522	
10914	051114	042524	005015		
10915	051120	044527	044124	053440	.ASCIZ /WITH WRITE LOCKED OUT WAS ATTEMPTED/
10916	051126	044522	042524	046040	
10917	051134	041517	042513	020104	
10918	051142	052517	020124	040527	
10919	051150	020123	052101	042524	
10920	051156	050115	042524	000104	
10921	051164	047105	041101	044514	EM44: .ASCII /ENABLING WRITES BY WRITE LOCK BUTTON CAUSED IMPROPER REGISTER CHANGE/<1
10922	051172	043516	053440	044522	
10923	051200	042524	020123	054502	
10924	051206	053440	044522	042524	
10925	051214	046040	041517	020113	

10926	051222	052502	052124	047117
10927	051230	041440	052501	042523
10928	051236	020104	046511	051120
10929	051244	050117	051105	051040
10930	051252	043505	051511	042524
10931	051260	020122	044103	047101
10932	051266	042507	005015	
10933	051272	047507	042117	042040
10934	051300	052101	020101	044507
10935	051306	042526	020123	044127
10936	051314	052101	051440	047510
10937	051322	046125	020104	042502
10938	051330	052040	042510	042522
10939	051336	005015		
10940	051340	042522	042503	053111
10941	051346	042105	042040	052101
10942	051354	020101	044507	042526
10943	051362	020123	042522	044507
10944	051370	052123	051105	041440
10945	051376	047117	042524	052116
10946	051404	020123	043101	042524
10947	051412	020122	051127	052111
10948	051420	020105	047514	045503
10949	051426	041040	052125	047524
10950	051434	006516	012	
10951	051437	105	040516	046102
10952	051444	042105	053440	044522
10953	051452	042524	000123	
10954	051456	051124	047101	043123
10955	051464	051105	044522	043516
10956	051472	047440	020116	040514
10957	051500	052123	041040	047514
10958	051506	045503	026440	041440
10959	051514	046131	047111	042504
10960	051522	020122	030464	026060
10961	051530	051440	041505	047524
10962	051536	020122	030462	020054
10963	051544	051124	041501	020113
10964	051552	034061	005015	
10965	051556	040503	051525	042105
10966	051564	044440	050115	047522
10967	051572	042520	020122	042522
10968	051600	044507	052123	051105
10969	051606	041440	040510	043516
10970	051614	006505	012	
10971	051617	107	047517	020104
10972	051624	040504	040524	043440
10973	051632	053111	051505	053440
10974	051640	040510	020124	044123
10975	051646	052517	042114	041040
10976	051654	020105	044124	051105
10977	051662	006505	012	
10978	051665	122	041505	044505
10979	051672	042526	020104	040504
10980	051700	040524	043440	053111
10981	051706	051505	051040	043505

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER WRITE LOCK BUTTON/<15><12>

.ASCIZ /ENABLED WRITES/

EM45: .ASCII /TRANSFERRING ON LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>

.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER TRANSFER/

MPINDEC-11-DERPU-B MACY11 27(732) 30-DEC-76 10:30 PAGE 214  
 DERPUB.P11 POWER DOWN AND UP ROUTINES

10982	051714	051511	042524	020122	
10983	051722	047503	052116	047105	
10984	051730	051524	040440	052106	
10985	051736	051105	052040	040522	
10986	051744	051516	042506	000122	
10987	051752	040504	040524	051040	EM46: .ASCII /DATA READ FROM LAST BLOCK - CYLINDER 410, SECTOR 21, TRACK 18/<15><12>
10988	051760	040505	020104	051106	
10989	051766	046517	046040	051501	
10990	051774	020124	046102	041517	
10991	052002	020113	020055	054503	
10992	052010	044514	042116	051105	
10993	052016	032040	030061	020054	
10994	052024	042523	052103	051117	
10995	052032	031040	026061	052040	
10996	052040	040522	045503	030440	
10997	052046	006470	012		
10998	052051	111	020123	047111	.ASCIZ /IS IN ERROR/
10999	052056	042440	051122	051117	
11000	052064	000			
11001	052065	124	040522	051516	EM47: .ASCII /TRANSFERRING DATA FROM NONEXISTANT SECTOR CAUSED IMPROPER /<15><12>
11002	052072	042506	051122	047111	
11003	052100	020107	040504	040524	
11004	052106	043040	047522	020115	
11005	052114	047516	042516	044530	
11006	052122	052123	047101	020124	
11007	052130	042523	052103	051117	
11008	052136	041440	052501	042523	
11009	052144	020104	046511	051120	
11010	052152	050117	051105	006440	
11011	052160	012			
11012	052161	122	043505	051511	.ASCII /REGISTER CHANGE, GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11013	052166	042524	020122	044103	
11014	052174	047101	042507	020054	
11015	052202	047507	042117	042040	
11016	052210	052101	020101	044507	
11017	052216	042526	020123	044127	
11018	052224	052101	051440	047510	
11019	052232	046125	020104	042502	
11020	052240	052040	042510	042522	
11021	052246	005015			
11022	052250	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED TRANSFER/
11023	052256	042105	042040	052101	
11024	052264	020101	044507	042526	
11025	052272	020123	042522	044507	
11026	052300	052123	051105	041440	
11027	052306	047117	042524	052116	
11028	052314	020123	043101	042524	
11029	052322	020122	052101	042524	
11030	052330	050115	042524	020104	
11031	052336	051124	047101	043123	
11032	052344	051105	000		
11033	052347	124	040522	051516	EM50: .ASCII /TRANSFERRING FROM NONEXISTANT SECTOR CAUSED DATA ERROR/<15><12>
11034	052354	042506	051122	047111	
11035	052362	020107	051106	046517	
11036	052370	047040	047117	054105	
11037	052376	051511	040524	052116	

11038	052404	051440	041505	047524	
11039	052412	020122	040503	051525	
11040	052420	042105	042040	052101	
11041	052426	020101	051105	047522	
11042	052434	006522	012		
11043	052437	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11044	052444	040504	040524	043440	
11045	052452	053111	051505	053440	
11046	052460	040510	020124	044123	
11047	052466	052517	042114	041040	
11048	052474	020105	044124	051105	
11049	052502	006505	012		
11050	052505	102	042101	042040	.ASCIZ /BAD DATA GIVES WHAT WAS IN BUFFER AFTER TRANSFER/
11051	052512	052101	020101	044507	
11052	052520	042526	020123	044127	
11053	052526	052101	053440	051501	
11054	052534	044440	020116	052502	
11055	052542	043106	051105	040440	
11056	052550	052106	051105	052040	
11057	052556	040522	051516	042506	
11058	052564	000122			
11059	052566	044507	044526	043516	EMS1: .ASCII /GIVING ILLEGAL FUNCTION CAUSED IMPROPER REGISTER CHANGE/<15><12>
11060	052574	044440	046114	043505	
11061	052602	046101	043040	047125	
11062	052610	052103	047511	020116	
11063	052616	040503	051525	042105	
11064	052624	044440	050115	047522	
11065	052632	042520	020122	042522	
11066	052640	044507	052123	051105	
11067	052646	041440	040510	043516	
11068	052654	006505	012		
11069	052657	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11070	052664	040504	040524	043440	
11071	052672	053111	051505	053440	
11072	052700	040510	020124	044123	
11073	052706	052517	042114	041040	
11074	052714	020105	044124	051105	
11075	052722	006505	012		
11076	052725	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ILLEGAL FUNCTION IS GIVEN/
11077	052732	042526	020104	040504	
11078	052740	040524	043440	053111	
11079	052746	051505	051040	043505	
11080	052754	051511	042524	020122	
11081	052762	047503	052116	047105	
11082	052770	051524	040440	052106	
11083	052776	051105	044440	046114	
11084	053004	043505	046101	043040	
11085	053012	047125	052103	047511	
11086	053020	020116	051511	043440	
11087	053026	053111	047105	000	
11088	053033	127	044522	042524	EMS2: .ASCII /WRITE DATA ON NONEXISTANT SECTOR CAUSED IMPROPER REGISTER CHANGE/<15><1
11089	053040	042040	052101	020101	
11090	053046	047117	047040	047117	
11091	053054	054105	051511	040524	
11092	053062	052116	051440	041505	
11093	053070	047524	020122	040503	

11094	053076	051525	042105	044440	
11095	053104	050115	047522	042520	
11096	053112	020122	042522	044507	
11097	053120	052123	051105	041440	
11098	053126	040510	043516	006505	
11099	053134	012			
11100	053135	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11101	053142	040504	040524	043440	
11102	053150	053111	051505	053440	
11103	053156	040510	020124	044123	
11104	053164	052517	042114	041040	
11105	053172	020105	044124	051105	
11106	053200	006505	012		
11107	053203	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED WRITE DATA/
11108	053210	042526	020104	040504	
11109	053216	040524	043440	053111	
11110	053224	051505	051040	043505	
11111	053232	051511	042524	020122	
11112	053240	047503	052116	047105	
11113	053246	051524	040440	052106	
11114	053254	051105	040440	052124	
11115	053262	046505	052120	042105	
11116	053270	053440	044522	042524	
11117	053276	042040	052101	000101	
11118	053304	042522	042101	044040	EM53: .ASCIZ /READ HEADER AND DATA AFTER A SEARCH CAUSED DATA ERROR/
11119	053312	040505	042504	020122	
11120	053320	047101	020104	040504	
11121	053326	040524	040440	052106	
11122	053334	051105	040440	051440	
11123	053342	040505	041522	020110	
11124	053350	040503	051525	042105	
11125	053356	042040	052101	020101	
11126	053364	051105	047522	000122	
11127	053372	052101	042524	050115	EM54: .ASCII /ATTEMPTING COMMAND WITH INVALID ADDRESS CAUSED IMPROPER REGISTER CHANGE
11128	053400	044524	043516	041440	
11129	053406	046517	040515	042116	
11130	053414	053440	052111	020110	
11131	053422	047111	040526	044514	
11132	053430	020104	042101	051104	
11133	053436	051505	020123	040503	
11134	053444	051525	042105	044440	
11135	053452	050115	047522	042520	
11136	053460	020122	042522	044507	
11137	053466	052123	051105	041440	
11138	053474	040510	043516	006505	
11139	053502	012			
11140	053503	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11141	053510	040504	040524	043440	
11142	053516	053111	051505	053440	
11143	053524	040510	020124	044123	
11144	053532	052517	042114	041040	
11145	053540	020105	044124	051105	
11146	053546	006505	012		
11147	053551	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/
11148	053556	042526	020104	040504	
11149	053564	040524	043440	053111	

11150	053572	051505	051040	043505
11151	053600	051511	042524	020122
11152	053606	047503	052116	047105
11153	053614	051524	040440	052106
11154	053622	051105	047440	042520
11155	053630	040522	044524	047117
11156	053636	000		
11157	053637	127	044522	044524
11158	053644	043516	047440	020122
11159	053652	042522	042101	047111
11160	053760	020107	044527	044124
11161	053666	042440	050130	041505
11162	053674	042524	020104	042101
11163	053702	051104	051505	020123
11164	053710	053117	051105	046106
11165	053716	053517	042440	051122
11166	053724	051117	005015	
11167	053730	040503	051525	042105
11168	053736	044440	050115	047522
11169	053744	042520	020122	042522
11170	053752	044507	052123	051105
11171	053760	041440	040510	043516
11172	053766	006505	012	
11173	053771	107	047517	020104
11174	053776	040504	040524	043440
11175	054004	053111	051505	053440
11176	054012	040510	020124	044123
11177	054020	052517	042114	041040
11178	054026	020105	044124	051105
11179	054034	006505	012	
11180	054037	122	041505	044505
11181	054044	042526	020104	040504
11182	054052	041524	043440	053111
11183	054060	051505	051040	043505
11184	054066	051511	042524	020122
11185	054074	047503	052116	047105
11186	054102	051524	040440	052106
11187	054110	051105	047440	042520
11188	054116	040522	044524	047117
11189	054124	000		
11190	054125	104	052101	020101
11191	054132	042522	042101	053440
11192	054140	052111	020110	047101
11193	054146	042440	050130	041505
11194	054154	042524	020104	042101
11195	054162	051104	051505	020123
11196	054170	053117	051105	046106
11197	054176	053517	042440	051122
11198	054204	051117	044440	020123
11199	054212	047111	047503	051122
11200	054220	041505	006524	012
11201	054226	127	051117	020104
11202	054232	047516	020056	020061
11203	054240	047524	031040	030066
11204	054246	051440	047510	046125
11205	054254	020104	042502	051040

EM55: .ASCII /WRITING OR READING WITH EXPECTED ADDRESS OVERFLOW ERROR/<15><12>

.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>

.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>

.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION/

EM56: .ASCII /DATA READ WITH AN EXPECTED ADDRESS OVERFLOW ERROR IS INCORRECT/<15><12>

.ASCII /WORD NO. 1 TO 260 SHOULD BE READ, WORD NO 261 TO 266 SHOULD/<15><12>

11206	054262	040505	026104	053440	
11207	054270	051117	020104	047516	
11208	054276	031040	030466	052040	
11209	054304	020117	033062	020066	
11210	054312	044123	052517	042114	
11211	054320	005015			
11212	054322	042502	041440	040510	.ASCIZ /BE CHANGED/
11213	054330	043516	042105	000	
11214	054335	101	052124	046505	EM57: .ASCII /ATTEMPTING DATA COMMAND WITH WRONG FORMAT BIT CAUSED/<15><12>
11215	054342	052120	047111	020107	
11216	054350	040504	040524	041440	
11217	054356	046517	040515	042116	
11218	054364	053440	052111	020110	
11219	054372	051127	047117	020107	
11220	054400	047506	046522	052101	
11221	054406	041040	052111	041440	
11222	054414	052501	042523	006504	
11223	054422	012			
11224	054423	111	050115	047522	.ASCII /IMPROPER REGISTER CHANGE/<15><12>
11225	054430	042520	020122	042522	
11226	054436	044507	052123	051105	
11227	054444	041440	040510	043516	
11228	054452	006505	012		
11229	054455	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11230	054462	040504	040524	043440	
11231	054470	053111	051505	053440	
11232	054476	040510	020124	044123	
11233	054504	052517	042114	041040	
11234	054512	020105	044124	051105	
11235	054520	006505	012		
11236	054523	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER ATTEMPTED DATA TRANSFER/
11237	054530	042526	020104	040504	
11238	054536	040524	043440	053111	
11239	054544	051505	051040	043505	
11240	054552	051511	042524	020122	
11241	054560	047503	052116	047105	
11242	054566	051524	040440	052106	
11243	054574	051105	040440	052124	
11244	054602	046505	052120	042105	
11245	054610	042040	052101	020101	
11246	054616	051124	047101	043123	
11247	054624	051105	000		
11248	054627	101	052124	046505	EM60: .ASCII /ATTEMPTING TO MODIFY REGISTER DURING AN OPERATION CAUSED IMPROPER/<15><
11249	054634	052120	047111	020107	
11250	054642	047524	046440	042117	
11251	054650	043111	020131	042522	
11252	054656	044507	052123	051105	
11253	054664	042040	051125	047111	
11254	054672	020107	047101	047440	
11255	054700	042520	040522	044524	
11256	054706	047117	041440	052501	
11257	054714	042523	020104	046511	
11258	054722	051120	050117	051105	
11259	054730	005015			
11260	054732	042522	044507	052123	.ASCII /REGISTER CHANGE. GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11261	054740	051105	041440	040510	

11262	054746	043516	027105	043440
11263	054754	047517	020104	040504
11264	054762	040524	043440	053111
11265	054770	051505	053440	040510
11266	054776	020124	044123	052517
11267	055004	042114	041040	020105
11268	055012	044124	051105	006505
11269	055020	012		
11270	055021	122	041505	044505
11271	055026	042526	020104	040504
11272	055034	040524	043440	053111
11273	055042	051505	051040	043505
11274	055050	051511	042524	020122
11275	055056	047503	052116	047105
11276	055064	051524	040440	052106
11277	055072	051105	047440	042520
11278	055100	040522	044524	047117
11279	055106	053440	051501	040440
11280	055114	052124	046505	052120
11281	055122	042105	005015	
11282	055126	047515	043104	047111
11283	055134	020107	042522	020107
11284	055142	044507	042526	020123
11285	055150	042101	051104	051505
11286	055156	020123	043117	051040
11287	055164	043505	051511	042524
11288	055172	020122	042502	047111
11289	055200	020107	047515	044504
11290	055206	054506	042105	053440
11291	055214	044510	044103	041440
11292	055222	052501	042523	020104
11293	055230	051105	047522	000122
11294	055236	042504	044526	042503
11295	055244	047040	052117	040440
11296	055252	040526	040514	046102
11297	055260	020105	042502	047506
11298	055266	042522	041440	046517
11299	055274	040515	042116	053440
11300	055302	051501	052040	020117
11301	055310	042502	043440	053111
11302	055316	042511	000116	
11303	055322	044122	051504	020061
11304	055330	047503	052116	047105
11305	055336	051524	042040	051125
11306	055344	047111	020107	047503
11307	055352	046515	047101	020104
11308	055360	040527	020123	047111
11309	055366	042440	051122	051117
11310	055374	000		
11311	055375	122	041505	046101
11312	055402	041111	040522	042524
11313	055410	041440	046517	040515
11314	055416	042116	041440	052501
11315	055424	042523	020104	046511
11316	055432	051120	050117	051105
11317	055440	051040	043505	051511

.ASCII /RECEIVED DATA GIVES REGISTER CONTENTS AFTER OPERATION WAS ATTEMPTED/<15

.ASCIZ /MODFING REG GIVES ADDRESS OF REGISTER BEING MODIFIED WHICH CAUSED ERROR

EM61: .ASCIZ /DEVICE NOT AVAlABLE BEFORE COMMAND WAS TO BE GIVIEN/

EM63: .ASCIZ /RHDS1 CONTENTS DURING COMMAND WAS IN ERROR/

EM64: .ASCII /RECALIBRATE COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>



11318	055446	042524	020122	044103	
11319	055454	047101	042507	005015	
11320	055462	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11321	055470	052101	020101	044507	
11322	055476	042526	020123	044127	
11323	055504	052101	051440	047510	
11324	055512	046125	020104	042502	
11325	055520	052040	042510	042522	
11326	055526	005015			
11327	055530	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
11328	055536	042105	042040	052101	
11329	055544	020101	044507	042526	
11330	055552	020123	042522	044507	
11331	055560	052123	051105	041440	
11332	055566	047117	042524	052116	
11333	055574	020123	043101	042524	
11334	055602	020122	047503	046515	
11335	055610	047101	000104		
11336	055614	047111	042524	051122	EM65: .ASCIZ /INTERRUPT FAILING/
11337	055622	050125	020124	040506	
11338	055630	046111	047111	000107	
11339					
11340					
11341					
11342	055636	042510	042101	051105	EM66: .ASCII /HEADER AND DATA COMMAND FOR HEAD SELECTION TEST/<15><12>
11343	055644	040440	042116	042040	
11344	055652	052101	020101	047503	
11345	055660	046515	047101	020104	
11346	055666	047506	020122	042510	
11347	055674	042101	051440	046105	
11348	055702	041505	044524	047117	
11349	055710	052040	051505	006524	
11350	055716	012			
11351	055717	103	052501	042523	.ASCII /CAUSED ERROR/<15><12>
11352	055724	020104	051105	047522	
11353	055732	006522	012		
11354	055735	122	042110	052123	.ASCII /RHDST GIVES WHAT TRACK WAS BEING WRITTEN OR READ/<15><12>
11355	055742	043440	053111	051505	
11356	055750	053440	040510	020124	
11357	055756	051124	041501	020113	
11358	055764	040527	020123	042502	
11359	055772	047111	020107	051127	
11360	056000	052111	042524	020116	
11361	056006	051117	051040	040505	
11362	056014	006504	012		
11363	056017	117	020116	054503	.ASCIZ /ON CYLINDER 0, SECTOR 0/
11364	056024	044514	042116	051105	
11365	056032	030040	020054	042523	
11366	056040	052103	051117	030040	
11367	056046	000			
11368	056047	122	040505	020104	EM67: .ASCII /READ HEADER AND DATA ERROR IN HEAD SELECTION TEST/<12><15>
11369	056054	042510	042101	051105	
11370	056062	040440	042116	042040	
11371	056070	052101	020101	051105	
11372	056076	047522	020122	047111	
11373	056104	044040	040505	020104	

11374	056112	042523	042514	052103	
11375	056120	047511	020116	042524	
11376	056126	052123	006412		
11377	056132	044506	051522	020124	.ASCII /FIRST FOUR WORD NUMBERS ARE HEADER/<12><15>
11378	056140	047506	051123	053440	
11379	056146	051117	020104	052516	
11380	056154	041115	051105	020123	
11381	056162	051101	020105	042510	
11382	056170	042101	051105	006412	
11383	056176	047527	042122	047040	.ASCII /WORD NUMBERS 5 TO 260 ARE DATA WORDS/<12><15>
11384	056204	046525	042502	051522	
11385	056212	032440	052040	020117	
11386	056220	033062	020060	051101	
11387	056226	020105	040504	040524	
11388	056234	053440	051117	051504	
11389	056242	006412			
11390	056244	047111	042040	052101	.ASCIZ /IN DATA WORDS BITS 4,5,6,7,8 GIVE TRACK NUMBER/
11391	056252	020101	047527	042122	
11392	056260	020123	044502	051524	
11393	056266	032040	032454	033054	
11394	056274	033454	034054	043440	
11395	056302	053111	020105	051124	
11396	056310	041501	020113	052516	
11397	056316	041115	051105	000	
11398	056323	122	040505	020104	EM70: .ASCII /READ HEADER AND DATA ERROR IN/<15><12>
11399	056330	042510	042101	051105	
11400	056336	040440	042116	042040	
11401	056344	052101	020101	051105	
11402	056352	047522	020122	047111	
11403	056360	005015			
11404	056362	044504	043106	051105	.ASCII /DIFFERENCE LINE TEST/<15><12>
11405	056370	047105	042503	046040	
11406	056376	047111	020105	042524	
11407	056404	052123	005015		
11408	056410	047527	042122	047040	.ASCII /WORD NOS 1-4 GIVE HEADER/<15><12>
11409	056416	051517	030440	032055	
11410	056424	043440	053111	020105	
11411	056432	042510	042101	051105	
11412	056440	005015			
11413					.ASCII
11414	056442	047527	042122	047040	.ASCIZ /WORD NOS 5-260 GIVE DATA WHICH IS THE CYLINDER ADDRESS/
11415	056450	051517	032440	031055	
11416	056456	030066	043440	053111	
11417	056464	020105	040504	040524	
11418	056472	053440	044510	044103	
11419	056500	044440	020123	044124	
11420	056506	020105	054503	044514	
11421	056514	042116	051105	040440	
11422	056522	042104	042522	051523	
11423	056530	000			
11424	056531	106	051117	044503	EM71: .ASCII /FORCING OPI BY 3 INDEX PULSES/<15><12>
11425	056536	043516	047440	044520	
11426	056544	041040	020131	020063	
11427	056552	047111	042504	020130	
11428	056560	052520	051514	051505	
11429	056566	005015			

11430	056570	040503	051525	042105	.ASCII /CAUSED IMPROPER REGISTER CHANGE/<15><12>
11431	056576	044440	050115	047522	
11432	056604	042520	020122	042522	
11433	056612	044507	052123	051105	
11434	056620	041440	040510	043516	
11435	056626	006505	012		
11436	056631	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11437	056636	040504	040524	043440	
11438	056644	053111	051505	053440	
11439	056652	040510	020124	044123	
11440	056660	052517	042114	041040	
11441	056666	020105	044124	051105	
11442	056674	006505	C12		
11443	056677	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER 3 INDEX PULSES/
11444	056704	042526	020104	040504	
11445	056712	040524	043440	053111	
11446	056720	051505	051040	043505	
11447	056726	051511	042524	020122	
11448	056734	047503	052116	047105	
11449	056742	051524	040440	052106	
11450	056750	051105	031440	044440	
11451	056756	042116	054105	050040	
11452	056764	046125	042523	000123	
11453	056772	044127	046111	020105	EM72: .ASCII /WHILE USING UNIBUS B/<15><12>
11454	057000	051525	047111	020107	
11455	057006	047125	041111	051525	
11456	057014	041040	005015		
11457	057020	042522	042101	042040	.ASCII /READ DATA CAUSED IMPROPER REGISTER CHANGE/<15><12>
11458	057026	052101	020101	040503	
11459	057034	051525	042105	044440	
11460	057042	050115	047522	042520	
11461	057050	020122	042522	044507	
11462	057056	052123	051105	041440	
11463	057064	040510	043516	006505	
11464	057072	012			
11465	057073	107	047517	020104	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11466	057100	040504	040524	043440	
11467	057106	053111	051505	053440	
11468	057114	040510	020124	044123	
11469	057122	052517	042114	041040	
11470	057130	020105	044124	051105	
11471	057136	006505	012		
11472	057141	122	041505	044505	.ASCIZ /RECEIVED DATA GIVES WHAT WAS THERE AFTER COMMAND/
11473	057146	042526	020104	040504	
11474	057154	040524	043440	053111	
11475	057162	051505	053440	040510	
11476	057170	020124	040527	020123	
11477	057176	044124	051105	020105	
11478	057204	043101	042524	020122	
11479	057212	047503	046515	047101	
11480	057220	000104			
11481	057222	044127	046111	020105	EM73: .ASCII /WHILE USING UNIBUS B/<15><12>
11482	057230	051525	047111	020107	
11483	057236	047125	041111	051525	
11484	057244	041040	005015		
11485	057250	042522	042101	042040	.ASCIZ /READ DATA INCORRECT/

11486	057256	052101	020101	047111	
11487	057264	047503	051122	041505	
11488	057272	000124			
11489	057274	044127	046111	020105	EM74: .ASCII /WHILE USING UNIBUS B/<15><12>
11490	057302	051525	047111	020107	
11491	057310	047125	041111	051525	
11492	057316	041040	005015		
11493	057322	051127	052111	020105	.ASCII /WRITE DATA COMMAND CAUSED IMPROPER REGISTER CHANGE/<15><12>
11494	057330	040504	040524	041440	
11495	057336	046517	040515	042116	
11496	057344	041440	052501	042523	
11497	057352	020104	046511	051120	
11498	057360	050117	051105	051040	
11499	057366	043505	051511	042524	
11500	057374	020122	044103	047101	
11501	057402	042507	005015		
11502	057406	047507	042117	042040	.ASCII /GOOD DATA GIVES WHAT SHOULD BE THERE/<15><12>
11503	057414	052101	020101	044507	
11504	057422	042526	020123	044127	
11505	057430	052101	051440	047510	
11506	057436	046125	020104	042502	
11507	057444	052040	042510	042522	
11508	057452	005015			
11509	057454	042522	042503	053111	.ASCIZ /RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/
11510	057462	042105	042040	052101	
11511	057470	020101	044507	042526	
11512	057476	020123	042522	044507	
11513	057504	052123	051105	041440	
11514	057512	047117	042524	052116	
11515	057520	020123	043101	042524	
11516	057526	020122	047503	046515	
11517	057534	047101	000104		
11518	057540	044127	046111	020105	EM75: .ASCII /WHILE USING UNIBUS B/<15><12>
11519	057546	051525	047111	020107	
11520	057554	047125	041111	051525	
11521	057562	041040	005015		
11522	057566	051127	052111	020105	.ASCIZ /WRITE DATA COMMAND CHANGED WRITE FROM BUFFER/
11523	057574	040504	040524	041440	
11524	057602	046517	040515	042116	
11525	057610	041440	040510	043516	
11526	057616	042105	053440	044522	
11527	057624	042524	043040	047522	
11528	057632	020115	052502	043106	
11529	057640	051105	000		
11530	057643	127	044510	042514	EM76: .ASCII /WHILE USING UNIBUS B/<15><12>
11531	057650	052440	044523	043516	
11532	057656	052440	044516	052502	
11533	057664	020123	006502	012	
11534	057671	127	044522	042524	.ASCII /WRITE CHECK CAUSED IMPROPER REGISTER CHANGE/<15><12>
11535	057676	041440	042510	045503	
11536	057704	041440	052501	042523	
11537	057712	020104	046511	051120	
11538	057720	050117	051105	051040	
11539	057726	043505	051511	042524	
11540	057734	020122	044103	047101	
11541	057742	042507	005015		

11542	057746	047507	042117	042040		.ASCII	/GOOD DATA GIVES WHAT SHOULD BE THERE/	<15><12>
11543	057754	052101	020101	044507				
11544	057762	042526	020123	044127				
11545	057770	052101	051440	047510				
11546	057776	046125	020104	042502				
11547	060004	052040	042510	042522				
11548	060012	005015						
11549	060014	042522	042503	053111		.ASCIZ	/RECEIVED DATA GIVES REGISTER CONTENTS AFTER COMMAND/	
11550	060022	042105	042040	052101				
11551	060030	020101	044507	042526				
11552	060036	020123	042522	044507				
11553	060044	052123	051105	041440				
11554	060052	047117	042524	052116				
11555	060060	020123	043101	042524				
11556	060066	020122	047503	046515				
11557	060074	047101	000104					
11558	060100	041520	020040	020040	DH1:	.ASCII	/PC	WAT BIT REG REG RHCS1/<15><12>
11559	060106	020040	040527	020124				
11560	060114	020040	020040	044502				
11561	060122	020124	020040	020040				
11562	060130	042522	020107	020040				
11563	060136	020040	042522	020107				
11564	060144	020040	020040	044122				
11565	060152	051503	006461	012				
11566	060157	040	020040	020040		.ASCIZ	/	PC EXPECT ADDRESS CONTENT CONTENTS/
11567	060164	020040	050040	020103				
11568	060172	020040	020040	042440				
11569	060200	050130	041505	020124				
11570	060206	040440	042104	042522				
11571	060214	051523	041440	047117				
11572	060222	042524	052116	041440				
11573	060230	047117	042524	052116				
11574	060236	000123						
11575	060240	041520	020040	020040	DH4:	.ASCII	/PC	WAT BIT REG TIME IN/<15><12>
11576	060246	020040	040527	020124				
11577	060254	020040	020040	044502				
11578	060262	020124	020040	020040				
11579	060270	042522	020107	020040				
11580	060276	020040	044524	042515				
11581	060304	020040	047111	005015				
11582	060312	020040	020040	020040		.ASCIZ	/	PC EXPECT ADDRESS 10 MSEC/
11583	060320	020040	041520	020040				
11584	060326	020040	020040	054105				
11585	060334	042520	052103	020040				
11586	060342	042101	051104	051505				
11587	060350	020123	030061	020040				
11588	060356	051515	041505	000				
11589	060363	120	020103	020040	DH5:	.ASCII	/PC	REG GOOD RECEIVED/<15><12>
11590	060370	020040	051040	043505				
11591	060376	020040	020040	043440				
11592	060404	047517	020104	020040				
11593	060412	051040	041505	044505				
11594	060420	042526	006504	012				
11595	060425	040	020040	020040		.ASCIZ	/	ADDRESS DATA DATA/
11596	060432	020040	040440	042104				
11597	060440	042522	051523	042040				



11654	061134	020040	053440	051117						
11655	061142	020104	020040	043440						
11656	061150	047517	020104	020040						
11657	061156	041040	042101	005015						
11658	061164	020040	020040	020040		.ASCIZ /	NO	DATA	DATA/	
11659	061172	020040	047040	020117						
11660	061200	020040	020040	040504						
11661	061206	040524	020040	020040						
11662	061214	040504	040524	000						
11663	061221	120	020103	020040	DH51:	.ASCII /PC	REG	GOOD	RECVD	ILLEGL/<15><12>
11664	061226	020040	051040	043505						
11665	061234	020040	020040	043440						
11666	061242	047517	020104	020040						
11667	061250	051040	041505	042126						
11668	061256	020040	044440	046114						
11669	061264	043505	006514	012						
11670	061271	040	020040	020040		.ASCIZ /	ADDRESS	DATA	DATA	FUNCTN/
11671	061276	020040	040440	042104						
11672	061304	042522	051523	042040						
11673	061312	052101	020101	020040						
11674	061320	042040	052101	020101						
11675	061326	020040	043040	047125						
11676	061334	052103	000116							
11677	061340	041520	020040	020040	DH60:	.ASCII /PC	REG	GOOD	RECVD	MODFING/<15><12>
11678	061346	020040	042522	020107						
11679	061354	020040	020040	047507						
11680	061362	042117	020040	020040						
11681	061370	042522	053103	020104						
11682	061376	020040	047515	043104						
11683	061404	047111	006507	012						
11684	061411	040	020040	020040		.ASCIZ /	ADDRESS	DATA	DATA	REG/
11685	061416	020040	040440	042104						
11686	061424	042522	051523	042040						
11687	061432	052101	020101	020040						
11688	061440	042040	052101	020101						
11689	061446	020040	051040	043505						
11690	061454	000								
11691	061455	120	020103	020040	DH61:	.ASCII /PC	PC OF	RHDS1		/<15><12>
11692	061462	020040	050040	020103						
11693	061470	043117	020040	051040						
11694	061476	042110	030523	020040						
11695	061504	006440	012							
11696	061507	040	020040	020040		.ASCIZ /	JSR	WAS	/	
11697	061514	020040	045040	051123						
11698	061522	020040	020040	053440						
11699	061530	051501	020040	020040						
11700	061536	000040								
11701	061540	041520	020040	020040	DH62:	.ASCII /PC	PC OF	RHCS1/		<15><12>
11702	061546	020040	041520	047440						
11703	061554	020106	020040	044122						
11704	061562	051503	006461	012						
11705	061567	040	020040	020040		.ASCIZ /	JSR	WAS/		
11706	061574	020040	045040	051123						
11707	061602	020040	020040	053440						
11708	061610	051501	000							
11709	061613	040	020040	041520	DH65:	.ASCII / PC	TEST	CONT	CONT	CONT /<15><12>





```

11766 062260 000000
11767 062262 000 000 000 DF1: .BYTE 0,0,0,0,0,0
11768 062265 000 000 000
11769 062270 000 000 000 DF4: .BYTE 0,0,0,0,0,1
11770 062273 000 000 001
11771 062276 000 000 000 DF5: .BYTE 0,0,0,0
11772 062301 000
11773 062302 000 000 000 DF6: .BYTE 0,0,0
11774 062305 000 000 000 DF7: .BYTE 0,0
11775 062307 000 000 000 DF10: .BYTE 0,0,0,0,0,0,0
11776 062312 000 000 000
11777 062315 000
11778 062316 000 000 000 DF26: .BYTE 0,0,0,0,0,0,0
11779 062321 000 000 000
11780 062324 000
11781 062325 000 000 000 DF30: .BYTE 0,0,0,0
11782 062330 000
11783 062331 000 000 000 DF51: .BYTE 0,0,0,0,0
11784 062334 000 000
11785 062336 000 000 000 DF60: .BYTE 0,0,0,0,0
11786 062341 000 000
11787 062343 000 000 000 DF61: .BYTE 0,0,0
11788 062346 000 000 000 DF62: .BYTE 0,0,0
11789 062351 000 000 000 DF65: .BYTE 0,0,0,0,0
11790 062354 000 000
11791 062356 000 000 000 DF66: .BYTE 0,0,0,0,0,0,0
11792 062361 000 000
11793 062364 000
11794 062366 .EVEN
11795 .EVEN
11796
11797 000001 .END

```

ACL	=	000040	1178#																		
ACU	=	100000	1138#																		
AOE	=	001000	1070#																		
AS	=	002312	1269#	3297*	10085	11762															
ATA	=	100000	1058#	2316	3200	3452	3701	3808	7030	7269	8571	8995									
ATABLE	=	004576	1331#	1693																	
ATTENT	=	004650	1359#	1691*	1693*	2308	2470	2644	2794	3026	3191	3473	3703	3721	3810						
			7021	7260	8575																
ATO	=	000001	1090#																		
AT1	=	000002	1091#																		
AT2	=	000004	1092#																		
AT3	=	000010	1093#																		
AT4	=	000020	1094#																		
AT5	=	000040	1095#																		
AT6	=	000100	1096#																		
AT7	=	000200	1097#																		
A16	=	000400	1033#																		
A17	=	001000	1034#																		
BA	=	002270	1258#	10013																	
BAI	=	000010	1004#	2449																	
BASECH	=	036714	1523	9499#																	
BEGIN	=	004712	153	1372#	10358																
BEGIN1	=	004662	161	1366#																	
BEGIN2	=	004676	159	1369#																	
BIT0	=	000001	125#																		
BIT00	=	000001	115#	125																	
BIT01	=	000002	114#	124																	
BIT02	=	000004	113#	123																	
BIT03	=	000010	112#	122																	
BIT04	=	000020	111#	121																	
BIT05	=	000040	110#	120																	
BIT06	=	000100	109#	119	9793																
BIT07	=	000200	108#	118																	
BIT08	=	000400	107#	117	9594																
BIT09	=	001000	106#	116	9602	9971															
BIT1	=	000002	124#																		
BIT10	=	002000	105#	9956																	
BIT11	=	004000	104#	9609																	
BIT12	=	010000	103#																		
BIT13	=	020000	102#	9963																	
BIT14	=	040000	101#	9580																	
BIT15	=	100000	100#	3815																	
BIT2	=	000004	123#																		
BIT3	=	000010	122#																		
BIT4	=	000020	121#	9077																	
BIT5	=	000040	120#																		
BIT6	=	000100	119#																		
BIT7	=	000200	118#																		
BIT8	=	000400	117#																		
BIT9	=	001000	116#																		
BPTVEC	=	000014	132#																		
CA	=	002306	1267#	10077																	
CC	=	002330	1276#	10117																	
CHECK	=	034614	1932	2033	2596	2767	3284	3509	8978#												
CHECKC	=	034652	6582	8986#																	
CHECKT	=	034636	2151	2264	2357	2903	3077	3305	3567	3725	3867	4013	4148	4300	4482						



DH30	061127	552	574	590	607	668	702	726	762	798	905	916	947	966
		11653#												
DH4	060240	326	11575#											
DH5	060363	337	422	434	448	462	474	485	492	502	511	522	545	566
		584	600	618	629	643	655	680	693	715	755	774	786	811
		852	864	927	940	958	978	11589#						
DH51	061221	737	11663#											
DH6	060462	348	11600#											
DH60	061340	824	11677#											
DH61	061455	834	11691#											
DH62	061540	842	11701#											
DH65	061613	872	11709#											
DH66	061734	888	11723#											
DH7	060541	357	11609#											
DIG8	= 000004	1045#												
DISPLA	001140	234#	1401*	9623*	9955*									
DISPRE	000174	149#	1401											
DLT	= 100000	1016#												
DL64	= 000020	1047#												
DMD	= 000001	1080#												
DPR	= 000400	1051#	1896	2085	3007	3175	3434	3676	3783	4052	4187	4339	4522	4673
		4850	4991	5175	5326	5523	5704	5937	6089	6321	6479	6718	6868	7005
		7244	7815	8039	8278	8996								
DRY	= 000200	1050#	1898	2085	2983	3159	3452	3652	3759	3883	3906	6599	6989	7190
		7228	7455	7908	8142	8996								
DST	002300	1264#	10069											
DSMR	= 177570	46#	233											
DS1	002316	1271#	10037	11746	11749	11762								
DT	002320	1272#	1718*	2483	10141									
DTE	= 010000	1073#												
DTSY	= 001000	1086#												
DT1	062022	295	307	318	11735#									
DT10	062106	372	386	400	416	11746#								
DT26	062126	535	11749#											
DT30	062146	556	575	591	608	669	703	727	763	799	906	917	948	967
		11752#												
DT4	062040	331	11738#											
DT5	062056	341	426	438	452	466	478	486	493	503	512	523	546	567
		585	601	619	630	644	656	681	694	716	756	775	787	812
		853	865	931	941	959	979	11741#						
DT51	062160	742	11754#											
DT6	062070	351	11743#											
DT60	062174	829	11756#											
DT61	062210	837	11758#											
DT62	062220	845	11760#											
DT65	062230	877	11762#											
DT66	062244	895	11764#											
DT7	062100	359	11745#											
DVA	= 004000	1036#	1757	1909	2076	2990	3166	3417	3659	3766	4043	4178	4330	4513
		4664	4841	4982	5166	5317	5514	5695	5928	6080	6312	6470	6709	6859
		6996	7235	7806	8030	8269	8988							
ECH	= 000100	1067#												
ECl	= 004000	1151#	2697	2846	3815	3999	4133	4134	4135	4285	4286	4287	4468	4619
		4620	4621	4796	4937	4938	4939	5121	5272	5273	5274	5465	5645	5646
		5647	5883	6035	6036	6037	6268	6425	6426	6427	6661	6813	6814	6815
		7100	7101	7102	7345	7346	7347	7563	7651	7652	7653	7760	7761	7762



EM67	056047	898	11368#											
EM7	043640	355	10426#											
EM70	056323	909	11398#											
EM71	056531	921	11424#											
EM72	056772	11453#												
EM73	057222	945	11481#											
EM74	057274	934	952	11489#										
EM75	057540	963	11518#											
EM76	057643	971	11530#											
ERFLGS	004644	1353#	9949#											
ERR	= 040000	1057#	3887	8573										
ERRVEC	= 000004	128#	1502*	1509*	1529*	9585	9586*	9588*	9591*					
ERWORD	004512	1314#	9474*	9480*	11752									
ER1	002276	1263#	10045	11746	11749	11764								
ER2	002302	1265#	10053	11749	11764									
ER3	002310	1268#	10061	11749	11764									
EXT1	= 000001	1105#												
EXT10	= 000010	1108#												
EXT2	= 000002	1106#												
EXT20	= 000020	1109#												
EXT4	= 000004	1107#												
EXT40	= 000040	1110#												
FEN	= 000200	1131#												
FER	= 000020	1065#												
FILL	035504	9247#												
FILLRE	034460	2433	2436	2450	2453	2456	2459	2462	2465	2474	2479	2488	2491	2683
		2686	2832	2835	3019	3022	3029	3032	3187	3813	4064	4067	4070	4199
		4202	4205	4351	4354	4357	4534	4537	4540	4685	4688	4691	4862	4865
		4868	5003	5006	5009	5187	5190	5193	5338	5341	5344	555	5561	5564
		5716	5719	5722	5949	5952	5955	6101	6104	6107	6333	6336	6339	6342
		6345	6491	6494	6497	6730	6733	6737	6740	6743	6880	6883	6886	7017
		7256	7604	7607	7610	7693	7696	7699	7827	7830	7833	8051	8054	8057
		8290	8293	8296	8555	8906#								
FINACC	004574	1326#	9052*											
FINALA	004572	1325#	9053*											
FIRST	004646	1354#	1414	1447*										
FLHEAD	034402	3953	3973	5808	5825	5845	5858	6009	6193	6210	6230	6243	6398	6613
		6632	6781	7070	7315	7466	7529	8849#						
FMT22	= 010000	1152#	2695	2844	3815	3998	3999	4000	4133	4134	4135	4285	4286	4287
		4467	4468	4469	4619	4620	4621	4795	4796	4797	4937	4938	4939	5120
		5121	5122	5272	5273	5274	5464	5465	5466	5645	5646	5647	5882	5883
		5884	6035	6036	6037	6267	6268	6269	6425	6426	6427	6660	6661	6662
		6813	6814	6815	7100	7101	7102	7345	7346	7347	7562	7563	7564	7651
		7652	7653	7760	7761	7762	7985	7986	7987	8219	8220	8221	8392	8393
		8394	8502	8503	8504	8633	8634	8635						
FUTABL	002334	1286#												
GNS	= ***** U	148	1421	1425	1429	1433	1437	1441	1445	1479	1517	1521	1569	1592
		1596	1600	1604	1632	1638	1644	1696	1703	1710	1851	1857	1884	1890
		2575	2584	3272	3276	3495	3501	3505	5544	5552	8436	8442	8696	8702
		8756	8762	9352	9358	9365	9369	9373	9380	9384	9502	9508	9519	9525
		9531	9535	9541	9547	9554	10003	10011	10019	10027	10035	10043	10051	10059
		10067	10075	10083	10091	10099	10107	10115	10123	10131	10139	10147	10312	10313
		10314	10315	10316	10317	10318	10319	10320						
GO	= 000001	1030#	1955	2057	2176	2288	2414	2667	2817	2972	2990	3149	3166	3290
		3398	3417	3461	3516	3641	3659	3748	3766	3875	3898	4025	4043	4160
		4178	4312	4330	4494	4513	4646	4664	4822	4841	4964	4982	5147	5166

		5299	5317	5495	5514	5677	5695	5910	5928	6062	6080	6294	6312	6452
		6470	6589	6687	6709	6841	6859	6978	6996	7119	7180	7216	7235	7364
		7444	7589	7678	7788	7806	7900	8012	8030	8134	8251	8269	8411	8525
		8649	3077											
GRV	= 000010	1046#												
HCE	= 000200	1068#												
HCI	= 002000	1150#	2699	2848	3815	4000	4135	4287	4469	4621	4797	4939	5122	5274
		5466	5647	5884	6037	6269	6427	6662	6815	7102	7347	7564	7653	7762
		7987	8221	8394	8504	8635								
HCRC	= 000400	1069#												
HT.	= 003323	1963#	1968	2067#	2072	2184#	2189	2296#	2301	2422#	2427	2675#	2680	2824#
		2829	2981#	2986	3157#	3162	3298#	3303	3408#	3413	3525#	3530	3650#	3655
		3757#	3762	3881#	3886	3904#	3909	4034#	4039	4169#	4174	4321#	4326	4504#
		4509	4655#	4660	4832#	4837	4973#	4978	5157#	5162	5308#	5313	5505#	5510
		5686#	5691	5919#	5924	6071#	6076	6303#	6308	6461#	6466	6597#	6602	6700#
		6705	6850#	6855	6987#	6992	7126#	7131	7188#	7193	7226#	7231	7371#	7376
		7453#	7458	7596#	7601	7685#	7690	7797#	7802	7906#	7911	8021#	8026	8140#
		8145	8260#	8265	8418#	8423	8532#	8537	8656#	8661				
IAE	= 002000	1071#												
IE	= 000100	1031#	1744	1784	2972	2990	2994	2995	3148	3166	3421	3422	3641	3659
		3663	3664	3748	3766	3770	3771	3875	3898	4025	4043	4160	4178	4312
		4330	4494	4513	4646	4664	4822	4841	4964	4982	5147	5166	5299	5317
		5495	5514	5677	5695	5910	5928	6062	6080	6294	6312	6452	6470	6589
		6687	6709	6841	6859	6978	6996	7119	7180	7216	7235	7364	7444	7589
		7678	7788	7806	7900	8012	8030	8134	8251	8269	8411	8525	8649	9103
		9115	9184	9198										
ILF	= 000001	1061#												
ILLEGL	002376	1304#	11754											
ILR	= 000002	1062#												
IOTVEC	= 000020	133#	1382#	1383#										
IR	= 000100	1007#	1867	2447	8543	8544								
IXE	= 004000	1135#												
KIPAR0	= 172340	196#												
KIPAR1	= 172342	197#												
KIPAR2	= 172344	198#												
KIPAR3	= 172346	199#												
KIPAR4	= 172350	200#												
KIPAR5	= 172352	201#												
KIPAR6	= 172354	202#												
KIPAR7	= 172356	203#												
KIPDR0	= 172300	185#												
KIPDR1	= 172302	186#												
KIPDR2	= 172304	187#												
KIPDR3	= 172306	188#												
KIPDR4	= 172310	189#												
KIPDR5	= 172312	190#												
KIPDR6	= 172314	191#												
KIPDR7	= 172316	192#												
LA	002332	1277#	10109											
LBT	= 002000	1053#	8995											
LT.	= 000105	1963#	1968	2067#	2072	2184#	2189	2296#	2301	2422#	2427	2675#	2680	2824#
		2829	2981#	2986	3157#	3162	3298#	3303	3408#	3413	3525#	3530	3650#	3655
		3757#	3762	3881#	3886	3904#	3909	4034#	4039	4169#	4174	4321#	4326	4504#
		4509	4655#	4660	4832#	4837	4973#	4978	5157#	5162	5308#	5313	5505#	5510
		5686#	5691	5919#	5924	6071#	6076	6303#	6308	6461#	6466	6597#	6602	6700#
		6705	6850#	6855	6987#	6992	7126#	7131	7188#	7193	7226#	7231	7371#	7376





PCLCSR	002256	1246*	1461	9050*	9077*													
PCLCTR	002262	1248*	9051															
PGE	= 002000	1011*																
PIP	= 020000	1056*	3007	3175	3434	3454	3676	3783	7005	7244								
PIRQ	= 177772	45*																
PIRQVE	= 000240	139*																
PKACK	002372	1302*	1934	1954	2035	2056	2075	3289	3515									
PLU	= 020000	1137*																
PRE	= 000020	1177*																
PRITEM	041404	1469*	2004*	2736*	3507*	8735*	8776*	9349*	9991*	10152*	10169	10173*						
PROG	= 001000	1052*																
PRO	= 000000	62*																
PR1	= 000040	63*																
PR2	= 000100	64*																
PR3	= 000140	65*																
PR4	= 000200	66*																
PR5	= 000240	67*																
PR6	= 000300	68*																
PR7	= 000340	69*																
PS	= 177776	42*	43	1407*	1743*	1783*	1838*	8753*	9348*									
PSEL	= 000000	1035*	5471	5495	5514	5653	5677	5695	8227	8251	8269							
PSU	= 000001	1174*																
PSW	= 177776	43*																
PUTREG	036472	1860	1893	9182	9197	9296	9403*											
PURVEC	= 000024	134*	1388*	1389*	10329*	10330*	10338*	10353*	10354*									
RDCHR	= 104405	9851	10317*															
RDLIN	= 104406	9896	10318*															
RDOCT	= 104407	1481	9375	9386	9510	9527	10319*											
RDY	= 000200	1032*	1744	1757	1784	1909	2076	2186	2298	2424	2826	2990	3166	3410				
		3417	3659	3766	4036	4171	4323	4506	4657	4834	4975	5159	5310	5507				
		5688	5921	6073	6305	6463	6702	6852	6996	7128	7235	7373	7598	7687				
		7799	8023	8262	8420	8534	8658	8988										
READAT	002360	1297*	4288	4311	4329	4622	4645	4663	4940	4963	4981	5275	5298	5316				
		5648	5676	5694	8636	8648												
READIN	002374	1303*	2647	2666	2797	2816												
RECALI	002340	1289*	2952	2971	2989	3127	3147	3165	3897	6588	7179							
REFOR	002362	1298*	4136	4159	4177	6038	6061	6079	6428	6451	6469	6816	6840	6858				
		7103	7118	7348	7363	7654	7677											
REGADR	004510	1313*	1546*	1920*	2082*	2091*	3004*	3013*	3172*	3181*	3431*	3440*	3673*	3682*				
		3780*	3789*	4049*	4058*	4184*	4193*	4336*	4345*	4519*	4528*	4670*	4679*	4847*				
		4856*	4988*	4997*	5172*	5181*	5323*	5332*	5520*	5529*	5701*	5710*	5934*	5943*				
		6086*	6095*	6318*	6327*	6476*	6485*	6715*	6724*	6865*	6874*	7002*	7011*	7241*				
		7250*	7812*	7821*	8036*	8045*	8275*	8284*	9305*	11741	11743	11754	11756					
REGSAV	041242	9948*																
REGSA1	041250	1500	9950*															
REINTO	003444	1308*	3974	3981	4097	4116	4129	4204	4234	4281	4356	4382	4446	4451				
		4566	4586	4591	4615	4690	4717	4779	4894	4914	4933	5008	5035	5099				
		5104	5219	5239	5244	5268	5343	5370	5443	5448	5591	5612	5617	5641				
		5721	5749	5846	5853	5859	5866	5982	6003	6031	6106	6137	6231	6238				
		6244	6251	6373	6392	6421	6496	6526	6773	6809	6885	6912	7064	7096				
		7137	7309	7341	7382	7545	7647	7698	7728	8610	8629	8667						
RELEAS	002344	1291*																
RESVEC	= 000010	129*																
RETCL	002370	1301*	3727	3747	3765													
RHAS	002234	1230*	1538	1571	2644*	2794*	3721*											
RHBA	002212	1218*	2437	4068	4203	4355	4538	4689	4866	5007	5191	5342	5562	5720				

RHCA	002230	5953	6105	6337	6495	6734	6884	7608	7697	7831	8055	8294	8541	9433*
RHCC	002252	1228*	2463	2684	2833	6340	6741	8926*	8944*	9430*				
RHCS1	002216	1237*	1918	1920	1922	2404	3020	3188	6343	6738	7018	7257	9052	9409
		1223*	1934*	1957*	2035*	2059*	2154*	2178*	2185	2266*	2290*	2297	2416*	2423
		2496	2647*	2669*	2797*	2819*	2825	2952*	2974*	3030	3127*	3150*	3203	3292*
		3378*	3400*	3409	3457	3466	3518*	3621*	3643*	3690	3727*	3750*	3797	3877*
		3900*	4027*	4035	4162*	4170	4314*	4322	4496*	4505	4648*	4656	4824*	4833
		4966*	4974	5149*	5158	5301*	5309	5471*	5497*	5506	5653*	5679*	5687	5912*
		5920	6064*	6072	6296*	6304	6454*	6462	6591*	6689*	6701	6843*	6851	6980*
		7033	7121*	7127	7182*	7218*	7272	7366*	7372	7446*	7591*	7597	7680*	7686
		7790*	7798	7902*	8014*	8022	8136*	8227*	8253*	8261	8413*	8419	8527*	8533
		8548	8651*	8657	8928*	8945*	8954*	8958	9103	9115	9184	9198	9439*	9504
		9537												
RHCS2	002214	1219*	1572	1625*	2443	5539	8542	8959	9436*					
RHDB	002206	1216*	1504	9511										
RHDST	002222	1225*	2454	2687	2836	4071	4206	4358	4541	4692	4869	5010	5194	5345
		5565	5723	5956	6108	6346	6498	6744	6887	7611	7700	7834	8058	8297
		8556	8927*	9431*										
RHDS1	002240	1232*	1964	1972	2068	2097	2312	2480	2676	2702	2851	2982	3033	3158
		3196	3266	3299	3448	3526	3651	3697	3758	3804	3882	3887	3905	6598
		6988	7026	7189	7227	7265	7454	7907	8141	8449	8567	8708	8960	
RHDT	002242	1233*	1626	1628	1640	1712	1716	1718						
RHEC1	002246	1235*	2489											
RHEC2	002250	1236*	2492											
RHER1	002220	1224*	1575	2451	8560	8961								
RHER2	002224	1226*	2457											
RHER3	002232	1229*	2466											
RHLA	002254	1238*	9053											
RHMR	002236	1231*	2475											
RHOF	002226	1227*	2460	2691	2840	3023	3814	8953*	9437*					
RHSN	002244	1234*	1705	1715	1717									
RHWC	002210	1217*	1940	2042	2161	2217	2273	2359	2434	2598	2653	2769	2803	2905
		2957	3079	3133	3308	3383	3569	3626	3733	4007	4065	4142	4200	4294
		4352	4476	4535	4628	4686	4804	4863	4946	5004	5129	5188	5281	5339
		5476	5559	5658	5717	5892	5950	6044	6102	6276	6334	6434	6492	6669
		6731	6823	6881	6963	7201	7571	7605	7660	7694	7770	7828	7994	8052
		8232	8291	8428	8510	8540	8687	8911	9222	9305	9407	9409	9432*	
RMR =	000004	1063*												
RPTRP1	010130	1739	1756*											
RPTRP2	010226	1779	1794*											
RPVEC	002204	987*	1409	1738	1778	1946*	2048*	2167*	2279*	2405*	2659*	2809*	2963*	3139*
		3389*	3632*	3739*	4016*	4151*	4303*	4485*	4637*	4813*	4955*	5138*	5290*	5485*
		5667*	5901*	6053*	6285*	6443*	6581*	6678*	6832*	6969*	7110*	7172*	7207*	7355*
		7436*	7580*	7669*	7779*	7892*	8003*	8126*	8241*	8402*	8516*	8640*	9521	9528*
		9543	9557*											
RPVECT	037516	1410	9551*	9557										
RP4VEC	004516	1316*	1464*	1468*	1946	2048	2167	2279	2405	2659	2809	2963	3139	3389
		3632	3739	4016	4151	4303	4485	4637	4813	4955	5138	5290	5485	5667
		5901	6053	6285	6443	6581	6678	6832	6969	7110	7172	7207	7355	7436
		7580	7669	7779	7892	8003	8126	8241	8402	8516	8640			
RUN	036532	3988	4123	4276	4458	4610	4786	4928	5111	5263	5455	5636	5872	6025
		6257	6415	6650	6803	7090	7335	7552	7641	7750	7976	8210	8383	8493
		8624	9430*											
RO =	%000000	50*	1409*	1410*	1411*	1503*	1506*	1574*	1578*	1612*	1615*	1620*	1653*	1692*
		1693	1738*	1739*	1740*	1778*	1779*	1780*	1906*	1907*	1909	1939*	1971*	1982*
		2041*	2063*	2078	2081	2096*	2108*	2160*	2194*	2217*	2218*	2220*	2222*	2223*

2225*	2227*	2229*	2231*	2233*	2235*	2239	2243*	2246*	2255*	2256	2259*	2272*
2311*	2323*	2359*	2360*	2362*	2364*	2365*	2367*	2369*	2371*	2373*	2375*	2377*
2381	2385*	2388*	2397*	2398	2401*	2433*	2436*	2442*	2450*	2453*	2456*	2459*
2462*	2465*	2474*	2479*	2488*	2491*	2495*	2507*	2598*	2600*	2602*	2604*	2605*
2607*	2609*	2611*	2613*	2615*	2617*	2621	2625*	2628*	2637*	2638	2641*	2652*
2683*	2686*	2690*	2701*	2713*	2769*	2771*	2773*	2775*	2776*	2778*	2780*	2782*
2784*	2786*	2788*	2790*	2791*	2802*	2832*	2835*	2839*	2850*	2862*	2905*	2907*
2909*	2911*	2912*	2914*	2916*	2918*	2920*	2922*	2924*	2928	2932*	2935*	2944*
2945	2948*	2956*	2977*	2993	3000	3003	3019*	3022*	3029*	3032*	3040*	3079*
3082*	3086*	3090*	3093*	3097*	3101*	3105*	3109*	3113*	3117*	3121*	3124*	3132*
3153*	3168	3171	3187*	3195*	3202*	3210*	3308*	3312*	3316*	3320*	3323*	3327*
3331*	3335*	3339*	3343*	3347*	3353	3357*	3360*	3369*	3370	3375*	3382*	3404*
3420	3427	3430	3447*	3456*	3465*	3478*	3522*	3569*	3571*	3574*	3577*	3579*
3582*	3585	3588*	3591*	3593*	3595*	3599	3603*	3606*	3615*	3616	3619*	3625*
3646*	3662	3669	3672	3689*	3696*	3709*	3732*	3753*	3769	3776	3779	3796*
3803*	3813*	3819*	3869*	3953*	3962*	3973*	3980*	3988*	4006*	4030*	4045	4048
4064*	4067*	4070*	4076*	4096*	4115*	4123*	4141*	4165*	4180	4183	4199*	4202*
4205*	4213*	4232*	4268*	4276*	4293*	4317*	4332	4335	4351*	4354*	4357*	4364*
4380*	4431*	4436*	4445*	4450*	4458*	4475*	4499*	4515	4518	4534*	4537*	4540*
4547*	4565*	4585*	4590*	4597*	4602*	4610*	4627*	4651*	4666	4669	4685*	4688*
4691*	4698*	4715*	4769*	4778*	4786*	4803*	4827*	4843	4846	4862*	4865*	4868*
4875*	4893*	4913*	4920*	4928*	4945*	4969*	4984	4987	5003*	5006*	5009*	5016*
5033*	5084*	5089*	5098*	5103*	5111*	5128*	5152*	5168	5171	5187*	5190*	5193*
5200*	5218*	5238*	5243*	5250*	5255*	5263*	5280*	5304*	5319	5322	5338*	5341*
5344*	5351*	5368*	5428*	5433*	5442*	5447*	5455*	5475*	5500*	5516	5519	5558*
5561*	5564*	5571*	5590*	5611*	5616*	5623*	5628*	5636*	5657*	5682*	5697	5700
5716*	5719*	5722*	5729*	5747*	5808*	5817*	5825*	5834*	5845*	5852*	5858*	5865*
5872*	5891*	5915*	5930	5933	5949*	5952*	5955*	5961*	5981*	6002*	6009*	6018*
6025*	6043*	6067*	6082	6085	6101*	6104*	6107*	6115*	6135*	6193*	6202*	6210*
6219*	6230*	6237*	6243*	6250*	6257*	6275*	6299*	6314	6317	6333*	6336*	6339*
6342*	6345*	6352*	6372*	6391*	6398*	6407*	6415*	6433*	6457*	6472	6475	6491*
6494*	6497*	6504*	6524*	6594*	6613*	6623*	6632*	6642*	6650*	6668*	6692*	6711
6714	6730*	6733*	6737*	6740*	6743*	6750*	6772*	6781*	6789*	6794*	6803*	6822*
6846*	6861	6864	6880*	6883*	6886*	6893*	6910*	6958*	6962*	6983*	6998	7001
7017*	7025*	7032*	7041*	7063*	7070*	7077*	7082*	7090*	7135*	7185*	7196*	7200*
7221*	7237	7240	7256*	7264*	7271*	7280*	7308*	7315*	7322*	7327*	7335*	7380*
7434*	7449*	7466*	7476*	7481*	7486*	7491*	7496*	7504*	7508	7509*	7515*	7516
7517*	7523*	7529*	7536*	7544*	7552*	7570*	7604*	7607*	7610*	7617*	7641*	7659*
7693*	7696*	7699*	7706*	7726*	7750*	7769*	7793*	7808	7811	7827*	7830*	7833*
7840*	7890*	7918*	7923*	7928*	7933*	7938*	7946*	7950	7951*	7957*	7958	7959*
7963*	7968*	7976*	7993*	8017*	8032	8035	8051*	8054*	8057*	8064*	8124*	8152*
8157*	8162*	8167*	8172*	8180*	8184	8185*	8191*	8192	8193*	8197*	8202*	8210*
8231*	8256*	8271	8274	8290*	8293*	8296*	8303*	8375*	8383*	8427*	8448*	8458*
8485*	8493*	8509*	8547*	8555*	8559*	8566*	8582*	8609*	8616*	8624*	8665*	8686*
8707*	8717*	8779*	8781	8813*	8816	8852	8853	8857	8862*	8882	8883	8884
8891*	8909	8910	8915*	8926	8927	8930*	8944	8946*	8953	8955*	9019	9020
9021	9028*	9078	9082*	9083	9085	9086	9087	9088	9089	9141*	9161	9163*
9164	9166	9167	9168	9169	9207*	9220	9221	9223	9225	9227	9232*	9252
9253	9254	9255	9265*	9290	9291	9292	9293	9294*	9322*	9404	9407*	9410
9415*	9430	9431	9432	9433	9435	9437	9439	9440*	9467	9468	9469	9470
9471*	9495*	9511*	9513*	9514*	9641	9651*	9655	9671	9672	9685*	9717	9718*
9719	9722*	9893	9897*	9898	9901	9921*	9924*	10155	10156*	10157*	10164*	10165*
10166*	10167*	10168*	10169	10171	10173	10174	10179	10185	10187*	10188	10202*	10295
10296*	10297	10298*	10299*	10300*	10301*	10331	10352*					
51*	1504*	1505	1511	1538*	1539*	1540	1544	1546	1571*	1580	1583	1621
1744*	1757	1784*	1906	1912	2063	2082	2977	3004	3153	3172	3404	3431

		3522	3646	3673	3753	3780	4030	4049	4165	4184	4317	4336	4499	4519
		4651	4670	4827	4847	4969	4988	5152	5172	5304	5323	5500	5520	5682
		5701	5915	5934	6067	6086	6299	6318	6457	6476	6594	6692	6715	6846
		6865	6983	7002	7185	7221	7241	7449	7503	7506*	7508*	7516*	7519*	7793
		7812	7945	7948*	7950*	7958*	7961*	8017	8036	8179	8182*	8184*	8192*	8195*
		8256	8275	8780*	8781	8784	8850	8852*	8857*	8861*	8879	8882*	8885*	8890*
		8907	8909*	8911*	8912*	8914*	8958*	8965*	8986	8991	9016	9019*	9022	9027*
		9079	9087*	9121*	9122	9131*	9132	9140*	9218	9220*	9222*	9225*	9227*	9231*
		9248	9252*	9257*	9264*	9285	9290*	9301	9303	9308	9321*	9405	9408*	9410*
		9414*	9462	9467*	9475	9478	9482	9494*	9512*	9515*	9642	9655*	9656	9660
		9684*	9894	9899*	9907*	9909*	9911*	9914*	9917	9920*	10184	10185*	10190	10194
R2	=%000002	10196	10201*	10332	10351*									
		52*	1505*	1572*	1573*	1577*	1868	1871	2241	2244*	2250*	2253*	2383	2386*
		2392*	2395*	2623	2626*	2632*	2635*	2930	2933*	2939*	2942*	3355	3358*	3364*
		3367*	3601	3604*	3610*	3613*	8851	8853*	8858*	8860*	8880	8883*	8886*	8889*
		8908	8910*	8912	8913*	8959*	8963*	8964*	9017	9020*	9022*	9026*	9080	9088*
		9121	9130*	9131	9139*	9219	9221*	9228*	9230*	9249	9253*	9259*	9263*	9286
		9291*	9301	9304	9308	9320*	9406	9409*	9411*	9413*	9463	9468*	9475	9479
		9482	9493*	9643	9654*	9658*	9661	3668*	9669*	9670	9675*	9683*	9895	9900*
R3	=%000003	9908*	9910*	9912*	9918	9919*	10333	10350*						
		53*	1613*	1614*	1617*	1647*	1861	1863	1894	1896	1898	1900	1902	2064
		2091	2580	2589	2978	3013	3154	3181	3405	3440	3523	3647	3682	3754
		3789	4031	4058	4166	4193	4318	4345	4500	4528	4652	4679	4828	4856
		4970	4997	5153	5181	5305	5332	5501	5529	5683	5710	5916	5943	6068
		6095	6300	6327	6458	6485	6595	6693	6724	6847	6874	6984	7011	7186
		7222	7250	7450	7794	7821	8018	8045	8257	8284	8881	8884*	8885	8888*
		8960*	8980	8985	8998	9018	9021*	9023*	9025*	9081	9092*	9095*	9097*	9100*
		9138*	9162	9172*	9175*	9177*	9180*	9206*	9250	9254*	9257	9258*	9262*	9287
		9292*	9313*	9319*	9464	9469*	9472	9480	9487*	9492*	9644	9652*	9653*	9667*
		9670*	9679*	9680*	9682*	9847	9848*	9849	9852*	9853	9857	9859	9861*	9863*
R4	=%000004	10242	10251*	10257*	10258*	10261*	10266*	10267*	10268	10277*	10334	10349*		
		54*	1575*	1576*	2240	2248*	2254*	2382	2390*	2396*	2622	2630*	2636*	2929
		2937*	2943*	3354	3362*	3368*	3600	3608*	3614*	8961*	9251	9255*	9258	9261*
		9288	9293*	9306	9318*	9465	9472*	9473*	9474	9491*	10243	10245*	10246*	10247*
R5	=%000005	10248	10249*	10263	10265*	10273*	10276*	10335	10348*					
		55*	1618*	1625	1634	1647	1649	1652*	2064*	2087	2090	2245*	2251*	2387*
		2393*	2627*	2633*	2934*	2940*	2978*	3009	3012	3154*	3177	3180	3359*	3365*
		3405*	3436	3439	3523*	3605*	3611*	3647*	3678	3681	3754*	3785	3788	4031*
		4054	4057	4166*	4189	4192	4318*	4341	4344	4500*	4524	4527	4652*	4675
		4678	4828*	4852	4855	4970*	4993	4996	5153*	5177	5180	5305*	5328	5331
		5501*	5525	5528	5683*	5706	5709	5916*	5939	5942	6068*	6091	6094	6300*
		6323	6326	6458*	6481	6484	6595*	6693*	6720	6723	6847*	6870	6873	6984*
		7007	7010	7186*	7222*	7246	7249	7450*	7505*	7510*	7794*	7817	7820	7947*
		7952*	8018*	8041	8044	8181*	8186*	8257*	8280	8283	9289	9298*	9300*	9305
		9317*	9466	9470*	9481	9490*	9645	9647*	9649*	9656*	9660*	9675	9681*	10244
		10250*	10252*	10254*	10255*	10256*	10257	10275*	10336	10347*				
R6	=%000006	56*	58	1377*	1378*	1379								
R7	=%000007	57*	59											
SAVER	034716	1939	2041	2160	2272	2652	2802	2956	3132	3382	3625	3732	4006	4141
		4293	4475	4627	4803	4945	5128	5280	5475	5657	5891	6043	6275	6433
		6668	6822	6962	7200	7570	7659	7769	7993	8231	8427	8509	8686	9015*
SAVERE	004522	1324*	1941	1984	2043	2110	2162	2196	2274	2305*	2308*	2325	2404*	2430*
		2439*	2440*	2471*	2483*	2485*	2509	2654	2715	2804	2864	2958	3026*	3042
		3134	3191*	3212	3384	3473*	3480	3627	3703*	3711	3734	3810*	3821	4008
		4078	4143	4215	4295	4366	4477	4549	4629	4700	4805	4877	4947	5018
		5130	5202	5282	5353	5477	5573	5659	5731	5893	5963	6045	6117	6277

SC	=	100000	6354	6435	6506	6670	6752	6824	6895	6964	7021*	7043	7202	7260*	7282
SC1	=	000100	7572	7619	7661	7708	7771	7842	7995	8066	8233	8305	8429	8460	8511
SC10	=	001000	8540*	8541*	8544*	8545*	8575*	8584	8688	8719	8912*	9225*	9227*	9297	
SC2	=	000200	1039*	1907	2997	3207	3424	3470	3666	3694	3773	3801	7037	7276	8552
SC20	=	002000	1111*												
SC4	=	000400	1114*												
SEECOM		002364	1112*												
SEEKCY		034526	1115*												
SELECT		004636	1113*												
SERCH		002346	1299*	6977	6995	7215	7234	7443	7899	8133	8945				
SKI	=	040000	6958	7196	7434	7890	8124	8944*							
SN		002322	1346*	1367*	1369*	1372*	1473	1659	8767						
SND1		005776	1292*	8928											
SP	=	%000006	1180*												
			1273*	1717*	2485	10149									
			1417	1447*											
			58*	1381*	1395*	1396*	1402	1403	1404	1482*	1483	1484	1499*	1510	1523*
			1537*	1634*	1640*	1649*	1684*	1698*	1705*	1712*	1736*	1756	1776*	1794	1831*
			1853*	1866*	1867*	1868	1886*	1954*	1955*	1957	2026*	2056*	2057*	2059	2075*
			2076*	2077	2078	2085*	2086	2087	2144*	2175*	2176*	2178	2239*	2240*	2241*
			2253	2254	2255	2287*	2288*	2290	2350*	2381*	2382*	2383*	2395	2396	2397
			2413*	2414*	2416	2468*	2470*	2471	2555*	2577*	2586*	2621*	2622*	2623*	2635
			2636	2637	2666*	2667*	2669	2760*	2816*	2817*	2819	2896*	2928*	2929*	2930*
			2942	2943	2944	2971*	2972*	2974	2989*	2990*	2995*	2996*	2997*	2999	3000
			3007*	3008	3009	3070*	3147*	3148*	3150	3165*	3166*	3167	3168	3175*	3176
			3177	3247*	3266*	3267*	3268	3289*	3290*	3292	3353*	3354*	3355*	3367	3368
			3369	3397*	3398*	3400	3416*	3417*	3422*	3423*	3424*	3426	3427	3434*	3435
			3436	3497*	3515*	3516*	3518	3550*	3599*	3600*	3601*	3613	3614	3615	3640*
			3641*	3643	3658*	3659*	3664*	3665*	3666*	3668	3669	3676*	3677	3678	3747*
			3748*	3750	3765*	3766*	3771*	3772*	3773*	3775	3776	3783*	3784	3785	3860*
			3874*	3875*	3877	3897*	3898*	3900	3943*	4024*	4025*	4027	4042*	4043*	4044
			4045	4052*	4053	4054	4159*	4160*	4162	4177*	4178*	4179	4180	4187*	4188
			4189	4259*	4311*	4312*	4314	4329*	4330*	4331	4332	4339*	4340	4341	4422*
			4493*	4494*	4496	4512*	4513*	4514	4515	4522*	4523	4524	4645*	4646*	4648
			4663*	4664*	4665	4666	4673*	4674	4675	4760*	4821*	4822*	4824	4840*	4841*
			4842	4843	4850*	4851	4852	4963*	4964*	4966	4981*	4982*	4983	4984	4991*
			4992	4993	5075*	5146*	5147*	5149	5165*	5166*	5167	5168	5175*	5176	5177
			5298*	5299*	5301	5316*	5317*	5318	5319	5326*	5327	5328	5417*	5494*	5495*
			5497	5513*	5514*	5515	5516	5523*	5524	5525	5676*	5677*	5679	5694*	5695*
			5696	5697	5704*	5705	5706	5798*	5909*	5910*	5912	5927*	5928*	5929	5930
			5937*	5938	5939	6061*	6062*	6064	6079*	6080*	6081	6082	6089*	6090	6091
			6183*	6293*	6294*	6296	6311*	6312*	6313	6314	6321*	6322	6323	6451*	6452*
			6454	6469*	6470*	6471	6472	6479*	6480	6481	6559*	6588*	6589*	6591	6686*
			6687*	6689	6708*	6709*	6710	6711	6718*	6719	6720	6840*	6841*	6843	6858*
			6859*	6860	6861	6868*	6869	6870	6977*	6978*	6980	6995*	6996*	6997	6998
			7005*	7006	7007	7118*	7119*	7121	7164*	7179*	7180*	7182	7215*	7216*	7218
			7234*	7235*	7236	7237	7244*	7245	7246	7363*	7364*	7366	7425*	7443*	7444*
			7446	7503*	7519	7588*	7589*	7591	7677*	7678*	7680	7787*	7788*	7790	7805*
			7806*	7807	7808	7815*	7816	7817	7880*	7899*	7900*	7902	7945*	7961	8011*
			8012*	8014	8029*	8030*	8031	8032	8039*	8040	8041	8107*	8133*	8134*	8136
			8179*	8195	8250*	8251*	8253	8268*	8269*	8270	8271	8278*	8279	8280	8354*
			8410*	8411*	8413	8438*	8524*	8525*	8527	8542*	8543*	8545	8648*	8649*	8651
			8698*	8758*	8764*	8772*	8808*	8850*	8851*	8860	8861	8879*	8880*	8881*	8888
			8889	8890	8907*	8908*	8913	8914	8978	8980*	8981*	8983	8985*	8986*	8987*
			8988	8995*	8996	9016*	9017*	9018*	9025	9026	9027	9078*	9079*	9080*	9081*
			9082	9089*	9138	9139	9140	9141	9161*	9162*	9163	9169*	9206	9207	9218*













SCM1 =	000006	245#	246#	247#	248#	249#	250#	251#						
SCM2 =	000014	245#	246#	247#	248#	249#	250#	251#						
SCM3 =	000006	243#	245#											
SCM4 =	000006	251#	252#	253#	254#	255#	256#	257#						
SCNTLC	041075	9803	9871#											
SCRFL	001221	261#	1648	1700	1707	1714	2579	2588	5546	5547	5554	5555	9362	9731
		9768	9871	9930	9966	9979	10154	10178	10183					
SDBLK	040244	9652	9686	9694#										
SDOAGN	034356	8803	8814	8820#										
SOTBL	040234	9655	9690#											
SENDAD	034346	155	8816#											
SENDOCT	034314	8805#												
SENDNG	034362	8771	8807	8822#										
SENULL	034377	8774	8810	8825#										
SEOP	034260	1526	1606	1610	8778	8795#								
SEOPCT	034306	8802#	8806											
SERFLG	001103	218#	9570	9598	9600	9606*	9627	9953*	9979					
SERMAX	001115	224#	1392*	9600	9622*	9627								
SERROR	041242	1384	1527	9945#										
SERRPC	001116	225#	9960*	9961*	9962	9979	10160	11735	11738	11741	11743	11745	11746	11749
		11752	11754	11756	11758	11760	11762	11764						
SERRTB	001224	279#	10168											
SERTY	041406	9965	9992#											
SERTTL	001112	222#	8764	8766*	9959*	9979								
SESCAP	001212	258#	1391*	9621*	9974	9976	9979							
SFILLC	001154	241#	9734	9768										
SFILLS	001153	240#	9768											
SGADR	001120	226#												
SGODAT	001124	228#	1545*	1921*	2077*	2086*	2999*	3008*	3167*	3176*	3426*	3435*	3668*	3677*
		3775*	3784*	4044*	4053*	4179*	4188*	4331*	4340*	4514*	4523*	4665*	4674*	4842*
		4851*	4983*	4992*	5167*	5176*	5318*	5327*	5515*	5524*	5696*	5705*	5929*	5938*
		6081*	6090*	6313*	6322*	6471*	6480*	6710*	6719*	6860*	6869*	6997*	7006*	7236*
		7245*	7807*	7816*	8031*	8040*	8270*	8279*	9303*	9478*	11741	11752	11754	11756
		8811#												
SGET42	034336	11												
SHO =	000000													
SHIOCT	041240	9918#	9929#											
SICNT	001104	219#	9613#	9614	9616*	9626								
SILLUP	043154	10329	10360#											
SITEMB	001114	223#	9962#	9979	10157									
SLF	001222	262#	9768	9862	9871	9930	9979							
SLPADR	001106	220#	1393*	1683*	3549*	9377*	9388	9604*	9619*	9624	9626			
SLPERR	001110	221#	1394*	8478*	9360	9387*	9604	9620*	9626	9973				
SMAIL =	***** U	1405	1499	1537	1563	1684	1732	1772	1831	2019	2144	2350	2555	2760
		2896	3070	3247	3550	3858	3943	4259	4422	4760	5075	5417	5798	6183
		6559	7164	7425	7880	8107	8354	8753	9619	9719	9968			
SMXCNT	040026	9617	9626#											
SNULL	001152	239#	9736	9768										
SNWTST=	000001	1493#	1495	1531#	1533	1551#	1553	1671#	1673	1723#	1725	1765#	1767	1803#
		1805	2009#	2011	2135#	2137	2342#	2344	2526#	2528	2738#	2740	2887#	2889
		3062#	3064	3225#	3227	3532#	3534	3851#	3853	3916#	3918	4247#	4249	4393#
		4395	4731#	4733	5046#	5048	5384#	5386	5767#	5769	6150#	6152	6540#	6542
		7151#	7153	7399#	7401	7859#	7861	8084#	8086	8327#	8329	8742#	8744	
SOCNT	042756	10241#	10270*	10283#										
SOMODE	042760	10236#	10240*	10245	10248*	10259*	10285#							
SOVER	040012	9581	9597	9605	9615	9623#								
SPASS	001100	216#	1843	2567	3259	8366	8770*	8772	8799*	8800*	8808	8822	9611	9627





ALLREG	21#	9999	10008	10016	10024	10032	10040	10048	10056	10064	10072	10080	10088	10096	10104
	10112	10120	10128	10136	10144										
CHANGR	21#	1970	2095	2307	2310	2441	2494	2689	2700	2838	2849	3025	3190	3194	3201
	3446	3455	3464	3471	3688	3695	3702	3795	3802	3809	7020	7024	7031	7259	7263
	7270	8447	8546	8558	8565	8574	8706								
CHECKD	21#	2151	2264	2357	2903	3077	3304	3567	3725	3867	4013	4148	4300	4482	4634
	4810	4952	5135	5287	5482	5664	5898	6050	6282	6440	6675	6829	7107	7352	7577
	7666	7776	8000	8238	8399										
CHECKV	21#	1932	2033	2596	2767	3284	3509								
CLEARA	21#	3962	3980	4115	4268	4431	4436	4445	4450	4585	4590	4597	4602	4769	4778
	4913	4920	5084	5089	5098	5103	5238	5243	5250	5255	5428	5433	5442	5447	5611
	5616	5623	5628	5817	5834	5852	5865	6002	6018	6202	6219	6237	6250	6391	6407
	6622	6641	6772	6788	6794	7062	7077	7082	7308	7322	7327	7476	7481	7486	7491
	7496	7523	7536	7544	7918	7923	7928	7933	7938	7963	7968	8152	8157	8162	8167
	8172	8197	8202	8374	8485	8609	8616								
CMPBLK	21#	4094	4230	4378	4563	4713	4891	5031	5216	5366	5588	5745	5979	6133	6370
	6522	6908	7134	7379	7724	8664									
CMREGI	21#	1981	2107	2193	2322	2506	2712	2861	3039	3209	3477	3708	3818	4075	4212
	4363	4546	4697	4874	5015	5199	5350	5570	5728	5960	6114	6351	6503	6749	6892
	7040	7279	7616	7705	7839	8063	8302	8457	8581	8716					
COMMEN	1#	11	140#												
DATAO	21#	3987	4122	4275	4457	4609	4785	4927	5110	5262	5454	5635	5871	6024	6256
	6414	6648	6801	7089	7334	7551	7640	7749	7975	8209	8382	8492	8623		
DISREG	21#														
DUM	21#	1934	2035	2154	2266	2647	2797	2952	3127	3378	3621	3727			
ENDCOM	1#	16	140#												
ERROR	40#	1512	1547	1750	1761	1795	1864	1872	1903	1913	1923	1990	2083	2092	2116
	2202	2331	2515	2721	2870	3005	3014	3048	3173	3182	3218	3432	3441	3486	3674
	3683	3717	3781	3790	3826	3889	4050	4059	4084	4104	4185	4194	4221	4240	4337
	4346	4372	4388	4520	4529	4555	4573	4671	4680	4706	4723	4848	4857	4883	4901
	4989	4998	5024	5041	5173	5182	5208	5226	5324	5333	5359	5376	5521	5530	5579
	5598	5702	5711	5737	5755	5935	5944	5969	5989	6087	6096	6124	6144	6319	6328
	6360	6380	6477	6486	6513	6533	6716	6725	6757	6866	6875	6901	6918	7003	7012
	7049	7143	7242	7251	7288	7388	7625	7715	7735	7813	7822	7848	8037	8046	8072
	8276	8285	8311	8466	8590	8673	8725	8992	8999	9105	9107	9117	9126	9135	9186
	9189	9201													
ESCAPE	1#	140#													
FIHEAD	21#	3952	3972	5807	5824	5844	5857	6008	6192	6209	6229	6242	6397	6611	6630
	6779	7069	7314	7465	7528										
FILLBL	21#														
FLSVRE	21#	2433	2436	2450	2453	2456	2459	2462	2465	2474	2478	2488	2491	2683	2686
	2832	2835	3019	3022	3029	3032	3187	3813	4064	4067	4070	4199	4202	4205	4351
	4354	4357	4534	4537	4540	4685	4688	4691	4862	4865	4868	5003	5006	5009	5187
	5190	5193	5338	5341	5344	5558	5561	5564	5716	5719	5722	5949	5952	5955	6101
	6104	6107	6333	6336	6339	6342	6345	6491	6494	6497	6730	6733	6736	6740	6743
	6880	6883	6886	7017	7256	7604	7607	7610	7693	7696	7699	7827	7830	7833	8051
	8054	8057	8290	8293	8296	8555									
GETPRI	1#														
GOO	21#	1953	2055	2174	2286	2412	2665	2815	2970	3146	3288	3396	3514	3639	3746
	3873	3896	4023	4158	4310	4492	4644	4820	4962	5145	5297	5493	5675	5908	6060
	6292	6450	6587	6685	6839	6976	7117	7178	7214	7362	7442	7587	7676	7786	7898
	8010	8132	8249	8409	8523	8647									
LOAD	21#	2218	2220	2222	2223	2225	2227	2229	2231	2233	2235	2237	2259	2360	2362
	2364	2365	2367	2369	2371	2373	2375	2377	2379	2401	2600	2602	2604	2605	2607
	2609	2611	2613	2615	2617	2619	2641	2771	2773	2775	2776	2778	2780	2782	2784
	2786	2788	2790	2791	2907	2909	2911	2912	2914	2916	2918	2920	2922	2924	2926





.SSAVE	1#	
.SSB20	1#	
.SSB20	1#	
.SSCOP	1#	9564
.SSIZE	1#	
.SSUPR	1#	
.STRAP	1#	10286
.STYPB	1#	
.STYPD	1#	9627
.STYPE	1#	9695
.STYPO	1#	10208
.S40CA	1#	



ADD	3842	6932	6934	6939	9130	9131	9258	9300	9376	9660	9723	9914	10168	10237	10247
ASL	7517	7959	8193	9907	9909	9911	10165	10166	10167	10299					
ASLB	9665														
BCC	1624	2247	2389	2629	2936	3361	3607	7518	7960	8194	9666				
BEQ	1475	1489	1542	1584	1587	1627	1629	1661	1759	1862	1870	1895	1897	1899	1901
	1911	1919	2080	2089	2563	2590	2992	3002	3011	3170	3179	3255	3269	3419	3429
	3438	3661	3671	3680	3768	3778	3787	3841	3888	3912	4047	4056	4182	4191	4334
	4343	4517	4526	4668	4677	4845	4854	4986	4995	5170	5179	5321	5330	5518	5527
	5540	5699	5708	5932	5941	6084	6093	6316	6325	6474	6483	6713	6722	6863	6872
	7000	7009	7239	7248	7810	7819	8034	8043	8118	8273	8282	8362	8768	8778	8782
	8814	8990	8997	9104	9116	9185	9199	9224	9302	9312	9476	9486	9595	9597	9599
	9603	9612	9726	9832	9902	9954	9957	9972	9975	10175	10180	10186	10197	10264	
BGE	9615	9760													
BGT	8803	9674	9904	10271											
BHI	9601														
BHIS	9125	9134													
BIC	1482	1907	2305	2308	2430	2470	2994	2995	3267	3421	3422	3473	3663	3664	3770
	3771	8543	8544	8800	8987	8995	9227	9310	9484	9513	9800	9913	9993	10261	
BIS	1867	1955	2057	2076	2176	2222	2288	2364	2414	2440	2604	2644	2667	2775	2794
	2817	2911	2972	2990	2996	2997	3026	3090	3148	3166	3191	3290	3320	3398	3417
	3423	3424	3516	3577	3641	3659	3665	3666	3703	3748	3766	3772	3773	3810	3875
	3898	4025	4043	4160	4178	4312	4330	4494	4513	4646	4664	4822	4841	4964	4982
	5147	5166	5299	5317	5471	5495	5514	5653	5677	5695	5910	5928	6062	6080	6294
	6312	6452	6470	6589	6687	6709	6841	6859	6978	6996	7021	7119	7180	7216	7235
	7260	7364	7444	7589	7678	7788	7806	7900	8012	8030	8134	8227	8251	8269	8411
	8525	8545	8575	8649	8953	8981	9225	9435	9514	9668	9669	10266	10267		
BISB	10157														
BIT	1513	1565	1586	1861	1894	1896	1898	1900	2580	2589	3887	5539	9093	9098	9103
	9115	9173	9178	9184	9198	9580	9594	9602	9609	9956	9963	9971			
BITB	3840	9748													
BLOS	9850														
BLT	9657	9673	9739	9906	10272										
BMI	9664														
BNE	1380	1415	1507	1514	1566	1579	1616	1654	1747	1787	1840	1842	1844	2252	2304
	2394	2429	2566	2568	2581	2634	2941	3258	3260	3366	3463	3612	3845	6942	7511
	7953	8187	8365	8367	8859	8887	9024	9094	9096	9099	9101	9174	9176	9179	9181
	9229	9260	9314	9412	9488	9516	9581	9610	9662	9720	9728	9735	9749	9756	9802
	9807	9815	9837	9854	9860	9964	9995	10158	10170	10189	10262	10346			
BPL	9648	9678	9714	9753	9969	10260									
BR	1368	1371	1399	1416	1420	1424	1428	1432	1436	1440	1444	1466	1478	1508	1516
	1520	1568	1591	1595	1599	1603	1631	1637	1643	1646	1695	1702	1709	1753	1791
	1845	1850	1856	1883	1889	2569	2574	2583	3261	3271	3275	3494	3500	3504	3847
	3914	5543	5551	6943	8368	8435	8441	8695	8701	8755	8761	8783	8982	9106	9111
	9120	9128	9188	9193	9204	9226	9351	9357	9364	9368	9372	9379	9383	9501	9507
	9518	9524	9530	9534	9540	9546	9553	9583	9589	9592	9605	9608	9659	9676	9716
	9732	9742	9751	9758	9856	9915	9928	9996	10002	10010	10018	10026	10034	10042	10050
	10058	10066	10074	10082	10090	10098	10106	10114	10122	10130	10138	10146	10163	10172	10192
	10199	10238	10253	10274	10340	10361									
CLC	7514	7956	8190												
CLR	1367	1370	1372	1373	1378	1390	1391	1545	1573	1607	1609	1618	1619	1691	1921
	2244	2386	2439	2626	2933	3358	3507	3604	5425	8766	8769	8797	8798	8965	9050
	9075	9076	9348	9607	9621	9651	9654	9787	9827	9828	9899	9900	10152	10156	10251
	10344														
CLRB	1582	9606	9680	9757	9861	9924									
CMP	1379	1402	1510	1626	1628	1756	1757	1794	1868	1909	2078	2087	3000	3009	3168
	3177	3268	3427	3436	3669	3678	3776	3785	4045	4054	4180	4189	4332	4341	4515

	4524	4666	4675	4843	4852	4984	4993	5168	5177	5319	5328	5516	5525	5697	5706
	5930	5939	6082	6091	6314	6323	6472	6481	6711	6720	6861	6870	6998	7007	7237
	7246	7808	7817	8032	8041	8271	8280	8781	8988	8996	9122	9132	9168	9301	9308
CMPB	9311	9475	9482	9485	9590	9614	9672	9801	9806	9814	9836	9849	9994	10169	10171
DEC	9596	9600	9725	9727	9734	9755	9759	9853	9859	9903	9905				
	1506	1578	1615	1653	1657	1746	1786	2251	2393	2633	2940	3365	3611	3910	6941
	7510	7952	8186	8777	8801	8858	8886	9023	9095	9100	9175	9180	9228	9259	9313
DECB	9411	9487	9515	9833	10164										
EMT	9738	9741	10259	10270											
40															
HALT	148	1859	1892	3278	3508	8444	8704	9549	9558	9715	9970	10339	10360		
INC	1577	1651	1652	2250	2392	2632	2939	3364	3610	6931	6933	6935	6936	6937	6938
	6940	8770	8799	9473	9613	9658	9811	9813	9835	9959	10265	10273	10345		
INCB	3839	9618	9761	9953											
IOT	41														
JMP	153	159	161	1417	1459	1526	1585	1588	1606	1610	1847	2564	2571	3256	3263
	3849	5424	5548	6944	8116	8119	8363	8370	8775	8785	8821	9805	9997		
JSR	1413	1564	1687	1737	1777	1834	1860	1878	1893	1928	1932	1939	1971	1982	2023
	2029	2033	2041	2096	2108	2147	2151	2160	2194	2213	2264	2272	2311	2323	2353
	2357	2433	2436	2442	2450	2453	2456	2459	2462	2465	2474	2479	2488	2491	2495
	2507	2558	2592	2596	2652	2683	2686	2690	2701	2713	2763	2767	2802	2832	2835
	2839	2850	2862	2899	2903	2956	3019	3022	3029	3032	3040	3073	3077	3132	3187
	3195	3202	3210	3250	3280	3284	3305	3382	3447	3456	3465	3478	3509	3553	3563
	3567	3625	3689	3696	3709	3725	3732	3796	3803	3813	3819	3835	3863	3867	3869
	3892	3946	3953	3962	3973	3980	3988	4006	4013	4064	4067	4070	4076	4096	4111
	4115	4123	4141	4148	4199	4202	4205	4213	4232	4262	4268	4276	4293	4300	4351
	4354	4357	4364	4380	4425	4431	4436	4445	4450	4458	4475	4482	4534	4537	4540
	4547	4565	4581	4585	4590	4597	4602	4610	4627	4634	4685	4688	4691	4698	4715
	4763	4769	4778	4786	4803	4810	4862	4865	4868	4875	4893	4909	4913	4920	4928
	4945	4952	5003	5006	5009	5016	5033	5078	5084	5089	5098	5103	5111	5128	5135
	5187	5190	5193	5200	5218	5234	5238	5243	5250	5255	5263	5280	5287	5338	5341
	5344	5351	5368	5420	5428	5433	5442	5447	5455	5475	5482	5558	5561	5564	5571
	5590	5607	5611	5616	5623	5628	5636	5657	5664	5716	5719	5722	5729	5747	5801
	5808	5817	5825	5834	5845	5852	5858	5865	5872	5891	5898	5949	5952	5955	5961
	5981	5998	6002	6009	6018	6025	6043	6050	6101	6104	6107	6115	6135	6186	6193
	6202	6210	6219	6230	6237	6243	6250	6257	6275	6282	6333	6336	6339	6342	6345
	6352	6372	6387	6391	6398	6407	6415	6433	6440	6491	6494	6497	6504	6524	6562
	6605	6613	6623	6632	6642	6650	6668	6675	6730	6733	6737	6740	6743	6750	6768
	6772	6781	6789	6794	6803	6822	6829	6880	6883	6886	6893	6910	6953	6958	6962
	7017	7025	7032	7041	7063	7070	7077	7082	7090	7107	7135	7167	7196	7200	7256
	7264	7271	7280	7304	7308	7315	7322	7327	7335	7352	7380	7428	7434	7460	7466
	7476	7481	7486	7491	7496	7523	7529	7536	7544	7552	7570	7577	7604	7607	7610
	7617	7636	7641	7659	7666	7693	7696	7699	7706	7726	7745	7750	7769	7776	7827
	7830	7833	7840	7883	7890	7913	7918	7923	7928	7933	7938	7963	7968	7976	7993
	8000	8051	8054	8057	8064	8110	8124	8147	8152	8157	8162	8167	8172	8197	8202
	8210	8231	8238	8290	8293	8296	8303	8357	8375	8383	8399	8427	8448	8458	8481
	8485	8493	8509	8547	8555	8559	8566	8582	8604	8609	8616	8624	8665	8686	8707
	8717	8816	9182	9197	9296	9306	9481	9733	9740	9747	9804	9965			
MOV	1366	1369	1377	1381	1382	1383	1384	1385	1386	1387	1388	1389	1393	1394	1395
	1396	1397	1400	1401	1403	1404	1407	1408	1409	1410	1411	1447	1463	1464	1467
	1468	1469	1483	1484	1491	1498	1499	1500	1502	1503	1504	1505	1509	1511	1523
	1527	1529	1536	1537	1538	1539	1544	1546	1562	1571	1572	1574	1575	1576	1608
	1612	1613	1614	1617	1620	1621	1625	1634	1640	1647	1649	1655	1656	1663	1682
	1683	1684	1685	1692	1698	1705	1712	1717	1718	1734	1736	1738	1739	1740	1743
	1744	1745	1774	1776	1778	1779	1780	1783	1784	1785	1831	1832	1838	1853	1863
	1866	1871	1886	1902	1906	1912	1920	1922	1934	1946	1954	1957	2004	2026	2027

2035	2048	2056	2059	2063	2064	2075	2077	2081	2082	2085	2086	2090	2091	2144
2145	2154	2167	2175	2178	2217	2218	2220	2223	2225	2227	2229	2231	2233	2235
2239	2240	2241	2243	2245	2248	2253	2254	2255	2259	2266	2279	2287	2290	2350
2351	2359	2360	2362	2365	2367	2369	2371	2373	2375	2377	2381	2382	2383	2385
2387	2390	2395	2396	2397	2401	2404	2405	2413	2416	2468	2471	2483	2485	2555
2556	2577	2586	2598	2600	2602	2605	2607	2609	2611	2613	2615	2617	2621	2622
2623	2625	2627	2630	2635	2636	2637	2641	2647	2659	2666	2669	2736	2760	2761
2769	2771	2773	2776	2778	2780	2782	2784	2786	2788	2790	2791	2797	2809	2816
2819	2896	2897	2905	2907	2909	2912	2914	2916	2918	2920	2922	2924	2928	2929
2930	2932	2934	2937	2942	2943	2944	2948	2952	2963	2971	2974	2977	2978	2989
2993	2999	3003	3004	3007	3008	3012	3013	3070	3071	3079	3082	3086	3093	3097
3101	3105	3109	3113	3117	3121	3124	3127	3139	3147	3150	3153	3154	3165	3167
3171	3172	3175	3176	3180	3181	3247	3248	3266	3289	3292	3308	3312	3316	3323
3327	3331	3335	3339	3343	3347	3353	3354	3355	3357	3359	3362	3367	3368	3369
3375	3378	3389	3397	3400	3404	3405	3416	3420	3426	3430	3431	3434	3435	3439
3440	3497	3515	3518	3522	3523	3549	3550	3551	3569	3571	3574	3579	3582	3585
3588	3591	3593	3595	3599	3600	3601	3603	3605	3608	3613	3614	3615	3619	3621
3632	3640	3643	3646	3647	3658	3662	3668	3672	3673	3676	3677	3681	3682	3721
3727	3739	3747	3750	3753	3754	3765	3769	3775	3779	3780	3783	3784	3788	3789
3858	3860	3861	3874	3877	3897	3900	3943	3944	4016	4024	4027	4030	4031	4042
4044	4048	4049	4052	4053	4057	4058	4151	4159	4162	4165	4166	4177	4179	4183
4184	4187	4188	4192	4193	4259	4260	4303	4311	4314	4317	4318	4329	4331	4335
4336	4339	4340	4344	4345	4422	4423	4485	4493	4496	4499	4500	4512	4514	4518
4519	4522	4523	4527	4528	4637	4645	4648	4651	4652	4663	4665	4669	4670	4673
4674	4678	4679	4760	4761	4813	4821	4824	4827	4828	4840	4842	4846	4847	4850
4851	4855	4856	4955	4963	4966	4969	4970	4981	4983	4987	4988	4991	4992	4996
4997	5075	5076	5138	5146	5149	5152	5153	5165	5167	5171	5172	5175	5176	5180
5181	5290	5298	5301	5304	5305	5316	5318	5322	5323	5326	5327	5331	5332	5416
5417	5418	5485	5494	5497	5500	5501	5513	5515	5519	5520	5523	5524	5528	5529
5541	5667	5676	5679	5682	5683	5694	5696	5700	5701	5704	5705	5709	5710	5798
5799	5901	5909	5912	5915	5916	5927	5929	5933	5934	5937	5938	5942	5943	6053
6061	6064	6067	6068	6079	6081	6085	6086	6089	6090	6094	6095	6183	6184	6285
6293	6296	6299	6300	6311	6313	6317	6318	6321	6322	6326	6327	6443	6451	6454
6457	6458	6469	6471	6475	6476	6479	6480	6484	6485	6559	6560	6570	6571	6572
6573	6574	6575	6576	6577	6578	6579	6581	6588	6591	6594	6595	6603	6678	6686
6689	6692	6693	6708	6710	6714	6715	6718	6719	6723	6724	6832	6840	6843	6846
6847	6858	6860	6864	6865	6868	6869	6873	6874	6969	6977	6980	6983	6984	6995
6997	7001	7002	7005	7006	7010	7011	7110	7118	7121	7164	7165	7172	7179	7182
7185	7186	7207	7215	7218	7221	7222	7234	7236	7240	7241	7244	7245	7249	7250
7355	7363	7366	7425	7426	7436	7443	7446	7449	7450	7503	7504	7505	7506	7508
7515	7516	7519	7580	7588	7591	7669	7677	7680	7779	7787	7790	7793	7794	7805
7807	7811	7812	7815	7816	7820	7821	7880	7881	7892	7899	7902	7945	7946	7947
7948	7950	7957	7958	7961	8003	8011	8014	8017	8018	8029	8031	8035	8036	8039
8040	8044	8045	8107	8108	8126	8133	8136	8179	8180	8181	8182	8184	8191	8192
8195	8241	8250	8253	8256	8257	8268	8270	8274	8275	8278	8279	8283	8284	8354
8355	8402	8410	8413	8438	8478	8516	8524	8527	8540	8541	8542	8640	8648	8651
8698	8735	8752	8753	8758	8764	8772	8776	8779	8780	8784	8804	8808	8813	8850
8851	8852	8853	8857	8860	8861	8879	8880	8881	8882	8883	8884	8885	8888	8889
8890	8907	8908	8909	8910	8912	8913	8914	8926	8927	8928	8944	8945	8954	8958
8959	8960	8961	8963	8964	8978	8980	8983	8985	8986	8991	8998	9016	9017	9018
9019	9020	9021	9022	9025	9026	9027	9051	9052	9053	9077	9078	9079	9080	9081
9082	9083	9085	9086	9087	9088	9089	9092	9097	9102	9114	9138	9139	9140	9141
9161	9162	9163	9164	9166	9167	9169	9172	9177	9183	9200	9206	9207	9218	9219
9220	9221	9230	9231	9248	9249	9250	9251	9252	9253	9254	9255	9257	9261	9262
9263	9264	9285	9286	9287	9288	9289	9290	9291	9292	9293	9294	9298	9303	9304
9305	9309	9317	9318	9319	9320	9321	9349	9354	9360	9377	9387	9388	9404	9405



.BYTE	217	218	223	224	239	240	241	242	1331	3990	3991	4125	4126	4278	4279
	4460	4461	4612	4613	4788	4789	4930	4931	5113	5114	5265	5266	5457	5458	5638
	5639	5874	5875	6027	6028	6259	6260	6417	6418	6652	6653	6805	6806	7092	7093
	7337	7338	7554	7555	7643	7644	7752	7753	7978	7979	8212	8213	8385	8386	8495
	8496	8626	8627	8825	9868	9869	10281	10282	10283	10284	11767	11769	11771	11773	11774
	11775	11778	11781	11783	11785	11787	11788	11789	11791						
.ENABL	1														
.END	11797														
.ENDC	6	15	19	29	31	32	33	40	126	140	154	182	193	204	206
	208	215	217	243	251	257	258	259	260	264	983	989	1023	1025	1198
	1377	1381	1382	1384	1386	1388	1390	1391	1393	1395	1405	1422	1426	1430	1434
	1438	1442	1446	1480	1494	1495	1496	1497	1498	1499	1518	1522	1532	1533	1534
	1535	1536	1537	1552	1553	1560	1561	1562	1563	1570	1593	1597	1601	1605	1633
	1639	1645	1672	1673	1680	1681	1682	1683	1684	1697	1704	1711	1724	1725	1730
	1731	1732	1766	1767	1770	1771	1772	1804	1805	1829	1830	1831	1852	1858	1885
	1891	1957	1961	1969	1971	1977	2010	2011	2017	2018	2019	2059	2063	2065	2077
	2095	2096	2102	2136	2137	2142	2143	2144	2178	2182	2190	2218	2220	2222	2223
	2225	2227	2229	2231	2233	2235	2237	2259	2261	2290	2294	2302	2310	2311	2317
	2343	2344	2348	2349	2350	2360	2362	2364	2365	2367	2369	2371	2373	2375	2377
	2379	2401	2403	2416	2420	2428	2443	2450	2495	2501	2527	2528	2553	2554	2555
	2576	2585	2600	2602	2604	2605	2607	2609	2611	2613	2615	2617	2619	2641	2643
	2669	2673	2681	2690	2698	2700	2701	2707	2739	2740	2758	2759	2760	2771	2773
	2775	2776	2778	2780	2782	2784	2786	2788	2790	2791	2793	2819	2822	2830	2839
	2847	2849	2850	2856	2878	2879	2894	2895	2896	2907	2909	2911	2912	2914	2916
	2918	2920	2922	2924	2926	2948	2950	2972	2974	2976	2977	2979	2999	3017	3029
	3063	3064	3068	3069	3070	3082	3084	3086	3088	3091	3093	3095	3097	3099	3101
	3103	3105	3107	3109	3111	3113	3115	3117	3119	3122	3124	3126	3148	3150	3152
	3153	3155	3167	3185	3194	3195	3201	3202	3208	3226	3227	3245	3246	3247	3273
	3277	3292	3296	3304	3312	3314	3316	3318	3321	3323	3325	3327	3329	3331	3333
	3335	3337	3339	3341	3343	3345	3347	3349	3373	3375	3377	3400	3404	3406	3426
	3444	3447	3455	3456	3462	3465	3471	3475	3496	3502	3506	3518	3522	3524	3531
	3533	3534	3547	3548	3549	3550	3571	3573	3574	3576	3578	3579	3581	3582	3584
	3585	3587	3588	3590	3593	3595	3597	3619	3621	3641	3643	3645	3646	3648	3668
	3686	3689	3695	3696	3702	3706	3748	3750	3752	3753	3755	3775	3793	3796	3802
	3803	3809	3813	3852	3853	3856	3857	3858	3871	3872	3875	3977	3879	3880	3887
	3898	3900	3902	3903	3910	3917	3918	3941	3942	3943	3957	3958	3959	3960	3977
	3978	3979	3980	3994	3998	3999	4000	4001	4025	4027	4029	4030	4032	4044	4062
	4129	4133	4134	4135	4136	4160	4162	4164	4165	4167	4179	4197	4248	4249	4257
	4258	4259	4281	4285	4286	4287	4288	4312	4314	4316	4317	4319	4331	4349	4394
	4395	4420	4421	4422	4463	4467	4468	4469	4470	4494	4496	4498	4499	4501	4514
	4532	4615	4619	4620	4621	4622	4646	4648	4650	4651	4653	4665	4683	4732	4733
	4758	4759	4760	4791	4795	4796	4797	4798	4822	4824	4826	4827	4829	4842	4860
	4933	4937	4938	4939	4940	4964	4966	4968	4969	4971	4983	5001	5047	5048	5073
	5074	5075	5116	5120	5121	5122	5123	5147	5149	5151	5152	5154	5167	5185	5268
	5272	5273	5274	5275	5299	5301	5303	5304	5306	5318	5336	5385	5386	5414	5415
	5416	5417	5460	5464	5465	5466	5467	5495	5497	5499	5500	5502	5515	5533	5545
	5553	5641	5645	5646	5647	5648	5677	5679	5681	5682	5684	5696	5714	5768	5769
	5796	5797	5798	5812	5813	5814	5815	5829	5830	5831	5832	5849	5850	5851	5852
	5862	5863	5864	5865	5878	5882	5883	5884	5885	5910	5912	5914	5915	5917	5929
	5947	6013	6014	6015	6016	6017	6018	6031	6035	6036	6037	6038	6062	6064	6066
	6067	6069	6081	6099	6151	6152	6181	6182	6183	6197	6198	6199	6200	6214	6215
	6216	6217	6234	6235	6236	6237	6247	6248	6249	6250	6263	6267	6268	6269	6270
	6294	6296	6298	6299	6301	6313	6331	6402	6403	6404	6405	6406	6407	6421	6425
	6426	6427	6428	6452	6454	6456	6457	6459	6471	6489	6541	6542	6557	6558	6559
	6589	6591	6593	6594	6596	6603	6617	6618	6619	6620	6636	6637	6638	6639	6656
	6660	6661	6662	6663	6687	6689	6691	6692	6694	6710	6728	6785	6786	6787	6788



	6809	6813	6814	6815	6816	6841	6843	6845	6846	6848	6860	6878	6978	6980	6982
	6983	6985	6997	7015	7024	7025	7031	7032	7038	7074	7075	7076	7077	7096	7100
	7101	7102	7103	7119	7121	7123	7124	7132	7152	7153	7162	7163	7164	7180	7182
	7184	7185	7187	7194	7216	7218	7220	7221	7223	7236	7254	7263	7264	7270	7271
	7277	7319	7320	7321	7322	7341	7345	7346	7347	7348	7364	7366	7368	7369	7377
	7400	7401	7423	7424	7425	7444	7446	7448	7449	7451	7459	7470	7471	7472	7473
	7533	7534	7535	7536	7558	7562	7563	7564	7565	7589	7591	7593	7594	7602	7647
	7651	7652	7653	7654	7678	7680	7682	7683	7691	7756	7760	7761	7762	7763	7788
	7790	7792	7793	7795	7807	7825	7860	7861	7878	7879	7880	7900	7902	7904	7905
	7912	7981	7985	7986	7987	7988	8012	8014	8016	8017	8019	8031	8049	8085	8086
	8105	8106	8107	8134	8136	8138	8139	8146	8215	8219	8220	8221	8222	8251	8253
	8255	8256	8258	8270	8288	8328	8329	8352	8353	8354	8388	8392	8393	8394	8395
	8411	8413	8415	8416	8424	8437	8443	8448	8454	8498	8502	8503	8504	8505	8525
	8527	8529	8530	8538	8547	8555	8559	8565	8566	8574	8578	8629	8633	8634	8635
	8636	8649	8651	8653	8654	8662	8697	8703	8707	8713	8742	8743	8744	8750	8751
	8752	8753	8757	8763	8787	8790	8791	8792	8794	8797	8803	8807	8811	8812	8812
	8821	8822	8825	8826	9353	9359	9366	9370	9374	9381	9385	9509	9520	9526	9526
	9532	9536	9542	9548	9555	9565	9571	9576	9580	9582	9593	9597	9598	9600	9600
	9602	9609	9613	9618	9619	9623	9626	9627	9628	9696	9719	9769	9806	9819	9840
	9841	9848	9850	9855	9870	9871	9872	9877	9876	9930	9931	9937	9953	9960	9965
	9966	9967	9968	9971	9978	9979	10004	10012	10020	10028	10036	10044	10052	10060	10068
	10076	10084	10092	10100	10108	10116	10124	10132	10140	10148	10209	10287	10296	10299	10312
	10313	10314	10315	10316	10317	10318	10319	10320	10325	10337	10347	10357	10359	10366	10312
.EQUIV	40	41	43	58	59	88	89	90	91	92	93	94	95	96	97
	116	117	118	119	120	121	122	123	124	125					
.EVEN	1422	1426	1430	1434	1438	1442	1446	1480	1518	1522	1570	1593	1597	1601	1605
	1633	1639	1645	1697	1704	1711	1852	1858	1885	1891	2576	2585	3273	3277	3496
	3502	3506	5545	5553	8437	8443	8697	8703	8757	8763	9353	9359	9366	9370	9374
	9381	9385	9503	9509	9520	9526	9532	9536	9542	9548	9555	9777	10004	10012	10020
	10028	10036	10044	10052	10060	10068	10076	10084	10092	10100	10108	10116	10124	10132	10140
	10148	10205	10365	11734	11794	11795									
.IF	2	12	16	29	30	31	32	33	38	98	126	151	182	193	204
	205	207	214	216	243	251	257	258	259	263	264	982	988	1022	1024
	1197	1377	1381	1382	1384	1386	1388	1390	1391	1393	1405	1421	1425	1429	1433
	1437	1441	1445	1479	1493	1495	1496	1498	1499	1517	1521	1531	1533	1534	1536
	1537	1551	1553	1560	1562	1563	1569	1592	1596	1600	1604	1632	1638	1644	1671
	1673	1680	1682	1683	1684	1696	1703	1710	1723	1725	1730	1732	1765	1767	1770
	1772	1803	1805	1829	1831	1851	1857	1884	1890	1955	1957	1959	1961	1969	1971
	1977	2009	2011	2017	2019	2057	2059	2061	2063	2073	2077	2096	2102	2135	2137
	2142	2144	2176	2178	2180	2182	2190	2218	2220	2222	2223	2225	2227	2229	2231
	2233	2235	2237	2259	2258	2290	2292	2294	2302	2308	2310	2311	2317	2342	2344
	2348	2350	2360	2362	2364	2365	2367	2369	2371	2373	2375	2377	2379	2401	2414
	2416	2418	2420	2428	2442	2448	2450	2495	2501	2526	2528	2553	2555	2575	2584
	2600	2602	2604	2605	2607	2609	2611	2613	2615	2617	2619	2641	2667	2669	2671
	2673	2681	2690	2696	2698	2700	2701	2707	2738	2740	2758	2760	2771	2773	2775
	2776	2778	2780	2782	2784	2786	2788	2790	2791	2817	2819	2820	2822	2830	2839
	2845	2847	2849	2850	2856	2887	2889	2894	2896	2907	2909	2911	2912	2914	2916
	2918	2920	2922	2924	2926	2948	2972	2976	2977	2987	2991	3026	3029	3062	3064
	3068	3070	3082	3086	3090	3091	3093	3097	3101	3105	3109	3113	3117	3121	3124
	3148	3152	3153	3163	3167	3191	3194	3195	3201	3202	3208	3225	3227	3245	3247
	3272	3276	3290	3292	3294	3296	3304	3312	3316	3320	3321	3323	3327	3331	3335
	3339	3343	3347	3351	3375	3398	3400	3402	3404	3414	3418	3447	3453	3455	3456
	3462	3465	3471	3473	3475	3495	3501	3505	3516	3518	3520	3522	3531	3532	3534
	3547	3549	3550	3571	3574	3577	3578	3579	3582	3585	3588	3593	3595	3597	3619
	3641	3645	3646	3656	3660	3689	3695	3696	3702	3703	3706	3748	3752	3753	3763
	3767	3796	3802	3803	3809	3810	3813	3851	3853	3856	3858	3871	3875	3879	3880

3887	3898	3902	3903	3910	3916	3918	3941	3943	3956	3957	3958	3959	3960	3976
3977	3978	3979	3980	3992	3994	3997	3998	3999	4000	4025	4029	4030	4040	4044
4127	4132	4133	4134	4135	4160	4164	4165	4175	4179	4247	4249	4257	4259	4280
4284	4285	4286	4287	4312	4315	4317	4327	4331	4393	4395	4420	4422	4462	4466
4467	4468	4469	4494	4498	4499	4510	4514	4614	4618	4619	4620	4621	4646	4650
4651	4661	4665	4731	4733	4758	4760	4790	4794	4795	4796	4797	4822	4826	4827
4839	4842	4932	4936	4937	4938	4939	4964	4968	4969	4979	4983	5046	5048	5073
5075	5115	5119	5120	5121	5122	5147	5151	5152	5163	5167	5267	5271	5272	5273
5274	5299	5303	5304	5314	5318	5384	5386	5414	5416	5417	5459	5463	5464	5465
5466	5495	5499	5500	5511	5515	5544	5552	5640	5644	5645	5646	5647	5677	5681
5682	5692	5696	5767	5769	5796	5798	5811	5812	5813	5814	5815	5828	5829	5830
5831	5832	5848	5849	5850	5851	5852	5861	5862	5863	5864	5865	5876	5878	5881
5882	5883	5884	5910	5914	5915	5925	5929	6012	6013	6014	6015	6016	6017	6018
6029	6034	6035	6036	6037	6062	6066	6067	6077	6081	6150	6152	6181	6183	6196
6197	6198	6199	6200	6213	6214	6215	6216	6217	6233	6234	6235	6236	6237	6246
6247	6248	6249	6250	6261	6263	6266	6267	6268	6269	6294	6298	6299	6309	6313
6401	6402	6403	6404	6405	6406	6407	6419	6424	6425	6426	6427	6452	6456	6457
6467	6471	6540	6542	6557	6559	6589	6593	6594	6603	6616	6617	6618	6619	6620
6635	6636	6637	6638	6639	6654	6656	6659	6660	6661	6662	6687	6691	6692	6706
6710	6784	6785	6786	6787	6788	6807	6812	6813	6814	6815	6841	6845	6846	6856
6860	6978	6982	6983	6993	6997	7021	7024	7025	7031	7032	7038	7073	7074	7075
7076	7077	7094	7099	7100	7101	7102	7119	7123	7124	7132	7151	7153	7162	7164
7180	7184	7185	7194	7216	7220	7221	7222	7236	7260	7263	7264	7270	7271	7277
7318	7319	7320	7321	7322	7339	7344	7345	7346	7347	7364	7368	7369	7377	7399
7401	7423	7425	7444	7448	7449	7459	7469	7470	7471	7472	7473	7532	7533	7534
7535	7536	7556	7558	7561	7562	7563	7564	7589	7593	7594	7602	7645	7650	7651
7652	7653	7678	7682	7683	7691	7754	7756	7759	7760	7761	7762	7788	7792	7793
7803	7807	7859	7861	7878	7890	7900	7904	7905	7912	7980	7984	7985	7986	7987
8012	8016	8017	8027	8031	8084	8086	8105	8107	8134	8138	8139	8146	8214	8218
8219	8220	8221	8251	8255	8256	8266	8270	8327	8329	8352	8354	8387	8391	8392
8393	8394	8411	8415	8416	8424	8436	8442	8448	8454	8497	8501	8502	8503	8504
8525	8529	8530	8538	8547	8553	8555	8559	8565	8566	8572	8574	8575	8578	8628
8632	8633	8634	8635	8649	8653	8654	8662	8696	8702	8707	8713	8741	8742	8744
8750	8752	8753	8756	8762	8786	8790	8791	8792	8793	8794	8796	8802	8805	8807
8811	8812	8821	8822	9352	9358	9365	9369	9373	9380	9384	9502	9508	9519	9525
9531	9535	9541	9547	9554	9564	9570	9575	9580	9592	9594	9595	9596	9598	9599
9600	9609	9611	9619	9620	9625	9626	9627	9695	9719	9768	9772	9801	9818	9840
9848	9849	9854	9870	9871	9876	9882	9898	9930	9936	9946	9956	9963	9965	9966
9968	9971	9978	9979	10003	10011	10019	10027	10035	10043	10051	10059	10067	10075	10083
10091	10099	10107	10115	10123	10131	10139	10147	10208	10286	10295	10299	10303	10313	10314
10315	10316	10317	10318	10319	10320	10324	10337	10347	10355	10357	10359	10363		
29	31	32	33	40	206	208	214	216	243	264	983	989	1023	1025
1198	1381	1494	1495	1497	1498	1499	1532	1533	1535	1536	1537	1552	1553	1561
1562	1563	1672	1673	1681	1682	1683	1724	1725	1731	1732	1766	1767	1771	1772
1804	1805	1830	1831	1971	2010	2011	2018	2019	2096	2136	2137	2143	2144	2218
2220	2223	2225	2227	2229	2231	2233	2235	2237	2259	2308	2310	2311	2343	2344
2349	2350	2360	2362	2365	2367	2369	2371	2373	2375	2377	2379	2401	2442	2495
2527	2528	2554	2555	2600	2602	2605	2607	2609	2611	2613	2615	2617	2619	2641
270	2701	2739	2740	2759	2760	2771	2773	2776	2778	2780	2782	2784	2786	2788
2790	2791	2839	2850	2888	2889	2895	2896	2907	2909	2912	2914	2916	2918	2920
2922	2924	2926	2948	3029	3063	3064	3069	3070	3082	3086	3091	3093	3097	3101
3105	3109	3113	3117	3121	3122	3124	3194	3195	3202	3226	3227	3246	3247	3312
3316	3321	3323	3327	3331	3335	3339	3343	3347	3351	3373	3375	3447	3456	3465
3473	3475	3533	3534	3548	3549	3571	3574	3578	3579	3582	3585	3588	3593	3595
3597	3619	3689	3696	3706	3796	3803	3813	3852	3853	3857	3858	3917	3918	3942
3943	3992	3994	3998	3999	4000	4127	4129	4133	4134	4135	4248	4249	4258	4259

.IFF

	4280	4285	4286	4287	4394	4395	4421	4422	4462	4467	4468	4469	4614	4619	4620
	4621	4732	4733	4759	4760	4790	4795	4796	4797	4932	4937	4938	4939	5047	5048
	5074	5075	5115	5120	5121	5122	5267	5272	5273	5274	5385	5386	5415	5416	5417
	5459	5464	5465	5466	5640	5645	5646	5647	5768	5769	5797	5798	5876	5878	5882
	5883	5884	6029	6031	6035	6036	6037	6151	6152	6182	6183	6261	6263	6267	6268
	6269	6419	6421	6425	6426	6427	6541	6542	6558	6559	6654	6656	6660	6661	6662
	6807	6809	6813	6814	6815	7024	7025	7032	7094	7096	7100	7101	7102	7152	7153
	7163	7164	7263	7264	7271	7339	7341	7345	7346	7347	7400	7401	7424	7425	7556
	7558	7562	7563	7564	7645	7647	7651	7652	7653	7756	7760	7761	7762	7860	7861
	7879	7880	7980	7985	7986	7987	8085	8086	8106	8107	8214	8219	8220	8221	8328
	8329	8353	8354	8387	8392	8393	8394	8448	8497	8502	8503	8504	8547	8559	8566
	8578	8628	8633	8634	8635	8707	8742	8743	8744	8751	8752	8753	8787	8793	8797
	8803	8806	8822	9565	9593	9596	9597	9600	9626	9628	9696	9769	9819	9840	9841
	9849	9854	9870	9877	9931	9936	9956	9978	9979	10209	10287	10296	10325	10357	
.IFT	1422	1426	1430	1434	1438	1442	1446	1480	1518	1522	1570	1593	1597	1601	1605
	1633	1639	1645	1697	1704	1711	1852	1858	1885	1891	1971	2096	2218	2220	2222
	2223	2225	2227	2229	2231	2233	2235	2237	2259	2308	2311	2360	2362	2364	2365
	2367	2369	2371	2373	2375	2377	2379	2401	2442	2495	2576	2585	2600	2602	2604
	2605	2607	2609	2611	2613	2615	2617	2619	2641	2690	2701	2771	2773	2775	2776
	2778	2780	2782	2784	2786	2788	2790	2791	2839	2850	2907	2909	2911	2912	2914
	2916	2918	2920	2922	2924	2926	2948	3026	3082	3086	3090	3091	3093	3097	3101
	3105	3109	3113	3117	3121	3124	3191	3195	3202	3273	3277	3312	3316	3320	3321
	3323	3327	3331	3335	3339	3343	3347	3351	3375	3447	3456	3465	3473	3496	3502
	3506	3571	3574	3577	3578	3579	3582	3585	3588	3593	3595	3597	3619	3689	3696
	3703	3796	3803	3810	3992	3994	3997	3998	3999	4000	4127	4132	4133	4134	4135
	4280	4284	4285	4286	4287	4462	4466	4467	4468	4469	4614	4618	4619	4620	4621
	4790	4794	4795	4796	4797	4932	4936	4937	4938	4939	5115	5119	5120	5121	5122
	5267	5271	5272	5273	5274	5459	5463	5464	5465	5466	5545	5553	5640	5644	5645
	5646	5647	5876	5878	5881	5882	5883	5884	6029	6034	6035	6036	6037	6261	6263
	6266	6267	6268	6269	6419	6424	6425	6426	6427	6654	6656	6659	6660	6661	6662
	6807	6812	6813	6814	6815	7021	7025	7032	7094	7099	7100	7101	7102	7260	7264
	7271	7339	7344	7345	7346	7347	7556	7558	7561	7562	7563	7564	7645	7650	7651
	7652	7653	7754	7756	7759	7760	7761	7762	7980	7984	7985	7986	7987	8214	8218
	8219	8220	8221	8387	8391	8392	8393	8394	8437	8443	8448	8497	8501	8502	8503
	8504	8547	8559	8566	8575	8628	8632	8633	8634	8635	8697	8703	8707	8757	8763
	9353	9359	9366	9370	9374	9381	9385	9503	9509	9520	9526	9532	9536	9542	9548
	9555	9608	9825	9903	9923	9930	9966	10004	10012	10020	10028	10036	10044	10052	10060
	10068	10076	10084	10092	10100	10108	10116	10124	10132	10140	10148				
.IFTF	1422	1426	1430	1434	1438	1442	1446	1480	1518	1522	1570	1593	1597	1601	1605
	1633	1639	1645	1697	1704	1711	1852	1858	1885	1891	2576	2585	3273	3277	3496
	3502	3506	5545	5553	8437	8443	8697	8703	8757	8763	9353	9359	9366	9370	9374
	9381	9385	9503	9509	9520	9526	9532	9536	9542	9548	9555	9606	9819	9899	9907
	9929	9965	10004	10012	10020	10028	10036	10044	10052	10060	10068	10076	10084	10092	10100
	10108	10116	10124	10132	10140	10148									
.IIF	1	6	11	26	27	28	29	32	33	34	35	148	263	1382	1384
	1390	1391	1393	1394	1706	1713	1935	1936	1958	1959	2036	2037	2060	2061	2155
	2156	2179	2180	2267	2268	2291	2292	2417	2418	2648	2670	2798	2820	2953	2954
	2975	2976	3128	3129	3151	3152	3293	3294	3379	3380	3401	3402	3519	3520	3622
	3644	3645	3728	3729	3751	3752	3878	3879	3901	3902	4002	4003	4028	4029	4137
	4138	4163	4164	4289	4290	4315	4316	4471	4472	4497	4498	4623	4624	4649	4650
	4799	4800	4825	4826	4941	4942	4967	4968	5124	5125	5150	5151	5276	5277	5302
	5303	5468	5469	5498	5499	5649	5650	5680	5681	5886	5887	5913	5914	6039	6040
	6065	6066	6271	6272	6297	6298	6429	6430	6455	6456	6592	6593	6664	6665	6690
	6691	6817	6818	6844	6845	6981	6982	7104	7105	7122	7123	7183	7184	7219	7220
	7349	7350	7367	7368	7447	7448	7566	7567	7592	7593	7655	7656	7681	7682	7764
	7765	7791	7792	7903	7904	7989	7990	8015	8016	8137	8138	8223	8224	8254	8255



	8396	8397	8414	8415	8506	8507	8528	8529	8637	8638	8652	8653	8791	8797	8798
	8809	8822	8826	9571	9572	9573	9574	9575	9576	9607	9608	9623	9626	9627	9768
	9772	9777	9853	9871	9872	9930	9937	9938	9939	9940	9941	9971	9979	10312	10313
.IRP	10314	10315	10316	10317	10318	10319	10320								
	1377	1493	1531	1551	1671	1723	1765	1803	2009	2135	2218	2227	2239	2255	2259
	2342	2360	2369	2381	2397	2401	2433	2450	2526	2600	2609	2621	2637	2641	2738
	2771	2780	2791	2887	2907	2916	2928	2944	2948	3062	3081	3108	3225	3311	3334
	3353	3369	3532	3570	3593	3599	3615	3851	3916	4247	4393	4731	5046	5384	5767
	6150	6540	7151	7399	7503	7519	7859	7945	7961	8084	8179	8195	8327	8742	8850
	8860	8879	8888	8907	8913	9016	9025	9078	9138	9161	9206	9218	9230	9248	9261
.LIST	9285	9317	9404	9413	9462	9490	9641	9681	9893	9919	9946	10331	10347		
	1	11	20	21	32	140	148	243	245	246	247	248	249	250	251
	252	253	254	255	256	257	1377	1395	1405	1422	1426	1430	1434	1438	1442
	1445	1480	1493	1498	1518	1522	1531	1536	1551	1562	1570	1593	1597	1601	1605
	1633	1639	1645	1671	1682	1684	1697	1704	1711	1723	1732	1765	1772	1803	1831
	1852	1858	1885	1891	1963	1968	1969	2009	2019	2026	2067	2072	2073	2135	2144
	2184	2189	2190	2296	2301	2302	2342	2350	2422	2427	2428	2526	2555	2576	2585
	2675	2680	2681	2738	2760	2824	2829	2830	2887	2896	2981	2986	2987	3062	3070
	3157	3162	3163	3225	3247	3273	3277	3298	3303	3304	3408	3413	3414	3496	3502
	3506	3525	3530	3531	3532	3549	3550	3650	3655	3656	3757	3762	3763	3851	3858
	3860	3881	3886	3887	3904	3909	3910	3916	3943	4034	4039	4040	4169	4174	4175
	4247	4259	4321	4326	4327	4393	4422	4504	4509	4510	4655	4660	4661	4731	4760
	4832	4837	4838	4973	4978	4979	5046	5075	5157	5162	5163	5308	5313	5314	5384
	5416	5417	5505	5510	5511	5545	5553	5686	5691	5692	5767	5798	5919	5924	5925
	6071	6076	6077	6150	6183	6303	6308	6309	6461	6466	6467	6540	6559	6597	6602
	6603	6700	6705	6706	6850	6855	6856	6987	6992	6993	7126	7131	7132	7151	7164
	7188	7193	7194	7226	7231	7232	7371	7376	7377	7399	7425	7453	7458	7459	7596
	7601	7602	7685	7690	7691	7797	7802	7803	7859	7880	7906	7911	7912	8021	8026
	8027	8084	8107	8140	8145	8146	8260	8265	8266	8327	8354	8418	8423	8424	8437
	8443	8532	8537	8538	8656	8661	8662	8697	8703	8742	8752	8757	8763	8797	9353
	9359	9366	9370	9374	9381	9385	9503	9509	9520	9526	9532	9536	9542	9548	9555
	9575	9840	9971	10004	10012	10020	10028	10036	10044	10052	10060	10068	10076	10084	10092
	10100	10108	10116	10124	10132	10140	10148	10303	10312	10313	10314	10315	10316	10317	10318
.MACRO	10319	10320	10321												
	1	21	33	207	1493	1531	1550	1670	1722	1764	1803	2008	2134	2341	2525
	2737	2886	3061	3225	3532	3851	3916	4246	4392	4730	5046	5384	5766	6149	6539
	7150	7399	7858	8083	8326	8742	10303								
.MCALL	1	140	1395												
.NLIST	1	11	20	21	32	140	148	243	245	246	247	248	249	250	251
	252	253	254	255	256	257	1377	1395	1405	1422	1426	1430	1434	1438	1442
	1446	1480	1493	1498	1518	1522	1531	1536	1551	1562	1570	1593	1597	1601	1605
	1633	1639	1645	1671	1682	1684	1697	1704	1711	1723	1732	1765	1772	1803	1831
	1852	1858	1885	1891	1963	1968	1969	2009	2019	2026	2067	2072	2073	2135	2144
	2184	2189	2190	2296	2301	2302	2342	2350	2422	2427	2428	2526	2555	2576	2585
	2675	2680	2681	2738	2760	2824	2829	2830	2887	2896	2981	2986	2987	3062	3070
	3157	3162	3163	3225	3247	3273	3277	3298	3303	3304	3408	3413	3414	3496	3502
	3506	3525	3530	3531	3532	3549	3550	3650	3655	3656	3757	3762	3763	3851	3858
	3860	3881	3886	3887	3904	3909	3910	3916	3943	4034	4039	4040	4169	4174	4175
	4247	4259	4321	4326	4327	4393	4422	4504	4509	4510	4655	4660	4661	4731	4760
	4832	4837	4838	4973	4978	4979	5046	5075	5157	5162	5163	5308	5313	5314	5384
	5416	5417	5505	5510	5511	5545	5553	5686	5691	5692	5767	5798	5919	5924	5925
	6071	6076	6077	6150	6183	6303	6308	6309	6461	6466	6467	6540	6559	6597	6602
	6603	6700	6705	6706	6850	6855	6856	6987	6992	6993	7126	7131	7132	7151	7164
	7188	7193	7194	7226	7231	7232	7371	7376	7377	7399	7425	7453	7458	7459	7596
	7601	7602	7685	7690	7691	7797	7802	7803	7859	7880	7906	7911	7912	8021	8026
	8027	8084	8107	8140	8145	8146	8260	8265	8266	8327	8354	8418	8423	8424	8437

	8443	8532	8537	8538	8656	8661	8662	8697	8703	8742	8752	8757	8763	8797	9353
	9359	9366	9370	9374	9381	9385	9503	9509	9520	9526	9532	9536	9542	9548	9555
	9575	9840	9971	10004	10012	10020	10028	10036	10044	10052	10060	10068	10076	10084	10092
	10100	10108	10116	10124	10132	10140	10148	10303	10312	10313	10314	10315	10316	10317	10318
.PAGE	207	263	981	1199	1250	1281	1365	8786	8919	8947	8967	9004	9209	9324	9417
.RADIX	9551	9563	9627	9930	10286	10324	10366								
	1968	1969	2072	2073	2189	2190	2301	2302	2427	2428	2680	2681	2829	2830	2986
	2987	3162	3163	3303	3304	3413	3414	3530	3531	3655	3656	3762	3763	3886	3887
	3909	3910	4039	4040	4174	4175	4326	4327	4509	4510	4660	4661	4837	4838	4978
	4979	5162	5163	5313	5314	5510	5511	5691	5692	5924	5925	6076	6077	6308	6309
	6466	6467	6602	6603	6705	6706	6855	6856	6992	6993	7131	7132	7193	7194	7231
	7232	7376	7377	7458	7459	7601	7602	7690	7691	7802	7803	7911	7912	8026	8027
.REPT	8145	8146	8265	8266	8423	8424	8537	8538	8661	8662					
.SBTTL	12	16	149	245	251										
	22	36	142	152	170	209	265	1202	1365	1493	1531	1551	1671	1723	1765
	1803	2009	2135	2342	2526	2738	2887	3062	3225	3532	3851	3916	4247	4393	4731
	5046	5384	5767	6150	6540	7151	7399	7859	8084	8327	8742	8788	8829	9566	9629
.TITLE	9697	9770	9878	9932	9980	10210	10288	10304	10326						
.WORD	1														
	148	149	150	216	219	220	221	222	225	226	227	228	229	230	231
	232	233	234	243	245	246	247	248	249	250	251	252	253	254	255
	256	1304	1339	1340	1342	1346	1347	1362	1363	1364	8802	8805	9323	9762	9772
	9773	9774	9926	9929	10177	10182	10285	10356	10358	11735	11738	11741	11743	11745	11746
	11749	11752	11754	11756	11758	11760	11762	11764							

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*.DERPUB/NL:TOC/SOL/CRF=DERPUB.SML, DERPUB.P11  
 RUN-TIME: 95 137 18 SECONDS  
 RUN-TIME RATIO: 348/250=1.3  
 CORE USED: 33K (65 PAGES)

